

=====

BPEL2PNML

=====

OVERVIEW:

- BPEL2PNML : A tool
(<http://www.bpm.fit.gut.edu.au/projects/babel/tools/BPEL2PNML.jar>)
for translating process definitions represented using the Business Process Execution Language for Web Services (BPEL) to Petri nets represented using the Petri Net Markup Language (PNML). The PNML code produced by this tool can be loaded into the WofBPEL tool below.

- WofBPEL: A tool
(<http://www.bpm.fit.gut.edu.au/projects/babel/tools/wofbpel.zip>)
for formal analysis of BPEL processes. This tool takes as input Petri net descriptions in PNML obtained as output from the BPEL2PNML tool.

WofBPEL checks for unreachable BPEL activities and for concurrent activities that may enter in competition for the same inbound message (a situation forbidden in BPEL). The tool also performs a reachability analysis to detect which inbound messages need to be preserved in the queue at each step of the BPEL process.

A technical report with details about WofBPEL can be found at:
<http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2005/BPM-05-13.pdf>

1) Input and Output behaviour of the program.

Input: BPEL File

Output: PNML File named: <Name_of_BPELFile>BPEL2PNML.xml

2) Usage

```
java -jar BPEL2PNML.jar [g] <nameofBPELFile>
```

Where g is optional to obtain short transition names (useful for the graphical representation). DO NOT use this Option if further analysis with WofBPEL is desired.

Examples:

BPEL2PNML HelloWorld.bpel

(=> useable for further analysis with WofBPEL)

BPEL2PNML g HelloWorld.bpel

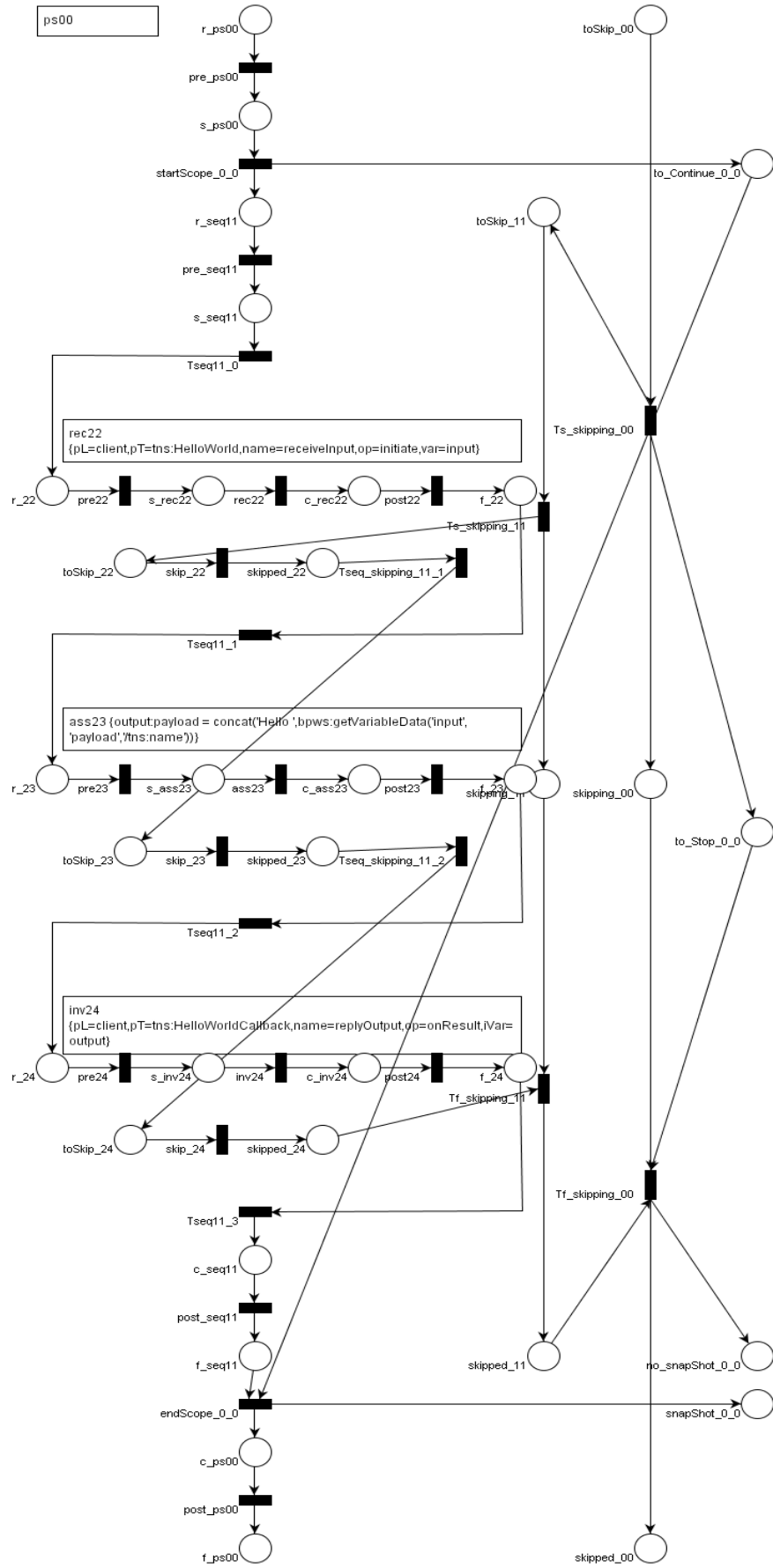
(=> not useable for WofBPEL but better for graphical representation using e.g. PIPE.)

The PNML file is used as input for the tool WofBPEL.

Please follow the instructions for an analysis of the produced Petri net as described in the WofBPEL documentation.

RECOMMENDATION:

To obtain a visual insight into the produced Petri net (in PNML format) the usage of the freely available Petri net Editor PIPE is recommended (<https://sourceforge.net/projects/petri-net/>). It is important to mention that the visualisation part is not perfect and complete. The reason is that the translation of a BPEL file into a Petri net is already a complex task. An automatic correct and nice visualisation of the produced Petri net would require some more programming. As an example of the visual output the following Petri net (HelloWorld of the Oracle BPEL examples, <http://www.oracle.com/technology/products/ias/bpel>) is provided (note that this example has been generated without including the bypass-Path for possible termination).



3) General Preconditions for the correct behaviour of the program:

- * Java Version 1.4.2._x
- * The input file is a syntactically correct XML File describing a business process in BPEL.

Otherwise: the program will terminate with no output or incorrect output and an error message.

4) General Limitations.

The program does currently not support:

1. Multiple simultaneous executions of the same event handler.
2. Serialisable Scopes

It is not planned to support features 1 and 2 of the above list. In addition the tool does not deal with process instantiation and thus the attribute “createInstance” is ignored.

5) Further Constraints.

Termination is currently only supported for basic activities. Compensation is only supported in a way as demonstrated in the compensation example. In particular: we assume a compensate command defining to which scope the compensation belongs and we assume that this scope exists. Additionally in the XML-file the compensate command is declared before the corresponding compensation-Handler.

6) Deviations and Extensions to the Specification described in the WofBPEL paper:

- Token Collector. To ensure to have exactly one active output place a token collector was introduced. Token get collected from the no_SnapShot and snapshot places of the top-level scope. Note that the process itself defines a scope.
- We generally provide a default-FaultHandler for a process.
- translation of the <while> construct follows the same schema as for the <pick> or <switch> activity, i.e. a not z (Boolean expression for the while loop) will connect to the skip-Path of the <while> activity.

7) Desirable Features. It would be desirable to include options to switch on and off standard behaviours like the bypass-Path to terminate basic activities in order to improve the visual output of the generated Petri-net. In case there is sufficient interest, i.e. a sufficient large number of people want this feature then we will implement such an option.

8) Error Reports and Problems.

This program is a research prototype following the theoretical specifications of the paper “Formal Semantics and Analysis of Control Flow in WS-BPEL” (<http://is.tm.tue.nl/staff/wvdaalst/BPMcenter/reports/2005/BPM-05-13.pdf>). During the programming phase tests have been performed for each concept (using files from the Oracle BPEL Tutorial and other hand-coded BPEL samples). However, the program has not been tested intensively. If you find any error or problems please follow these steps:

- i) If an error occurred during the translation process then:
 - Produce a screen dump containing the complete error-message of the Output on the shell/dos-prompt.
 - Else if the translation worked but a semantical error in the output is detected then describe this error in a txt-file.
- ii) Send the screen dump or txt-file and the BPEL file as an attachment to the following email-address: sw.breutel@qut.edu.au
Generally I will try to fix the error and send a reply within 5 working days.

Third-Party Software Acknowledgements

(a) Software developed by the JDOM Project
(<http://www.jdom.org/>).

JDOM Beta 9, Copyright © 2000-2003 Jason Hunter & Brett McLaughlin. All rights reserved. This software is subject to the terms of the licence available at <http://www.jdom.org/>.

Copyright 2005 BPM-Program@QUT

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing,
software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
See the License for the specific language governing permissions
and
limitations under the License.