# Transpath2CPN QVT Transformation
# Quick Start

Abel Gómez

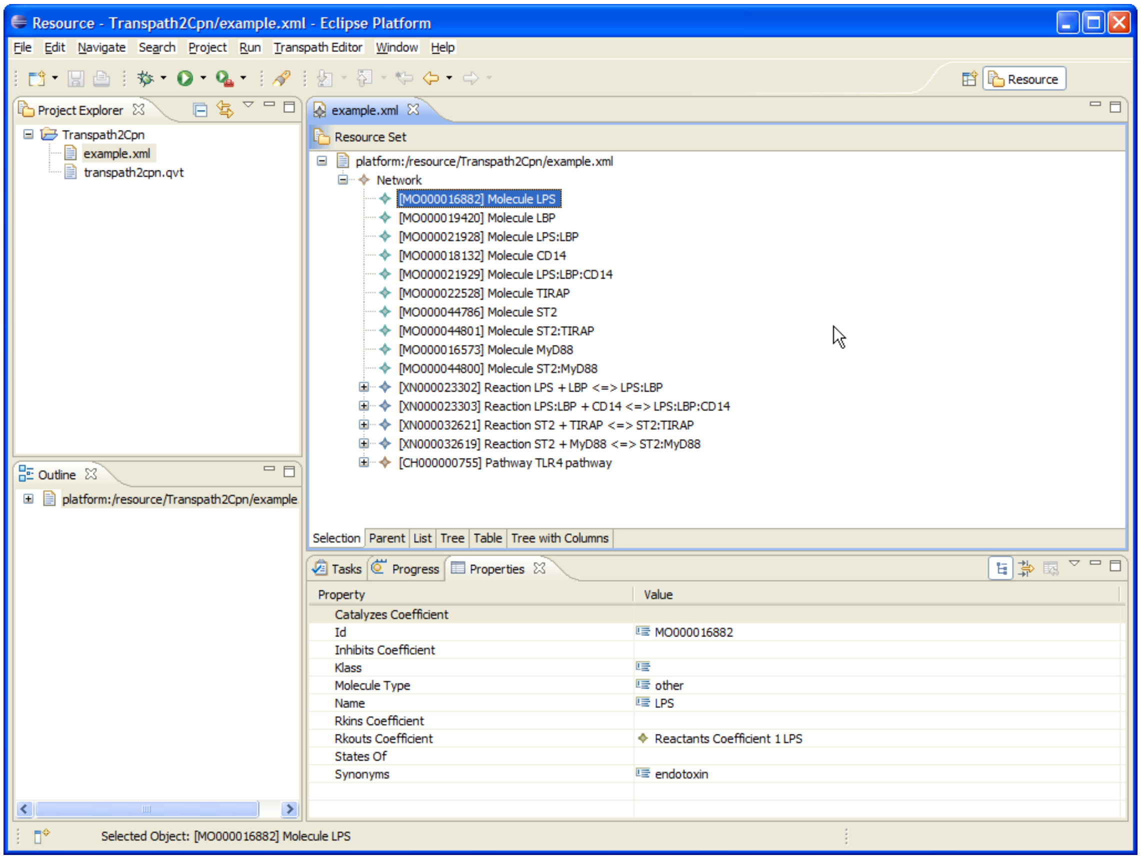
## 1 Getting the example tool

The tool can be downloaded from `ftp://lujuria.dsic.upv.es/transpath2cpn`. The compressed file `eclipse-platform-3.3.2-win32-Transpath2Cpn-Minimal.zip` contains all the needed files to execute the transformation. It also includes some example files.

## 2 The workspace

The tool provides two example files:

- `example.xml`. Is a XML file extracted from the transpath database. It must contain the information about a pathway.
- `transpath2cpn.qvt`. Contains the QVT transformation to transform from the transpath domain to the CPN Tools domain.


Resource - Transpath2Cpn/example.xml - Eclipse Platform
File Edit Navigate Search Project Run Transpath Editor Window Help
Resource
Project Explorer
Transpath2Cpn
example.xml
transpath2cpn.qvt
Outline
platform:/resource/Transpath2Cpn/example
example.xml
Resource Set
Selection Parent List Tree Table Tree with Columns
Tasks Progress Properties

platform:/resource/Transpath2Cpn/example.xml
Network
[MO000016882] Molecule LPS
[MO000019420] Molecule LBP
[MO000021928] Molecule LPS:LBP
[MO000018132] Molecule CD14
[MO000021929] Molecule LPS:LBP:CD14
[MO000022528] Molecule TIRAP
[MO000044786] Molecule ST2
[MO000044801] Molecule ST2:TIRAP
[MO000016573] Molecule MyD88
[MO000044800] Molecule ST2:MyD88
[XN000023302] Reaction LPS + LBP <=> LPS:LBP
[XN000023303] Reaction LPS:LBP + CD14 <=> LPS:LBP:CD14
[XN000032621] Reaction ST2 + TIRAP <=> ST2:TIRAP
[XN000032619] Reaction ST2 + MyD88 <=> ST2:MyD88
[CH000000755] Pathway TLR4 pathway

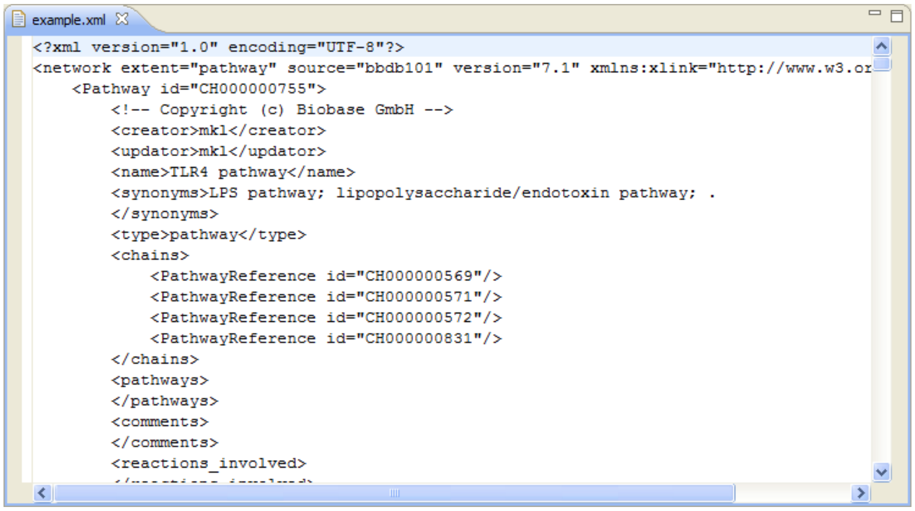
| Property | Value |
|---|---|
| Catalyzes Coefficient | |
| Id | MO000016882 |
| Inhibits Coefficient | |
| Klass | |
| Molecule Type | other |
| Name | LPS |
| Rkins Coefficient | |
| Rkouts Coefficient | Reactants Coefficient 1 LPS |
| States Of | |
| Synonyms | endotoxin |

Selected Object: [MO000016882] Molecule LPS

Figure 1: Workspace with the example files.

Figure 1 shows the example workspace when the tool starts. The file `example.xml` is open with its default editor (the transpath model editor). This editor is able to represent the information of the TRANSPATH database as instances of the transpath metamodel.

Figure 2 shows the actual contents of the `example.xml` file. As can be seen, `example.xml` is a XML directly obtained from the TRANSPATH database.



```
<?xml version="1.0" encoding="UTF-8"?>
<network extent="pathway" source="bbdb101" version="7.1" xmlns:xlink="http://www.w3.or
    <Pathway id="CH000000755">
        <!-- Copyright (c) Biobase GmbH -->
        <creator>mkl</creator>
        <updator>mkl</updator>
        <name>TLR4 pathway</name>
        <synonyms>LPS pathway; lipopolysaccharide/endotoxin pathway; .
        </synonyms>
        <type>pathway</type>
        <chains>
            <PathwayReference id="CH000000569"/>
            <PathwayReference id="CH000000571"/>
            <PathwayReference id="CH000000572"/>
            <PathwayReference id="CH000000831"/>
        </chains>
        <pathways>
        </pathways>
        <comments>
        </comments>
        <reactions_involved>
        </reactions_involved>
```

Figure 2: Actual contents of the `example.xml` file.

# 3 Running the example

In order to execute the transformation, the user must select Run as → 1 QVT Transformation as the figure 3 shows.

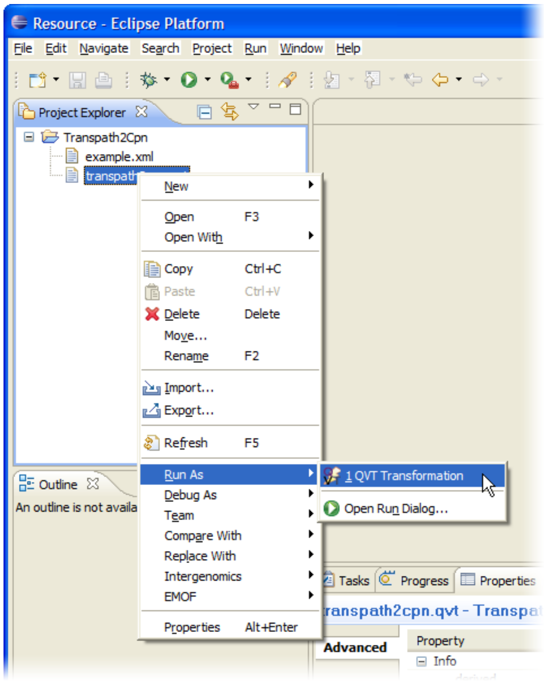
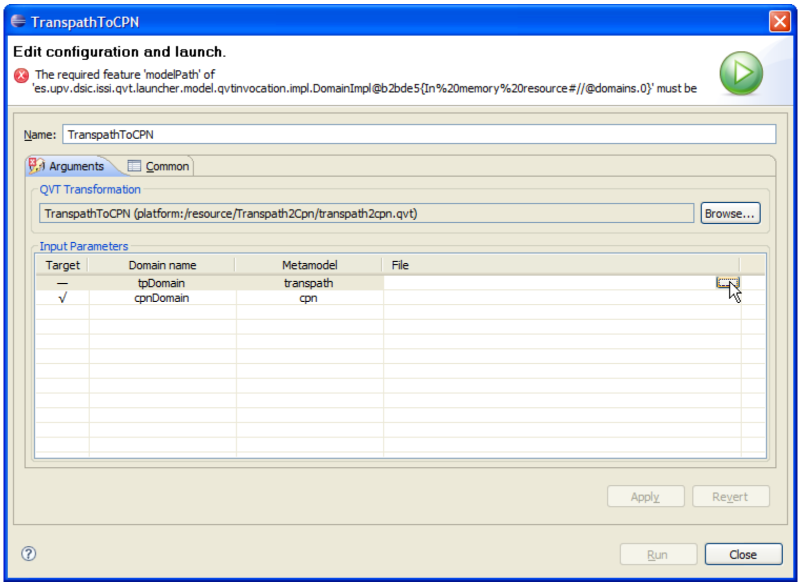


Figure 3: Running the QVT transformation.

Figure 4 shows the dialog used to configure the transformations. The upper editable field contains the name that identifies the selected transformation. This name is initialized to the transformation name. The second field (which has the Browse... button on the right) indicates the path of the file that contains the transformation. The Browse... button is used to specify the transformation file to be executed (`*.qvt`). When the Run as dialog is opened by selecting a QVT file from the context menu, this value is stablished automatically to the selected file.



Edit configuration and launch.

The required feature 'modelPath' of 'es.upv.dsic.issi.qvt.launcher.model.qvtinvocation.impl.DomainImpl@b2bde5{In%20memory%20resource#//@domains.0}' must be

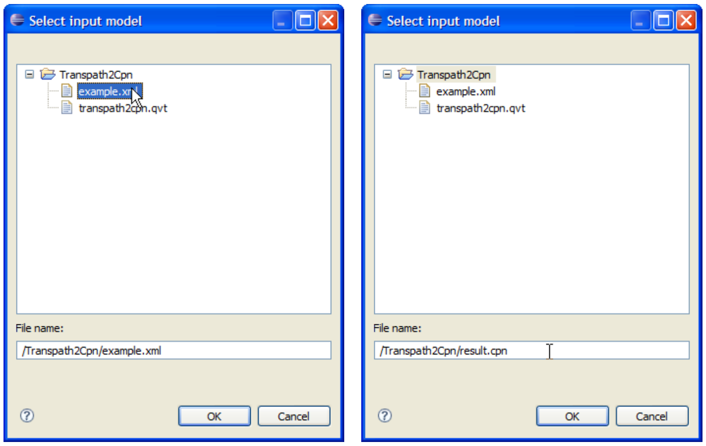
Name: TranspathToCPN

TranspathToCPN (platform:/resource/Transpath2Cpn/transpath2cpn.qvt)

| Target | Domain name | Metamodel | File |
|---|---|---|---|
| — | tpDomain | transpath | |
| √ | cpnDomain | cpn | |

Figure 4: Configuring the transformation.

Once a QVT file has been selected, the central table can be used to specify which files correspond to each one of the domains of the transformation, as the figure 4 shows. Figures 5a and 5b show the dialogs to specify the transpath and the cpn domains repectively.



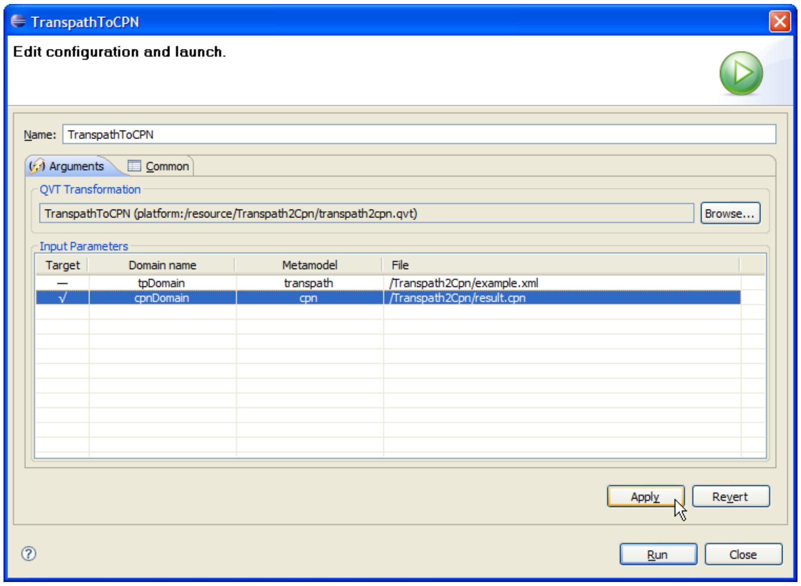
File name: /Transpath2Cpn/example.xml

File name: /Transpath2Cpn/result.cpn

(a) Input file of the transformation (b) Output file of the transformation

Figure 5: Arguments of the transformation.

Figure 6 shows the dialog with all the information needed to execute the transformation. The direction of the transformation is specified by clicking in the target cell of the target domain. By default, the target domain will be the last domain of the transformation.



Edit configuration and launch.

Name: TranspathToCPN

QVT Transformation

TranspathToCPN (platform:/resource/Transpath2Cpn/transpath2cpn.qvt)

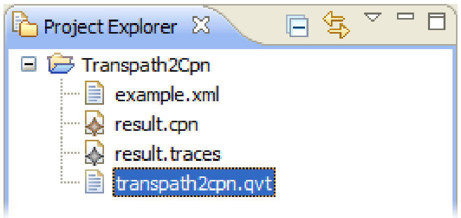
Input Parameters

| Target | Domain name | Metamodel | File |
|---|---|---|---|
| — | tpDomain | transpath | /Transpath2Cpn/example.xml |
| √ | cpnDomain | cpn | /Transpath2Cpn/result.cpn |

Figure 6: The transformation and its arguments.

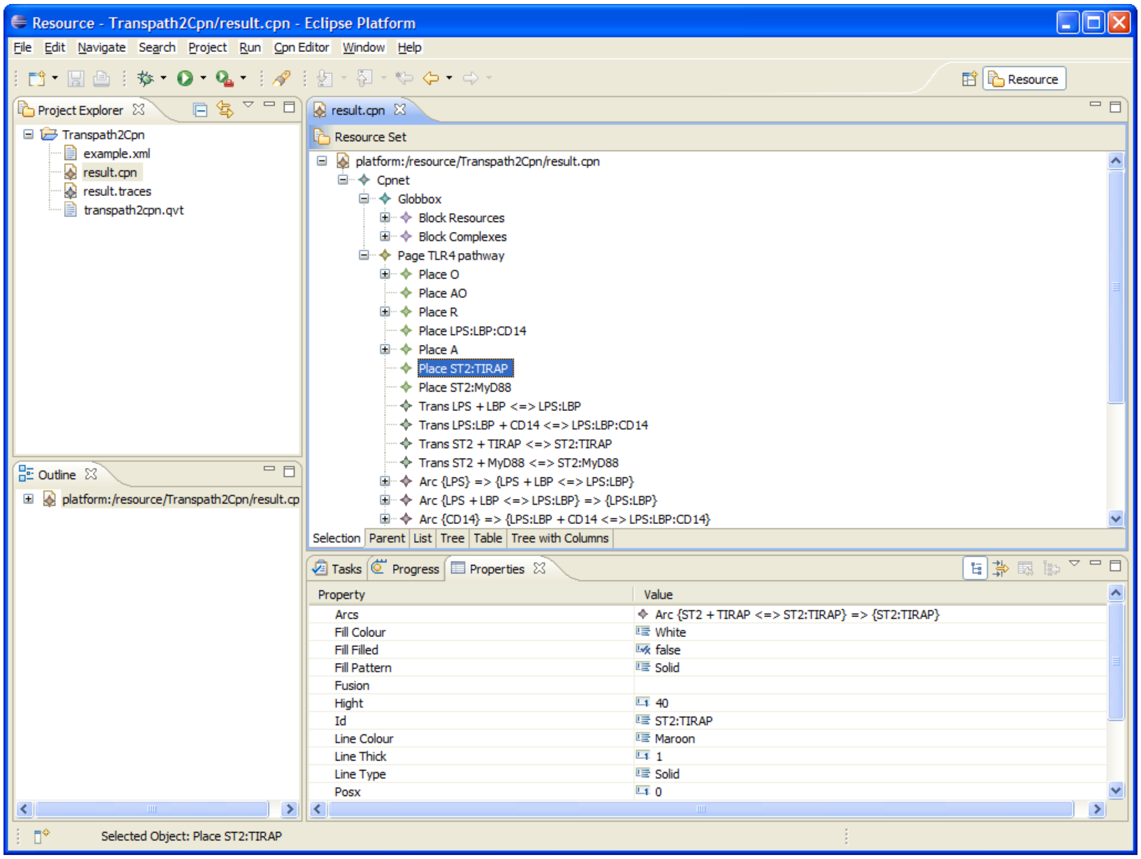
## 4 Result files

Once the transformation has been configured, the Run button can be pressed, and the execution of the transformation begins. When the transformation ends, two new files appear in the workspace. The first one is the result model (`result.cpn` for this example) and the second one is the traces model (`result.traces`, which filename is built from the result model). Figure 7 shows the Project Explorer view with the result files.



Transpath2Cpn
example.xml
result.cpn
result.traces
transpath2cpn.qvt

Figure 7: Result files.

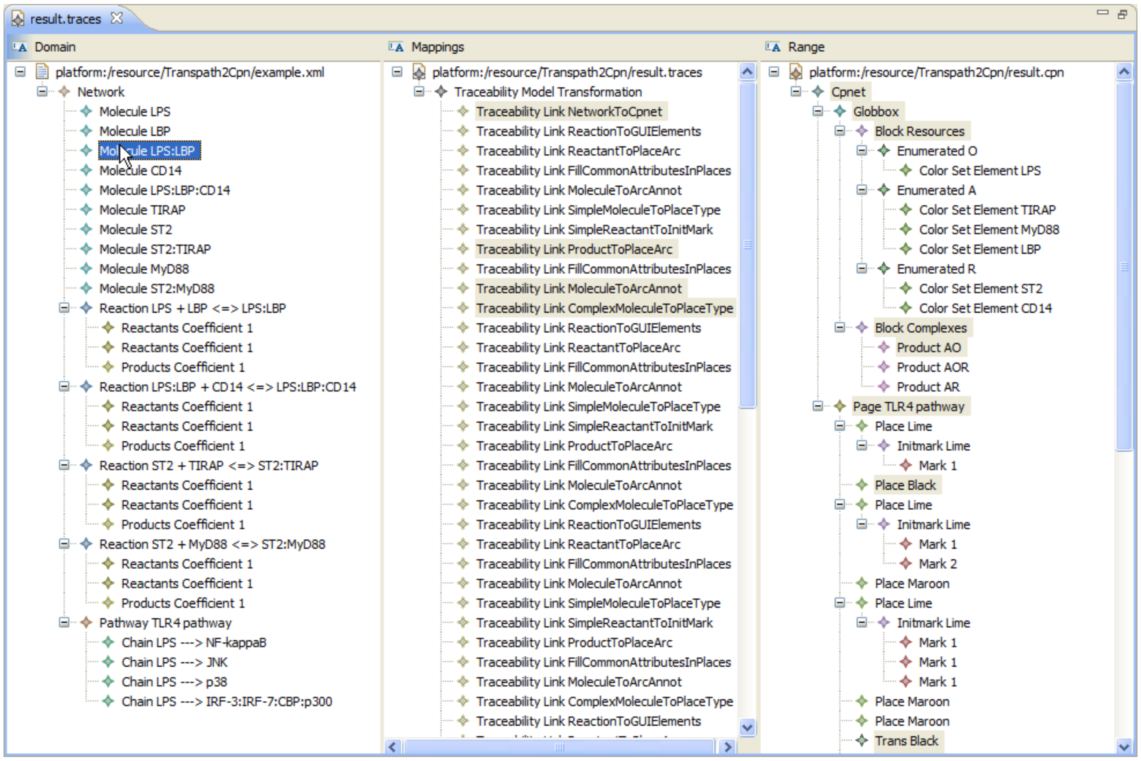

Resource - Transpath2Cpn/result.cpn - Eclipse Platform
File Edit Navigate Search Project Run Cpn Editor Window Help
Resource
Project Explorer
Transpath2Cpn
example.xml
result.cpn
result.traces
transpath2cpn.qvt
Outline
platform:/resource/Transpath2Cpn/result.cp
result.cpn
Resource Set
platform:/resource/Transpath2Cpn/result.cpn
Cpnet
Globbox
Block Resources
Block Complexes
Page TLR4 pathway
Place O
Place AO
Place R
Place LPS:LBP:CD14
Place A
Place ST2:TIRAP
Place ST2:MyD88
Trans LPS + LBP <=> LPS:LBP
Trans LPS:LBP + CD14 <=> LPS:LBP:CD14
Trans ST2 + TIRAP <=> ST2:TIRAP
Trans ST2 + MyD88 <=> ST2:MyD88
Arc {LPS} => {LPS + LBP <=> LPS:LBP}
Arc {LPS + LBP <=> LPS:LBP} => {LPS:LBP}
Arc {CD14} => {LPS:LBP + CD14 <=> LPS:LBP:CD14}
Selection Parent List Tree Table Tree with Columns
Tasks Progress Properties

| Property | Value |
|---|---|
| Arcs | Arc {ST2 + TIRAP <=> ST2:TIRAP} => {ST2:TIRAP} |
| Fill Colour | White |
| Fill Filled | false |
| Fill Pattern | Solid |
| Fusion | |
| Hight | 40 |
| Id | ST2:TIRAP |
| Line Colour | Maroon |
| Line Thick | 1 |
| Line Type | Solid |
| Posx | 0 |

Selected Object: Place ST2:TIRAP

Figure 8: Cpn model editor.

The result file is an XMI file which is instance of the CPN Tools metamodel. Figure 8 shows the `result.cpn` file in the Cpn model editor, and figure 9 shows the traces model with the default traces editor. This traces editor shows the correspondences between the source and the target domains.



- platform:/resource/Transpath2Cpn/example.xml
  - Network
    - Molecule LPS
    - Molecule LBP
    - Molecule LPS:LBP
    - Molecule CD14
    - Molecule LPS:LBP:CD14
    - Molecule TIRAP
    - Molecule ST2
    - Molecule ST2:TIRAP
    - Molecule MyD88
    - Molecule ST2:MyD88
    - Reaction LPS + LBP <=> LPS:LBP
      - Reactants Coefficient 1
      - Reactants Coefficient 1
      - Products Coefficient 1
    - Reaction LPS:LBP + CD14 <=> LPS:LBP:CD14
      - Reactants Coefficient 1
      - Reactants Coefficient 1
      - Products Coefficient 1
    - Reaction ST2 + TIRAP <=> ST2:TIRAP
      - Reactants Coefficient 1
      - Reactants Coefficient 1
      - Products Coefficient 1
    - Reaction ST2 + MyD88 <=> ST2:MyD88
      - Reactants Coefficient 1
      - Reactants Coefficient 1
      - Products Coefficient 1
    - Pathway TLR4 pathway
      - Chain LPS ---> NF-kappaB
      - Chain LPS ---> JNK
      - Chain LPS ---> p38
      - Chain LPS ---> IRF-3:IRF-7:CBP:p300

- platform:/resource/Transpath2Cpn/result.traces
  - Traceability Model Transformation
    - Traceability Link NetworkToCpnet
    - Traceability Link ReactionToGUIElements
    - Traceability Link ReactantToPlaceArc
    - Traceability Link FillCommonAttributesInPlaces
    - Traceability Link MoleculeToArcAnnot
    - Traceability Link SimpleMoleculeToPlaceType
    - Traceability Link SimpleReactantToInitMark
    - Traceability Link ProductToPlaceArc
    - Traceability Link FillCommonAttributesInPlaces
    - Traceability Link MoleculeToArcAnnot
    - Traceability Link ComplexMoleculeToPlaceType
    - Traceability Link ReactionToGUIElements
    - Traceability Link ReactantToPlaceArc
    - Traceability Link FillCommonAttributesInPlaces
    - Traceability Link MoleculeToArcAnnot
    - Traceability Link SimpleMoleculeToPlaceType
    - Traceability Link SimpleReactantToInitMark
    - Traceability Link ProductToPlaceArc
    - Traceability Link FillCommonAttributesInPlaces
    - Traceability Link MoleculeToArcAnnot
    - Traceability Link ComplexMoleculeToPlaceType
    - Traceability Link ReactionToGUIElements
    - Traceability Link ReactantToPlaceArc
    - Traceability Link FillCommonAttributesInPlaces
    - Traceability Link MoleculeToArcAnnot
    - Traceability Link SimpleMoleculeToPlaceType
    - Traceability Link SimpleReactantToInitMark
    - Traceability Link ProductToPlaceArc
    - Traceability Link FillCommonAttributesInPlaces
    - Traceability Link MoleculeToArcAnnot
    - Traceability Link ComplexMoleculeToPlaceType
    - Traceability Link ReactionToGUIElements

- platform:/resource/Transpath2Cpn/result.cpn
  - Cpnet
    - Globbox
      - Block Resources
        - Enumerated O
          - Color Set Element LPS
        - Enumerated A
          - Color Set Element TIRAP
          - Color Set Element MyD88
          - Color Set Element LBP
        - Enumerated R
          - Color Set Element ST2
          - Color Set Element CD14
      - Block Complexes
        - Product AO
        - Product AOR
        - Product AR
    - Page TLR4 pathway
      - Place Lime
        - Initmark Lime
          - Mark 1
      - Place Black
      - Place Lime
        - Initmark Lime
          - Mark 1
          - Mark 2
      - Place Maroon
      - Place Lime
        - Initmark Lime
          - Mark 1
          - Mark 1
          - Mark 1
      - Place Maroon
      - Place Maroon
      - Trans Black

Figure 9: Traces model editor.

In order to be able to open the result file in the CPN Tools application, this file must be converted to a valid CPN Tools XML file. This step is done by using the context menu (see figure 10). In this step, also a layout algorithm can be applied if the layout step is not performed in the previous transformation step.

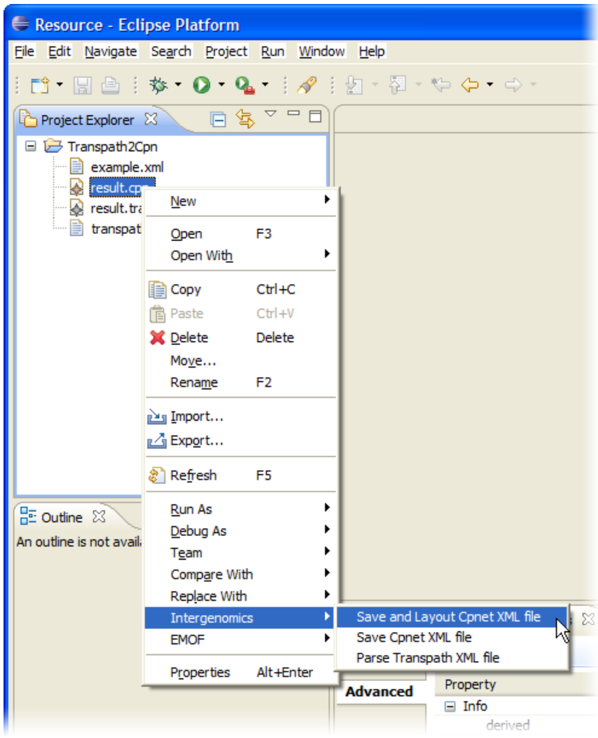
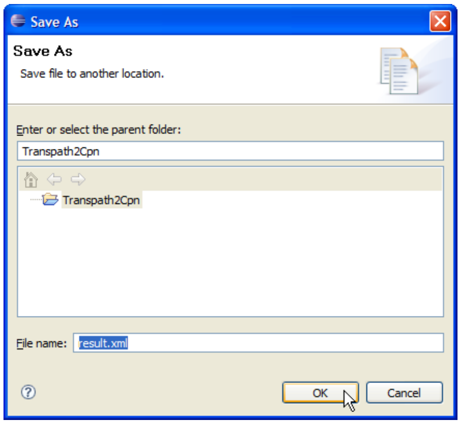


Figure 10: Save to CPN Tools.

In figure 11 the dialog to set the XML file name is shown.


Save As

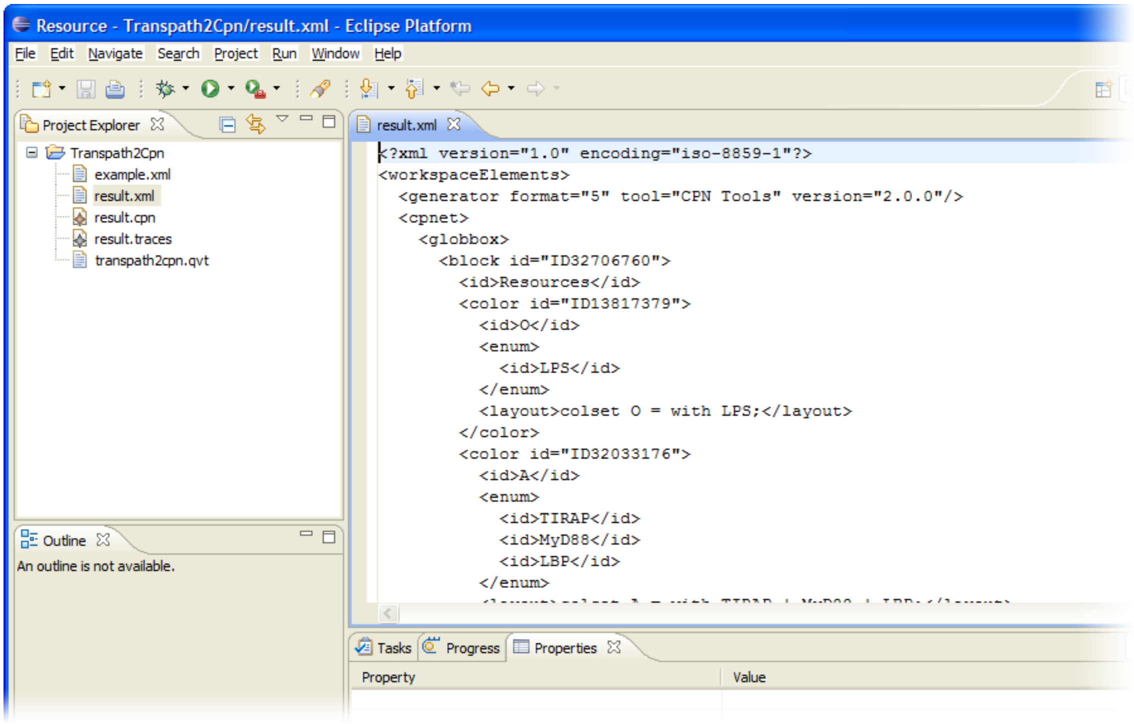
Save file to another location.

Enter or select the parent folder: Transpath2Cpn

File name: result.xml

Figure 11: Save file as. . . .

Figure 12 shows the final XML file in the Project Explorer view. The file can be opened in a text editor, and, as can be seen on the right part of the figure, it contains a valid XML file that can be directly opened in CPN Tools.



An outline is not available.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<workspaceElements>
  <generator format="5" tool="CPN Tools" version="2.0.0"/>
  <cpnet>
    <globbox>
      <block id="ID32706760">
        <id>Resources</id>
        <color id="ID13817379">
          <id>O</id>
          <enum>
            <id>LPS</id>
          </enum>
          <layout>colset O = with LPS;</layout>
        </color>
        <color id="ID32033176">
          <id>A</id>
          <enum>
            <id>TIRAP</id>
            <id>MyD88</id>
            <id>LBP</id>
          </enum>
          <layout>colset A = with TIRAP | MyD88 | LBP;</layout>
```

Figure 12: Contents of the final XML file.

## 5 Result file in CPN Tools

Finally, figure 13 shows how the final petri net looks like. The position of the places and transitions can vary depending on how the layout algorithm is applied. The obtained net can be directly simulated by CPN Tools.

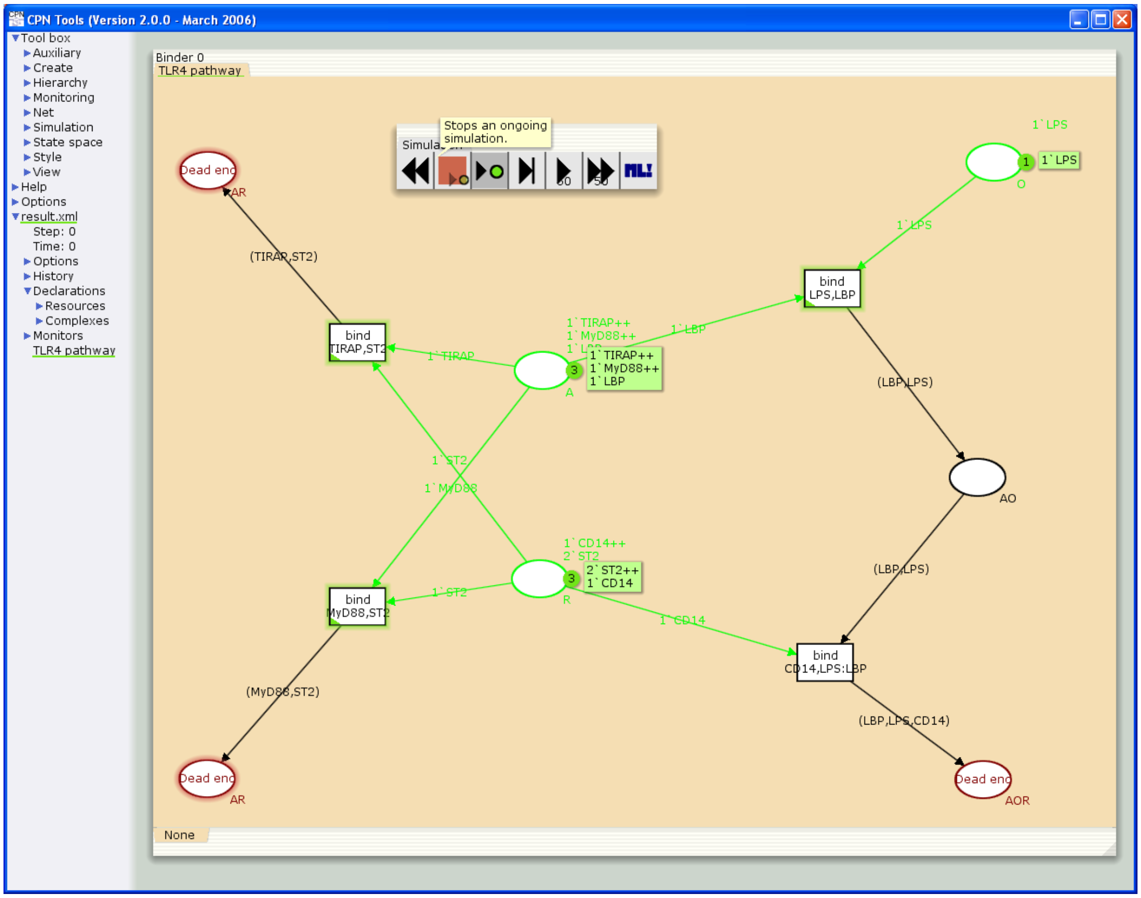


Figure 13: Final result in CPN Tools.