Wizards for the OMNeT++ IDE

András Varga

OMNeT++ Workshop
March 19, 2010
Malaga, Spain

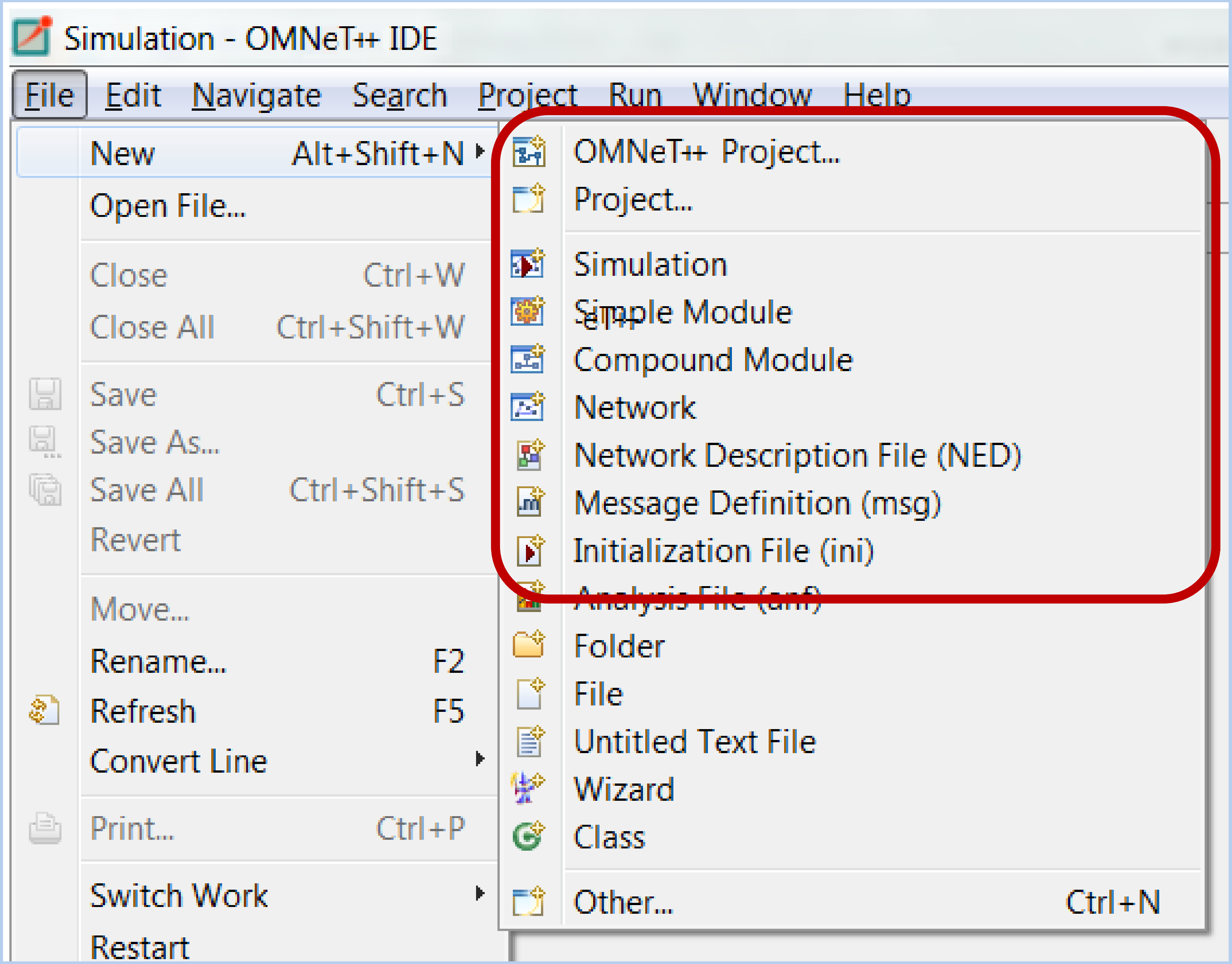
Motivation

- New users often find the IDE and the INET / INETMANET frameworks overwhelming
  - Why not help them to make the first steps?
  - They want to get a first *simple* simulation up and running quickly, so that they can start tweaking it
  - And: why not show off features? (IPv6, Ad-hoc, Mobility, MPLS,…)
- Eclipse was built to be extended
  - New tools, new editors, new views,…
  - New wizards!

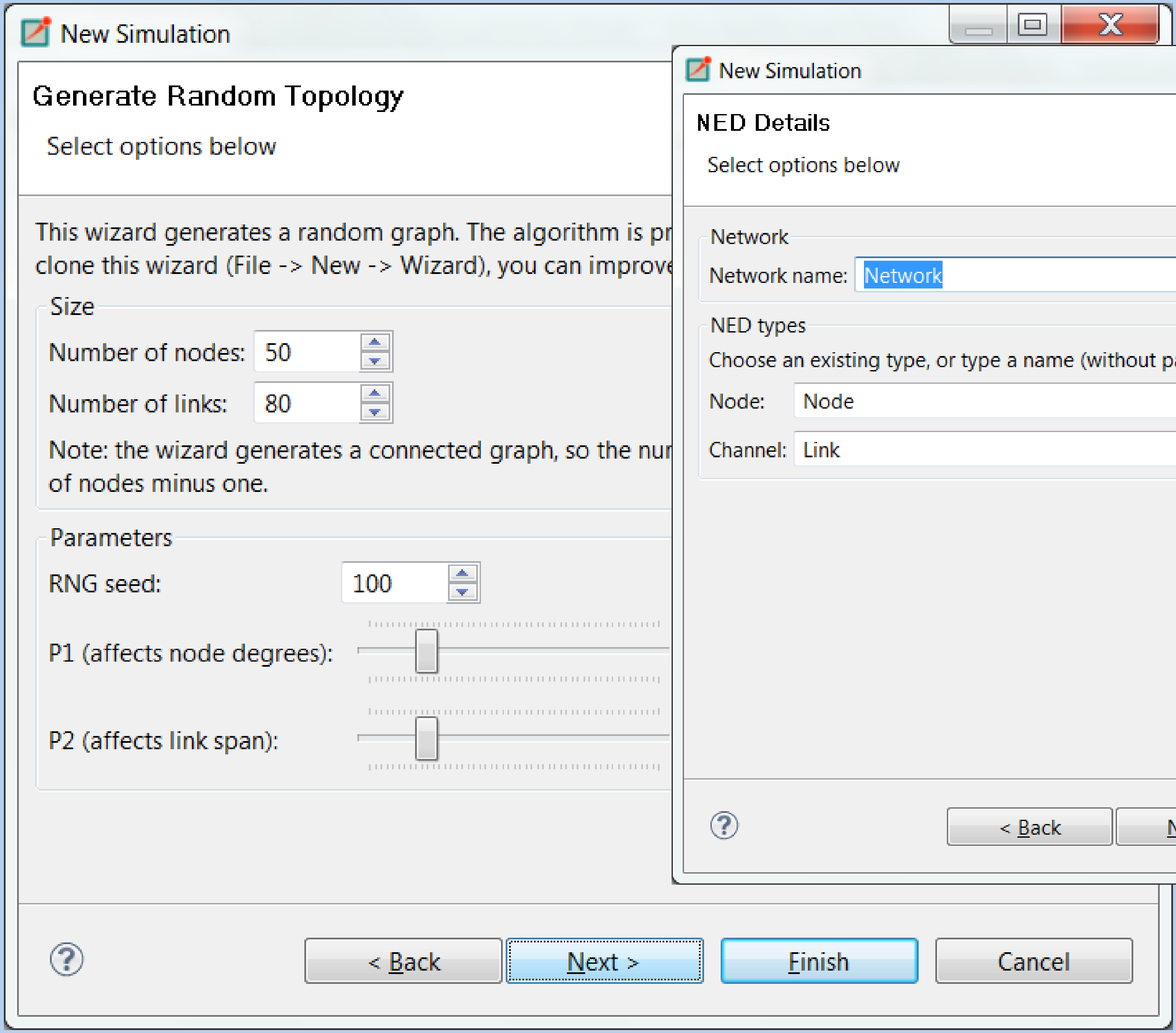
IDE Extensibility

1. Eclipse extensibility
   - features and plug-ins installed in the normal way, Help | Install New software…
2. OMNeT++ IDE loads plug-ins from projects
   - When user opens e.g. the INET project, jars in its plugins/ folder get loaded automatically!
   - Users of the project automatically get the UI extension, no extra installation step required!
   - But writing Eclipse plug-ins is hard
3. <u>Wizards contributed to the File|New dialogs</u>
   - Easy to write (little/no programming required)
   - Can be distributed with the project
   - Also automatically activated when project gets opened

Wizards in the Menu



Example: Topology Generation



New Simulation

Generate Random Topology

Select options below

This wizard generates a random graph. The algorithm is pr
clone this wizard (File -> New -> Wizard), you can improve

Size

Number of nodes: 50

Number of links: 80

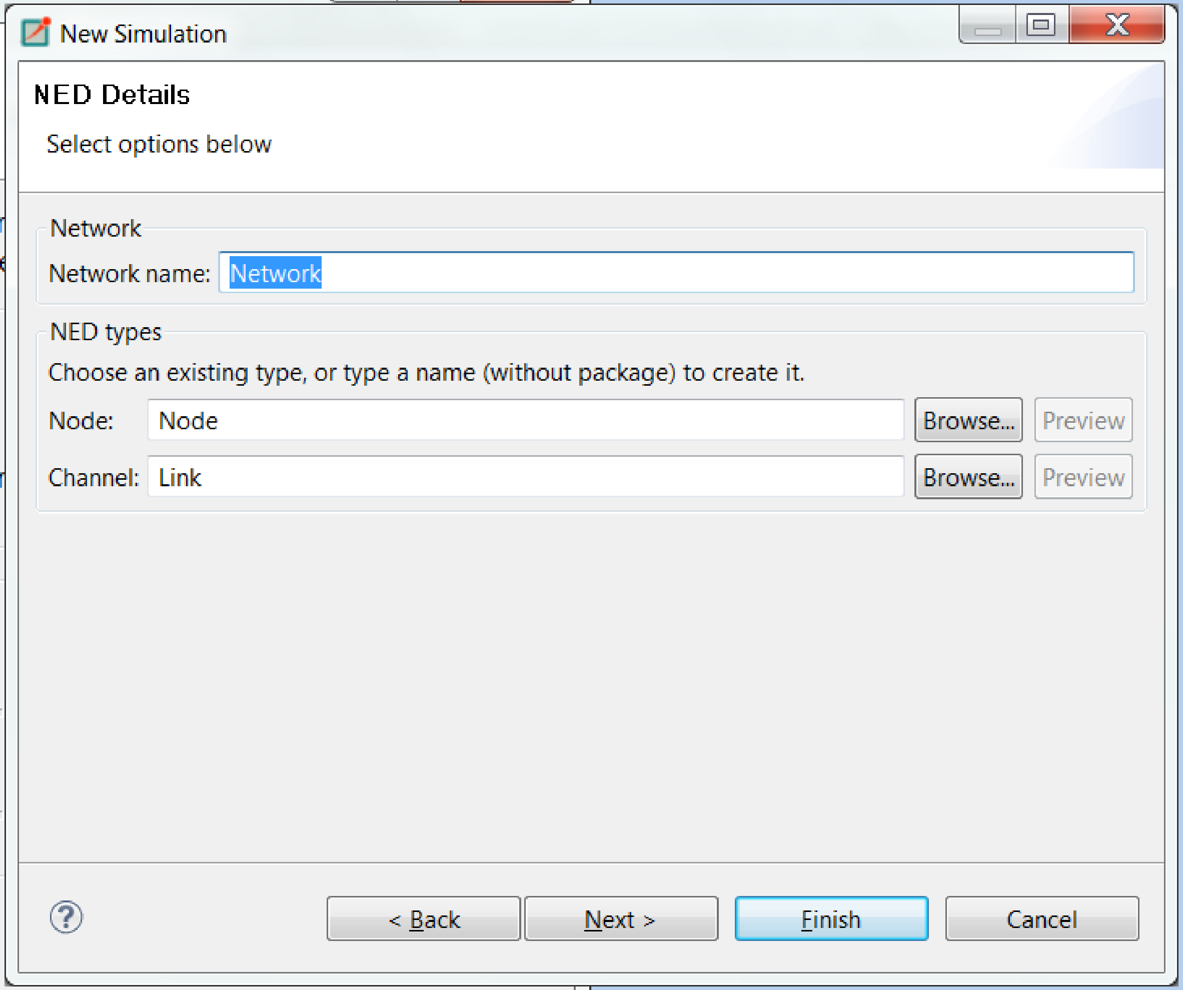
Note: the wizard generates a connected graph, so the nur
of nodes minus one.

Parameters

RNG seed: 100

P1 (affects node degrees):

P2 (affects link span):



New Simulation

NED Details

Select options below

Network

Network name: Network

NED types

Choose an existing type, or type a name (without package) to create it.

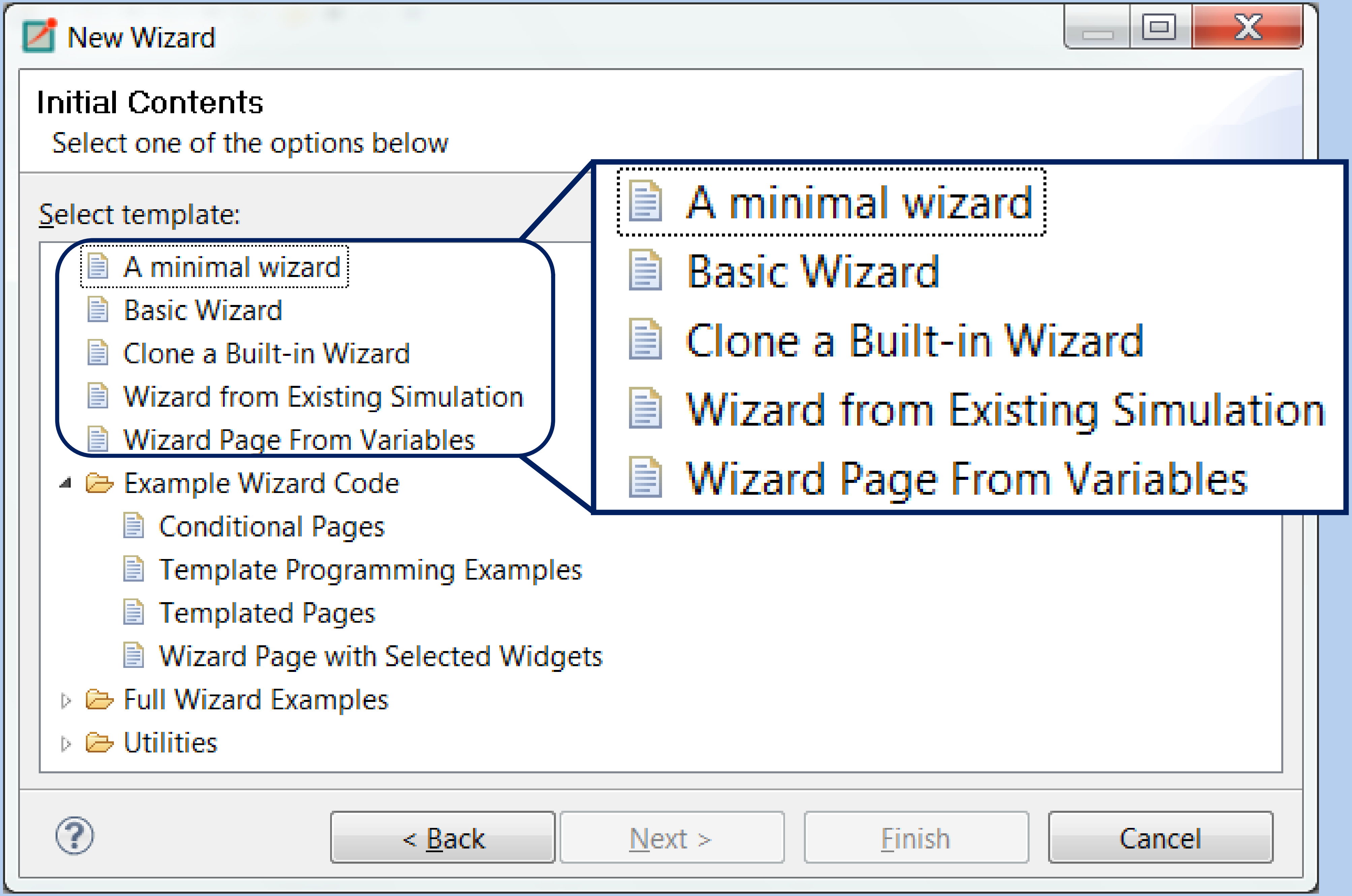
Node: Node

Channel: Link

What is a Wizard?

- Technically:
  - A templates/<*wizardname*> folder in the project
  - With a bunch of text files in it:
    - **template.properties**: declares wizard name, type, pages, etc.
    - **.xswt files:** XML files that describe the UI of wizard pages
    - **.ftl files:** will be turned into content, after substituting $variables and #if, #list, etc. constructs (ftl=FreeMarker Template Language)

Wizard-Creation Wizards



New Wizard

Initial Contents

Select one of the options below

Select template:

- A minimal wizard
- Basic Wizard
- Clone a Built-in Wizard
- Wizard from Existing Simulation
- Wizard Page From Variables
- Example Wizard Code
  - Conditional Pages
  - Template Programming Examples
  - Templated Pages
  - Wizard Page with Selected Widgets
- Full Wizard Examples
- Utilities

Creating a Wizard

Let us create a simple “New Simulation” wizard!

- It should prompt for:
  - network name
  - number of hosts
  - traffic type
- Files:
  - In folder inet/templates/newwizard:
    - template.properties
    - wizardpage.xswt
    - network.ned.ftl
    - omnetpp.ini.ftl

Example: Properties File

- template.properties:

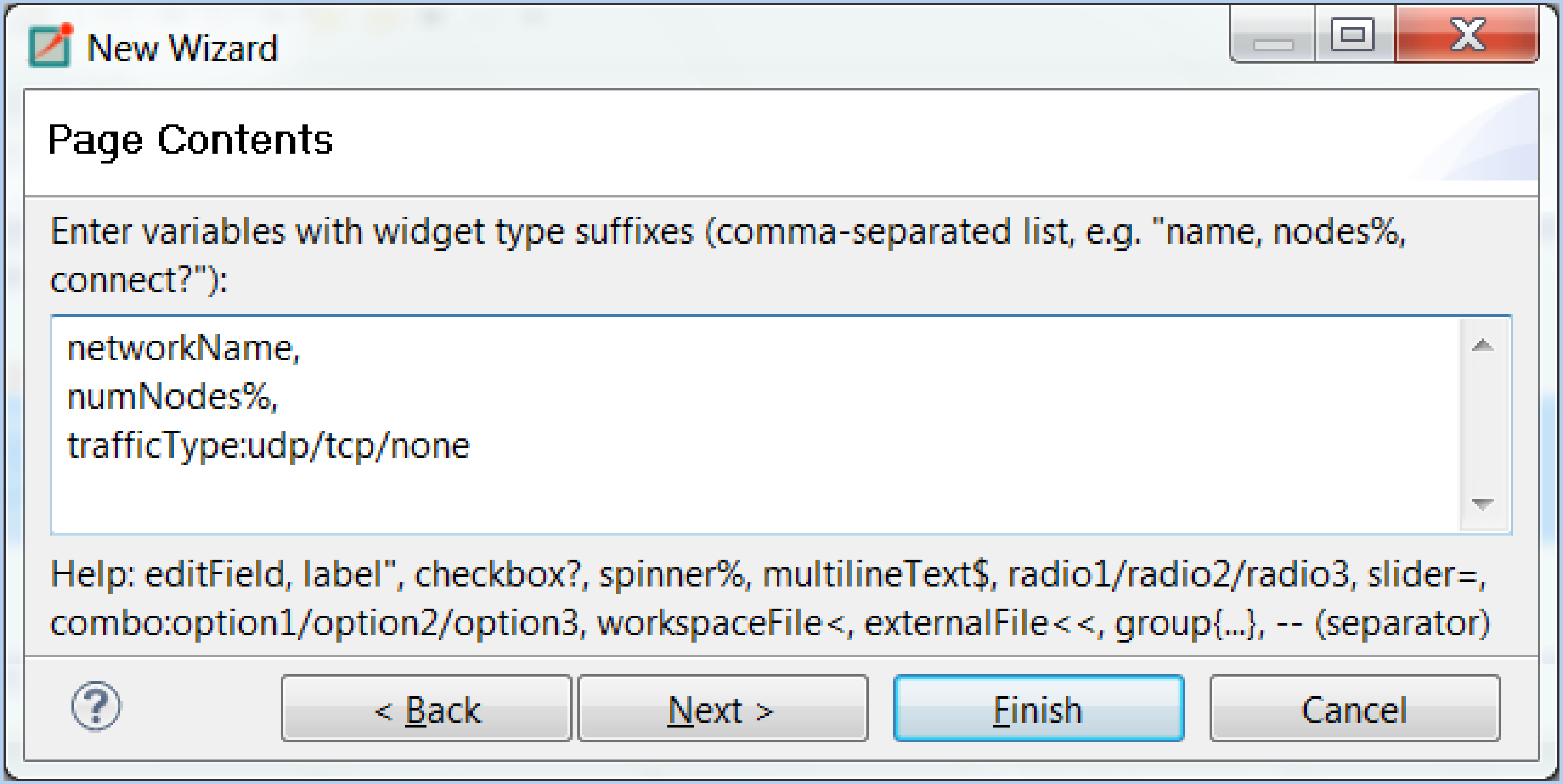
```
templateName = New Network Wizard
templateDescription = Wizard with a single input page
templateCategory = INET
supportedWizardTypes = simulation, project

# custom wizard pages
page.1.file = wizardpage.xswt
page.1.title = New Network
page.1.description = Select options below

# variables
networkName = Network
numNodes = 10
trafficType = none
```

Example: Wizard Page

Let us use the “Wizard page from variables” wizard:



New Wizard

Page Contents

Enter variables with widget type suffixes (comma-separated list, e.g. "name, nodes%, connect?"):

```
networkName,
numNodes%,
trafficType:udp/tcp/none
```

Help: editField, label", checkbox?, spinner%, multilineText$, radio1/radio2/radio3, slider=, combo:option1/option2/option3, workspaceFile<, externalFile<<, group{...}, -- (separator)

The generated wizardpage.xswt (simplified)

```
<?xml version="1.0"?>
<xswt>
  ...
  <x:children>
    <label x:text="This is a generated wizard page.."/>

    <label text="Network name:"/>
    <text x:id="networkName"/>

    <label text="Num nodes:"/>
    <spinner x:id="numNodes" minimum="0" maximum="100"/>

    <label text="Traffic:"/>
    <combo x:id="trafficType">
      <add x:p0="udp"/>
      <add x:p0="tcp"/>
      <add x:p0="none"/>
    </combo>

  </x:children>
</xswt>
```

Example: Templated Content

- omnetpp.ini.ftl:

```
[General]
network = ${networkName}
<#if trafficType=="tcp">
…
<#elseif trafficType=="udp">
…
</#if>
```

- network.ned.ftl

```
<@setoutput path=${networkName}+".ned"/>

network ${networkName} {
    submodules:
        host[${numNodes}]: Host;
        …
}
```

Extensibility

When the FreeMarker template language is not enough:

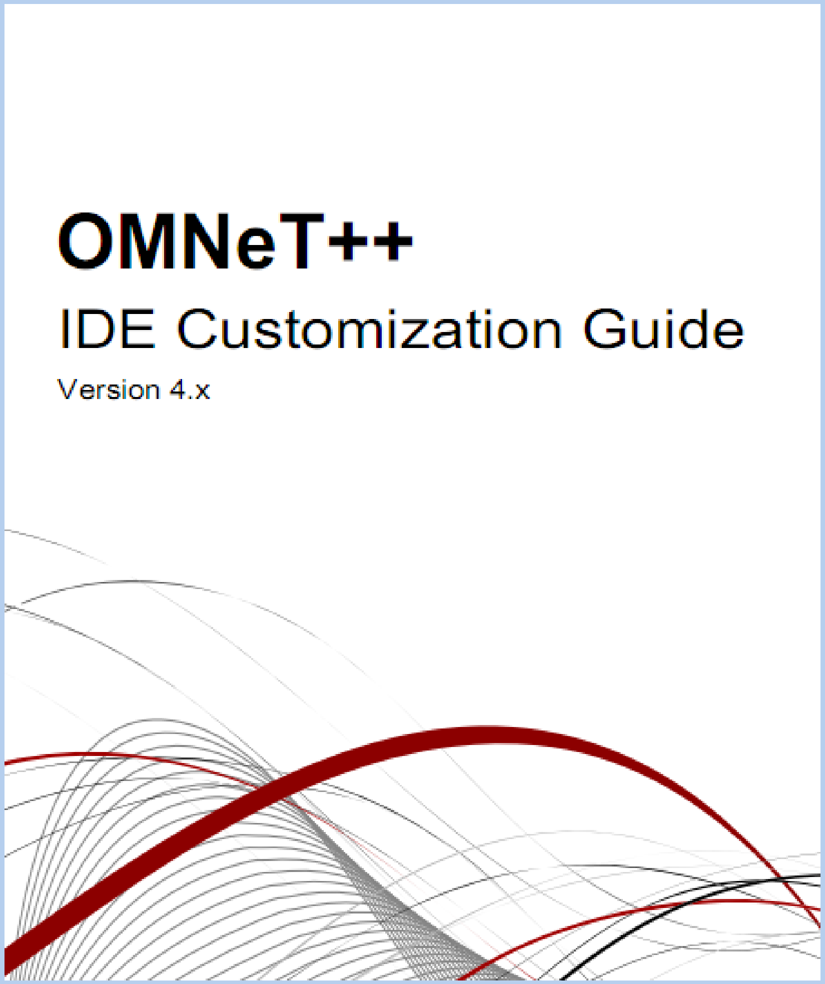
- **Java**: you can write the code in Java, copy the JAR file into the templates/ folder, and invoke it from FreeMarker
- **External programs:** you can run external programs from FreeMarker, and let them do the job and/or capture their output
  - Easy way to incorporate C/C++ code

When XSWT or provided widgets are not enough:

- **Custom widgets:** you can write custom widgets (also compound widgets like table+buttons) in Java
- **Custom pages:** you can write whole custom pages in Java
  - `page.1.class = org.example.foo.MyWizardPage`

Documentation

“IDE Customization Guide”



OMNeT++

IDE Customization Guide

Version 4.x