FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

BPMN2Asbru: Condition Wrapper Data

project documentation (description of architecture)

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, April 15th, 2013

Table of Contents

1. Functionality (What is the program doing?)

The BPMN2Asbru is a program to parse plain-text phrases for logical conditions and translates them into the Asbru language.

The input is a XML-valid file that contains these assumed condition phrases marked with the xml-tag “ConditionExpression”.

The program uses GATE (http://gate.ac.uk/) to process the input texts. In this case process means, doing the xml handling, annotating different words, phrases and other concepts like numbers, symbols and others and extracting these information to reuse it in logical conditions.

Since the demanded conditions are contained in clinical practice guidelines and therefore containing many medical concepts, some component was necessary to identify these medical concepts. For this reason the UMLS (unified medical language system) is used as metathesaurus for the “medical language”. The UMLS offers a program called MetaMap, which in turn includes a JAVA API than can be used as a plug-in for gate. This MetaMap plug-in for gate communicates with servers that process the input texts and returns the found medical concepts. The location of these servers can be configured in the BPMN2Asbru program! It is recommended to download and setup the servers locally, to provide quick processing times.

After the text annotations are made with gate and the MetaMap plug-in, the program annotates some more words, word phrases and symbols, to get a basic idea of the text content. For example words that express logic operations (AND, OR, XOR) are annotated separately. Braces are also annotated separately to identify nested concepts. Some more additional annotations are made that should help to get the semantics of the contained conditions.

After that, the program uses the annotations for trying to interpret the texts correctly and to filter the key information that represent the logical conditions.

After that, these found conditions are translated into the Asbru language, which is fully automatic processable by machines, since being based completely on the XML standard.

The result of the program is an output file, containing the conditions formulated in the Asbru language.

To change the input file, the output file, the MetaMap configuration and to use other functionalities, like reinitializing gate, the program offers a menue in the console during runtime.

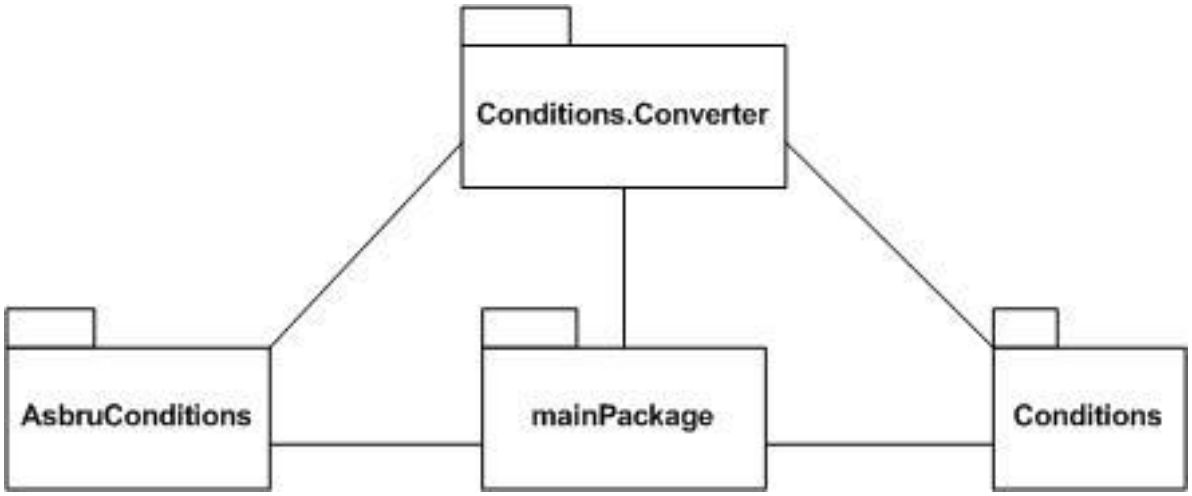
2. Further work (What is the program not yet doing?)

The main task for the future is to improve the process of identifying the right conditions in the texts. This is a very complex task and therefore room for improvement is always given.

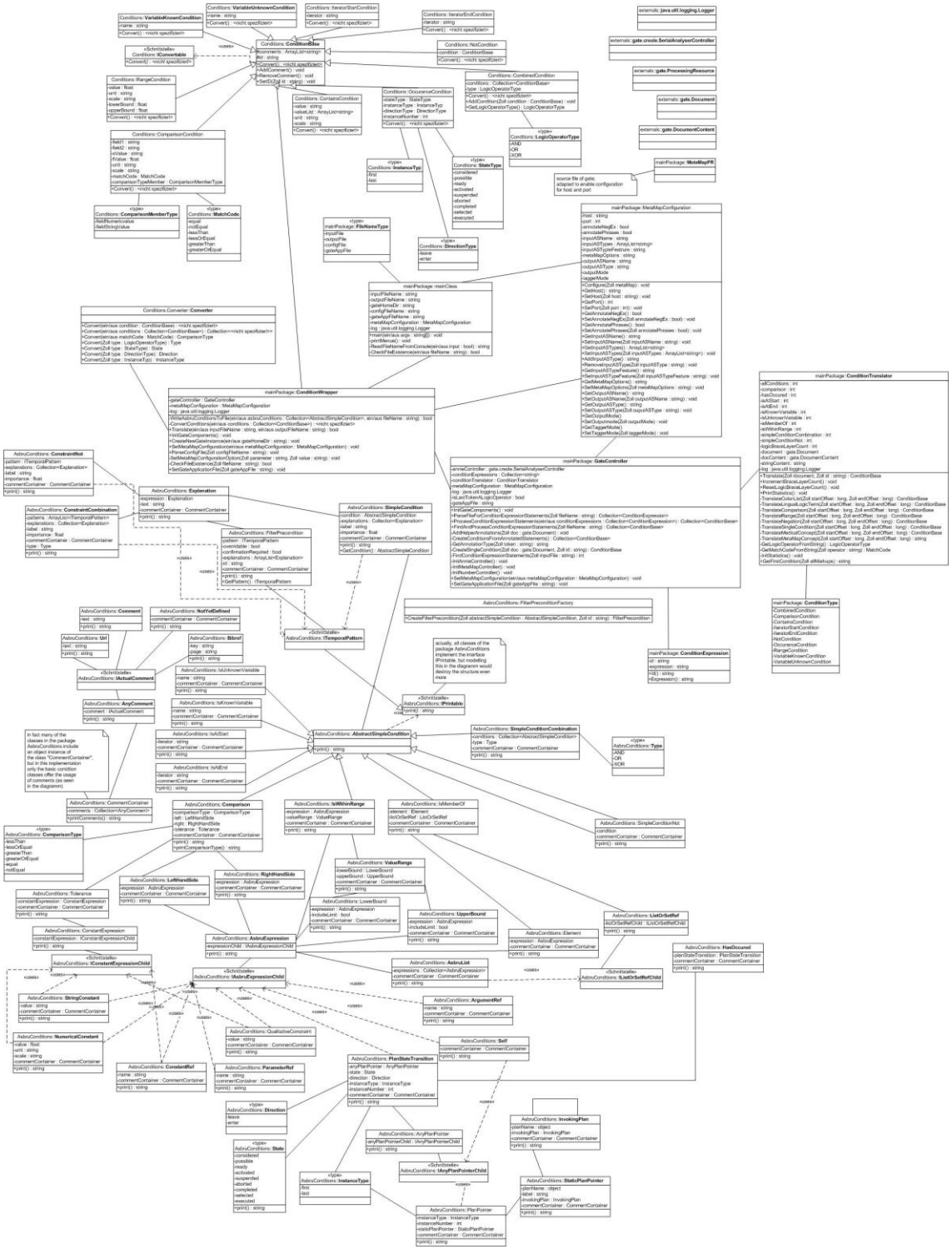
To improve the identification of the conditions in the text, the process of annotating the text, which is a pre-process of identification, can also be improved, for example by creating more relations between the annotations.

At the moment the program offers the possibility of configuring MetaMap. Depending on the MetaMap configuration, the results given by the MetaMap server differ in bulk and accuracy. At the moment the implementation always uses the medical annotation with the highest score, calculated by MetaMap. A future work could be taking a closer look on the results with lower scores. Maybe one of them could fit the wanted medical concept better.

3. Package Overview



4. Graphical Overview



5. Program Inputs (command line arguments)

- input file name:
  - mandatory
  - format: xml
  - content: bpmn-modelled clinical practice guideline
  - request: valid xml
- file name of a gate application:
  - optional
  - no definition of a gate application file: a gate application with all necessary resources and modules is created at runtime
  - format: .gapp; .xgapp
  - content: configuration of a gate application (in xml)
- metamap configuration file name:
  - optional
  - no definition of a configuration file: standard configuration is used
  - format: cfg
  - content: MetaMap configuration (server, port, operating parameters [search options]): template is given in the data folder of the project folder (../BPMN2Asbru/data/MetaMap.Config_regExp.txt) and (../BPMN2Asbru/data/MetaMapConfig.cfg)
- output file name:
  - optional
  - no definition of an output file: result_asbruconditions.xml is created in the project folder
  - format: xml
  - content: found conditions modeled in the Asbru language
- gate home directory:
  - optional
  - no explicit definition of gate home directory: standard installation directory is used (C:\Program Files\GATE_Developer_7.1)
  - format: String
  - content: gate home directory

6. Program Outputs

output file: as written in Chapter 5. (Program Inputs) the output of the program is one file of format xml containing the found conditions translated into the Asbru language and formatted as valid xml. The file name and the folder in which it is stored can be configured by the user by defining the output file via the programs' command line arguments (Chapter 5. Program Inputs). If no output file is configured, the results are written into the file "output.xml" in the project folder.