# 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.

E Resource - Transpath2Cpn/example.xm	I - Eclipse Platform			
<u>File Edit Navigate Search Project Run Trans</u>	spath Editor <u>Wi</u> ndow <u>H</u> elp			
📬 • 🖩 👜   🏇 • O • 🏊 •   🛷	2 - 日 - 寺 今 - 今 -	😫 🔂 Resource		
Project Explorer 🛛 📄 🔩 🌄 🗖	😥 example.xml 🕴			
	Construction         Construction           Resource Set         Proceeding           Participation         Metwork           Point         Proceeding           Proceeding         Proceeding			
	Selection Parent List Tree Table Tree with Columns			
	🐼 Tasks 🖉 Progress 🔲 Properties 🕱 🗧 🗄			
	Property	Value		
	Catalyzes Coefficient			
	Id	IIII MO000016882		
	Inhibits Coefficient			
	Klass	12 <u>-</u>		
	Molecule Type	u≣ other		
	Name	비로 LPS		
	Rkins Coefficient			
	Rkouts Coefficient	Reactants Coefficient 1 LPS		
	States Of			
	Synonyms			
< S				
Selected Object: [MO000016882] Mol	ecule 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.



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

#### 3 Running the example

In order to execute the transformation, the user must select <u>R</u>un as  $\rightarrow \underline{1}$  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.

t configur The required es.upv.dsic.is	ation and launch. feature 'modelPath' of si.gvt.launcher.model.gvtir	nvocation.impl.DomainIm	회@b2bde5{In%20memory%20resc	ource#//@domains.0}' must be
me: Transp	athToCPN			
Arguments				
QVT Transfor	rmation			
TranspathTe	oCPN (platform:/resource/	Transpath2Cpn/transpat	12cpn.qvt)	Browse
Input Paramo	eters			
Target	Domain name	Metamodel	File	
-	tpDomain	transpath		R
×	UNDOM:	Lpri		- w
				ApplyRevert

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.

Select input model	Select input model
Transpath2Cpn	☐ ☐ Transpath2Cpn example.xml transpath2cpn.qvt
File name:	File name:
/Transpath2Cpn/example.xml	/Transpath2Cpn/result.cpn
⑦ CK Cancel	⑦ OK Cancel

(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.

it configur	ation and launch.			D
me: Transp	athToCPN			_
Arguments				
QVT Transfo	rmation			
TranspathT	oCPN (platform:/resource/1	Transpath2Cpn/transpat	h2cpn.qvt) Brow	se
1				
Target	Domain name	Metamodel	Fie	
- Target	toDomain	transpath	/Transpath2Con/example.xml	-
√	cpnDomain	cpn	/Transpath2Cpn/result.cpn	
				_
			Apply Re	vert
			·v	
			Run	Close

Figure 6: The transformation and its arguments.

### 4 Result files

Once the transformation has been configured, the <u>Run</u> 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.



Figure 7: Result files.

E Resource - Transpath2Cpn/result.cpn -	Eclipse Platform		
<u>File Edit N</u> avigate Se <u>a</u> rch Project <u>R</u> un <u>C</u> pn E	iditor <u>W</u> indow <u>H</u> elp		
i 🗂 • 🖩 🖻 i 🏇 • O • 🏊 • i 🛷	월 - 월 - 박 <b>승 -</b> 수 -		E Resource
🎦 Project Explorer 🛛 📄 🔄 🏹 🖓 🖓	🚱 result.cpn 🖾		- 8
Transpath2Cpn Result.cpn Result.cpn Result.cpn Result.cpn Result.cpn Result.cpn.qvt Result.cpn.		an 11.BP:CD14 RAP V988 JPS1LDP	
		4 <=> LPS:LBP:CD 14}	<u>×</u>
	Selection Farent List Tree Table Tree With Colur	mins	
	Tasks C Progress Properties 🛛		
	Property	Value	~
	Arts Fill Colour Fill Filled Fullon Hight Id Line Colour Line Thick Line Type Posx	<ul> <li>✓ Art (512 + 11KAP &lt;=&gt; 512:11KAP) =&gt; (512:11KAP)</li> <li>✓ Art (512 + 11KAP &lt;=&gt; (512:11KAP)</li> <li>✓ False</li> <li>☑ Fold</li> <li>☑ Fold</li> <li>☑ F1 1</li> <li>☑ Sold</li> <li>☑ 0</li> </ul>	
< S	<	i og torefore linde	>
Selected Object: Place ST2:TIRAP			



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.



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.

Resource - Ecli	pse Platform		State of Control of Co	
Eile Edit Navigate	Search Project Run	Windov	/ Help	
i 📬 • 🖾 🖾 i	<b>☆・Q・Q</b> ・	1	왕 - 장 -	$\diamond \diamond \cdot \diamond \cdot$
Project Explorer	8 🗐 🛱 🗸 🗖			
😑 छ Transpath20	on xml			
result.cp	New		ľ.	
📄 transpat	Open F3 Open Wit <u>h</u>	٠		
	Ctrl+C Paste Ctrl+V			
	X Delete Delete Mo <u>v</u> e Rena <u>m</u> e F2			
	<mark>≧n I</mark> mport ☑ Exp <u>o</u> rt			
	શ Refresh 🛛 F5			
An outline is not avail,	<u>R</u> un As <u>D</u> ebug As T <u>e</u> am	) ) )		
	Comp <u>a</u> re With Rep <u>l</u> ace With	:		
	Intergenomics	•	Save and La	yout Cpnet XML file
	EMOF	*	Save Cpnet	XML file
	Properties Alt+En	ter 7	Parse Trans	ратл хиль тне
		-	Advanced	Property Info

Figure 10: Save to CPN Tools.

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

OK Cancel

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.



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.