<table>
<thead>
<tr>
<th>Work Package Number:</th>
<th>WP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Package Title:</td>
<td>Decision rules and evidence</td>
</tr>
<tr>
<td>Nature of Deliverable:</td>
<td>Software deliverable</td>
</tr>
<tr>
<td>Dissemination Level:</td>
<td>Public</td>
</tr>
<tr>
<td>Version:</td>
<td>1.1 Final</td>
</tr>
<tr>
<td>Delivery Date From Annex 1:</td>
<td>M51</td>
</tr>
<tr>
<td>Principal Authors:</td>
<td>Derek Corrigan (RCSI)</td>
</tr>
<tr>
<td>Contributing Authors:</td>
<td>-</td>
</tr>
<tr>
<td>Partner Institutions:</td>
<td>Royal College of Surgeons in Ireland (RCSI)</td>
</tr>
<tr>
<td></td>
<td>Trinity College Dublin (TCD)</td>
</tr>
<tr>
<td></td>
<td>Imperial College London (IC)</td>
</tr>
<tr>
<td></td>
<td>Wroclaw University of Technology (WRUT)</td>
</tr>
<tr>
<td></td>
<td>Mediterranean Institute of Primary Care (RCSI)</td>
</tr>
<tr>
<td></td>
<td>Kings College London (KCL)</td>
</tr>
</tbody>
</table>
This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 247787 [TRANSFoRm].

**Document Update History**

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Date</th>
<th>Revision Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 Draft</td>
<td>18/04/2014</td>
<td>Initial draft for discussion released</td>
</tr>
<tr>
<td>1.1 Draft</td>
<td>20/05/2014</td>
<td>Revised based on review feedback from B Delaney, V Curcin.</td>
</tr>
<tr>
<td>1.1 Final</td>
<td>23/05/2014</td>
<td>Final document release</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** .................................................................................................................................................. 9  
   1.1. Actionable knowledge to address patient safety requirements ......................................................... 10  
   1.2. Document outline .................................................................................................................................. 11  
   1.3. Supplementary Documents .................................................................................................................. 12  

2. **Functional design requirements for the support of actionable evidence in TRANSFoRm** ...................................................................................................................................................... 13  
   2.1. Characteristics of the Learning Healthcare System .............................................................................. 13  
   2.2. Diagnostic decision support requirements developed from WP2 outputs ................................... 15  
   2.3. A formalised clinical evidence diagnostic strategy ............................................................................ 16  
      2.3.1. Two Phase Diagnostic Process ....................................................................................................... 16  
      2.3.2. ‘Suggesting’ Phase Diagnostic Strategies ........................................................................................ 18  
      Presenting Complaint ...................................................................................................................................... 18  
      Self Labelling ................................................................................................................................................ 19  
      Pattern Recognition Trigger ........................................................................................................................ 19  
      2.3.3. ‘Alerting’ Phase Diagnostic Strategies .............................................................................................. 19  
      Pattern Recognition Fit .................................................................................................................................. 19  
      Red Flag Cues ............................................................................................................................................... 19  
      Severity of conditions ................................................................................................................................. 19  
      Restricted Rule Outs (Exclusion) .................................................................................................................. 19  
      Probabilistic Reasoning .............................................................................................................................. 20  
   2.4. Summary of functional requirements ..................................................................................................... 20  

3. **Clinical evidence ontology design structures** ................................................................................................. 21  
   3.1. Ontology Design Methodology .................................................................................................................. 21  
   3.2. Ontology concepts required to support diagnostic strategies ............................................................ 21  
      3.2.1. RFE Concept ...................................................................................................................................... 22
3.2.2. Diagnosis Concept .......................................................... 22
3.2.3. Cue Concept (including Sign, Symptom, Test, Risk) ....................... 23
3.2.4. Quantification Concept ....................................................... 23
3.2.5. Prevalence Concept .......................................................... 23
3.2.6. Demographic Concept ....................................................... 23
3.2.7. Binding Concept (incorporating sub concepts ReadCodeBinding, ICPC2Binding, ICD10Binding, UMLSBinding, SNOMEDBinding) ....................... 23
3.2.8. Modeling Co-morbidity ...................................................... 24

3.3. Ontology Development Tools ................................................ 25

4. **Clinical evidence service architecture and platform** ..................... 27

4.1. Triple store performance selection considerations .......................... 29
4.2. Sesame triple store platform ................................................... 30
4.3. Development of SPARQL Queries ............................................ 31
4.4. Sesame application programming interface (API) ........................... 31
4.5. Sesame REST API support for ad-hoc queries ............................... 32
4.6. Jersey Java Implementation of REST Services .............................. 32
4.7. Clinical evidence service interfaces and methods .......................... 32
4.8. Data Mining ........................................................................ 33
    4.8.1. Data mining steps .......................................................... 34
    4.8.2. Association rule derivation using KNIME ................................ 34
    4.8.3. Calculation and validation of quality measures using KNIME ........ 36
    4.8.4. Clinical association rule review ........................................ 36
    4.8.5. Rule review and curation in practice ................................... 38
    4.8.6. Association rule export .................................................. 40
    4.8.7. Ontology Rule Import and Update ..................................... 41
    4.8.8. Rule Availability through web service .................................. 43
4.9. Evidence service client description to implement decision support tool
requirements .................................................................................................................. 43

5. Conclusions ............................................................................................................... 48

6. Future Work ............................................................................................................... 49

7. References ............................................................................................................... 50

8. Abbreviations .......................................................................................................... 52

9. Appendix A – WP2 patient safety use case differentials ........................................ 53

10. Appendix B – Search criteria for WT 4.1 reviews of literature ............................ 56

11. Appendix C – Sample clinical evidence ontology design competency questions .......................................................... 59

12. Appendix D – Example of SPARQL query execution using Sesame API 60

13. Appendix E – Example of REST end point path definition ................................. 63

14. Appendix F – Clinical Evidence Service REST endpoints .................................. 64

15. Appendix G – Definitions of association rules quality measures ......................... 66

16. Appendix H – XML schema for association rules ............................................... 68

17. Appendix I – C# evidence client core classes for implementing decision support functions .................................................................................................................. 71

18. Appendix J – C# evidence client methods implemented by the ClinicalEvidenceWebInterface class .................................................................................................................. 75

19. Appendix K – C# evidence client methods implemented by the PatientCase class .................................................................................................................. 78

20. Appendix L – The PatientEvidenceSet C# class ................................................. 79

List of Figures

Figure 1 – Knowledge types and characteristics supported by TRANSFoRm ........... 9

Figure 2 – WP4 components as part of the broader decision support solution ....... 11

Figure 3 – Generation of clinical knowledge as part of the learning healthcare system
Figure 4 – Interactions in 2 Phases of a formal diagnostic strategy ......................................................... 17
Figure 5 – Core evidence model ontology concepts and relationships ..................................................... 22
Figure 6 – Model b implemented as a model of co-morbidity. Courtesy of Kruger et al. ......................................................... 24
Figure 7 – Defining ontology concepts in Protégé .................................................................................. 25
Figure 8 – Defining ontology relationships in Protégé ........................................................................... 26
Figure 9 – Defining ontology class diagnosis instances in Protégé ......................................................... 27
Figure 10 – Architecture implementation technologies .............................................................................. 28
Figure 11 – Clinical evidence architecture overview ................................................................................. 29
Figure 12 – Sesame triple store workbench browser .................................................................................. 30
Figure 13 – Sesame SPARQL query testing ............................................................................................ 31
Figure 14 – TransHIS episode of care structure ...................................................................................... 33
Figure 15 – Distinct steps in the TRANSFoRm data mining process ....................................................... 34
Figure 16 – A workflow to derive association rules in KNIME ................................................................. 35
Figure 17 – Structure of derived association rules .................................................................................... 35
Figure 18 – Web rule viewer selection of rules filters ................................................................................ 37
Figure 19 – The main rule viewer details screen ....................................................................................... 37
Figure 20 – Generation of association rules export XML from the web rule sender ................................ 40
Figure 21 – Flowchart for ontology update of evidence from XML export rule files ............................... 42
Figure 22 – REST evidence output of imported evidence rules ............................................................... 43
Figure 23 – The WP5 decision support tool window on top of the Vision 3 EHR ............................... 44
Figure 24 – Workflow and interactions of decision support components .................. 46

List of Tables

Table 1 – LHS functional requirements for TRANSFoRm ........................................... 14
Table 2 – WP2 diagnostic strategy requirements for TRANSFoRm ............................ 16
Table 3 – WP2 clinical evidence requirements for TRANSFoRm .............................. 16
Table 4 – Diagnostic phase evidence requirements for TRANSFoRm ....................... 18
Table 5 – UTI predictors for Netherlands data ......................................................... 39
Executive Summary

This document describes the components that collectively provide a fully functional backend clinical evidence service upon which TRANSFoRm diagnostic decision support is provided. The solution is based on ontological models that are made available using a service oriented architecture and are entirely based on open source components and open standards such as XML, JSON and RDF. The data mining process allows for the dynamic generation of actionable clinical evidence derived from coded sources of electronic primary care data. This evidence is used to populate the ontology content. The update process is described here using a specific clinical case example. A specific implementation of a client library for consuming the service is described based on the requirements of the TRANSFoRm diagnostic decision support tool developed in WP5.

The key contributions of this deliverable are:

- design and implementation of a flexible ontology based model of evidence with support for terminology bindings
- implementation of data mining process and curation tools to generate clinically meaningful evidence that is empirically measured based on coded sources of primary care EHR data
- the provision of fully functional service oriented evidence service that can be consumed by any third party tool using widely available open standards.
1. Introduction

The TRANSFoRm (Translational Research and Patient Safety in Europe) project provides an electronic platform to allow for the representation, sharing and re-use of three distinct types of clinical knowledge related to the primary care setting as shown in figure 1. These distinct types of knowledge are:

- research knowledge (in the form of clinical trial data)
- routinely collected healthcare knowledge (in the form of individual or aggregated sources of electronic healthcare records)
- actionable knowledge (in the form of derived knowledge obtained through the analysis of research or routinely collected knowledge).

Figure 1 – Knowledge types and characteristics supported by TRANSFoRm

The deliverable described in this document fully describes the work completed as part of Work Package 4 (WP4 – Decision Rules and Evidence). This deliverable (4.1 - Ontology of Diagnostic evidence and DSS service) specifically addresses the
requirements of generation, representation, curation and deployment of actionable knowledge within the TRANSFoRm project.

1.1. Actionable knowledge to address patient safety requirements

Diagnostic error has been shown to be a major threat to patient safety within the context of the primary care setting (1-2). TRANSFoRm addresses this patient safety issue through the provision of reusable modular components that are used in the prototype diagnostic decision support tool for use in primary care developed in WP5.

The evidence service is deployed in the clinical context of supporting three patient safety use cases developed as part of WP2. These patient safety use cases are based on a defined list of selected differential diagnosis to be considered of a patient presenting with a general clinical complaint of chest pain, abdominal pain or dyspnoea. The rationale for selection of these differentials was based on the creation of diagnostic scenarios that are constructed as part of WP2 to provide different levels of diagnostic challenge based on the diagnostic characteristics that those differentials exhibit.

The requirements for WP4 are therefore to provide a functional backend evidence service that can be consumed by a separately implemented diagnostic decision support tool interface (WP5). This is provided with a view to supporting the diagnostic clinical evidence requirements of the three defined patient safety use cases that will be used for evaluation of the diagnostic performance of the tool itself.

The core components of the evidence service are shown in figure 2. These are based on the functional objectives of the clinical evidence service that can be summarized as follows:

- provide an ontology based diagnostic model of evidence for the representation of clinical evidence supporting the three patient safety use cases developed in WP2
- provide for the population of the ontology content using clinical evidence derived from evidence based literature sources supporting patient safety use cases
- provide for the population of the ontology content using clinical evidence derived from data-mining of electronic sources of primary care EHR data
supporting patient safety uses

- make the ontology models available as a web based evidence service that can be directly queried to ask diagnostic questions supporting the requirements of the decision support tool developed for WP5
- support open technical standards that provide for high levels of both system and semantic interoperability.

![Decision Support Components]

*Figure 2 – WP4 components as part of the broader decision support solution*

### 1.2. Document outline

This document fully describes the work that has been done for the provision of the WP4 clinical evidence service as part of TRANSFoRm. This document relates specifically to the backend evidence service from WP4 and does not describe or
address the specification or development of the actual decision support tool interface itself that consumes the evidence service provided as part of work done in WT5.2b.

In section 2 we place the work done as part of WP4 within the broader context of the Learning Healthcare System (LHS) initiative(3-4). Section 2 will also define high level functional requirements for the service through review of the literature concerning cognitive diagnostic strategies that can be supported by this evidence service implementation. This is done with a view to supporting the outputs of WP2 that have defined patient safety use cases and previously compared the effectiveness of ‘early’ or ‘late’ diagnostic decision support. In section 3 we describe the core design and technical implementation of the diagnostic evidence ontology. Section 4 provides a description of the technical solution for making the evidence ontology available as part of fully functional web-based clinical evidence service. This describes web service implementation along with the population of evidence from data mining evidence sources. This section also describes the functionality provided by the evidence client implemented to specifically support the requirements of the WP5 decision support tool interface. Sections 5 and 6 will provide conclusions and suggestions for future work to more fully develop the models of actionable knowledge that have been described here.

1.3. Supplementary Documents

This document is informed by clinical evidence reviews carried out as part of WT 4.1 and WT 4.2. and already presented at previous review meetings. In addition to this deliverable document, the following accompanying technical document is also available:

TRANSFoRm WP4 – Clinical Evidence Service Software Installation and User Guide – This document describes the general software requirements for installing and configuring the clinical evidence service along with a description of the evidence service web functions. Also provided is reference to the on-line user documentation describing the specific use of the evidence client DLL for use with the TRANSFoRm decision support tool interface.
2. Functional design requirements for the support of actionable evidence in TRANSFoRm

Three core areas were considered as a basis for informing the functional requirements for the design and development of this deliverable. These are:

- the requirements for knowledge translation in the context of its role in the broader Learning Healthcare System
- clinical evidence requirements of patient safety use cases and decision support interventions as developed and recommended from WP2
- a review of diagnostic strategies commonly employed in primary care practice.

2.1. Characteristics of the Learning Healthcare System

The requirements of the TRANSFoRm clinical evidence service need to be considered in the broader context of its role in providing a middleware for the Learning Healthcare System (3-4). It performs this role through support for the conduct of clinical research through provision of an electronic research trial infrastructure combined with supporting tools for the generation, representation and curation of actionable knowledge from electronic sources as shown in figure 3.

![Figure 3 – Generation of clinical knowledge as part of the learning healthcare system](image)

Definitions have been proposed for what constitutes the LHS. These definitions have
been expanded to describe high-level capabilities of the LHS needed to support these definitions. One specific capability identified is that the LHS is ‘capable of engendering a virtuous cycle of health improvement’(4). Individual requirements identified to support this knowledge translation capability include that the LHS supports:

- ‘Generating valid clinical knowledge’
- ‘Packaging and curation of knowledge so it is widely accessible and actionable and putting knowledge to use to effect change’
- ‘Development of meaningful use of the EHR to support diagnostic and therapeutic support based on evidence’
- ‘Development of a computable representation of research evidence and make that available to EHR systems as a Web service’
- ‘Develop a means of providing diagnostic or therapeutic prompts within an EHR that works across a variety of EHR systems’.

Based on these we created the initial set of functional requirements shown in table 1:

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Clinical knowledge will be deployed as part of a service oriented architecture using open standards that support semantic and system interoperability</td>
</tr>
<tr>
<td>2</td>
<td>The clinical evidence service must support the ability to interact with multiple EHR vendors by allowing evidence to be bound to different terminological coding schemes</td>
</tr>
<tr>
<td>3</td>
<td>Any clinical knowledge generated must be accessible using a tool to support curation, clinical review and approval of generated knowledge with respect to a clinical ‘gold standard’, before that knowledge can be considered for deployment in the ‘live’ evidence service environment</td>
</tr>
<tr>
<td>4</td>
<td>The clinical evidence service must support a defined diagnostic query interface to allow third party tools to ask meaningful diagnostic questions of the knowledgebase</td>
</tr>
<tr>
<td>5</td>
<td>The clinical evidence service must support a defined update interface to allow data mining tools to insert or update data mined evidence discovered from electronic sources after clinical review into the ‘live’ evidence service</td>
</tr>
</tbody>
</table>

Table 1 – LHS functional requirements for TRANSFoRm
2.2. Diagnostic decision support requirements developed from WP2 outputs

Two outputs from WP2 are important in defining clinical requirements for the development of our evidence service deliverable. These outputs are:

- a study that investigated the timing points of when to provide diagnostic decision support as part of the patient consultation, carried out in WT 2.2
- development of patient safety use cases that describe the clinical content that will be used to test and evaluate the final prototype decision support tool, carried out in WT 2.1.

The work carried out in WT 2.2. examined the timing points in the primary care consultation with a view to establishing if interventions at an early or later stage were most effective for providing diagnostic decision support. Two basic strategies for diagnostic intervention were identified and tested:

- early intervention in the consultation providing a list of differential diagnoses to consider that have been included or excluded based on the presenting patient complaint (known as the reason for encounter (RFE)), patient demographics (age, sex), and patient history available from the EHR – this strategy is termed as ‘suggesting’
- later intervention in the consultation that further refines and ranks the probability of likely diagnoses based on supplementary additional diagnostic cues suggested and obtained by the decision support tool as part of the full patient diagnostic workup (signs, symptoms, risk factors, clinical tests) – this strategy is termed as ‘alerting’.

The conclusion of this deliverable was that both ‘suggesting’ and ‘alerting’ are potentially effective strategies for providing diagnostic decision support and should be considered as part of the implementation of the clinical evidence service as shown in table 2.
The clinical content to be populated in the clinical evidence service has been informed by the development of three patient safety scenarios as part of work done in WT 2.1. These scenarios are based on patients presenting with chest pain, abdominal pain or dyspnoea. For each presentation, WP2 developed a set of diagnostic scenarios of different degrees of difficulty. Evidence for the scenarios was provided partially through systematic reviews of literature evidence carried out in WT4.1. For an example of the search strategy employed, refer to Appendix B. Table 3 summarises evidence requirements.

Table 2 – WP2 diagnostic strategy requirements for TRANSFoRm

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>The clinical evidence service must support a formalised diagnostic strategy that allows for both ‘suggesting’ at an early stage of the consultation and also ‘alerting’ at a later stage of the consultation</td>
</tr>
</tbody>
</table>

Table 3 – WP2 clinical evidence requirements for TRANSFoRm

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>The clinical evidence service must support the concept of the ‘Reason for Encounter’ (RFE) as a trigger for evaluating patient evidence related to each of the associated differentials and their underlying diagnostic cues for each RFE</td>
</tr>
<tr>
<td>8</td>
<td>The clinical evidence service must support evidence manually populated from literature reviews carried out in WT 4.1. to support the presenting patient complaints and associated differential diagnoses as listed in Appendix A - WP2 patient safety use case differentials</td>
</tr>
</tbody>
</table>

2.3. A formalised clinical evidence diagnostic strategy

2.3.1. Two Phase Diagnostic Process

Based on the previous defined requirements, the TRANSFoRm evidence service
supports a formal two-phase approach to the diagnostic decision support that is triggered by a patient reason for encounter. As previously described in WP2 the two phases refer to the timings of decision support intervention and are referred to as “suggesting” and “alerting”. The evidence service provides for interactions between the clinician, the clinician EMR system (with integrated decision support tool developed in WP5) and the clinical evidence service developed in WP4 as shown in figure 4.

**A two-phase approach needed**

---

**Figure 4 – Interactions in 2 Phases of a formal diagnostic strategy**

The first “suggesting” phase as previously discussed is triggered by the presenting complaint(s), or ‘reason for encounter’. A number of differential diagnoses may be associated with any presenting reason for encounter. These can then be filtered based on some core demographics that might be obtained from the electronic health record (age, sex, ethnicity, nationality) and underlying patient risk factors. The key output from this phase is a suggested list of differentials to consider based on the
underlying patient reason for encounter.

The second “alerting” phase will involve more detailed diagnostic workup by looking at the underlying patient diagnostic cues and comparing them against the clinical evidence available for each diagnosis. The key output from this process will be a quantified and ranked list of competing diagnoses ordered by relative likelihood and disease severity. Requirements based on this approach are summarised in Table 4.

<table>
<thead>
<tr>
<th>Requirement No.</th>
<th>Requirement Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>The diagnostic process will be initiated through the initial submission of patient details obtained from the EMR that provides the initial RFE, patient demographics and patient risk factors. The output of this initial ‘suggesting’ process is a filtered list of initial problem differentials to consider.</td>
</tr>
<tr>
<td>10</td>
<td>The diagnostic process allows an iterative cycle of diagnostic workup initiated through the update of patient diagnostic cues entered through the decision support interface. The initial differentials to consider will be refined based on the presence or absence of diagnostic cues and a ranked list based on likely differential diagnoses will be returned categorised as ‘most likely’, ‘not to miss’ and ‘others’.</td>
</tr>
</tbody>
</table>

Table 4 – Diagnostic phase evidence requirements for TRANSFoRm

The clinical evidence service supports a number of identified diagnostic strategies identified from literature(5). These diagnostic strategies can be deployed and used within the ‘suggesting’ phase or ‘alerting’ phase of the diagnostic process as appropriate.

2.3.2. ‘Suggesting’ Phase Diagnostic Strategies

**Presenting Complaint** - the patient expresses a general expression of the underlying reason for their clinical presentation e.g. “doctor I have a pain in my abdomen”. The reason for encounter (RFE) is captured as a formal concept in the general model of clinical evidence. The RFE may be considered a general presenting problem or a defined diagnosis label. This RFE concept is associated through ontological relationships with defined differential diagnoses to consider.
**Self Labelling** – the patient expresses a definition of the problem that may also be a diagnostic label e.g. “doctor I think I have a Urinary Tract Infection”. The RFE may be considered a defined diagnosis label as suggested by the patient.

**Pattern Recognition Trigger** – identification of a potential differential diagnosis to consider based on diagnostic pattern recognition and experience on the part of the clinician. This will involve the identification of a potential diagnosis using a collection of diagnostic cues that support the presence or absence of a condition. The diagnoses to consider may be filtered based on patient demographics or underlying lifestyle risk factors as defined in the patient EMR.

### 2.3.3. **‘Alerting’ Phase Diagnostic Strategies**

**Pattern Recognition Fit** - identification of a potential differential diagnosis to consider based on consideration of additional diagnostic cues collected and entered as part of the clinical consultation and diagnostic workup with the patient. These additional cues may change the differentials to be considered by suggesting the presence or absence of any particular condition.

**Red Flag Cues** – specific combinations of strongly associated diagnostic cues may give rise to a stronger probability of the presence of a particular diagnosis(6). An example are the NICE clinical guidelines that identify 4 key red flags for the detection of ovarian cancer in primary care(7).

**Severity of conditions** – certain diagnoses are inherently more severe or life-threatening to the potential safety of a patient. It is therefore a requirement to record an ontological relationship of “hasDiseaseSeverity” against each diagnosis in the ontology. Ovarian cancer for example would have an associated disease severity of “High”. These disease severities can be used to classify “not to miss” diagnoses when presenting differential diagnoses for consideration.

**Restricted Rule Outs (Exclusion)** – exclusion is a useful diagnostic strategy to rule out potential diagnoses that may not fit a patient case. The utility of supporting negative ontological statements as well as positive associations in our ontology should allow for this e.g. IsNotSymptomOf.
**Probabilistic Reasoning** – probabilistic reasoning may be implemented as a technology based diagnostic strategy to calculate and rank the relative likelihood of competing differentials for a presenting RFE(8). The implementation of Bayes rule depends on knowledge of the pre-test probability of a particular condition (or prevalence of the disease in the study population). By combining likelihood ratios calculated for each RFE with an associated differential, or each diagnosis and an associated diagnostic cue, a post-test probability can be calculated. Different prevalence rates for a condition can occur in different demographic populations. This variation can be also depend on the clinical work setting (primary care, secondary care). The use of data mining and analysis techniques on populations with defined demographic contexts, can allow for more accurate calculation of pre-test probabilities based on empirical aggregated primary care data from EHRs.

### 2.4. Summary of functional requirements

We have identified requirements for the development of a clinical evidence service from three core areas: functional requirements of knowledge generation within the LHS itself, clinical content requirements for supporting defined patient safety use cases, and diagnostic strategy requirements describing how we wish to use our clinical evidence service to provide delivery of diagnostic decision support. We will now describe in section 3 how these functional requirements are implemented as technical solutions as part of this deliverable.
3. Clinical evidence ontology design structures

Based on the previous functional requirements a model of clinical evidence was designed and implemented. The ontology design methodology and concepts implemented are described in this section. The Protégé ontology file is available for download from the TRANSFoRm wiki under the ‘models’ section.

3.1. Ontology Design Methodology

The methodology for developing the clinical evidence ontology has been developed based on the ontology design practices advocated by the work of Noy and McGuinness, Allemang and Hendler, Gruninger and Fox (9-11). The steps for ontology design are:

- define clinical competency questions for ontology design (examples of clinical competency question are provided in Appendix C)
- identify core ontology features required to express competency questions
- define ontology queries to test ability to express ontology competency questions and validate query results

3.2. Ontology concepts required to support diagnostic strategies

The ontological concepts and relationships defined between them define the representation of clinical evidence in a manner that is fit for purpose to implement the diagnostic strategies previously described. A summary of the core ontology concepts that can support the 2 phase approach to decision support is now described and shown in figure 5. The ontology models support ontological relationships that are bidirectional in nature. This means that either a “top-down” approach, starting from a reason for encounter down to individual diagnostic cues, or a “bottom-up” approach starting from a diagnostic cue and working back to diagnoses can be employed. Although not explicitly shown in figure 5, each individual relationship has corresponding inverse relationships to support querying in either direction. For example the “hasSign” relationship has an inverse relationship of “isSignOf”, and the “hasDifferentialDiagnosis” relationship has an inverse
relationship of “isDifferentialDiagnosisOf”.

![Core evidence model ontology concepts and relationships](image)

**Figure 5 – Core evidence model ontology concepts and relationships**

### 3.2.1. RFE Concept
A single reason for encounter as expressed by the patient that triggers “top-down” reasoning e.g. “AbdominalPain”.

### 3.2.2. Diagnosis Concept
A diagnostic outcome or label for which evidence can be supported. A diagnosis can be a differential diagnosis of particular RFES. Each Diagnosis will have associated Cues. E.g. “UrinaryTractInfection” “hasSymptom” “Dysuria”.

22
3.2.3. **Cue Concept (including Sign, Symptom, Test, Risk)**

These represent the underlying patient reported symptoms, clinician observed signs, patient risk factors and clinical tests that are associated as clinical evidence to support a diagnosis. e.g. “Dysuria” is a clinical symptom.

3.2.4. **Quantification Concept**

This concept represents the strength of the association of an individual RFE or diagnostic cue with a particular diagnosis. An example is a positive or negative likelihood ratio obtained through data mining. Quantifications will also have an associated demographic context for the population from which the quantification was calculated e.g. age, sex, ethnicity, region.

3.2.5. **Prevalence Concept**

This concept represents the prevalence rate for a diagnosis as calculated through data mining at a point in time. Prevalence may also have an associated demographic context for the population from which the prevalence was calculated e.g. age, sex, region. The prevalence can be used as the pre-test probability in a Bayesian calculation to calculate the post-test probability of a diagnosis based on a presenting RFE or based on underlying cues present (their contribution represented by the quantification concept containing likelihood ratios for each individual RFE or cue associated with a diagnosis).

3.2.6. **Demographic Concept**

These concepts describe key demographic contexts that are associated with the prevalence of a diagnosis, with the quantification of diagnostic cues e.g. “Male”, “Female”, “Caucasian”, “Ireland”.

3.2.7. **Binding Concept (incorporating sub concepts ReadCodeBinding, ICPC2Binding, ICD10Binding, UMLSBinding, SNOMEDBinding)**

A terminological code binding can be associated with a RFE, cue or diagnosis. This concept allows the capture of clinical terminology codes for ICPC2, Read Codes, UMLS, ICD10 or SNOMED. Each instance has a code, a code description and a code version e.g. ‘165..11 – Fever Symptoms’.

23
3.2.8. Modeling Co-morbidity

A number of distinct models of co-morbidity can be implemented (12). These include:

- **associated liabilities model** establishes relations between individual diagnostic cues
- **multiformity model** establishes relations between diagnostic cues and diagnoses with a cue being associated with potentially many (perhaps co-morbid) conditions
- **causal model** – relations between diagnoses
- **independence model** – co-morbid diagnoses modelled as separate individual entities
- **hybrid model** – mix of associated liabilities and multiformity models.

They are represented pictorially in figure 6.

![Diagram of co-morbidity models](image)

*Figure 6 – Model b implemented as a model of co-morbidity. Courtesy of Kruger et al.*

For the purposes of representing clinical evidence found in both literature and from data mining, the multiformity model was most suitable and has been implemented as part of the general model of clinical evidence. Individual diagnostic cues may be associated (and quantified) with multiple diagnoses.
3.3. Ontology Development Tools

Protégé is a well established tool for development of ontologies. After reviewing available tools it was considered as suitable choice for the development of the clinical evidence ontology(13). Protégé version 4.3 has been used for the ontology modelling work. The ontology concepts previously identified were implemented along with associated relationships and instances of each class defined by the clinical content required for the patient safety use cases.

The steps required for construction of the ontology were:

- defining the ontology class structure (figure 7)
- defining object relationships existing between classes (figure 8)
- defining instances of classes to represent our clinical knowledge related to our specific patient safety use cases (figure 9)

![Figure 7 – Defining ontology concepts in Protégé](image-url)
Protégé uses web ontology language (OWL) and resource description framework (RDF) standards to represent the constructed ontology as a text file that can then be loaded into an ontology triple store (14-15).

Figure 8 – Defining ontology relationships in Protégé
4. Clinical evidence service architecture and platform

The clinical evidence service consists of three implementation layers:

- the persistence layer that provides multi user access and dynamic content update through a host data store for the protégé ontology upon which the evidence service is built around
- a service layer providing a fully functional web based interface with defined query methods to access questions of the ontology content from any third party tool
- a client layer provided as a client side library used by the TRANSFoRm decision support interface to provide for exchange of patient data from the CDSS to the evidence service and return of evidence recommendations to the CDSS interface

The implementation technologies for the three layers are summarised in figure 10.
The fully implemented architecture and components are shown in figure 11 and described in the subsequent sections.
4.1. Triple store performance selection considerations

In order to provide a platform for multi-user access and dynamic update of ontology
content, the Protégé ontology is hosted on a Sesame triple store. Sesame provides a rich persistence platform around which the evidence service can be developed and hosted. A number of triple stores were initially considered but Sesame was selected based on its open source nature and its favourable performance characteristics when compared to other available platforms (16-17).

4.2. Sesame triple store platform

The Sesame platform provides a fully functional web based workbench allowing for import of OWL based ontology content into a triple store for hosting (figure 12). The platform allows for ontology browsing and query development using an ontology query language called SPARQL (18).

![Sesame triple store workbench browser](image-url)
4.3. Development of SPARQL Queries

The Sesame platform allows for the formulation and testing of SPARQL queries that allow access to any of the ontology concepts defined in the clinical evidence ontology as shown in figure 13. Using the named ontology relationships that link concepts, these flexible queries allow implementation of ‘top-down’ diagnostic strategies (e.g. evd:UrinaryTractInfection evd:hasSymptom ?anySymptom) or ‘bottom-up’ diagnostic strategies (e.g. ?anySymptom evd:isSymptomOf evd:UrinaryTractInfection).

![Workbench](Image)

**Figure 13 – Sesame SPARQL query testing**

4.4. Sesame application programming interface (API)

The Sesame platform provides a fully functional API to allow for rich application development. This provides default methods for connecting to Sesame repositories.
and executing SPARQL queries as part of a programmable application. For a complete example of code that uses the Sesame API to execute a SPARQL query refer to Appendix D.

4.5. Sesame REST API support for ad-hoc queries

The clinical evidence service implements fixed web service REST methods that implement standard queries useful for querying the ontology content. Sesame provides flexibility beyond this by providing standard functionality to submit SPARQL queries directly against the server through its own web accessible interface. With knowledge of the clinical evidence ontology structure this allows for any third party application to formulate and submit SPARQL queries for the purposes of ad-hoc querying directly against the server returning results in RDF or XML formats.

4.6. Jersey Java Implementation of REST Services

In order to provide a fully functional evidence service without the need for underlying knowledge of SPARQL or the ontology structure, we have implemented a broad range of parameterised REST based methods that can be used to run standard diagnostic queries against the evidence server. These methods are accessible as structured web based URLs that can return XML, JSON or RDF formats to any third party tool that wishes to use it. The Jersey Java implementation of REST services was selected for this. Appendix D provides an example of using the Sesame API to execute a SPARQL query. Appendix E demonstrates a coded example of defining a REST based URL path to access this functionality and return XML results.

4.7. Clinical evidence service interfaces and methods

All the currently defined REST endpoints are available from the evidence server using the following root URL and adding the appropriate REST path from the tables shown in Appendix F to obtain the required query information:

http://phaedrus.scss.tcd.ie/munnellg/ClinicalEvidenceRESTService/interfaces/query/

The table in Appendix F documents all of the methods that are currently defined by the Clinical Evidence REST Service.
4.8. Data Mining

A data mining platform has been implemented to generate actionable knowledge that from electronic sources of coded primary care data that can be transferred into the clinical evidence ontology and subsequently queried using the REST based query interface.

In particular we have used anonymised primary care data captured from the Netherlands and Malta using the TransHIS EHR(19). This is a freely available open source EHR that was developed as part of the Transition project and captures International Classification of Primary Care 2 (ICPC2) coded values for RFE, diagnostic cues, demographic variables and outcome diagnoses. The TransHIS EHR structure links related clinical encounters in an episode of care (EOC) structure that may occur over multiple time points as shown in figure 14.

![Figure 14 – TransHIS episode of care structure](image)
4.8.1. Data mining steps

The distinct steps in the data mining process are shown in figure 15. These steps involve:

- derivation of association rules linking RFE, diagnostic cues and demographics to a recorded encounter diagnosis from the first encounter of new episodes of care
- calculation of association rule quality measures to determine the relative strength of each rule association derived
- filtering of association rules to allow selection of ‘high-quality’ association rules
- clinical review of selected rules to assess clinical validity of rules with respect to wider clinical body of evidence
- transfer of reviewed ‘high-quality’ association rules to the clinical evidence ontology

![Figure 15 – Distinct steps in the TRANSFoRm data mining process](image)

4.8.2. Association rule derivation using KNIME

An open source data analysis tool, the Konstanz Information Miner (KNIME) has been used to define workflows that pre-process the TransHIS record data and derive
association rules based on ICPC2 codes (an example is shown in figure 16).

Figure 16 – A workflow to derive association rules in KNIME

These rules identify all possible combinations of RFE, diagnostic cues and demographic variables (antecedent variables) that are linked with a recorded diagnostic outcome (consequent variable) as shown in figure 17. After cleaning (first encounter only from new episodes) 393,169 records were loaded into KNIME: 55,821 for Malta and 337,348 for the Netherlands. In total 6,380 association rules were extracted from the data: 2,937 for Malta and 3,443 for the Netherlands.

Figure 17 – Structure of derived association rules
4.8.3. Calculation and validation of quality measures using KNIME

In order to assess the usefulness of each derived association rule it was necessary to define and calculate a number of association rule quality measure using KNIME. These quality measures are standard statistical measures that define the relative strength of each association found (the measures and associated calculations defined are shown in Appendix G). These measures include calculations of positive and negative likelihood ratios that can be used along with prevalence rates to implement Bayesian calculations to calculate the probability of a diagnosis given a presenting RFE.

In order to validate the likelihood ratios generated, the figures calculated from the KNIME tool were cross referenced to previously published case studies using TransHIS that calculated the likelihood ratios with 95% confidence intervals of diagnoses based on a presenting complaint of cough or sadness for both Malta and the Netherlands. All likelihood ratios generated from KNIME tool were successfully cross-referenced with the original work from where the methodology and calculations defined in KNIME were originally obtained from (21).

4.8.4. Clinical association rule review

A web based association rule viewer allows curation and clinical review of all generated association rules from the KNIME tool. The web rule viewer (login and password ‘user’) is available at:

http://156.17.131.239/RulesAssessment/

Selecting the ‘Rule Viewer’ menu item brings up the main rule display. The tool allows filtering of all rules generated from KNIME on any of the coded ICPC2 antecedent variables (RFEs or diagnostic cues/anams, and demographics). The outcome diagnosis being examined can be filtered by selecting a consecutive variable.

In addition thresholds can be set on the defined quality measures to filter based on the strength of the rules required (support defined the number of cases for example).
A scenario name can be entered that identifies a particular snapshot of rules run on a certain date. This allows multiple copies of rules at different points in time to be stored and retrieved based on a scenario label. The labels ‘NL_automated’ and ‘ML_automated’ can be used to retrieve data for Netherlands and Malta respectively as shown in figure 18.

![Web rule viewer selection of rules filters](image)

*Figure 18 – Web rule viewer selection of rules filters*

The ‘filter’ button selects the required rules into the main rule viewer screen in the centre of the screen. By highlighting a particular rule the associated rule descriptions are shown at the bottom of the screen along with 95% confidence intervals. The column headings allow sorting of rules in ascending or descending order based on any of the quality measures as shown in figure 19.

![The main rule viewer details screen](image)

*Figure 19 – The main rule viewer details screen*
Rules of interest can be selected for deployment to the ontology by selecting the ‘deploy’ button beside each rule on the main screen.

4.8.5. Rule review and curation in practice

In this section we discuss an example of clinical review and single variable rule selection from the rule viewer tool. The tool was used to select single variable association rules of interest related to a consecutive diagnosis of U71 (urinary tract infection). Cut offs for a support level > 5 cases and a positive likelihood ratio greater 2 were filtered.

Using the 95% confidence intervals the rules were classified as follows based on a methodology previously used and published with TransHIS data(21):

- Strong predictors, in red were classified as LR+ > 8 with a tight confidence interval (value of observation is greater than width of interval)
- Weak predictors, in green were classified as LR+ >= 2 and LR+ <= 8 with a tight confidence interval (value of observation is greater than width of interval)
- Predictors for exclusion of diagnosis, in yellow were classified as LR+ <= 0.5

Based on these classifications the associations shown in table 5 were selected for deployment to the clinical evidence ontology.
<table>
<thead>
<tr>
<th>ICPC2 Code</th>
<th>ICPC2 Description</th>
<th>LR+ Netherlands</th>
<th>LR+ JAMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>U71</td>
<td>Cystitis/urinary infection, other</td>
<td>185.80 (165.50 – 208.58)</td>
<td>4.0 (2.9 – 5.5)</td>
</tr>
<tr>
<td>U01</td>
<td>Dysuria/painful urination</td>
<td>84.02 (77.87 – 90.67)</td>
<td>1.5 (1.2 – 2.0)</td>
</tr>
<tr>
<td>U27</td>
<td>Fear of urinary disease</td>
<td>44.86 (38.7 – 52.02)</td>
<td></td>
</tr>
<tr>
<td>U02</td>
<td>Urinary frequency/urgency</td>
<td>44.65 (41.59 – 47.94)</td>
<td>1.8 (1.1 – 3.0)</td>
</tr>
<tr>
<td>U06</td>
<td>Haematuria</td>
<td>22.40 (18.63 – 26.94)</td>
<td>2.0 (1.3 – 2.9)</td>
</tr>
<tr>
<td>U29</td>
<td>Urinary symptom/complaint</td>
<td>17.16 (10.8 – 27.29)</td>
<td></td>
</tr>
<tr>
<td>U07</td>
<td>Urine symptom/complaint</td>
<td>16.34 (11.51 – 23.21)</td>
<td></td>
</tr>
<tr>
<td>U01 and U02</td>
<td>Dysuria with Urinary Frequency</td>
<td>193.87 (165.54 – 227.05)</td>
<td></td>
</tr>
<tr>
<td>U05</td>
<td>Urination problems, other</td>
<td>6.35 (4.81 – 8.4)</td>
<td></td>
</tr>
<tr>
<td>U04</td>
<td>Incontinence urine</td>
<td>6.27 (5.04 – 7.80)</td>
<td></td>
</tr>
<tr>
<td>D06</td>
<td>Abdominal pain localized, other</td>
<td>2.59 (2.33 – 2.88)</td>
<td>1.1 (0.9 – 1.4)</td>
</tr>
<tr>
<td>L05</td>
<td>Flank/axilla symptom/complaint</td>
<td>2.08 (1.58 – 2.75)</td>
<td>1.1 (0.9 – 1.4)</td>
</tr>
<tr>
<td>X15</td>
<td>Vaginal symptom/complaint, other</td>
<td>0.48 (0.26 – 0.9)</td>
<td></td>
</tr>
<tr>
<td>X14</td>
<td>Vaginal Discharge</td>
<td>0.13 (0.06 – 0.32)</td>
<td>0.3(0.1-0.9)</td>
</tr>
</tbody>
</table>

Table 5 – UTI predictors for Netherlands data
A comparison of these predictors can be done against a gold standard literature review for comparison such as a JAMA review (22). In this case the identified predictors compare favourably indicating similar strong predictors in the form of urinary frequency, haematuria and dysuria. Also agreeing with the JAMA review, self-labelling by patients was also shown as a strong predictor for UTI and the presence of vaginal discharge was considered a possible excluding factor.

4.8.6. Association rule export

Association rules that have been selected for deployment from the main screen can then be exported in an XML format using the 'rule sender' menu. A list of selected rules is presented and selecting the 'send' option generates an XML output for the selected rules that can be saved onto the clinical evidence server and imported. An example of rules export generation is shown in figure 20. The XML schema for rules is shown in Appendix H.

```
<?xml version="1.0" encoding="UTF-8"?>
<RuleSet RuleSetSource="RulesAssessment.xml" RuleSetDate="Tue Apr 08 17:46:22 CEST 2014" RuleSetComment="Rule exported">
  <Rule RuleScenario="NL_automated" RuleProvenance="Netherlands">
    <RuleAntecedent Description="Dysuria/painful urination" AntecedentType="RFE">U01</RuleAntecedent>
    <RuleConsecutive Description="Cystitis/urinary infection, other" ConsecutiveEncoding="ICPC2">U71</RuleConsecutive>
    <RuleRemark/>
    <RuleScore ScoreType="Support">1014,000</RuleScore>
    <RuleScore ScoreType="Confidence">66,400</RuleScore>
    <RuleScore ScoreType="Lift">28,918</RuleScore>
    <RuleScore ScoreType="Specificity">0.997</RuleScore>
    <RuleScore ScoreType="Sensitivity">0.234</RuleScore>
    <RuleScore ScoreType="LR+">94,023</RuleScore>
    <RuleScore ScoreType="LR-">0.768</RuleScore>
    <RuleScore ScoreType="Odds">109,426</RuleScore>
  </Rule>
  <Rule RuleScenario="NL_automated" RuleProvenance="Netherlands">
    <RuleAntecedent Description="Flank/axilla symptom/complaint" AntecedentType="RFE">L05</RuleAntecedent>
    <RuleConsecutive Description="Cystitis/urinary infection, other" ConsecutiveEncoding="ICPC2">U71</RuleConsecutive>
    <RuleRemark/>
    <RuleScore ScoreType="Support">57,000</RuleScore>
    <RuleScore ScoreType="Confidence">4,700</RuleScore>
    <RuleScore ScoreType="Lift">2,030</RuleScore>
    <RuleScore ScoreType="Specificity">0.997</RuleScore>
    <RuleScore ScoreType="Sensitivity">0.007</RuleScore>
    <RuleScore ScoreType="LR+">2,080</RuleScore>
    <RuleScore ScoreType="LR-">0.997</RuleScore>
    <RuleScore ScoreType="Odds">2,080</RuleScore>
  </Rule>
</RuleSet>
```

Figure 20 – Generation of association rules export XML from the web rule sender
4.8.7. **Ontology Rule Import and Update**

The Ontology Updater tool is a command line tool developed as part of the TRANSFoRm project for the purpose of processing the generated XML rules and updating the clinical evidence ontology with evidence. XML files are saved into a specified directory on the evidence server and a command line tool is executed or can be scheduled to run periodically:

```
java -jar ontology_updater.jar [path to update file]
```

First, each of a rules antecedents are examined to check if they describe a reason for encounter as indicated by the AntecedentType attribute in the XML document. In the event that an RFE is identified, the tool will check the clinical evidence ontology to ensure that it contains the item based on the associated code binding. If not, then a new RFE is generated in the ontology and populated with the data from the antecedent. If the reason for encounter is found to exist, then the process simply continues scanning antecedents for RFEs until it reaches the end of the list for the given rule.

Similarly to RFEs in the antecedents, the consecutive is treated as an object in the clinical evidence ontology. Thus, if it cannot be found in the ontology based on its code binding, then the tool will generate and populate it using the data in the XML file. Once this is done, the tool is ready to update the quantification.

A quantification is defined by a unique combination of antecedents, consecutives and a country. If a quantification with the precise combination of these elements is not found in the ontology, then a new one is generated with a random GUID as an identifier. This quantification is then linked with the relevant RFE objects and differential diagnosis objects in the ontology.

Upon completion of this task, the tool checks if there are any other rules to be processed. If so, the operation repeats. Otherwise, the tool will terminate its execution. An illustration of the update process is given in the figure 21.
Figure 21 – Flowchart for ontology update of evidence from XML export rule files
4.8.8. Rule Availability through web service

Rules imported into the ontology can be accessed using the appropriate REST endpoint to retrieve the imported quantifications from the evidence service as shown in figure 22.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<quantificationCollection>
  - <quantification>
    - <context>
      <contextType>Sex</contextType>
      <contextValue>Female</contextValue>
    </context>
    - <context>
      <contextType>Country</contextType>
      <contextValue>Netherlands</contextValue>
    </context>
    - <value>
      <quantificationType>NegativeLR</quantificationType>
      <value>0.99</value>
    </value>
    - <value>
      <quantificationType>PositiveLR</quantificationType>
      <value>70.95</value>
    </value>
    - <value>
      <quantificationType>Confidence</quantificationType>
      <value>62.50</value>
    </value>
    - <value>
      <quantificationType>EvidenceSource</quantificationType>
      <value>a_130613</value>
    </value>
    - <value>
      <quantificationType>Support</quantificationType>
      <value>65</value>
    </value>
  <reason>AbdominalPainLocalisedOtherRFE</reason>
  <reason>DysuriaRFE</reason>
  <quantifyDiagnosis>UrinaryTractInfection</quantifyDiagnosis>
  <quantifyID>QuantifyRFE1</quantifyID>
</quantificationCollection>
```

*Figure 22 – REST evidence output of imported evidence rules*

4.9. Evidence service client description to implement decision support tool requirements

As part of work being done in WP5 a decision support tool has been implemented that consumes the clinical evidence service produced by WP4. The detailed requirements specification for this tool are defined in a separate deliverable and not
described in detail here. The decision support tool interface can be considered to be a third party tool that consumes the evidence service. It is linked to a third party EHR (Vision 3) and can be called as a separate decision support window sitting on top it (as shown in figure 23).

![Figure 23 – The WP5 decision support tool window on top of the Vision 3 EHR](image)

The tool is used on a primary care practitioners computer and the interactions between the decision support tool (integrated with an EHR), the ontology and primary care practitioner have been described at a high level as part of the two phase diagnostic process (in figure 4 previously).

We can expand on this workflow describing the interactions between the EHR (using the CDIM model based data connector), the evidence service (using a REST client DLL) and the decision support tool itself. The initial trigger for the process is the entering of a patient RFE which results in an initial list of differential diagnoses to
consider, filtered based on the initial patient details extracted from the EHR. An iterative process of cue gathering takes place through the decision support tool. This patient evidence set is submitted to the evidence service resulting in a filtering process of differentials to consider that is returned back to the CDSS interface. This workflow process is summarised in figure 24.
Figure 24 – Workflow and interactions of decision support components
The decision support tool is implemented using C#. In order to implement the diagnostic workflow an evidence client DLL has been created and can be linked into the decision support interface. This DLL implements a number of classes and methods that provide C# wrappers for the REST based web service calls previously described. For a full list of the methods and C# classes implemented refer to the following on-line documentation or the summaries in Appendix I onwards:

http://phaedrus.scss.tcd.ie/munnellg/

The most important class used by the decision support tool is the ClinicalEvidenceWebInterface class which provides the methods that can be called by the decision support interface itself. These methods are summarised in Appendix J. In addition a PatientCase class accepts an XML serialisable C# PatientEvidenceSet that allows the CDSS interface to submit or update the exchange of a set of patient evidence obtained from a combination of the EHR and the decision support interface, using the ‘UpdatePatientCase’ method. This method returns a ranked list of differentials to consider listed as ‘most likely’, ‘not to miss’ and ‘others’. The PatientCase class and a portion of the C# PatientEvidence class are shown Appendix K and L.
5. Conclusions

The generation and representation of actionable clinical knowledge from electronic sources of patient data is a challenging task. The platform and associated models developed as part of TRANSFoRm have demonstrated that it is feasible to generate and make openly available clinically meaningful knowledge that complements available literature based evidence for diagnostic purposes. This is achieved by quantifying the actual empirical contribution of each piece of clinical evidence as a basis for informing and reaching a particular diagnosis. This is important if we wish to deliver on our goal of improving patient safety outcomes and informing clinical decisions by making them truly more ‘evidence-based’.

The open models and service oriented architecture that we have developed are flexible enough to allow any third party tool using these open standards to ask diagnostic questions using standardised URL structures and to write their own client implementation based on the REST endpoints that have already been provided. Flexibility is further enhanced through the provision of ad-hoc querying against the Sesame server using its own REST interface for submitting SPARQL queries directly against the server. The service provides for semantic interoperability by defining ontology concepts that allow for terminological bindings to be associated with ontology content.

The design of the ontology is general and flexible enough to be populated dynamically from clinical content based on other coded EHR sources and need not be limited to just the clinical scenarios defined by TRANSFoRm. The models developed here are generalisable enough that they could be extended to include additional clinical content beyond the three presenting clinical conditions presented as part of this work, or to other clinical contexts in addition to just primary care data. The challenge is to identify suitable patient data sources in large enough volumes from which evidence of a sufficiently high level of confidence can be generated and represented. The open implementation and integration of evidence from empirically measured EHR sources described here is consistent with the complex goals of
developing a learning healthcare system. The potential for using large collections of aggregated electronic patient data to support these goals appears underutilised at present and requires further research and development.

6. Future Work

Data mining has been carried out to produce association rules based on the first encounter of new episode of care. Subsequent work will look at doing analysis at further encounters and timing points in addition to the first encounter of an episode of care. This should allow for better analysis of more complex conditions which may take more than one encounter to formulate a definitive diagnosis. This could also allow for time-based concepts to be added to the ontology by comparing rules generated for conditions at different points in time.

Vocabulary binding has been done using manual configuration of the evidence ontology which is not sustainable in the longer term. Future work to be done in conjunction with the University of Warwick and the Mayo Clinic will provide tighter integration with the TRANSFoRm vocabulary service. This will allow for a vocabulary enrichment process by using pivot terminology bindings manually defined in the ontology to identify and populate mappings to other terminologies that can be dynamically populated into the ontology content.
7. References


15. Web Ontology Language Standard Homepage. Available from:
http://www.w3.org/2004/OWL/.


19. Transition Project Homepage.


### 8. Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFO</td>
<td>Basic Formal Ontology</td>
</tr>
<tr>
<td>CDIM</td>
<td>Clinical Data Integration Model</td>
</tr>
<tr>
<td>CDSS</td>
<td>Clinical Decision Support System</td>
</tr>
<tr>
<td>DLL</td>
<td>Dynamic Linked Library</td>
</tr>
<tr>
<td>EHR</td>
<td>Electronic Health Record</td>
</tr>
<tr>
<td>EMR</td>
<td>Electronic Medical Record</td>
</tr>
<tr>
<td>ICPC2</td>
<td>International Classification of Primary Care Version 2</td>
</tr>
<tr>
<td>ICD10</td>
<td>International Classification of Diseases Version 10</td>
</tr>
<tr>
<td>JSON</td>
<td>Javascript Object Notation</td>
</tr>
<tr>
<td>KNIME</td>
<td>The Konstanz Information Miner</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>REST</td>
<td>Representational state transfer</td>
</tr>
<tr>
<td>RFE</td>
<td>Reason for Encounter</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol and RDF Query Language</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SNOMED</td>
<td>The Systematized Nomenclature of Medicine</td>
</tr>
<tr>
<td>UMLS</td>
<td>Unified Medical Language System</td>
</tr>
<tr>
<td>UTI</td>
<td>Urinary Tract Infection</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
9. Appendix A – WP2 patient safety use case differentials

Reason for Encounter - Abdominal pain

<table>
<thead>
<tr>
<th>No.</th>
<th>Differential Diagnoses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Urinary tract infection</td>
</tr>
<tr>
<td>2</td>
<td>Pyelonephritis</td>
</tr>
<tr>
<td>3</td>
<td>Pelvic inflammatory disease</td>
</tr>
<tr>
<td>4</td>
<td>Gastroenteritis</td>
</tr>
<tr>
<td>5</td>
<td>Appendicitis</td>
</tr>
<tr>
<td>6</td>
<td>Crohn’s disease</td>
</tr>
<tr>
<td>7</td>
<td>Ectopic Pregnancy</td>
</tr>
<tr>
<td>8</td>
<td>Irritable bowel syndrome</td>
</tr>
<tr>
<td>9</td>
<td>Ovarian cancer</td>
</tr>
<tr>
<td>10</td>
<td>Renal stone</td>
</tr>
<tr>
<td>11</td>
<td>Uterine fibroids</td>
</tr>
<tr>
<td>12</td>
<td>Diabetes</td>
</tr>
<tr>
<td>13</td>
<td>Hepatitis</td>
</tr>
<tr>
<td>14</td>
<td>Colon cancer</td>
</tr>
<tr>
<td>15</td>
<td>Renal cancer</td>
</tr>
<tr>
<td>16</td>
<td>Anxiety and mood disorders</td>
</tr>
<tr>
<td>17</td>
<td>Diverticular disease</td>
</tr>
<tr>
<td>18</td>
<td>Endometrial cancer</td>
</tr>
<tr>
<td>19</td>
<td>Liver tumour</td>
</tr>
<tr>
<td>20</td>
<td>Pancreatic cancer</td>
</tr>
<tr>
<td>21</td>
<td>Stomach cancer</td>
</tr>
<tr>
<td>22</td>
<td>GORD / peptic ulceration</td>
</tr>
</tbody>
</table>
### Reason for Encounter - Chest pain

<table>
<thead>
<tr>
<th>No.</th>
<th>Differentials to be considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pneumothorax</td>
</tr>
<tr>
<td>2</td>
<td>Pneumonia</td>
</tr>
<tr>
<td>3</td>
<td>Tuberculosis</td>
</tr>
<tr>
<td>4</td>
<td>Angina</td>
</tr>
<tr>
<td>5</td>
<td>Mesothelioma</td>
</tr>
<tr>
<td>6</td>
<td>Pulmonary Embolism</td>
</tr>
<tr>
<td>7</td>
<td>Aortic valve regurgitation</td>
</tr>
<tr>
<td>8</td>
<td>Aortic valve stenosis</td>
</tr>
<tr>
<td>9</td>
<td>LRTI, bacterial</td>
</tr>
<tr>
<td>10</td>
<td>Musculoskeletal chest pain</td>
</tr>
<tr>
<td>11</td>
<td>Hyperthyroidism</td>
</tr>
<tr>
<td>12</td>
<td>Rib fracture</td>
</tr>
<tr>
<td>13</td>
<td>Chest wall injury</td>
</tr>
<tr>
<td>14</td>
<td>Congestive cardiac failure (CCF)</td>
</tr>
<tr>
<td>15</td>
<td>COPD</td>
</tr>
<tr>
<td>16</td>
<td>LRTI, Viral</td>
</tr>
<tr>
<td>17</td>
<td>Lung cancer</td>
</tr>
<tr>
<td>18</td>
<td>Diabetes</td>
</tr>
<tr>
<td>19</td>
<td>Asthma</td>
</tr>
<tr>
<td>20</td>
<td>Cardiac arrhythmia</td>
</tr>
<tr>
<td>21</td>
<td>Anxiety and mood disorders</td>
</tr>
<tr>
<td>22</td>
<td>Herpes zoster</td>
</tr>
<tr>
<td>23</td>
<td>GORD / Peptic ulceration</td>
</tr>
</tbody>
</table>
## Reason for Encounter - Dyspnoea

<table>
<thead>
<tr>
<th>No.</th>
<th>Differentials to be considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cor pulmonale (right heart failure)</td>
</tr>
<tr>
<td>2</td>
<td>COPD exacerbation / COPD</td>
</tr>
<tr>
<td>3</td>
<td>Acute Bronchitis</td>
</tr>
<tr>
<td>4</td>
<td>Childhood Asthma</td>
</tr>
<tr>
<td>5</td>
<td>LRTI, bacterial</td>
</tr>
<tr>
<td>6</td>
<td>LRTI, viral</td>
</tr>
<tr>
<td>7</td>
<td>Musculoskeletal chest pain</td>
</tr>
<tr>
<td>8</td>
<td>Tonsillitis</td>
</tr>
<tr>
<td>9</td>
<td>URTI</td>
</tr>
<tr>
<td>10</td>
<td>Angina</td>
</tr>
<tr>
<td>11</td>
<td>Aortic valve regurgitation</td>
</tr>
<tr>
<td>12</td>
<td>Aortic valve stenosis</td>
</tr>
<tr>
<td>13</td>
<td>Atrial fibrillation</td>
</tr>
<tr>
<td>14</td>
<td>Congestive cardiac failure (CCF)</td>
</tr>
<tr>
<td>15</td>
<td>Allergic reactions</td>
</tr>
<tr>
<td>16</td>
<td>Lung cancer</td>
</tr>
<tr>
<td>17</td>
<td>Mitral valve stenosis</td>
</tr>
<tr>
<td>18</td>
<td>Mitral valve regurgitation</td>
</tr>
<tr>
<td>19</td>
<td>Myocardial infarction</td>
</tr>
<tr>
<td>20</td>
<td>Pulmonary embolism</td>
</tr>
<tr>
<td>21</td>
<td>Tuberculosis</td>
</tr>
<tr>
<td>22</td>
<td>Anaemia</td>
</tr>
<tr>
<td>23</td>
<td>Lymphoma</td>
</tr>
<tr>
<td>24</td>
<td>Multiple myeloma</td>
</tr>
<tr>
<td>25</td>
<td>Obstructive sleep apnea</td>
</tr>
</tbody>
</table>
10. Appendix B – Search criteria for WT 4.1 reviews of literature

The search was carried out focussing on clinical papers containing information relating to the following categories:

- Risk Factors
- Signs and symptoms
- Clinical Prediction Rules
- Prevalence Information

Step 1
An initial search was carried out for the target diagnosis on the HRB Centre for Primary Care Research database of clinical documents which was created to identify clinical papers containing clinical prediction rules or treatment algorithms from a selection of 28 primary care focussed medical journals. This database contains 70,000 clinical papers covering 1980-2008.

An example of the search string used for creating this database from pubmed for the period 2008 is:

AND decision[All Fields]) OR (validation[All Fields] AND rule[All Fields]) OR "validation score"[All Fields] OR (derivation[All Fields] AND validation[All Fields]) OR (("sensitivity and specificity"[MeSH Terms] OR ("sensitivity"[All Fields] AND "specificity"[All Fields]) OR "sensitivity and specificity"[All Fields]) OR "sensitivity"[All Fields] AND "specificity"[All Fields]) AND ("sensitivity and specificity"[MeSH Terms] OR ("sensitivity"[All Fields] AND "specificity"[All Fields]) OR "sensitivity and specificity"[All Fields] OR "specificity"[All Fields])) OR ("diagnosis"[Subheading] OR "diagnosis"[All Fields] OR "symptoms"[All Fields] OR "diagnosis"[MeSH Terms] OR "symptoms"[All Fields]) AND ("diagnosis"[Subheading] OR "diagnosis"[All Fields] OR "signs"[All Fields] OR "diagnosis"[MeSH Terms] OR "signs"[All Fields])) AND "humans"[MeSH Terms]

Step 2
The following evidence based websites were searched for articles relating to the diagnosis:

- JAMA Evidence
- British Medical Journal Clinical Evidence
- The Cochrane Medical Reviews
- NHS Clinical Knowledge Summaries
- NICE Guidelines
- SIGN Guidelines

Step 3
A search of pubmed and embase. An example search string used was:

‘tuberculosis'/de AND 'diagnosis':ab,ti AND ([article]/lim OR [review]/lim) AND [humans]/lim AND [english]/lim

References related to retrieved articles were systematically searched.
11. Appendix C – Sample clinical evidence ontology design competency questions

<table>
<thead>
<tr>
<th>Question No</th>
<th>Competency Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>For a particular coded reason for encounter, provide a list of differential diagnoses that should be considered for investigation for that RFE</td>
</tr>
<tr>
<td>2</td>
<td>For a patient presenting with a defined list of signs, symptoms, risk factors and clinical test results, what are the initial differential diagnoses that should be considered for investigation</td>
</tr>
<tr>
<td>3</td>
<td>For a particular diagnosis, present a list of the key clinical diagnostic cues that should be considered as clinical evidence to support the inclusion or exclusion of that diagnosis</td>
</tr>
<tr>
<td>4</td>
<td>When diagnostic cues associated with a diagnosis for investigation are compared to the diagnostic cues presented by a particular patient instance, what is the probability of that diagnosis occurring for that patient</td>
</tr>
<tr>
<td>5</td>
<td>Describe the population characteristics from which the clinical criteria to be applied to diagnostic cues indicative of a particular clinical diagnosis were derived</td>
</tr>
<tr>
<td>6</td>
<td>For a particular diagnostic cue, what are the potential diagnoses that are indicated by clinical criteria applied to that particular diagnostic cue</td>
</tr>
</tbody>
</table>
12. Appendix D – Example of SPARQL query execution using Sesame API

```java
public static EvidenceDiagnosis query(String rfe)
throws OpenRDFException {
    // Create evidence diagnosis object
    EvidenceDiagnosis evdDiag = new EvidenceDiagnosis();

    // Create array for holding diagnosis list returned from SPARQL query
    ArrayList<EvidenceDiagnosis> nameList = new ArrayList<EvidenceDiagnosis>();

    // Create Sesame repository pointing to the clinical evidence/ repository ID
    Repository myRepository = new HTTPRepository(Constants.sesameServer,
                                                    Constants.repositoryID);

    try {
        // Initialise the Sesame repository connection
        myRepository.initialize();
    } catch (RepositoryException e1) {
        e1.printStackTrace();
    }

    try {
        // Establish the connection to the repository connection
        RepositoryConnection con = myRepository.getConnection();

        try{
            // Create SPARQL query string to retrieve differential diagnoses based on presenting RFE
            String queryString = Constants.queryString +
                "SELECT DISTINCT ?anyDifferentialDiagnosis
                WHERE{evd:" + rfe + " evd:hasDifferentialDiagnosis
                     ?anyDifferentialDiagnosis.}";

            System.out.println(queryString);
        }
    }
}
```
// Execute the SPARQL query against the evidence repository

TupleQuery tupleQuery = con.prepareTupleQuery(QueryLanguage.SPARQL, queryString);

//System.out.println(tupleQuery);
TupleQueryResult result = tupleQuery.evaluate();

//System.out.println(result);
System.out.println("Tuple Query result");

try {
    // Iterate through each differential diagnosis and extract diagnosis properties
    while (result.hasNext()) {
        BindingSet bindingSet = result.next();
        Value anyDiffDiag = bindingSet.getValue("anyDifferentialDiagnosis");
        String diffDiagnosis = anyDiffDiag.stringValue();
        diffDiagnosis = diffDiagnosis.substring(66);
        evdDiag.setEvidenceDiagnosis(diffDiagnosis);
        // Add the diagnosis object to the evidence diagnosis list to be returned
        nameList.add(evdDiag);
    }
}

finally {
    // Close SPARQL result set
    result.close();
}

finally {
    // Close Sesame connection
    con.close();
}

catch (OpenRDFException e) {
    System.err.println("Main Exceptions");
}
}  
}  
return evDiag;

}//end of query( ) function
13. Appendix E – Example of REST end point path definition

```java
package web;

import java.util.ArrayList;

// Import java REST libraries
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

// Import Sesame libraries
import org.openrdf.OpenRDFException;

// Import evidence service business objects
import business.QueryRFEBO;
import model.EvidenceDiagnosis;

// Set the REST based query path used to execute the underlying SPARQL // query
// The query takes one parameter, a specified RFE id and returns an XML // representation of the evidence diagnosis list associated with the RFE @Path("/query/differentials/{rfe}")

public class Rfe {

    // Specify the return type of the REST query, can be XML or JSON // format along with the input parameter taken from the REST url // which is the RFE in this case

    @GET
    @Produces({MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON})
    public EvidenceDiagnosis getXML(@PathParam("rfe") String rfe) throws OpenRDFException {

        // Create evidence diagnosis list object
        EvidenceDiagnosis ed = new EvidenceDiagnosis();

        // Execute the underlying SPARQL command based on the passed // RFE returning the XML representation for display at the // REST endpoint defined above
        ed = QueryRFEBO.getXML(rfe);
        return ed;
    }
}
```
## 14. Appendix F – Clinical Evidence Service REST endpoints

<table>
<thead>
<tr>
<th>REST URL Path</th>
<th>Data Obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>interfaces/query/rfes</td>
<td>Return a list of all reasons for encounter defined in the ontology</td>
</tr>
<tr>
<td>interfaces/query/differentials</td>
<td>Return a list of all differential diagnoses defined in the ontology</td>
</tr>
<tr>
<td>interfaces/query/cues</td>
<td>Return a list of all diagnostic cues defined in the ontology</td>
</tr>
<tr>
<td>interfaces/query/signs</td>
<td>Return a list of all diagnostic signs defined in the ontology</td>
</tr>
<tr>
<td>interfaces/query/symptoms</td>
<td>Return a list of all diagnostic symptoms defined in the ontology</td>
</tr>
<tr>
<td>interfaces/query/tests</td>
<td>Return a list of all diagnostic tests defined in the ontology</td>
</tr>
<tr>
<td>interfaces/query/risks</td>
<td>Return a list of all diagnostic risk factors defined in the ontology</td>
</tr>
<tr>
<td>interfaces/query/rfes/quantifications/{rfe}</td>
<td>Return a list of all quantifications which are linked to the specific RFE with ontology identifier {rfe} e.g. AbdominalPainLocalisedOtherRFE</td>
</tr>
<tr>
<td>interfaces/query/rfes/differentials/{rfe}</td>
<td>Return a list of all differential diagnoses which are linked to the specific RFE with ontology identifier {rfe} e.g. AbdominalPainLocalisedOtherRFE</td>
</tr>
<tr>
<td>interfaces/query/rfes/cues/{rfe}</td>
<td>Return a list of all diagnostic cues which are linked to the specific RFE with ontology identifier {rfe} e.g. AbdominalPainLocalisedOtherRFE</td>
</tr>
<tr>
<td>interfaces/query/rfes/signs/{rfe}</td>
<td>Return a list of all diagnostic signs which are linked to the specific RFE with ontology identifier {rfe} e.g. AbdominalPainLocalisedOtherRFE</td>
</tr>
<tr>
<td>interfaces/query/rfes/symptoms/{rfe}</td>
<td>Return a list of all diagnostic symptoms which are linked to the specific RFE with ontology identifier {rfe} e.g. AbdominalPainLocalisedOtherRFE</td>
</tr>
<tr>
<td>interfaces/query/rfes/tests/{rfe}</td>
<td>Return a list of all diagnostic tests which are linked to the specific RFE with ontology identifier {rfe} e.g. AbdominalPainLocalisedOtherRFE</td>
</tr>
<tr>
<td>interfaces/query/rfes/risks/{rfe}</td>
<td>Return a list of all diagnostic risk factors which are linked to the specific RFE with ontology identifier {rfe} e.g. AbdominalPainLocalisedOtherRFE</td>
</tr>
<tr>
<td>interfaces/query/differentials/rfes/{diff}</td>
<td>Return a list of all rfes which are linked to the differential diagnosis with ontology identifier {diff} e.g. UrinaryTractInfection</td>
</tr>
<tr>
<td>URL Path</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>interfaces/query/differentials/cues/{diff}</td>
<td>Return a list of all diagnostic cues which are linked to the differential</td>
</tr>
<tr>
<td></td>
<td>diagnosis with ontology identifier {diff} e.g. UrinaryTractInfection</td>
</tr>
<tr>
<td>interfaces/query/differentials/signs/{diff}</td>
<td>Return a list of all diagnostic signs which are linked to the differential</td>
</tr>
<tr>
<td></td>
<td>diagnosis with ontology identifier {diff} e.g. UrinaryTractInfection</td>
</tr>
<tr>
<td>interfaces/query/differentials/symptoms/{diff}</td>
<td>Return a list of all diagnostic symptoms which are linked to the differential</td>
</tr>
<tr>
<td></td>
<td>diagnosis with ontology identifier {diff} e.g. UrinaryTractInfection</td>
</tr>
<tr>
<td>interfaces/query/differentials/tests/{diff}</td>
<td>Return a list of all diagnostic tests which are linked to the differential</td>
</tr>
<tr>
<td></td>
<td>diagnosis with ontology identifier {diff} e.g. UrinaryTractInfection</td>
</tr>
<tr>
<td>interfaces/query/differentials/risks/{diff}</td>
<td>Return a list of all diagnostic risk factors which are linked to the</td>
</tr>
<tr>
<td></td>
<td>differential diagnosis with ontology identifier {diff} e.g.</td>
</tr>
<tr>
<td></td>
<td>UrinaryTractInfection</td>
</tr>
<tr>
<td>interfaces/query/differentials/redflags/{diff}</td>
<td>Return a list of all diagnostic red flag cues which are linked to the</td>
</tr>
<tr>
<td></td>
<td>differential diagnosis with ontology identifier {diff} e.g.</td>
</tr>
<tr>
<td></td>
<td>OvarianCancer</td>
</tr>
<tr>
<td>interfaces/query/cues/rfes/{cue}</td>
<td>Return a list of all rfes which are linked to the diagnostic cue with</td>
</tr>
<tr>
<td></td>
<td>ontology identifier {cue} e.g. Fever</td>
</tr>
<tr>
<td>interfaces/query/cues/differentials/{cue}</td>
<td>Return a list of all differential diagnoses which are linked to the</td>
</tr>
<tr>
<td></td>
<td>diagnostic cue with ontology identifier {cue} e.g. Fever</td>
</tr>
</tbody>
</table>
## 15. Appendix G - Definitions of association rules quality measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Description</th>
<th>Formulae</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apriori or prior probability (AP)</td>
<td>Probability of having positive the condition expressed in the consequent</td>
<td>( \frac{N^+_c}{N_c} )</td>
</tr>
<tr>
<td></td>
<td>Variables characterization, to be computed per each variable in the antecedent</td>
<td>( \frac{N^+_c + a}{N^+_c} )</td>
</tr>
<tr>
<td></td>
<td>Posterior probability (PP)</td>
<td>( \frac{N^+_c + u}{N^+_c} )</td>
</tr>
<tr>
<td></td>
<td>Likelihood ratio (LRx)</td>
<td>( \frac{N^+_c + v}{N^+_c} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \frac{N^+_c + u}{N^+_c} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( LR^+ = \frac{Sen}{1 - Spe} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( LR^- = \frac{1 - Sen}{Spe} )</td>
</tr>
<tr>
<td>Support (sup)</td>
<td>Proportion of transactions which contain the itemset ( A \cup C )</td>
<td>( \frac{N^+_c + a + \hat{C}}{N} )</td>
</tr>
<tr>
<td>Lift (or interest) (Lift)</td>
<td>How many times more often ( A ) and ( C ) occur together than expected if they were statistically independent ( (lift) more times appears complain=AbdominalPain, ..., PrimaryCare = ClinicalEnvironment associated to Diagnosis=UrinaryTractInfection, that we could expect if they were independent events.)</td>
<td>( \text{conf}(A \rightarrow C) / \text{supp}(C) = \frac{N^+_c}{N_c} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{conf}(C \rightarrow A) / \text{supp}(A) = \frac{N^+_c}{N_c} )</td>
</tr>
<tr>
<td>Confidence(or strength) (Conf)</td>
<td>Probability of seeing the rule’s consequent (C) under the condition that the transactions also contain the antecedent (A) ( (\text{conf}) percentage of the patients having complain=AbdominalPain, ..., PrimaryCare = ClinicalEnvironment, has been diagnosed with UrinaryTractInfection).</td>
<td>( \text{supp}(A \rightarrow C) / \text{supp}(C) = \frac{N^+_c}{N^+_c + a + \hat{C}} )</td>
</tr>
<tr>
<td>Conviction (Conv)</td>
<td>Compares the probability that A appears without C if they were dependent with the actual frequency of the appearance of ( A \text{without} C ). In that respect it is similar to lift, however, it contrast to lift it is a directional ( (\text{conv}(A \rightarrow C) \neq \text{conv}(C \rightarrow A)) ) measure since it also uses the information of the absence of the consequent</td>
<td>( \frac{(1 - \text{supp}(C)) / (1 - \text{conf}(A \rightarrow C))}{N_c} )</td>
</tr>
<tr>
<td>Sensitivity (Sen)</td>
<td>Probability of the antecedent being positive given the consequent is positive</td>
<td>( \frac{N^+_c}{N^+_c + a^-} )</td>
</tr>
<tr>
<td>Specificity (Spe)</td>
<td>Probability of the antecedent being negative given the consequent is negative</td>
<td>( \frac{N^+_c}{N^+_c + a^-} )</td>
</tr>
</tbody>
</table>
A = association rule antecedent

C = association rule consequent

Rule structure is A \rightarrow C

Nc + = number of occurrences of patients with a selected condition as expressed in the rule consequent C

Nc - = number of occurrences of patients without a selected condition as expressed in the rule consequent C

N = total number of patients

V^x = a selected diagnostic cue or variable V expressed in the rule antecedent
16. Appendix H – XML schema for association rules

```xml
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Rule">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="RuleAntecedent" maxOccurs="unbounded"/>
        <xs:element ref="RuleConsecutive"/>
        <xs:element ref="RuleRemark" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="RuleScore" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="RuleScenario" type="xs:string" use="required"/>
      <xs:attribute name="RuleProvenance" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:element name="RuleSet">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Rule" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="RuleSetID" type="xs:string" use="required"/>
      <xs:attribute name="RuleSetDate" type="xs:string" use="required"/>
      <xs:attribute name="RuleSetSource"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
<xs:complexType name="RuleSet">
  <xs:complexContent>
    <xs:extension base="xs:string">
      <xs:attribute name="RuleSetComment" type="xs:string" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="RuleScore">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:string">
        <xs:attribute name="ScoreType" type="xs:string" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<xs:element name="RuleRemark" type="xs:string"/>

<xs:element name="RuleAntecedent">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Description" type="xs:string" use="required"/>
        <xs:attribute name="AntecedentType" type="xs:string" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>

<xs:element name="RuleConsecutive">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:string">
        <xs:attribute name="Description" type="xs:string" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:attribute name="ConsecutiveEncoding"
type="xs:string"
use="required"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:schema>
17. Appendix I– C# evidence client core classes for implementing decision support functions

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ClinicalEvidenceOntologyObject</strong></td>
<td>An abstract class which is at the root of all objects extracted from the clinical evidence REST service. At present, it simply handles serialization of objects back to XML</td>
</tr>
<tr>
<td><strong>ClinicalEvidenceWebInterface</strong></td>
<td>The interface to the TRANSFoRm clinical evidence web service. Defines a number of static methods which may be called to execute queries across the ontology that is stored by the REST service.</td>
</tr>
<tr>
<td><strong>CodeBinding</strong></td>
<td>A code which can be used to specifically identify a particular RFE, cue or diagnosis</td>
</tr>
<tr>
<td><strong>Cue</strong></td>
<td>A cue is data that is recorded by the clinician in session with the patient. It could be a symptom or perhaps some observation of the patient’s lifestyle. Ultimately, a cue will be used to identify a probable diagnosis. In future implementations, cue will have a number of child classes which refer to specific types of cues such as symptom, risk, sign, etc. For now, these children are dissociated classes.</td>
</tr>
<tr>
<td><strong>CueDifferentialsList</strong></td>
<td>An indexable collection of evidence diagnosis. Required for serialisation, but will be replaced by EvidenceDiagnosisList or something more general at a later date</td>
</tr>
<tr>
<td><strong>CueList</strong></td>
<td>A serializable list of EvidenceDiagnosis objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>Demographic</strong></td>
<td>A demographic describes some contextual aspect of a patient's condition, such as age, weight, country, etc.</td>
</tr>
<tr>
<td><strong>DifferentialCueMap</strong></td>
<td>Maintains a mapping between differential identifiers and lists of cues associated with a particular differential</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>DifferentialCueMap.DiffCueItem</strong></td>
<td>A temporary class. This is required to serialize and deserialize a dictionary to and from XML as C# does not support this operation natively. This class represents a single key value pair in the dictionary.</td>
</tr>
<tr>
<td><strong>EvidenceDiagnosis</strong></td>
<td>An evidence diagnosis is a specific diagnosis of a patient's condition based on an examination of the various cues and rfe's that are present</td>
</tr>
<tr>
<td><strong>EvidenceDiagnosisList</strong></td>
<td>A serializable list of EvidenceDiagnosis objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>PatientCase</strong></td>
<td>A class which handles a single patient's case. Aggregates information about the patient's symptoms and uses them to generate lists of potential diagnoses</td>
</tr>
<tr>
<td><strong>PatientEvidenceSet</strong></td>
<td>A collection of evidence gathered about a patient's condition</td>
</tr>
<tr>
<td><strong>Quantification</strong></td>
<td>A quantification is a measure of how strongly related a RFE is to a particular differential diagnosis</td>
</tr>
<tr>
<td><strong>QuantificationContext</strong></td>
<td>The context of the patient described by the quantification, e.g. sex, age, country of origin</td>
</tr>
<tr>
<td><strong>QuantificationList</strong></td>
<td>A serializable list of Quantification objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>QuantificationValue</strong></td>
<td>A specific value that is associated with a quantification</td>
</tr>
<tr>
<td><strong>Rfe</strong></td>
<td>A reason for encounter (RFE) is the reason for an initial interaction between the patient and the clinician. It is the reason why the patient presented themselves to the clinic for examination.</td>
</tr>
<tr>
<td><strong>RfeList</strong></td>
<td>A serializable list of Rfe objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>Risk</strong></td>
<td>A risk describes some condition of the patient's lifestyle or history which may make them more susceptible to a particular disease, e.g. family history of heart disease, long term smoker, etc.</td>
</tr>
<tr>
<td><strong>RiskList</strong></td>
<td>A serializable list of Risk objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>Sign</strong></td>
<td>A sign is a cue which is measured by the clinician, e.g. temperature</td>
</tr>
<tr>
<td><strong>SignList</strong></td>
<td>A serializable list of Sign objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>Symptom</strong></td>
<td>A symptom is a cue which the clinician observes, e.g. pale skin, cough, etc. A symptom may be the same as a reason for encounter, however, in the knowledge ontology, the two are not linked.</td>
</tr>
<tr>
<td><strong>SymptomList</strong></td>
<td>A serializable list of Symptom objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>Test</strong></td>
<td>A test is a type of cue which describes how to check for a particular diagnosis. For example, it may suggest a dipstick urinalysis test in order to check for the presence of the disease.</td>
</tr>
<tr>
<td><strong>TestList</strong></td>
<td>A serializable list of Test objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td><strong>Demographic</strong></td>
<td>A demographic describes some contextual aspect of a patient's condition, such as age, weight, country, etc.</td>
</tr>
<tr>
<td><strong>DifferentialCueMap</strong></td>
<td>Maintains a mapping between differential identifiers and lists of cues associated with a particular differential.</td>
</tr>
</tbody>
</table>
| **DifferentialCueMap.DiffCueItem** | A temporary class. This is required to serialize and deserialize a dictionary to and from XML as C# does not support this operation natively. This class represents a single key value pair in the dictionary.
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EvidenceDiagnosis</td>
<td>An evidence diagnosis is a specific diagnosis of a patient's condition based on an examination of the various cues and rfes that are present</td>
</tr>
<tr>
<td>EvidenceDiagnosisList</td>
<td>A serializable list of EvidenceDiagnosis objects. Has an indexer as a transparent layer between the calling function and the underlying list container.</td>
</tr>
<tr>
<td>PatientCase</td>
<td>A class which handles a single patient's case. Aggregates information about the patients symptoms and uses them to generate lists of potential diagnoses</td>
</tr>
</tbody>
</table>
18. **Appendix J – C# evidence client methods**
implemented by the ClinicalEvidenceWebInterface class

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getAssociatedDifferentialCues</td>
<td>Gets the differentials that are associated with a particular RFE. Then retrieves the cues that are associated with each differential. The result is returned as a dictionary which is indexed by the name of a differential and the values are lists of cues that are associated with that particular differential.</td>
</tr>
<tr>
<td>GetCueDifferentials</td>
<td>Get a list of differential diagnosis that are associated with a particular cue. Returns an indexable list of differential diagnosis.</td>
</tr>
<tr>
<td>GetCueRfes</td>
<td>Get a list of all reasons for encounter associated with a particular cue.</td>
</tr>
<tr>
<td>GetDifferentialCues</td>
<td>Get a list of cues that are associated with a particular differential diagnosis. Results are returned as an indexable object.</td>
</tr>
<tr>
<td>GetDifferentialRedFlags</td>
<td>Get a list of symptoms which are associated with a differential diagnosis that are considered strong indicators of its presence. Returns an indexable list of symptoms.</td>
</tr>
<tr>
<td>GetDifferentialRfes</td>
<td>Get a list of reasons for encounter that are associated with a particular differential diagnosis.</td>
</tr>
<tr>
<td>GetDifferentialRisks</td>
<td>Get a list of risks that are associated with a particular evidence diagnosis. The results are returned as an indexable object.</td>
</tr>
<tr>
<td>GetDifferentialSigns</td>
<td>Gets a list of signs that are associated with a particular evidence diagnosis. The results are returned as an indexable object.</td>
</tr>
<tr>
<td>GetDifferentialSymptoms</td>
<td>Get a list of the symptoms that are associated with a particular differential diagnosis. The results are returned as an indexable list of objects.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetDifferentialTests</td>
<td>Get a list of tests that can be performed to check for a particular differential diagnosis. The results are returned as an indexable list of objects.</td>
</tr>
<tr>
<td>GetRfeCues</td>
<td>Get a list of all cues associated with a reason for encounter</td>
</tr>
<tr>
<td>GetRfeDifferentials</td>
<td>Get the differentials (possible diagnosis) for a reason for encounter</td>
</tr>
<tr>
<td>GetRfeQuantifications</td>
<td>Get a list of the quantifications that are associated with the RFE that is passed as an argument. Result is returned as an indexable object</td>
</tr>
<tr>
<td>GetRfeRisks</td>
<td>Get a list of all risks associated with a reason for encounter</td>
</tr>
<tr>
<td>GetRfeSigns</td>
<td>Get a list of all signs associated with a reason for encounter</td>
</tr>
<tr>
<td>GetRfeSymptoms</td>
<td>Get a list of all symptoms associated with a reason for encounter</td>
</tr>
<tr>
<td>GetRfeTests</td>
<td>Get a list of all tests associated with a reason for encounter</td>
</tr>
<tr>
<td>GetRiskDifferentials</td>
<td>Get a list of differential diagnosis associated with a particular risk</td>
</tr>
<tr>
<td>GetRiskRfes</td>
<td>Get a list of all reasons for encounter associated with a particular risk</td>
</tr>
<tr>
<td>GetSignDifferentials</td>
<td>Get a list of differential diagnosis associated with a particular sign</td>
</tr>
<tr>
<td>GetSignRfes</td>
<td>Get a list of all reasons for encounter associated with a particular sign</td>
</tr>
<tr>
<td>GetSymptomDifferentials</td>
<td>Get a list of differential diagnosis associated with a particular symptom</td>
</tr>
<tr>
<td>GetSymptomRfes</td>
<td>Get a list of all reasons for encounter associated with a particular symptom</td>
</tr>
<tr>
<td>GetTestDifferentials</td>
<td>Get a list of differential diagnosis associated with a particular test</td>
</tr>
<tr>
<td>GetTestRfes</td>
<td>Get a list of all reasons for encounter associated with a particular test</td>
</tr>
<tr>
<td>ListCues</td>
<td>Get a list of all cues in the clinical evidence ontology</td>
</tr>
<tr>
<td>Service</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ListDifferentials</td>
<td>Get a list of all differential diagnosis in the clinical evidence ontology</td>
</tr>
<tr>
<td>ListRfes</td>
<td>Retrieves a list of all RFEs that are stored in the clinical evidence ontology</td>
</tr>
<tr>
<td>ListRisks</td>
<td>Get a list of all risks in the clinical evidence ontology</td>
</tr>
<tr>
<td>ListSigns</td>
<td>Get a list of all signs in the clinical evidence ontology</td>
</tr>
</tbody>
</table>
19. Appendix K – C# evidence client methods implemented by the PatientCase class

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UpdateCase(PatientEvidenceSet)</code></td>
<td>Accept patient evidence set data from the CDSS relating to a patient case as a new evidence set. Use this data to re-filter and re-rank the returned list of possible diagnoses to consider to the CDSS interface.</td>
</tr>
</tbody>
</table>
20. **Appendix L – The PatientEvidenceSet C# class**

```csharp
public partial class PatientEvidenceSet {

    private List<RFE> rFESField;
    private List<Demographic> demographicsField;
    private List<Risk> risksField;
    private List<Sign> signsField;
    private List<Symptom> symptomsField;
    private ushort patientIDField;

    public PatientEvidenceSet() {
        this.symptomsField = new List<Symptom>();
        this.signsField = new List<Sign>();
        this.risksField = new List<Risk>();
        this.demographicsField = new List<Demographic>();
        this.rFESField = new List<RFE>();
    }

    public List<RFE> RFEs {
        get {
            return this.rFESField;
        }
        set {
            this.rFESField = value;
        }
    }

    public List<Demographic> Demographics {
        get {
            return this.demographicsField;
        }
        set {
            this.demographicsField = value;
        }
    }
}
```