Project Number: 295397

VARIES

VARiability **I**n safety-critical **E**mbedded **S**ystems

ARTEMIS-2011-1

D4.3 BVR Tool

Start date of project: May, 1st 2012
Duration: 36 months

Deliverable	**D4.3 BVR Tool**		
Confidentiality	PU	**Type**	R
Project	VARIES	**Date**	2015-01-29
Status	FINAL	**Version**	01
File Identifier	VARIES_D4.3_v01_PU_FINAL		

Contact Person	**Øystein Haugen**
Organisation[1]	SINTEF
Phone	+47 913 90 914
E-Mail	oystein.haugen@sintef.no

[1] See Appendix I (Partner identification) on page 24.

THE RESEARCH LEADING TO THESE RESULTS HAS RECEIVED FUNDING FROM THE SVENTH FRAMEWORK PROGRAMME AND ARTEMIS-JU UNDER GRANT AGREEMENT N° 295397

AUTHORS

Name	Organization1	E-Mail
Øystein Haugen	SINTEF	oystein.haugen@sintef.no
Anatoly Vasilevskiy	SINTEF	anatoly.vasilevsiy@sintef.no

REVIEWERS
(of the current version)

Name	Organization1	E-Mail
Geert Vanstraelen	Macq	Geert.vanstraelen@macq.eu

CHANGE HISTORY

Version	Date	Reason for Change	Sections / Pages Affected
01	2015-01-29	First official version	All

CONTENT

LIST OF FIGURES

LIST OF TABLES

Appendix

1 Introduction

1.1 Overview, Purpose and Scope

This document describes our BVR Tool. The deliverable is indeed the tool itself, but this document serves as a placeholder and documentation.

The goal of this document is to serve as documentation to installation and usage of the BVR Tool.

For references to the work plan, see [VARIES-TA], B3.3.

1.2 Partner involvement

Table 1. Partner involvement

Partner	Involvement
SINTEF	Creators of the tool
Macq	Internal review
Autronica	Users of early versions of the tool. Requirements for the tool. Usecases.

2 Installing the software

This chapter tells you how to install the tool.

2.1 The Eclipse Base

Please install an Eclipse modeling tools version.

We have done most experiments on Eclipse version Kepler:

http://eclipse.org/downloads/packages/eclipse-modeling-tools/keplerr

The plugins should also work now on Linux as well as Windows. Make sure to have Java 8. The VSpec editor and the resolution editor should work also on Eclipse Luna, but the realization on Papyrus is dependent on Kepler and Papyrus 0.9. We will port the base model coloring interface also to Papyrus 1.0 in the follow-up version.

2.2 BVR Tool Update site

The BVR Tool update site can be found at:

http://bvr.modelbased.net/update/site.xml

PLEASE look at this video before you install the BVR Tool Bundle and follow it closely:
http://heim.ifi.uio.no/anatolyv/demo/installation.swf

3 RUNNING THE BVR TOOL

You may now run the software. Please look at: Demo1 http://heim.ifi.uio.no/anatolyv/demo/demo.swf and Demo2 http://bvr.modelbased.net/demo/

3.1 THE VERY START – MAKING A PROJECT

- Create your own workspace
- Create any project
- Rightclick the project, select *New / Other* and find *BVR model* under *Example EMF Model Creation Wizards* (The easiest way is to search for "BVR" once you have selected New/Other.)
- Fill in your own preferred name(s)
- Select BVR Model from dropdown menu "Model object"
- You will get a very simple tree browser with a minimal model up, close that tab

3.2 THE VSPEC EDITOR (MVC)

MVC means Model View Controller. We have also had other plugins for VSpec editor, but in the current version we only include the MVC editor. Its main asset is that the whole tree structured is rearranged for every modification to preserve the visual aesthetics.

3.2.1 VSPEC EDITING

We will in the next pages show how a VSpec model is created and edited.

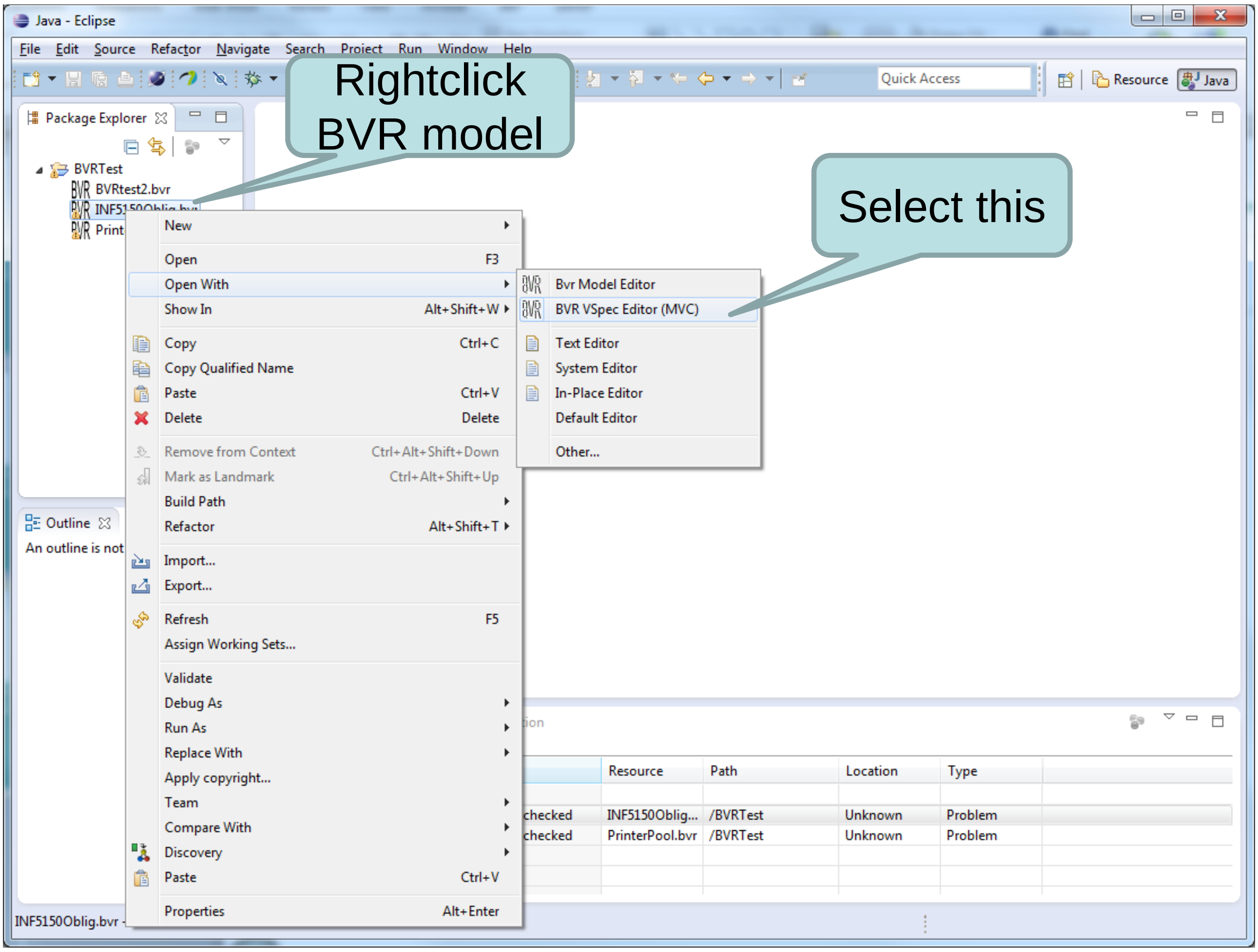


Figure 1 Opening the VSpec (MVC) editor

When the (almost) empty model has been opened the editor looks like Figure 2.

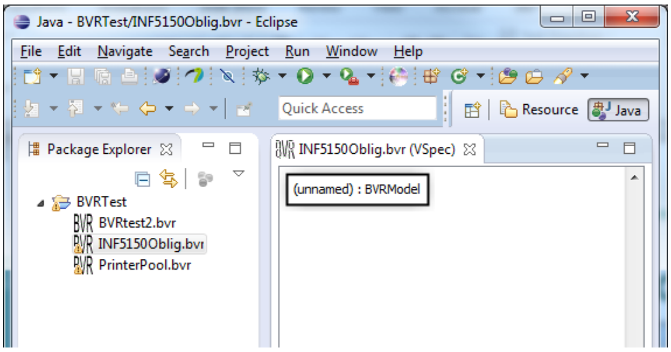
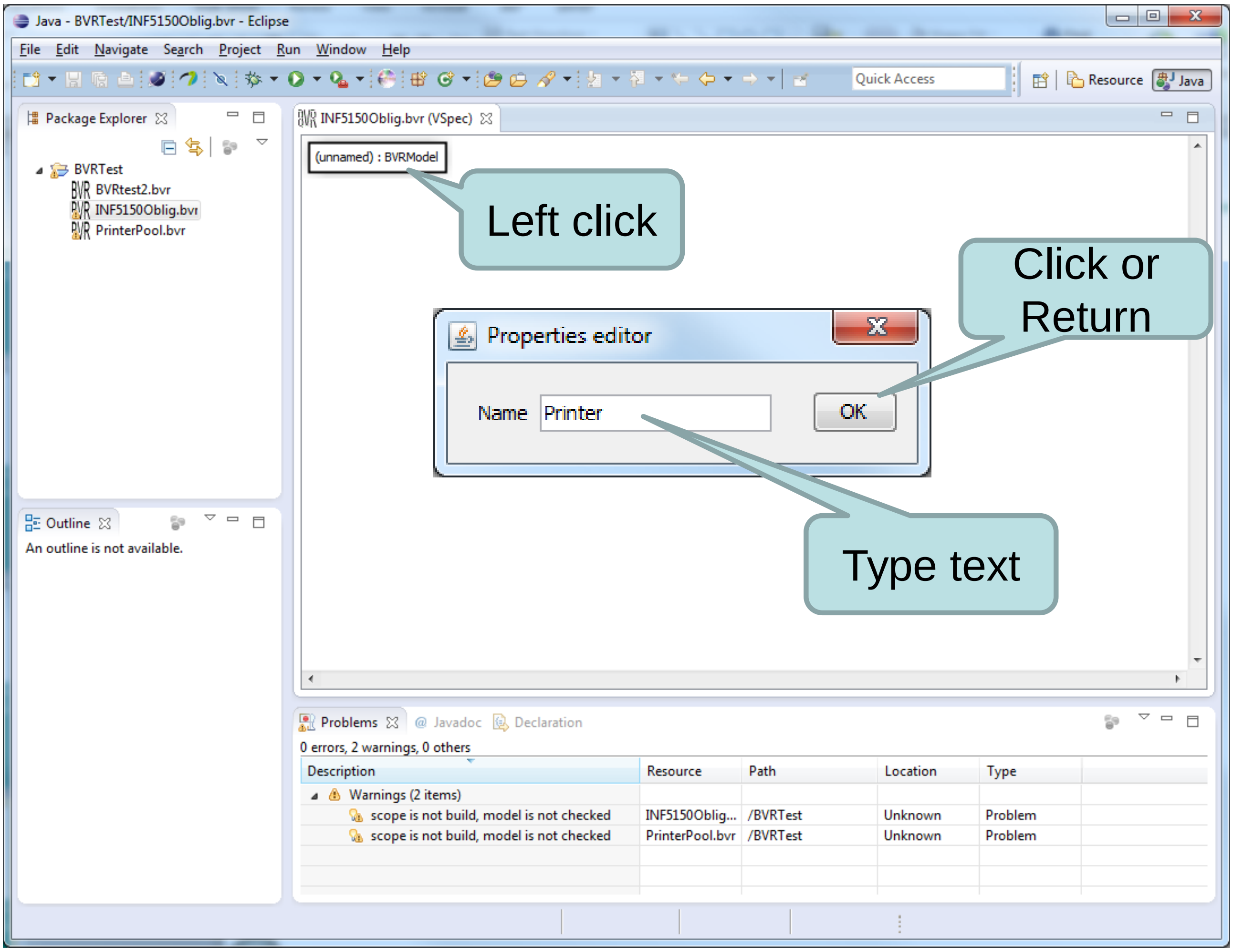


Figure 2 The almost empty model in VSpec editor

Now you may leftclick on the object and change the name (and other properties) as shown in Figure 3.



Name Printer

An outline is not available.

0 errors, 2 warnings, 0 others

Description	Resource	Path	Location	Type
Warnings (2 items)				
scope is not build, model is not checked	INF5150Oblig...	/BVRTest	Unknown	Problem
scope is not build, model is not checked	PrinterPool.bvr	/BVRTest	Unknown	Problem

Figure 3 Changing name by leftclicking

You may now start to build the tree from the starting node by rightclicking.

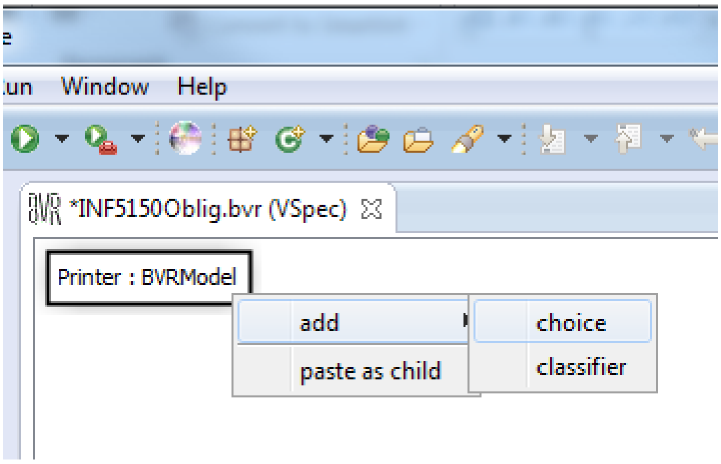


Figure 4 Rightclicking to modify the tree

The menu selection "add" will give a context specific pop-up menu with those alternatives that are syntactically eligible at this point in the tree.

A leftclick on the created element allows to change its name.

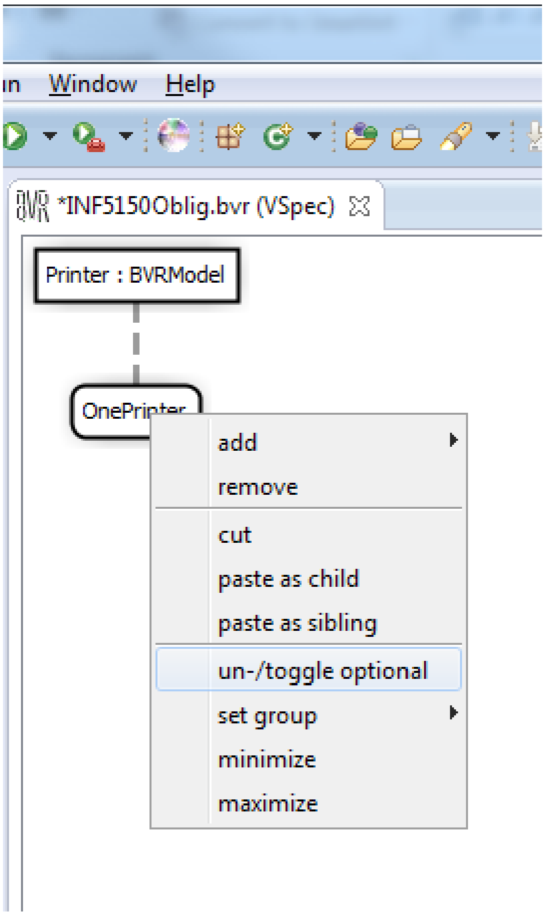


Figure 5 Optional or mandatory VSpecs

To define whether a VSpec child shall be optional (with dashed line) or mandatory (with solid line) can be toggled by a menu selection.

To change the group multiplicity select another menu option, namely "set group" as shown in Figure 6.

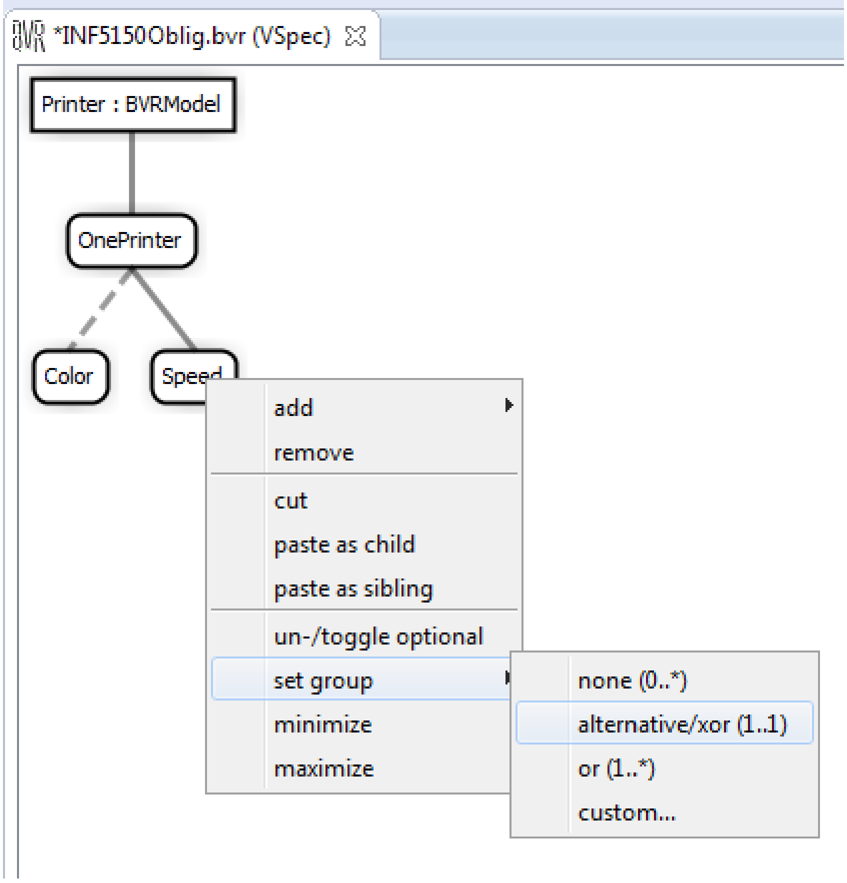


Figure 6 Selecting group multiplicity (here: 1..1)

This results in the appearance of a small triangle to denote the group multiplicity.

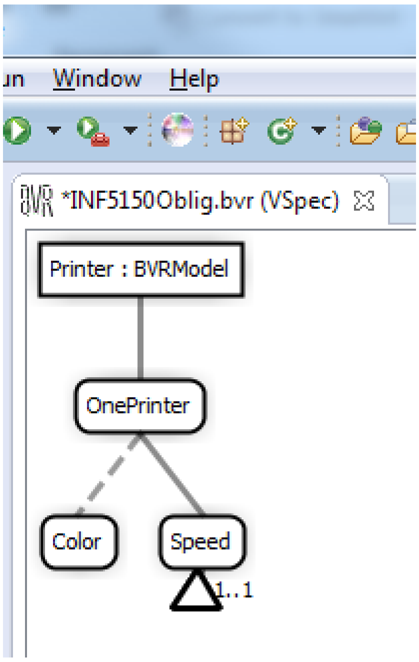


Figure 7 Group Multiplicity notation inserted

Now the user can add choices (say: *X20* and *X22*) below *Speed* following the procedure for *Add* given above from Figure 4. They will become alternatives where only one can eventually be selected.

Having created the choices, we can go on to defining some explicit constraints.

Constraints are placed to the left of the children and the content of the constraint is directly parsed. If the constraint content is not parsed to be correct, no changes will be done to the constraint, and you will still have the chance to edit.

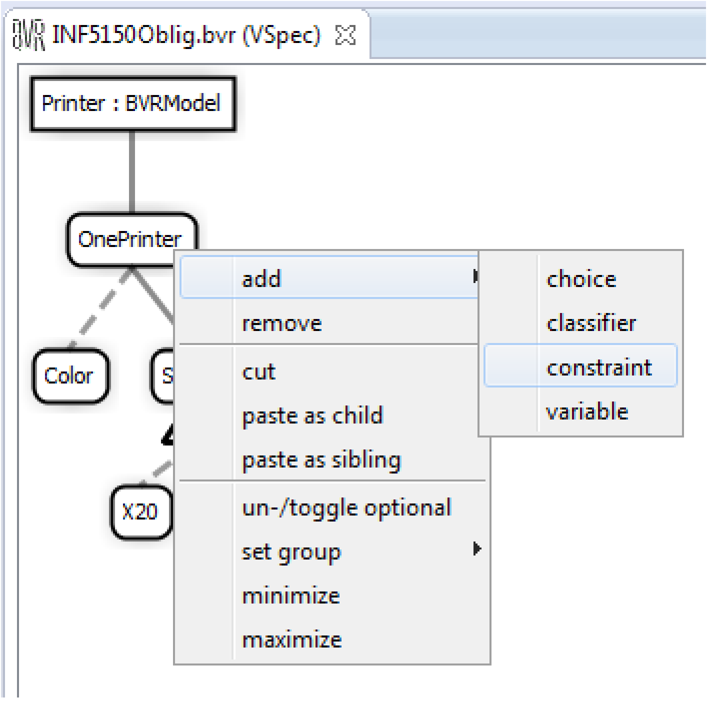
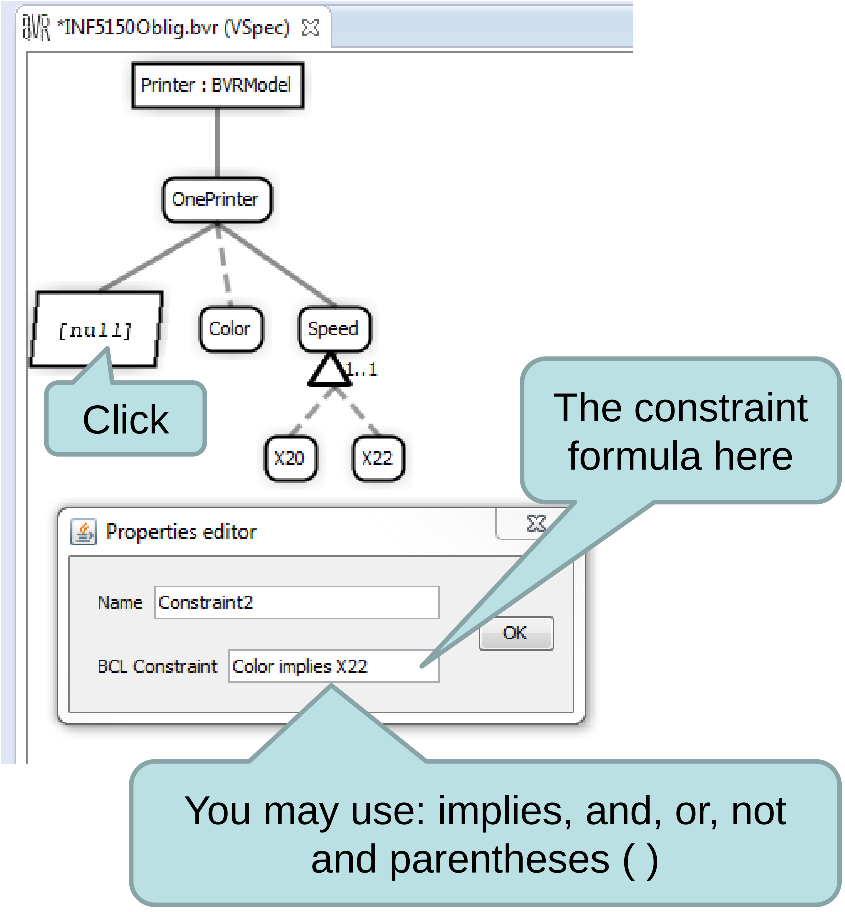


Figure 8 Inserting constraint

Figure 9 shows the editing of a constraint.



*INF5150Oblig.bvr (VSpec)
Printer : BVRModel
OnePrinter
[null]
Color
Speed
1..1
X20
X22
Click
The constraint formula here
Properties editor
OK

Name: Constraint2

BCL Constraint: Color implies X22

You may use: implies, and, or, not and parentheses ( )

Figure 9 Editing a constraint

3.2.2 Saving on BVR-format and Outputting a diagram to PNG format

If you want to print or otherwise apply the diagrams for visualization, you can produce a png-file by following the procedure indicated in Figure 10.

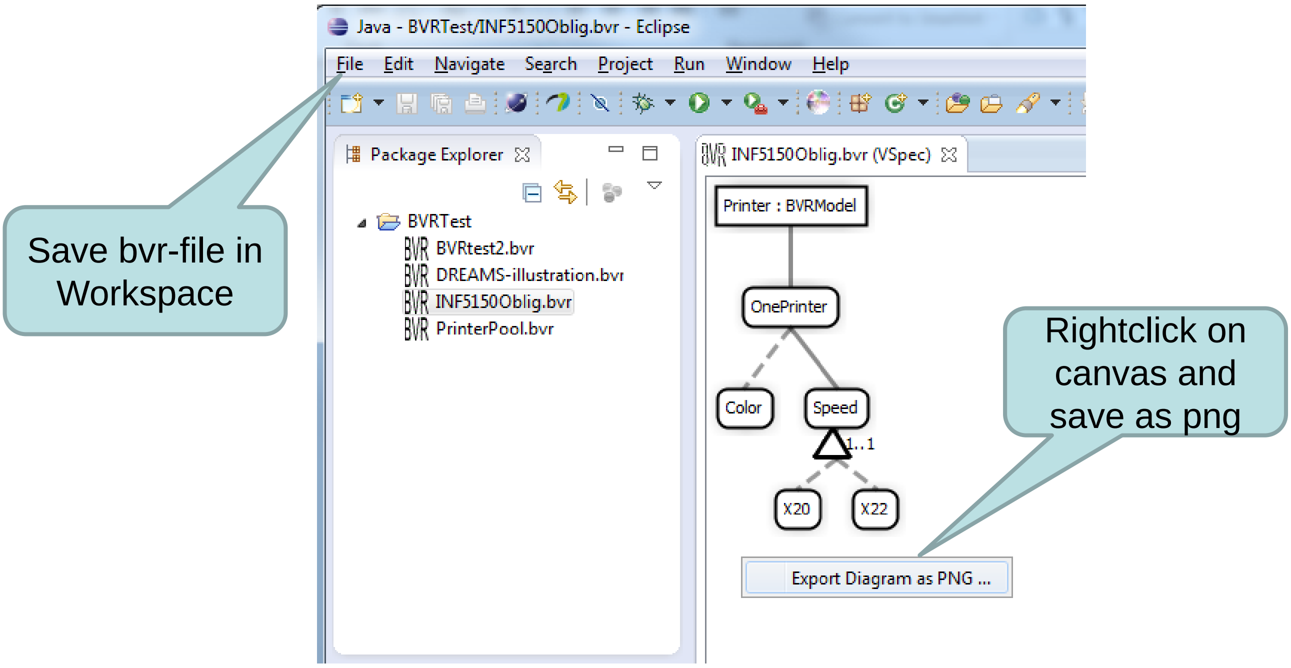

Java - BVRTest/INF5150Oblig.bvr - Eclipse
File Edit Navigate Search Project Run Window Help
Package Explorer
BVRTest
BVRtest2.bvr
DREAMS-illustration.bvr
INF5150Oblig.bvr
PrinterPool.bvr
INF5150Oblig.bvr (VSpec)
Printer : BVRModel
OnePrinter
Color
Speed
1..1
X20
X22
Export Diagram as PNG ...
Save bvr-file in Workspace
Rightclick on canvas and save as png

Figure 10 Saving and outputting to picture

3.2.3 Practical view options for better overview

We have also included mechanisms to collapse and expend subtrees such that the user can focus on exactly what he needs.

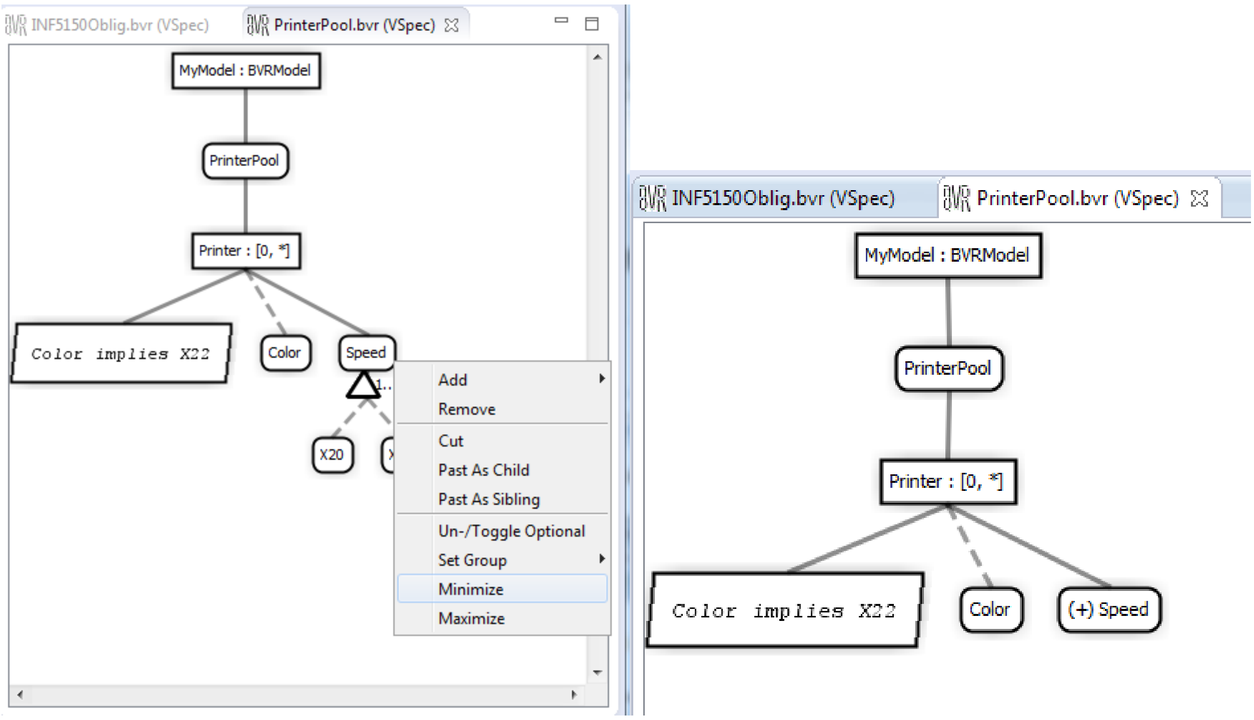


Figure 11 Minimize

The 'maximize' function will restore the full subtree.

In Figure 11 we use an example where *Printer* is a VClassifier. VClassifier is a VSpec that includes instance multiplicities meaning that a range is given to indicate the number of possible instances of the VClassifier.

3.3 The Resolution Editor

Once the variability model (the VSpec model) has been created, you may define what configurations (or resolutions) you would like. This is done by the resolution editor.

Here is how you create the resolution model from a VSpec model.

- While still retaining the variability model in one diagram, open the very same model in the Resolution Editor (MVC),
- right-click on the canvas in that opened pane and select "new". This creates at least one top element of the resolution model. More elements are also generated to correspond to decisions that have to be taken. VClassifers are instantiated according to their lower instance multiplicity bound.
- Then arrange the panes beside each other on the Eclipse canvas like shown in Figure 12.

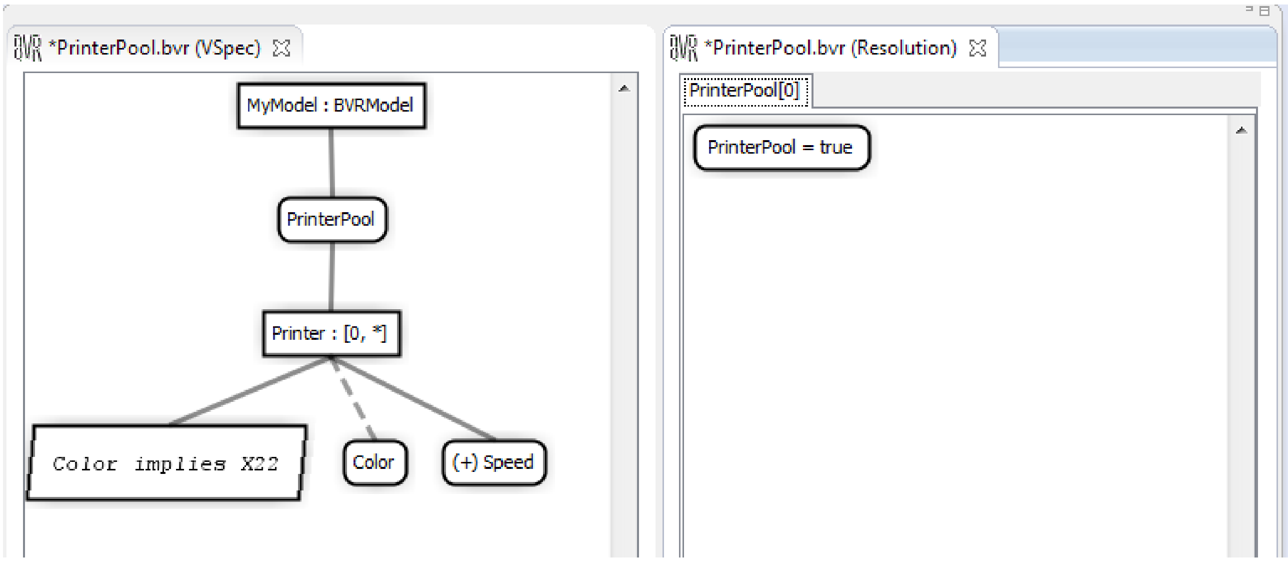


Figure 12 Opening and creating a resolution model start

3.3.1 Manually creating resolution elements

Rightclicking on an element in the resolution model will give you a menu where the possible VSpec elements to resolve are shown.

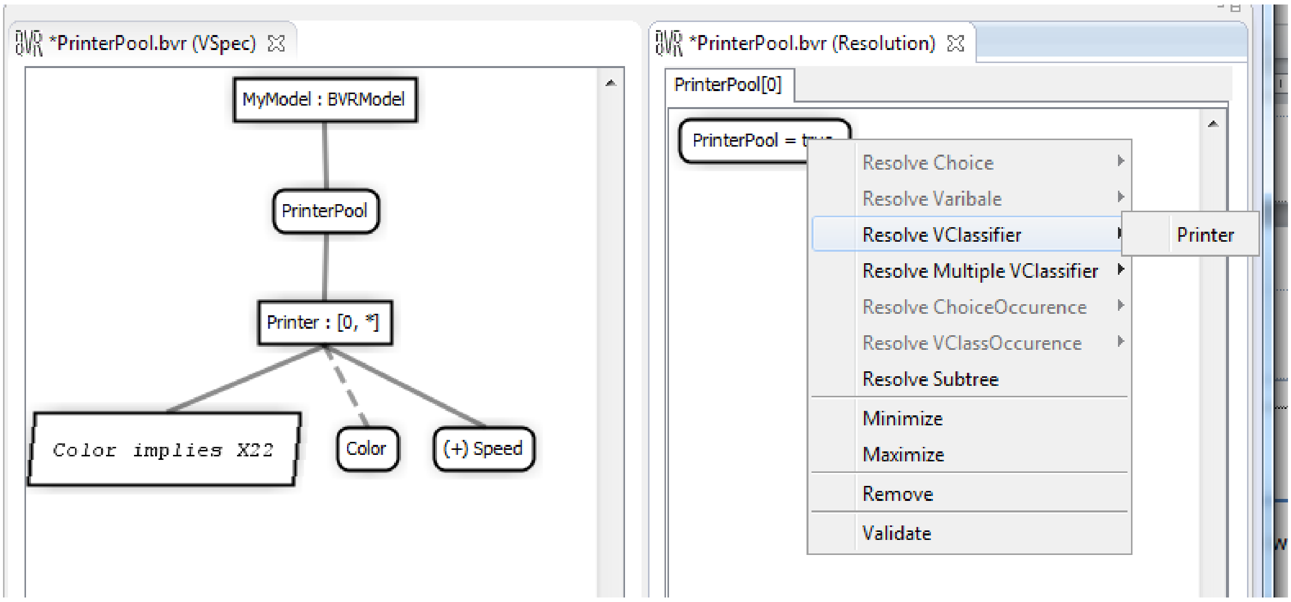


Figure 13 Manual resolution

In Figure 13 we show that the VClassifier Printer is resolved (one instance created) below the PrinterPool on which the user rightclicks. The result is shown in Figure 14.

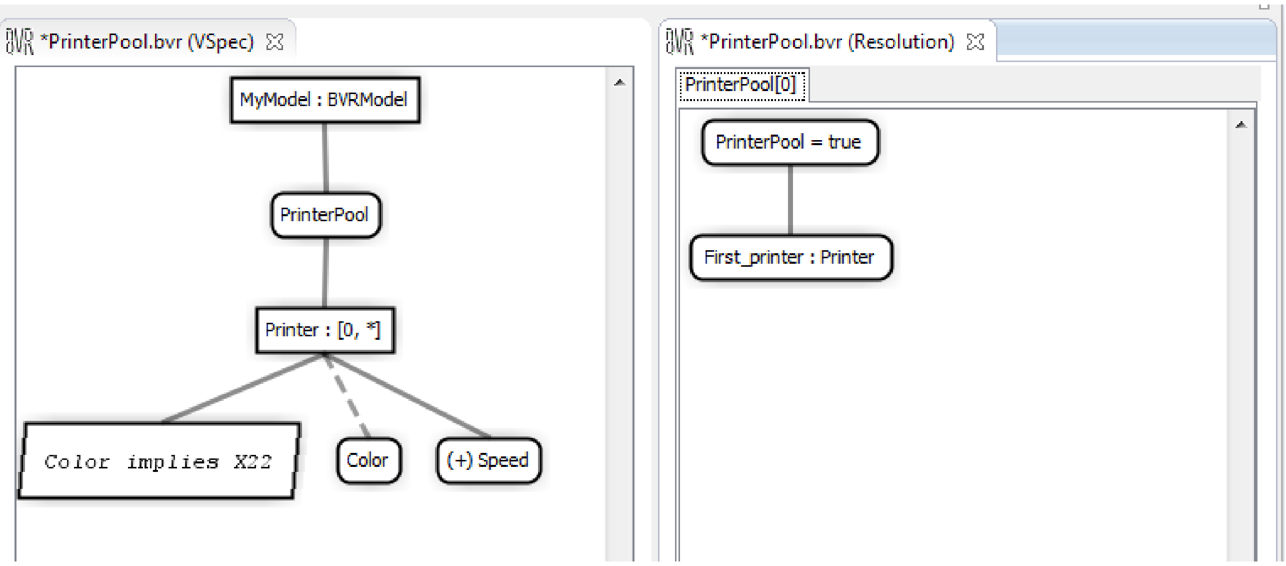


Figure 14 Created First_printer (name given by leftclicking)

3.3.2 Generate a default resolution start

Now that we have created the first printer as an instance of a VClassifier, we can take advantage of a mechanism to resolve the subtree below that VClassifier as far as possible. This is done by the menu option 'Resolve subtree' from rightclicking on a node. With variability models containing only choices, the resolution to defaults are done on first generation. The result is a resolution subtree where all choices are resolved to True, but all choices to be true may not be according to the constraints and to see where there are still some manual decisions to be taken the user may rightclick on the canvas and select the menu option 'Show/hide grouping', and red triangles will appear on roots which have not been properly resolved on their group multiplicity. An example is shown in Figure 15. Notice that the consequences of explicit constraints are not covered by this visualization.

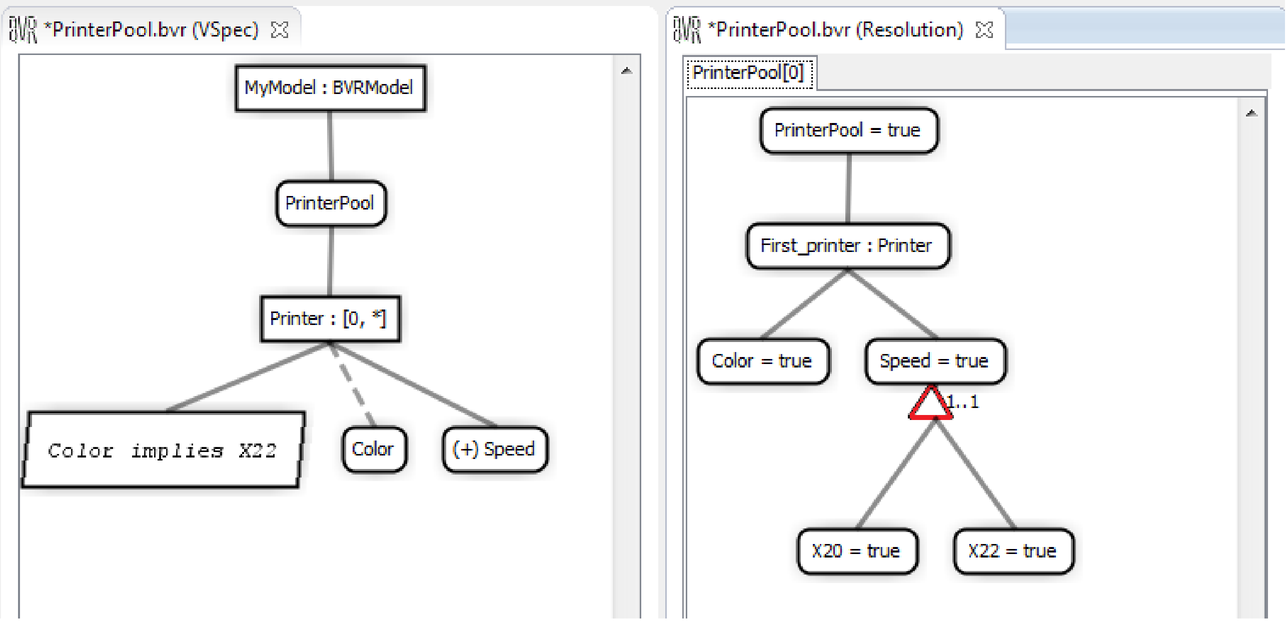


Figure 15 Resolution model where needed decision on group multiplicity is indicated

What we need to do here is to select one speed. The explicit constraint says that when Color is selected we must select X22, but this is not enforced by the tool directly.

Toggling the value of a choice resolution is done by simply leftclicking on the node.

3.3.3 Validation

Once a resolution model has been made, or even a part of it, it is possible to provide some validation that the resolution corresponds to the variability model.

Caveat: The validation operation only works for models that contain only Choices and BCL Constraints. This means that with models including VClassifiers and Variables, you will get an error message.

With the variability models of only choices you may rightclick on a node and perform validate. The node's subtree will be validated against the corresponding part of the variability model.

The local validation of a part (or the whole) of a resolution model is executed by rightclicking on a node and selecting the Validate operation. The result of the operation will be shown in a separate dialog box which in case of error indicates where the error can be found.

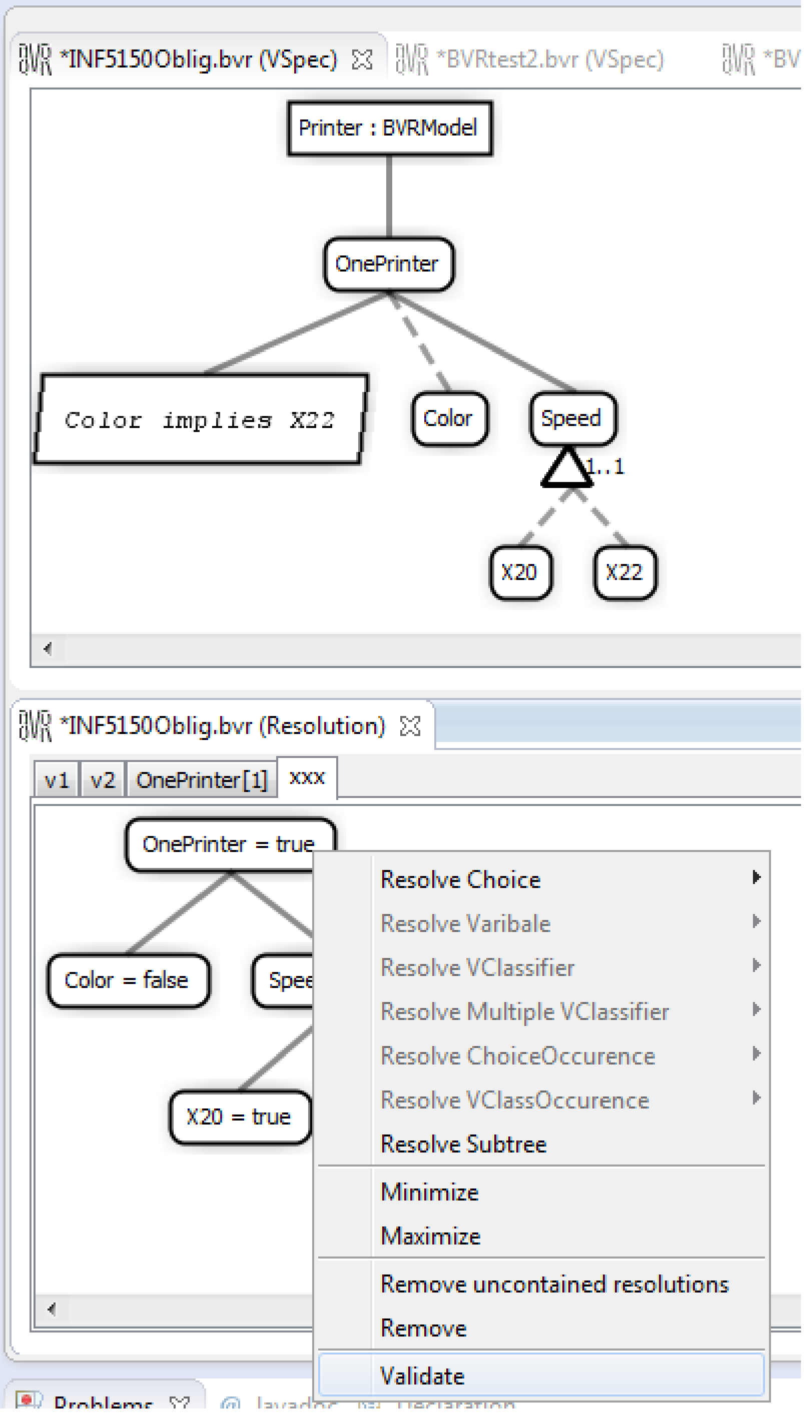
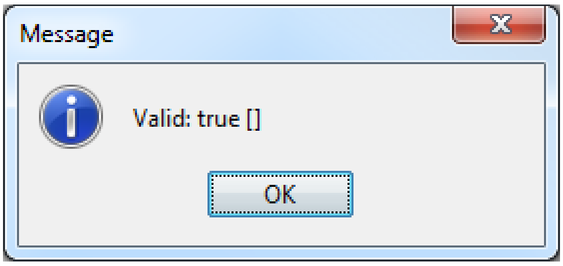


Figure 16 Local validation

In Figure 16 we see how the local validation menu looks. In this case the node is the top one of a resolution and the result of the validation is shown in a dialog as depicted in Figure 17.



Valid: true []

Figure 17 Response dialog box – here the subtree is a valid resolution

We have a few interesting global operations as well. These are reached by rightclicking on the bar beside the resolution model tabs. These operations are global and refer to all the resolutions.

Within the BVRTool there is a special tool (called SPLCA) that makes it possible to generate a covering array that is either 1-wise, 2-wise or 3-wise. This means that a kind of optimal set of resolutions are generated, typically for the purpose of testing the product line. We show in Figure 18 such an operation which is reached from a blank canvas or the bar beside existing resolutions. In the case of existing resolutions, they will be scratched when the new set of optimal resolutions is generated.

The ICPL resolution selection tool is made by Martin F. Johansen in conjunction with his PhD work. Please refer to his thesis available at http://duo.uio.no/ (Permanent link to his thesis: http://urn.nb.no/URN:NBN:no-40483)

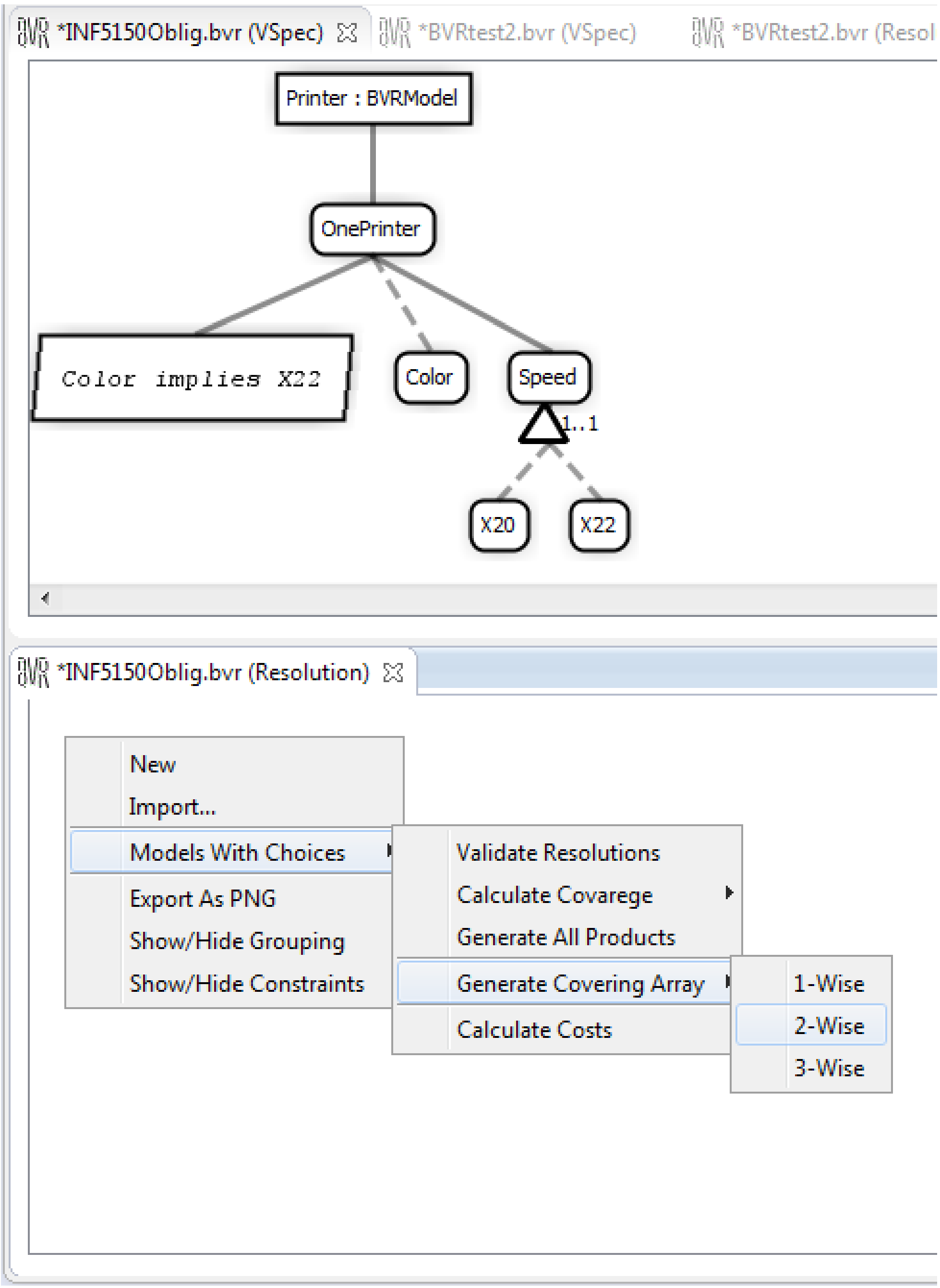


Figure 18 Generating covering arrays

As can be seen from the menu depicted in Figure 18we can also generate all products, but one should be very careful because there may be a lot more resolutions possible than one imagines.

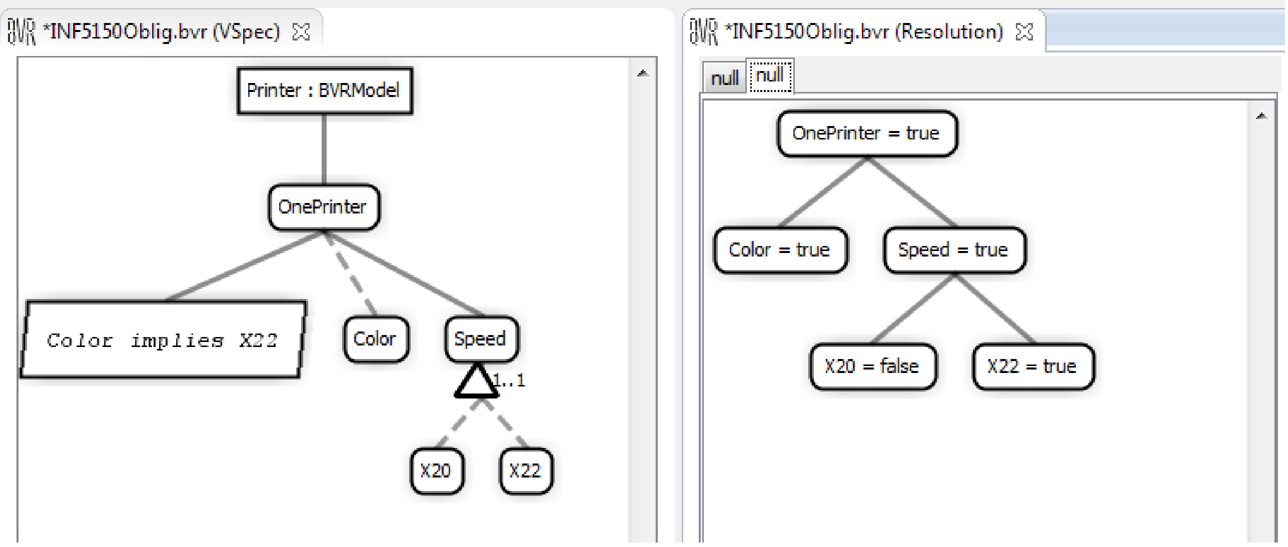


Figure 19 SPLCA Tool for resolution generation

In the variability model of Figure 19 we have that every possible resolution amounts to 3, while a 1-wise generation yields 2. When numbers are this small, the differences are small, too, but when the number of features increases the coverage array generation becomes increasingly difficult and the effect of using it is also increasing.

3.4 The Realization Editor and Materialization

Once the resolution models have been finalized and we want to generate the corresponding product models, we need to have defined the mappings between the variability model and the base model.

The main idea is that the base model(s) will be maintained with tools specific to these base languages while the BVR Tool should be able to connect seamlessly to these base language editors.

The BVR Tool requires that 4 functions are implemented in the base language tool for highlighting and selection. The BVR Tool then can effectively cooperate with the base editor.

The following video tries to explain how the realization and materialization works with the BVR Tool:

http://bvr.modelbased.net/demo/

Please go to the middle of that short video if you do not want to get a repetition of variability and resolution modeling. The tool version is slightly older than this one, but for the realization modeling there is hardly any visible changes.

We have made a connection to the older Papyrus 0.9 and will soon make one for Papyrus 1.0. The old interface will not work for Papyrus 1.0.

The coloring scheme does work for a BVR Tool enabled editor for the generic EMF tree editor.

4 Abbreviations and Definitions

Term	Definition
BVR	Base Variability Resolution

5 References

Author, Year	Authors; *Title*; Publication data (document reference)
VARIES-TA	ARTEMIS Call 2011, ARTEMIS-2011-1, 295397 VARIES: VARiability In safety-critical Embedded Systems. Technical Annex.
VARIES D4.2	The BVR Language Manual.

6 APPENDIX

Appendix I. PARTNER IDENTIFICATION

Short name	Full name	Type	Country
Atego DE	Atego GmbH	SME	DE
Atego UK	Atego Systems Ltd	SME	UK
Autronica	Autronica Fire & Security AS	LE	NO
Barco	Barco NV	LE	BE
B&M	Berner & Mattner Systemtechnik GmbH	LE	DE
FMTC	Flanders' Mechatronics Technology Centre vzw	RES	BE
Fraunhofer	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.	RES	DE
HI-Iberia	HI-Iberia Ingenieria y Proyectos SL	SME	ES
HiQ	HiQ Finland Oy	LE	FI
IMDEA	Fundation IMDEA Software	RES	ES
ITU	IT University of Copenhagen	RES	DK
Macq	Macq	SME	BE
Metso	Metso Automation Oy	LE	FI
Mobisoft	Mobisoft Oy	SME	FI
pure-systems	pure-systems GmbH	SME	DE
SINTEF	STIFTELSEN SINTEF	RES	NO
Sirris	Sirris c.d.g.	RES	BE
SKS	SoftKinetic Sensors	SME	BE
Spicer	Spicer Off-Highway NV (Dana)	LE	BE
TECNALIA	TECNALIA Corporación Tecnológica	RES	ES
TÜV	TÜV SÜD AG	LE	DE
Vlerick	Vlerick Business School	RES	BE
VTT	VTT technical research centre of Finland	RES	FI