FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

BPMN2Asbru: Condition Wrapper Data

project proposal for the course

188.947 project medical informatics

for the master studies

medical informatics (066 936)

submitted by

Christian Hinterer
matriculation number 0927843

at the
faculty of informatics at the university of technology vienna

supervision
supervising tutor: Mag. Dr.rer.soc.oec. Katharina Kaiser

Vienna, October 16th, 2012

Table of Contents

Abbreviations

BPMN – Business Process Model and Notation
GATE – General Architecture for Text Engineering
GUI – Graphical User Interface
IDE – Integrated Development Environment
NLM – U.S. National Library of Medicine
UMLS – Unified Medical Language System
XML – Extensible Markup Language

1. Problem Description

The Business Process Model and Notation (BPMN) is a specification and notation for specifying and describing business processes in a business process model. Since it is not executable, several included information slots can be filled with plain text. If there is the need to execute the described processes for some reason, they have to be transformed into other notations that make this possible. These other notations in turn are required to be structured, so they are able to be interpreted correctly.

At the medical sector, documents that include processes described with the BPMN are for example the clinical practice guidelines. To find and filter medical concepts in these clinical practice guidelines a thesaurus can be used, for example the Unified Medical Language System (UMLS) by the U.S. National Library of Medicine (NLM). The UMLS is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems.

Asgaard/Asbru is a project at the university of technology in Vienna with the aim of designing a set of tasks that support the design and the execution of skeletal plans by a human executing agent. The Asgaard/Asbru project wants to use and further process the information that is contained in the clinical practice guidelines. Therefore the needed information has to be found in the clinical practice guidelines and has to be transformed into a format that can be processed, namely the Asbru language, which is a structured and defined language using XML.

Currently there is no existing application that parses the clinical practice guidelines that are described with the BPMN, filters it for concepts of the UMLS and transforms the required information into a format that can be processed by Asbru.

The main goal of this project is to develop an application that parses the clinical practice guidelines for UMLS concepts and creates a format, which is understood by Asbru, out of it. In this project the UMLS concepts will be limited to conditions.

2. Expected Results

The result of this project will be the implementation of a Java-based library that includes a parser-logic. This parser-logic will offer methods to hand over free-text condition handling data from BPMN, will generate conditions for the Asbru language out of them and store these conditions in a file that can be processed by Asbru. This implementation will handle mainly conditions that contain exact data values and not time-related values.

Furthermore the results will include a standalone Java-based application that uses the library and offers a GUI to operate and configure the parser.

For example a condition specified with the BPMN could look like this:

```
<conditionExpression xsi:type="tFormalExpression" id="cond">
        age &lt;60
</conditionExpression>
```

After the use of the parser, it will look like this:

```
<simple-condition>
  <comparison type="less-than">
    <left-hand-side>
      <parameter-ref name="age" />
    </left-hand-side>
    <right-hand-side>
      <numerical-constant value="60"/>
    </right-hand-side>
  </comparison>
</simple-condition>
```

3. Method

The parser will be implemented using Java 6 as platform and Eclipse as integrated development environment (IDE).
There are already existing tools and software components to parse texts for UMLS concepts.

The NLM offers an application named MetaMap that can be used to indentify UMLS concepts in texts. There is also an existing Java API of MetaMap. Furthermore there is a plug-in for General Architecture of Text Engineering (GATE) that allows not only to identify the concepts but also to process them. GATE is an open-source-software developed by the University of Sheffield to process texts and to solve text processing problems.

The GATE plug-in will be used to identify the UMLS concepts in question. The logic to involve the GATE plug-in, to distinguish the different conditions and to create a file

that includes the information in a format that can be processed by Asbru, will be implemented within this project.

The first step will be to identify the concepts that include the conditions in the medical practice guidelines.

After that the extracted conditions will be analyzed. On the basis of this analysis the conditions will be classified into different groups. This is a very complex procedure, because the slug of the conditions is plain text. That makes it quite complicated to distinguish between the different conditions.

Then it will be evaluated which groups of conditions are able to be translated into the Asbru language rather easy, which ones are more difficult to translate, respectively in conjunction with more effort or under consideration of more complex circumstances and which ones are not able to be translated with this parser implementation.

In the end the conditions, that are identified as UMLS concepts and that can be modeled in the Asbru language, will be translated.

The following list includes the conditions that can be modeled with the Asbru language. As mentioned before, this parser will only process conditions for data values, but no time-oriented aspects.

Since the Asbru is a time-oriented, intention-based, skeletal plan-specification representation language, the conditions remind of the conditions of a programming language with its procedural style.

The following list intentionally does not include the whole Asbru structure of the conditions, but only an enumeration of some conditions:

- simple conditions:
    - <comparison>: compares values and variables with operators (less-than, less-or-equal, greater-than, greater-or-equal, equal, not-equal)
    - <is-within-range>: checks if value is in specific range
    - <is-member-of>: evaluates if something is part of a specified set
    - <is-at-end>: evaluates if the iterator of a set, list or multi-set is at the end
    - <is-at-start>: evaluates if the iterator of a set, list or multi-set is at the beginning
    - <has-occurred>: evaluates if the plan state transition given as a child tag, has already occurred
    - <simple-condition-combination>: combines simple conditions with the logical operators and, or, xor
    - <simple-condition-not>: negation of its child element
- conditions:

  - <filter-precondition>: if not fulfilled, the plan is rejected
  - <setup-precondition>: must hold before execution; can be achieved by physician's actions
  - <suspend-condition>: plan suspended, until reactivation
  - <reactivate-condition>: reactivates a suspended plan
  - <complete-condition>: if fulfilled and all mandatory subplans too, plan completed
  - <abort-condition>: if fulfilled, plan aborted
- iterative plans (for-each,…):
  - <termination-condition>: ends an iterative plan, if fulfilled
- cyclical plans:
  - <cyclical-complete-condition>: specifies the time a cyclical plan should end based on end-time, until-condition or fixed number of times-completed
  - <until-condition>: if fulfilled, ends the cyclical plan

4. References

[Kaiser, 2012] Dr.rer.soc.oec. Katharina Kaiser. BPMN2Asbru: Condition Wrapper Data Project Description. Created At: February 23, 2012. Retrieved At: October 4, 2012. http://www.cvast.tuwien.ac.at/node/428

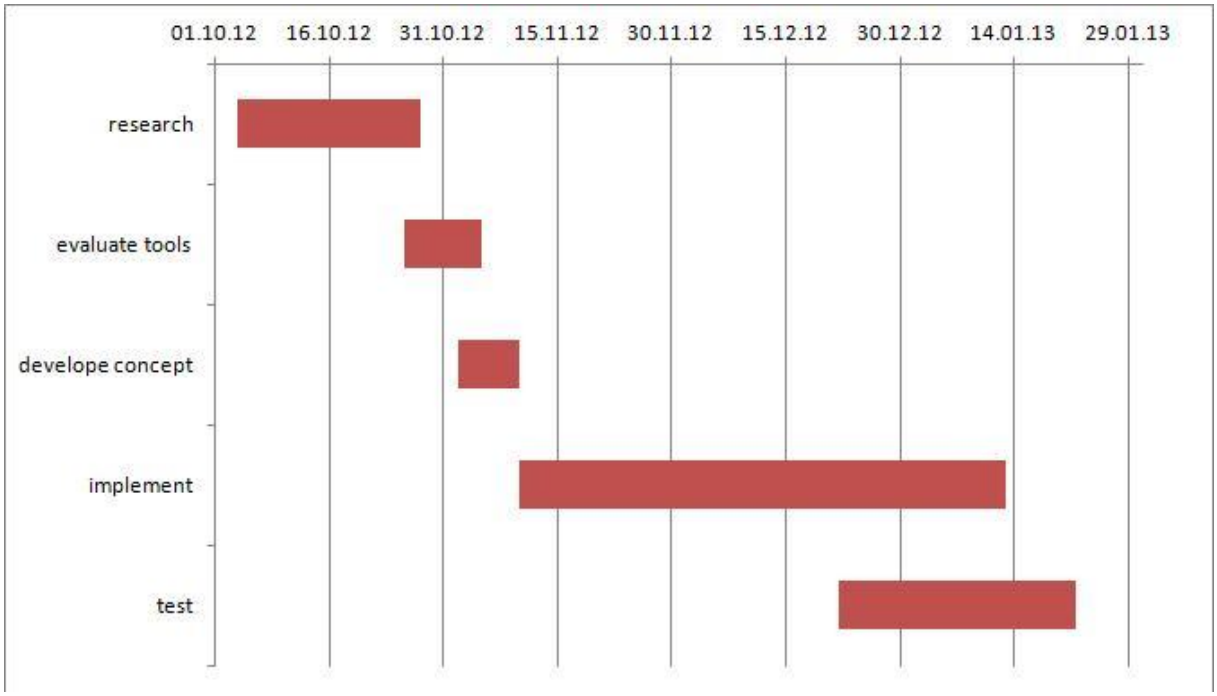
The Asgaard/Asbru Project. Retrieved At: October 16, 2012. http://www.asgaard.tuwien.ac.at/about/project.html

Unified Medical Language System. Retrieved At: Octuber 16, 2012. http://www.nlm.nih.gov/research/umls/

MetaMap. Retrieved At: October 16, 2012. http://metamap.nlm.nih.gov/

GATE. Retrieved At: October 16, 2012. http://gate.ac.uk/

Appendix 1 – time schedule



Kick-Off meeting: October 4, 2012

Phase 1 (research; Oct. 4, 2012 – Oct. 28, 2012; app. 24d): Research on problem, state of the art, existing standards, existing solutions and existing tools

Phase 2 (tool evaluation; Oct. 26, 2012 – Nov. 5, 2012; app. 10d): Evaluation of existing tools with regard to applicability in my implementation

Phase 3 (concept development; Nov. 2, 2012 – Nov. 10, 2012; app. 8d): Developing a concept of the implementation, including components, architecture and used technologies

Phase 4 (implementation; Nov. 10, 2012 – Jan. 13, 2013; app. 64d): Implementing the parser, including documentation

Phase 5 (testing; Dec. 22, 2012 – Jan. 22, 2013; app. 31d): Testing the implemented application

Hand-In final version: January 31, 2013

Like in every other software project the phases 4 and 5 will alternately be part of an iterative process.

Appendix 2 – seminar of medical informatics

Topic description of the course 188.948 seminar of medical informatics:

The topic of the seminar is tight connected with the content of the project.

The work for the seminar will include a detailed description how medical practice guidelines can be analyzed regarding to information extraction. At this version the information that will be extracted will be limited to conditions and more exactly only to that ones, that handle data values, but not time-related ones.

It will be described how the conditions can be filtered out of the guidelines and further more how they can be arranged in different classes depending on the including semantics.

It will also include how the conditions can be analyzed regarding to their practicability into the Asbru language. If the conditions can be translated into the Asbru language, it will be described how this can be done and also under which circumstances this translation is possible.

The works for the seminar will also include a prototype concept for translating complex conditions out of the medical practice guidelines and into the Asbru language.

In further works this prototype concept can possibly being used to create a more seamless specification for translating the respective conditions into the Asbru language.

Just for the steak of completeness some tools that support the described processes will be mentioned.

The results of the seminar will be a “state-of-the-art” document and a presentation on the document’s content and the acquired knowledge.