ProFiDo XML Configuration Format Specification

Falko Bause, Philipp Gerloff, Alparslan Kirman, Jan Kriege, Daniel Scholtyssek
Informatik IV, TU Dortmund
profido@ls4.cs.uni-dortmund.de

July 24, 2014

ProFiDo (Processes Fitting Toolkit Dortmund) [4] is a graphical toolkit that provides a consistent use of commandline-oriented tools for the fitting of stochastic processes and distributions. ProFiDo itself provides only a core functionality to handle graphs including support for storing and loading graphs and the parameter specification of graph elements. All fitting programs which might be supported by ProFiDo are not hard-coded into ProFiDo's Java code. Instead on each start of ProFiDo a configuration file in XML format is parsed which completes the GUI's functionality. This document describes the structure and tags of this XML configuration file.

1 General Document Structure of the Configuration Format

The document description should begin with an XML declaration like for example `<?xml version="1.0"?>`. The single root element of the configuration description is called settings with the start tag `<settings>` and the end tag `</settings>`. The whole specification is expected between these tags.
The XML configuration file consists of five parts:

- a part specifying software directories enclosed in tags `<config>` and `</config>` (cf. Sect. 2).
- a part containing a list of programs enclosed in tags `<software>` and `</software>` (cf. Sect. 3), each program being able to read and write data in the XML interchange format (see [5]). Usually these programs are shell scripts encapsulating the commandline-based fitting tools.
- a part specifying a list of converter tools enclosed in tags `<tools>` and `</tools>` (cf. Sect. 4), being able to convert data given in some other format into the XML interchange format or vice versa.
- a part specifiying a list of edge viewer entries enclosed in tags `<edgeviewer>` and `</edgeviewer>` (cf. Sect. 5).
- a part specifying a list of report entries enclosed in tags `<report>` and `</report>` (cf. Sect. 6).

REMARK:
In the following we do not mention closing tags explicitly. I.e. if we describe that some information is specified by tag `<tagname>`, we implicitly mean that the information is specified within tags `<tagname>` and `</tagname>`.

The overall structure of a configuration file is sketched in the following example.

Example 1 (Sketch of XML Configuration File)

```
<?xml version="1.0"?>
<settings>
  <config>
    [...]
  </config>
  <software>
    [...]
  </software>
  <tools>
    [...]
  </tools>
  <edgeviewer>
    [...]
  </edgeviewer>
  <report>
    [...]
  </report>
</settings>
```

In the following we describe these five parts in more detail. An example of a complete XML configuration file is given in appendix A.

2 General Configuration

This section describes the `<config>` part in detail.

2.1 Specification of Software Directories

After having specified a workflow with ProFiDo, the user can export a shell script which executes the modelled workflow. ProFiDo also generates a further shell script for setup purposes. This shell script copies necessary programs and scripts to local directories, so that they can be used exclusively by the user in directories where he has full permissions. The reason for copying programs is that a fitting tool might expect input data, like e.g. a trace file, to be located in the program's installation directory or that a program might use fixed file names during runtime.

The source directories of software and scripts are defined by two tags:
`<softwaresource>` and `<auxiliaryfolder>`.

Example 2 (Defining software directories)

```
<?xml version="1.0"?>
<settings>
 <config>
  <softwaresource>/home/fred/Software</softwaresource>
  <auxiliaryfolder>
      ${SOFTWARESOURCE}/scripts
  </auxiliaryfolder>
  <defaultsuffix>.xml</defaultsuffix>
  <validinputcharacters>
     [...]
  </validinputcharacters>
 </config>
<software>
[...]
</settings>
```

which results in the following setup shell script provided only job types *Gfit* and *Plot* are used in the user specified workflow and */home/fred/experiments* has been selected as the export folder in ProFiDo's GUI.

Example 3 (Example setup shell script)

```
#!/bin/sh
# ————————————————————————————————
#    Script for workflow: test
# ————————————————————————————————

# This workflow was created with ProFiDo 1.3

#Helper function to check if a file exists.
function checkIfFileExists {
 if [ ! −f $1 ]; then
 echo "Error! The file ${1} does not exist!"
 fi
}

# Before any further steps, check if all required external
# binaries are present in the path.
requirements="sh wish"

# Further requirements can be added to the above
# variable separated by blanks. Example: "ls cat head"
missingCommands=""
for curCmd in $requirements; do
 if ! [ ‘command −v $curCmd‘ ]; then
```

```
  missingCommands="$missingCommands\n$curCmd"
 fi
done
if ! [ -z "$missingCommands" ]; then
 echo -e "Warning, the following required external commands
   were not found on your system."
 echo -e "Please ensure their presence before executing
   the workflow. $missingCommands"
fi

# Before the workflow can be used, this script sets up the
# environment.

# First, all folders are set.
export SOFTWARESOURCE="/home/fred/Software"
export TARGETDIR="/home/fred/experiments"

# Creater Folders
\mkdir -p "${TARGETDIR}/Results/Tmp/"
\mkdir -p "${TARGETDIR}/Software/"

# Copy Software
\cp -f -R "${SOFTWARESOURCE}/Gfit" "${TARGETDIR}/Software/Gfit"
\cp -f -R "${SOFTWARESOURCE}/Plot" "${TARGETDIR}/Software/Plot"
\cp -f -R "${SOFTWARESOURCE}/scripts" "${TARGETDIR}/Software"
```

Note that the shell variable SOFTWARESOURCE is used within the `<auxiliaryfolder>` tag and therefore used when creating the copy command collecting the scripts.

2.2 Specification of the default suffix

With the `<defaultsuffix>` tag the default extension for all files that are represented by edges can be defined. Per default ProFiDo uses ".xml".

2.3 Specification of valid input characters

If one wants to prevent the use of specific characters, because some of the programs or tools can not handle them, it is possible to specify valid input characters with the `<validinputcharacters>` tag. Within the GUI, the user won't be able to use any other than the specified characters.
In case the `<validinputcharacters>` tag is not set by the user, ProFiDo allows the use of the following characters:

- Word characters: a-z, A-Z, 0-9.
- Whitespace characters.
- Brackets: (,), {, }.
- Additional characters: period, Slash, Backslash, Underscore, $, Quotation marks.

3 Specification of Program List

All programs are specified within the `<software>` tags. Program specifications are started by the tag `<program>`. They consist of parts considering

- general information specified by tags `<general>`,
- input/output data specified by tags `<input>` and `<output>`
- and parameters of a program call by tag `<parameterlist>`.

ProFiDo allows for the specification of dependencies, i.e. additional tools that are required to execute the programs. The dependencies are expected within the `<requires>` tag separated by a comma. The `<requires>` tag may be used within the `<software>` tag if a tool is required for all programs or within the `<general>` tag of a program specification if it is only a dependency of a single program (see below). The following example shows a skeleton specification of a program list. The `<requires>` tag used in the example defines that the two shells sh and wish are required for all the following programs. When exporting a workflow as a script this information will be used to provide dependency checks for all requirements valid for the specific workflow.

Example 4 (Skeleton Program List)

```xml
<?xml version="1.0"?>
<settings>
[...]
<software>
<requires>sh, wish</requires>
<program>
  <general>
    [...]
  </general>
  <input>
    [...]
  </input>
  <output>
    [...]
  </output>
  <parameterlist>
    [...]
  </parameterlist>
</program>
<program>
  [...]
</program>
[...]
<program>
  [...]
</program>
</software>
[...]
</settings>
```

3.1 General Part of a Program Specification

The general part of a program specification is described by tags `<name>`, `<article>`, `<binary>` and `<requires>`. In tag `<name>` a name for the program must be given, which identifies the node type in ProFiDo's GUI. Tag `<article>` is an optional possibility to give further information, e.g. a reference, and the mandatory tag `<binary>` gives the location of the program relative to the software directory specified by tag `<softwaresource>` (see Sect. 2). Using the `<requires>` tag one can specifiy additional tools that are necessary, i.e. have to be installed in order to execute the program.

Example 5 (General Part for G-FIT)

```
<?xml version="1.0"?>
<settings>
 <config>
  <softwaresource>/home/fred/Software</softwaresource>
  <auxiliaryfolder>\${SOFTWARESOURCE}/scripts</auxiliaryfolder>
 </config>
<software>
<program>
  <general>
    <name>G-FIT</name>
    <article>Axel Thuemmler, Peter Buchholz, and Miklos Telek.
       A Novel Approach for Phase-Type Fitting with the EM
       Algorithm. IEEE Trans. Dependable Sec. Comput.,
       3(3):245--258, 2006.
    </article>
    <binary>Gfit/gfit.sh</binary>
  </general>
  <input>
[...]
</settings>
```

Example 6 (General Part for ARIMA Fitting) *The ARIMA Fitting job node is based on an implementation in R and requires an installation of the Rscript interpreter:*

```
<?xml version="1.0"?>
<settings>
 <config>
  <softwaresource>/home/fred/Software</softwaresource>
  <auxiliaryfolder>\${SOFTWARESOURCE}/scripts</auxiliaryfolder>
 </config>
<software>
[...]
<program>
  <general>
    <name>ARIMA Fitting</name>
    <article></article>
    <binary>arimafit/arimafit.r</binary>
    <requires>Rscript</requires>
  </general>
  <input>
[...]
</settings>
```

3.2 Input/Output Part of a Program Specification

After the general part the type of permitted inputs and generated outputs is specified by tags `<type>`, `<parameter>` and the optional `<converter>`. This description is enclosed in the corresponding `<input>`,`<output>` tags.

Tag `<type>` specifies the model type that is expected as input or generated as output of the tool, respectively. E.g., possible types might be XML−Trace or XML−MAP for tools that can work with a Trace or a MAP respectively. Note, that for two tools that are connected in a worklow the output type of the first tool must match an input type of the second tool.

If the input needs to be processed before it is passed to the tool this can be done by specifying a converter script with the tag `<converter>`. For input files these converters are executed before and for an output file after the execution of the program.

The tag `<parameter>` contains an identifier of a commandline parameter as described in Sect. 3.3.

Example 7 (Input/Output Part for G-FIT)

```
[...]
</general>
<input>
  <type>XML–Trace</type>
  <parameter>Input Filename</parameter>
</input>
<output>
  <type>XML–MAP</type>
  <parameter>Output Filename</parameter>
</output>
<parameterlist>
[...]
```

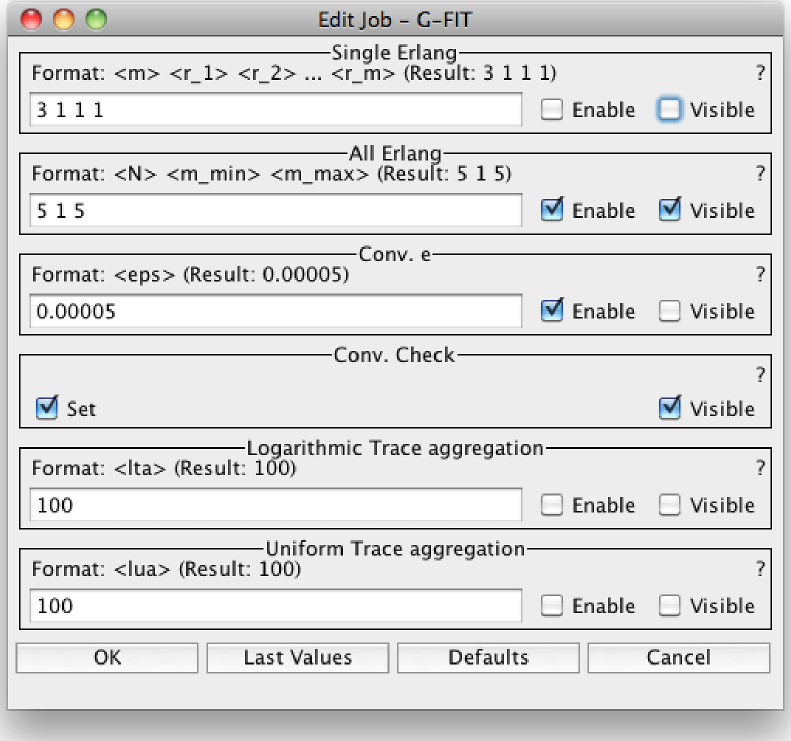
Example 8 (Input/Output Part for Plot) *The following part of the XML specification for the Plot node demonstrates the usage of the* `<converter>` *tags.*

```
[...]
</general>
<input>
  <type>XML–Trace</type>
  <converter>
    scripts/CreateLinkToXMLReferenceTarget.tcl
  </converter>
  <parameter>Trace Input Filenames</parameter>
</input>
[...]
<input>
```

```
    <type>XML–ARTA</type>
    <converter>scripts/XMLToARTA.tcl</converter>
    <parameter>ARTA Input Filenames</parameter>
  </input>
  [...]
```

3.3 Parameter Part of a Program Specification

The parameter part of a program is described by tag `<parameterlist>`. A list of parameters consists of an enumeration of parameter groups, each specified by tag `<parametergroup>`. ProFiDo displays all parameter groups of a program in a corresponding property window displaying all groups in a separate box (cf. Fig. 1).



Edit Job – G-FIT

Single Erlang
Format: <m> <r_1> <r_2> ... <r_m> (Result: 3 1 1 1) ?
3 1 1 1 □ Enable □ Visible

All Erlang
Format: <N> <m_min> <m_max> (Result: 5 1 5) ?
5 1 5 ☑ Enable ☑ Visible

Conv. e
Format: <eps> (Result: 0.00005) ?
0.00005 ☑ Enable □ Visible

Conv. Check ?
☑ Set ☑ Visible

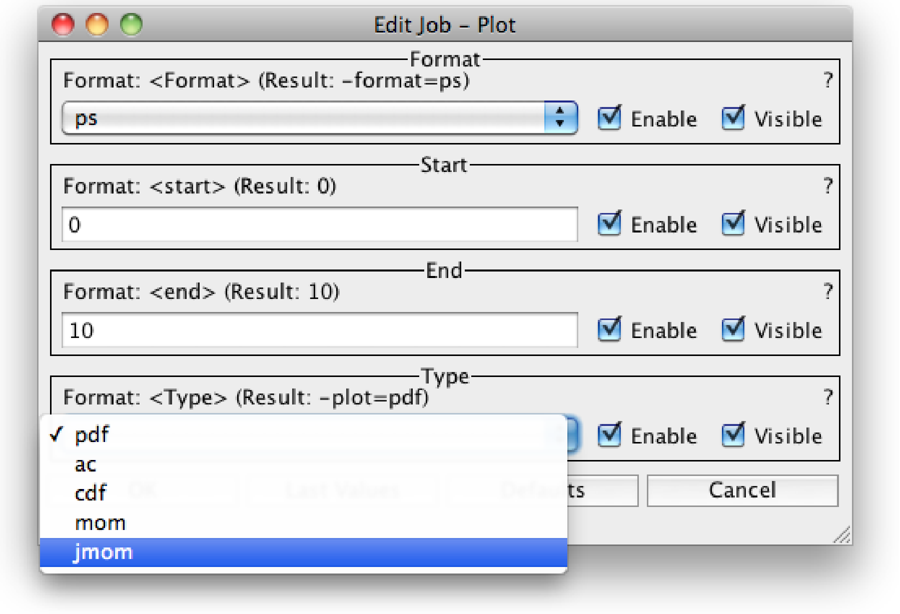
Logarithmic Trace aggregation
Format: <lta> (Result: 100) ?
100 □ Enable □ Visible

Uniform Trace aggregation
Format: <lua> (Result: 100) ?
100 □ Enable □ Visible

Figure 1: Properties window of a G-FIT Job in ProFiDo's GUI. Here the user can select the corresponding call parameters

Example 9 (Skeleton of Parameter List Entries)

```
  [...]
```



Edit Job – Plot

Format

Format: <Format> (Result: -format=ps)

ps ☑ Enable ☑ Visible

Start

Format: <start> (Result: 0)

0 ☑ Enable ☑ Visible

End

Format: <end> (Result: 10)

10 ☑ Enable ☑ Visible

Type

Format: <Type> (Result: -plot=pdf)

☑ Enable ☑ Visible

Figure 2: Properties window of a Plot Job in ProFiDo's GUI

```
</output>
<parameterlist>
  <parametergroup visible="false" type="text" disabled="true">
  [...]
  </parametergroup>
  <parametergroup visible="true" type="text">
  [...]
  </parametergroup>
  [...]
</parameterlist>
[...]
```

The XML specification of a parameter group might have the following attributes:

visible: This boolean attribute specifies the default value of the visibility attribute of this parameter group. A visible parameter group is displayed in the workflow of ProFiDo's canvas.

type: Types might be filename, bool, text or select. Depending on the type the attribute is displayed in a different way. For text attributes an input box is generated where the user can enter text. Single Erlang, All Erlang and Conv. e in Fig. 1 are all of type text. bool results in a checkbox for this attribute (cf. Conv. Check in Fig. 1). For an attribute of type select a selection box is generated where the user can select between the given options (cf. Type in Fig. 2) . filename is a special type used for the input and output parameters. These parameters are not displayed in the properties window, since the filenames are determined by incoming and outgoing arcs. Moreover, the tag <name> (see below) must contain the same identifier as one of the <parameter> tags specified as <input> or <output> (cf. Sect. 3.2).

For parameters of the type filename the additional attribute multi is used to specify that several parameters of the same type may be used for specification (cf. the definition of Plot in appendix A where several trace files are allowed as input for this job node).

disabled: This boolean attribute specifies the default value of the “Enable” attribute (cf. Fig. 1). Disabling a parameter group implies that the whole parameter group is neglected when generating the tool specific call.

A parameter group is specified by tags <name>, <description>, <key> and <**default**>. Depending on the type attribute additional tags are required. For text parameter groups these tags are followed by one or several parameters each specified by tag <parameter>. For select parameter groups an additional tag <values> is required. <name> contains an identifier for the attribute. The content of the tag <description> is used to display information about that attribute as help for the user. The tag <key> is used for the generation of the call to the program when exporting the workflow as a script for execution.[1] If the program expects a specific key to be put in front of a parameter value this key can be given by the tag <key>. <**default**> is used to declare the default value for an attribute.

Since tools might allow a call using several (enabled) parameter groups the <successor> tag might give the name of the next parameter group being allowed when constructing the call for the corresponding program.

Example 10 (A Parameter Group for G-FIT)

```
[...]
</output>
<parameterlist>
  <parametergroup visible="false" type="text" disabled="true">
    <name>Single Erlang</name>
    <description>Fit a single hyper-Erlang distribution
                 to the given data trace.</description>
    <key>-f </key>
    <default>3 1 1 1</default>
    <parameter type="static">
      <name>m</name>
      <description>Number of different Erlang
                   branches.</description>
      <successor>r_i</successor>
    </parameter>
    <parameter type="dynamic">
      <name>r_i</name>
```

[1] E.g. -r is a possible key for the Unix command ls.

```
      <dynamicpart>i</dynamicpart>
      <dynamicref>m</dynamicref>
      <description>Number of phases of the i−th Erlang
                   branch (i = 1,...,m).</description>
      <successor></successor>
    </parameter>
    <successor>All Erlang</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
  [...]
</parameterlist>
[...]
```

Each parameter of a text parameter group is specified by a name (tag <name>) and an optional description (tag <description>), which is displayed when the user clicks on the associated question mark (see Fig. 1). Since tool calls might depend on a specific sequence of parameters the <successor> tag gives the name of the next parameter being relevant when constructing the call for the corresponding program.

A parameter might be of type <static> or <dynamic>. In contrast to static parameters, dynamic parameters have two additional mandatory tags <dynamicpart> and <dynamicref>. <dynamicref> references the parameter (being assumed to be of type integer) which determines the number of values of the dynamic parameter. Tag <dynamicpart> specifies that part of the parameter's name which will be enumerated when being displayed in the property window (cf. Fig. 1).

Example 11 (Select Parameter Group for Plot (cf. Fig. 2))

```
[...]
<parametergroup visible="true" type="select">
  <name>Type</name>
  <description> Define what information to plot.</description>
  <key>—plot=</key>
  <default> </default>
  <values>
    <value>pdf</value>
    <value>ac</value>
    <value>cdf</value>
    <value>mom</value>
    <value>jmom</value>
  </values>
  <successor>Output Filename</successor>
</parametergroup>
[...]
```

The tag <values> is used to list the options the user can select for a select parameter group. Each option is enclosed in tags <value> and </value>.

4 Specification of Converter Scripts

All converter scripts are specified within the `<tools>` tag. Each converter is described by the `<converter>` tag. Converters might be of type input or output and are associated with an input or output node within the ProFiDo GUI. Similar to the program specification the `<requires>` tag can be used to specify dependencies globally as well as for each converter.

Example 12 (Converter of Type Input)

```
[...]
<tools>
<requires> [...] </requires>
<converter type="input">
[...]
```

Each converter has one input (tag `<input>`) and one output (tag `<output>`) specification and a tag `<binary>` which defines the location of the corresponding script relative to the directory specified by tag `<softwaresource>` (cf. Sect. 2).

Example 13 (Converter Specification)

```
[...]
<tools>
<requires>sh, wish</requires>
<converter type="input">
  <input>
    <type>Trace</type>
    <suffix>.trace</suffix>
  </input>
  <output>
    <type>XML-Trace</type>
    <suffix>.xml</suffix>
  </output>
  <binary>scripts/TraceToXMLTrace.sh</binary>
</converter>
<converter type="output">
  <input>
    <type>XML-MAP</type>
    <suffix>.xml</suffix>
  </input>
  <output>
    <type>Map</type>
    <suffix>.map</suffix>
  </output>
  <binary>scripts/CreateCopyOfXMLReferenceTarget.tcl</binary>
```

```
</converter>
[...]
</tools>
[...]
```

The input and output specifications of each converter contain a tag to describe the type (<type>) and one to define the suffix (<suffix>). The types should correspond to the type specifications for the input/output of programs (cf. Sect. 3.2). The <suffix> defines the corresponding extension for each file type.

5 Specification of EdgeViewer Entries

Edge viewers can be used to display intermediate results of the tools/programs within ProFiDo. All edge viewers are specified within the <edgeviewer> tag. Each viewer is described by the <viewer> tag. Multiple viewers can be defined within the <edgeviewer> tag with one tag for each viewer (see Sect. 7 for how to specify a viewer object).

6 Specification of Report Entries

Report viewers are used to create the overall report of a workflow. For most parts a report viewer creates LaTeX code. All report viewers are specified within the <report> tag. Each viewer is described by the <viewer> tag. Multiple viewers can be defined within the <report> tag with one tag for each viewer (see Sect. 7 for how to specify a viewer object).

7 Specification of Viewer Objects

A viewer is specified within the <viewer> tag. A viewer contains the following tags:

- Several <type> tags which specify the supported types of that viewer.
- A <name> tag, so that the name will be displayed withing ProFiDo.
- An <intern> or <extern> tag which determine if the content should be shown within the GUI (intern) or whether it should be shown using an external application (extern). These tags are only applicable within the <edgeviewer> section (cf. Sect. 5).
- A <description> tag which can be used to describe the viewer object. This tag is optional.
- A <program> tag which can be used to specify a viewer job (cf. Sect. 7.1). This tag is optional.
- A <converter> tag which can be used to specify a converter that should be executed after the viewer job (cf. Sect. 7.2). The converter is mostly used to

convert the output format of the viewer job to an output format which is needed by the viewer program (e.g. it might be used to convert the output to LaTeX code). This tag is optional.

7.1 Specification of a Viewer Job

In order to create a viewer job, one can use the `<program>` tag. The viewer job will be used to display results. It consists of the following tags:

- A `<name>` tag giving the name as defined in the corresponding program tag (see Sect. 3.1).
- A `<preset>` tag, which can be used to set various presets for the parameters of the used program.

The `<preset>` tag itself contains the `<parametergroup>` tag, which might have the attribute fixed = "true". The attribute indicates whether or not the value can be changed by the user. Within the `<parametergroup>` tag, one needs to specify the `<name>` and `<value>` tag. The name represents the name of the parameter group as defined within the corresponding program tag (see Sect. 3.1). The value is the target value one wants to preset.

7.2 Specification of a Converter

A converter contains two tags:

- `<input>` specifies which type of input the converter will get.
- `<output>` specifies which type of output the converter will convert to.

Which converter will be used is defined by the signature of input and output (see Sect. 4).

Example 14 (Example of an edge viewer)

```
<edgeviewer>
 <viewer>
  <type>XML–MAP</type>
  <type>XML–Trace</type>
  <type>XML–PH</type>
  <type>XML–ARTA</type>
  <type>XML–ARIMA</type>
  <type>XML–CAPP</type>
  <name>Plot Details</name>
  <program>
   <name>Plot</name>
   <preset>
    <parametergroup fixed="true">
     <name>Format</name>
     <value>png</value>
    </parametergroup>
   </preset>
  </program>
  <converter>
   <input>XML–PNG</input>
   <output>PNG</output>
  </converter>
  <intern>image</intern>
 </viewer>
</edgeviewer>
```

References

[1] Falko Bause, Peter Buchholz, and Jan Kriege. ProFiDo - The Processes Fitting Toolkit Dortmund. In *Proc. of the 7th International Conference on Quantitative Evaluation of SysTems (QEST) 2010*, pages 87–96, 2010.

[2] Falko Bause, Philipp Gerloff, Alparslan Kirman, Jan Kriege, and Daniel Scholtyssek. ProFiDo XML Configuration Format Specification, 2014. http://www4.cs.uni-dortmund.de/profido.

[3] Falko Bause, Philipp Gerloff, Alparslan Kirman, Jan Kriege, and Daniel Scholtyssek. ProFiDo XML Workflow Format Specification, 2014. http://www4.cs.uni-dortmund.de/profido.

[4] Falko Bause, Philipp Gerloff, and Jan Kriege. ProFiDo - A Toolkit for Fitting Input Models. In Bruno Müller-Clostermann, Klaus Echtle, and Erwin P. Rathgeb, editors, *Proceedings of the 15th International GI/ITG Conference on Measurement, Modelling and Evaluation of Computing Systems and Dependability and Fault Tolerance (MMB & DFT 2010)*, volume 5987 of *LNCS*, pages 311–314. Springer, 2010.

[5] Falko Bause and Jan Kriege. ProFiDo XML Interchange Format Specification, 2014. http://www4.cs.uni-dortmund.de/profido.

A Example of a ProFiDo Configuration File

```
<settings>
 <config>
  <softwaresource>${HOME}/ProFiDo/software</softwaresource>
  <auxiliaryfolder>${SOFTWARESOURCE}/scripts</auxiliaryfolder>
  <defaultsuffix>.xml</defaultsuffix>
  <validinputcharacters><![CDATA[([\w\s]|[äüöÄÜÖß]|
  [\{\}]|[\"\.\-\+\(\)\_\/\\])]]>
  </validinputcharacters>
 </config>
 <software>
```

```
<requires>sh, wish</requires>
<program>
 <general>
  <name>G-FIT</name>
  <article>A Novel Approach for Phase-Type Fitting
   with the EM Algorithm.
  </article>
  <binary>Gfit/gfit.sh</binary>
 </general>
 <input>
  <type>XML-Trace</type>
  <parameter>Input Filename</parameter>
 </input>
 <output>
  <type>XML-MAP</type>
  <type>XML-PH</type>
  <parameter>Output Filename</parameter>
 </output>
 <parameterlist>
  <parametergroup visible="true" type="filename">
   <name>Input Filename</name>
   <description>Define Input Filename.</description>
   <key></key>
   <default>trace.xml</default>
   <successor>Output Filename</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename">
   <name>Output Filename</name>
   <description>Define Output Filename.</description>
   <key></key>
   <default>em_map.xml</default>
   <successor>Single Erlang</successor>
  </parametergroup>
  <parametergroup visible="false" type="text" disabled="true">
   <name>Single Erlang</name>
   <description>Fit a single hyper-Erlang distribution to the given data
   trace.
   </description>
   <key>-f </key>
   <default>3 1 1 1</default>
   <parameter type="static">
    <name>m</name>
    <description>Number of different Erlang branches.</description>
    <successor>r_i</successor>
   </parameter>
   <parameter type="dynamic">
    <name>r_i</name>
    <dynamicpart>i</dynamicpart>
    <dynamicref>m</dynamicref>
    <description>Number of phases of the i-th
     Erlang branch (i = 1,...,m).
    </description>
    <successor></successor>
```

```
     </parameter>
     <successor>All Erlang</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>All Erlang</name>
     <description>Fit all settings of a hyper—Erlang distribution and
      return the best fitted distribution.
     </description>
     <key>—a </key>
     <default>5 1 5</default>
     <parameter type="static">
      <name>N</name>
      <description>Number of the overall number of (internal) states used
        by the hyper—Erlang distribution.
      </description>
      <successor>m_min</successor>
     </parameter>
     <parameter type="static">
      <name>m_min</name>
      <description>Minimal number of Erlang branches to be used.
      </description>
      <successor>m_max</successor>
     </parameter>
     <parameter type="static">
      <name>m_max</name>
      <description>Maximal number of Erlang branches to be used.
      </description>
      <successor></successor>
     </parameter>
     <successor>Conv. e</successor>
    </parametergroup>
    <parametergroup visible="false" type="text">
     <name>Conv. e</name>
     <description>Set convergence epsilon.</description>
     <key>—e </key>
     <default>0.00005</default>
     <parameter type="static">
      <name>eps</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Conv. Check</successor>
    </parametergroup>
    <parametergroup visible="true" type="bool">
     <name>Conv. Check</name>
     <description>Set convergence check based on maximal difference of
      parameter values.
     </description>
     <key>—par </key>
     <default>1</default>
     <successor>Logarithmic Trace aggregation</successor>
    </parametergroup>
    <parametergroup visible="false" type="text" disabled="true">
```

```
   <name>Logarithmic Trace aggregation</name>
   <description>Number of intervals for logarithmic Trace aggregation
   </description>
   <key>-lta </key>
   <default>100</default>
   <parameter type="static">
    <name>lta</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Uniform Trace aggregation</successor>
  </parametergroup>
  <parametergroup visible="false" type="text" disabled="true">
   <name>Uniform Trace aggregation</name>
   <description>Number of intervals for uniform Trace aggregation
   </description>
   <key>-uta </key>
   <default>100</default>
   <parameter type="static">
    <name>lua</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor></successor>
  </parametergroup>
 </parameterlist>
</program>
<program>
 <general>
  <name>Momfit</name>
  <article>A Heuristic Approach for Fitting MAPs to Moments and
   Joint Moments
  </article>
  <binary>momfit/momfit.sh</binary>
 </general>
 <input>
  <type>XML-Trace</type>
  <parameter>Input Filename</parameter>
 </input>
 <output>
  <type>XML-MAP</type>
  <type>XML-PH</type>
  <parameter>Output Filename</parameter>
 </output>
 <parameterlist>
  <parametergroup visible="true" type="filename">
   <name>Input Filename</name>
   <description>Define Input Filename.</description>
   <key></key>
   <default>trace.xml</default>
   <successor>Output Filename</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename">
   <name>Output Filename</name>
   <description>Define Output Filename.</description>
   <key></key>
   <default>momfit_ph.xml</default>
   <successor>Moments</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>Moments</name>
   <description>Number of moments.</description>
   <key></key>
   <default>5</default>
   <parameter type="static">
    <name>moments</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>States</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>States</name>
   <description>Number of states.</description>
   <key></key>
   <default>3</default>
   <parameter type="static">
    <name>states</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Iterations</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>Iterations</name>
   <description>Number of iterations.</description>
   <key></key>
   <default>10000</default>
   <parameter type="static">
    <name>iter</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Epsilon</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>Epsilon</name>
   <description>Convergence criterion.</description>
   <key></key>
   <default>1.0e-7</default>
   <parameter type="static">
    <name>eps</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Time</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>Time</name>
   <description>Max. time for fitting.</description>
   <key></key>
   <default>100</default>
   <parameter type="static">
    <name>time</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Seed</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>Seed</name>
   <description>Seed for random number generator.</description>
   <key></key>
   <default>13</default>
   <parameter type="static">
    <name>seed</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor></successor>
  </parametergroup>
 </parameterlist>
</program>
<program>
 <general>
  <name>MAP_EM</name>
  <article></article>
  <binary>EM/EM.sh</binary>
 </general>
 <input>
  <type>XML-Trace</type>
  <parameter>Input Filename</parameter>
 </input>
 <input>
  <type>XML-MAP</type>
  <parameter>MAP Input Filename</parameter>
 </input>
 <output>
  <type>XML-MAP</type>
  <parameter>Output Filename</parameter>
 </output>
 <parameterlist>
  <parametergroup visible="true" type="filename">
   <name>Input Filename</name>
   <description>Define Input Filename.</description>
   <key></key>
   <default>trace.xml</default>
   <successor>MAP Input Filename</successor>
  </parametergroup>
```

```
    <parametergroup visible="true" type="filename">
     <name>MAP Input Filename</name>
     <description>Define Input Filename</description>
     <key></key>
     <default></default>
     <successor>Output Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Output Filename</name>
     <description>Define Output Filename.</description>
     <key></key>
     <default></default>
     <successor>Conv. e</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Conv. e</name>
     <description>Set convergence epsilon.</description>
     <key></key>
     <default>0.005</default>
     <parameter type="static">
      <name>eps</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>#Iteration</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>#Iteration</name>
     <description>Set maximum number of iterations.</description>
     <key></key>
     <default>20</default>
     <parameter type="static">
      <name>iter</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Map Dims</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Map Dims</name>
     <description>Define dimensions of MAP.</description>
     <key></key>
     <default>3</default>
     <parameter type="static">
      <name>dim</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Alpha</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Alpha</name>
     <description>Set alpha randomizing parameter.</description>
```

```
   <key></key>
   <default>0.5</default>
   <parameter type="static">
    <name>alpha</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Seed</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>Seed</name>
   <description>Set seed used to initialize the pseudorandom
    number generator.
   </description>
   <key></key>
   <default>123</default>
   <parameter type="static">
    <name>seed</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor></successor>
  </parametergroup>
 </parameterlist>
</program>
<program>
 <general>
  <name>MAP_MOEA</name>
  <article></article>
  <binary>map_moea/moea.sh</binary>
 </general>
 <input>
  <type>XML-Trace</type>
  <parameter>Input Filename</parameter>
 </input>
 <output>
  <type>XML-MAP</type>
  <parameter>Output Filename</parameter>
 </output>
 <parameterlist>
  <parametergroup visible="true" type="filename">
   <name>Input Filename</name>
   <description>Add Input Filename</description>
   <key></key>
   <default></default>
   <successor>Output Filename</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename">
   <name>Output Filename</name>
   <description>Define Output Filename</description>
   <key></key>
   <default></default>
   <successor>Number of Generation</successor>
```

```
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Number of Generation</name>
     <description>Set the number of Generation</description>
     <key></key>
     <default>300</default>
     <parameter type="static">
      <name>maxgen</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Alpha</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Alpha</name>
     <description>Set alpha parameter</description>
     <key></key>
     <default>60</default>
     <parameter type="static">
      <name>alpha</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Mu</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Mu</name>
     <description>Define the Mu Paramater</description>
     <key></key>
     <default>30</default>
     <parameter type="static">
      <name>mu</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Lambda</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Lambda</name>
     <description>Define the Lambda paramater</description>
     <key></key>
     <default>30</default>
     <parameter type="static">
      <name>lambda</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Epsilon</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Epsilon</name>
     <description>Set Epsilon parameter</description>
     <key></key>
     <default>1e-3</default>
     <parameter type="static">
      <name>epsilon</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>#Iteration</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>#Iteration</name>
     <description>Maximum number of Iteration</description>
     <key></key>
     <default>100</default>
     <parameter type="static">
      <name>iter</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Dimension</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Dimension</name>
     <description>Define the Dimension</description>
     <key></key>
     <default>5</default>
     <parameter type="static">
      <name>dim</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Alpha2</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Alpha2</name>
     <description>2nd alpha value</description>
     <key></key>
     <default>5.0</default>
     <parameter type="static">
      <name>alpha2</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Seed</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Seed</name>
     <description>Set seed to initialize the pseudorandom number
     </description>
     <key></key>
     <default>111</default>
     <parameter type="static">
      <name>seed</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Autocorrelation</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Autocorrelation</name>
     <description>Set the Autocorrelation</description>
     <key></key>
     <default>10</default>
     <parameter type="static">
      <name>acov</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor></successor>
    </parametergroup>
   </parameterlist>
  </program>
  <program>
   <general>
    <name>JMomfit</name>
    <article>A Heuristic Approach for Fitting MAPs to Moments and
     Joint Moments
    </article>
    <binary>jmomfit/jmomfit.sh</binary>
   </general>
   <input>
    <type>XML-Trace</type>
    <parameter>Input Filename Trace</parameter>
   </input>
   <input>
    <type>XML-PH</type>
    <parameter>Input Filename PH</parameter>
   </input>
   <output>
    <type>XML-MAP</type>
    <parameter>Output Filename</parameter>
   </output>
   <parameterlist>
    <parametergroup visible="true" type="filename">
     <name>Input Filename Trace</name>
     <description>Define Input Filename.</description>
     <key></key>
     <default>trace.xml</default>
     <successor>Input Filename PH</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Input Filename PH</name>
     <description>Define Input Filename.</description>
     <key></key>
     <default>momfit_ph.xml</default>
     <successor>Output Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Output Filename</name>
     <description>Define Output Filename.</description>
     <key></key>
     <default>jmomfit_map.xml</default>
     <successor>Joint Moments</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Joint Moments</name>
     <description>Number of joint moments.</description>
     <key></key>
     <default>3</default>
     <parameter type="static">
      <name>jmoments</name>
      <description></description>
      <successor></successor>
     </parameter>
    </parametergroup>
   </parameterlist>
  </program>
  <program>
   <general>
    <name>ACfit</name>
    <article>An Empirical Comparison of MAP Fitting Algorithms</article>
    <binary>acfit/acfit.sh</binary>
   </general>
   <input>
    <type>XML-Trace</type>
    <parameter>Input Filename Trace</parameter>
   </input>
   <input>
    <type>XML-PH</type>
    <parameter>Input Filename PH</parameter>
   </input>
   <output>
    <type>XML-MAP</type>
    <parameter>Output Filename</parameter>
   </output>
   <parameterlist>
    <parametergroup visible="true" type="filename">
     <name>Input Filename Trace</name>
     <description>Define Input Filename.</description>
     <key></key>
     <default>trace.xml</default>
     <successor>Input Filename PH</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Input Filename PH</name>
     <description>Define Input Filename.</description>
     <key></key>
     <default>momfit_ph.xml</default>
     <successor>Output Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Output Filename</name>
     <description>Define Output Filename.</description>
     <key></key>
     <default>acfit_map.xml</default>
     <successor>Autocorrelation</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Autocorrelation</name>
     <description>Number of lag-k autocorrelation coefficients.
     </description>
     <key></key>
     <default>10</default>
     <parameter type="static">
      <name>acorr</name>
      <description></description>
      <successor></successor>
     </parameter>
    </parametergroup>
   </parameterlist>
  </program>
  <program>
   <general>
    <name>MEPfit</name>
    <article></article>
    <binary>mepfit/mepfit.exe</binary>
   </general>
   <input>
    <type>XML-Trace</type>
    <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
    <parameter>Input Filename Trace</parameter>
   </input>
   <output>
    <type>XML-MAP</type>
    <parameter>Output Filename</parameter>
   </output>
   <parameterlist>
    <parametergroup visible="true" type="filename">
     <name>Input Filename Trace</name>
     <description>Define Input Filename.</description>
     <key></key>
     <default>trace.xml</default>
     <successor>Order</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Order</name>
     <description>Order of MAP.</description>
     <key></key>
     <default>3</default>
     <parameter type="static">
      <name>order</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Output Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Output Filename</name>
     <description>Define Output Filename.</description>
     <key></key>
     <default>mepfit_map.xml</default>
     <successor></successor>
    </parametergroup>
   </parameterlist>
  </program>
  <program>
   <general>
    <name>ARIMA Fitting</name>
    <article></article>
    <binary>arimafit/arimafit.r</binary>
    <requires>Rscript</requires>
   </general>
   <input>
    <type>XML-Trace</type>
    <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
    <parameter>Input Filename</parameter>
   </input>
   <output>
    <type>XML-ARIMA</type>
    <parameter>Output Filename</parameter>
   </output>
   <parameterlist>
    <parametergroup visible="true" type="filename">
     <name>Input Filename</name>
     <description>Define Input Filename.</description>
     <key></key>
     <default>trace.xml</default>
     <successor>Output Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Output Filename</name>
     <description>Define Output Filename.</description>
     <key></key>
     <default>arimafit_arima.xml</default>
     <successor>AR Order</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>AR Order</name>
     <description>Number of autoregressive coefficients.</description>
     <key></key>
     <default>3</default>
     <parameter type="static">
      <name>p</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Degree of Differencing</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Degree of Differencing</name>
     <description>Degree of Differencing.</description>
     <key></key>
     <default>0</default>
     <parameter type="static">
      <name>d</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>MA Order</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>MA Order</name>
     <description>Number of moving average coefficients.</description>
     <key></key>
     <default>2</default>
     <parameter type="static">
      <name>q</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor></successor>
    </parametergroup>
   </parameterlist>
  </program>
  <program>
   <general>
    <name>ARTAfit</name>
    <article></article>
    <binary>artafit/artafit.exe</binary>
   </general>
   <input>
    <type>XML−Trace</type>
    <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
    <parameter>Trace Input Filename</parameter>
   </input>
   <input>
    <type>XML−Dist</type>
    <parameter>Dist Input Filename</parameter>
   </input>
   <output>
    <type>XML−ARTA</type>
    <parameter>Output Filename</parameter>
   </output>
   <parameterlist>
    <parametergroup visible="true" type="filename">
     <name>Trace Input Filename</name>
     <description>Define Input Filename.</description>
     <key>−trace=</key>
     <default>trace.xml</default>
```

```
     <successor>Dist Input Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Dist Input Filename</name>
     <description>Define Input Filename.</description>
     <key>–dist=</key>
     <default>dist.xml</default>
     <successor>Output Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Output Filename</name>
     <description>Define Output Filename.</description>
     <key>–output=</key>
     <default>artafit_artaa.xml</default>
     <successor>Autocorrelations</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Autocorrelations</name>
     <description>Number of autocorrelation coefficients.</description>
     <key>–ac=</key>
     <default>10</default>
     <parameter type="static">
      <name>ac</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>min. AR order</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>min. AR order</name>
     <description>minimal number of AR coefficients for base process.
     </description>
     <key>–pmin=</key>
     <default>2</default>
     <parameter type="static">
      <name>pmin</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>max. AR order</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>max. AR order</name>
     <description>maximal number of AR coefficients for base process.
     </description>
     <key>–pmax=</key>
     <default>5</default>
     <parameter type="static">
      <name>pmax</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>min. MA order</successor>
```

```
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>min. MA order</name>
   <description>minimal number of MA coefficients for base process.
   </description>
   <key>–qmin=</key>
   <default>0</default>
   <parameter type="static">
    <name>qmin</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>max. MA order</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>max. MA order</name>
   <description>maximal number of MA coefficients for base process.
   </description>
   <key>–qmax=</key>
   <default>2</default>
   <parameter type="static">
    <name>qmax</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor></successor>
  </parametergroup>
 </parameterlist>
</program>
<program>
 <general>
  <name>CAPP-Fit</name>
  <article></article>
  <binary>cappfit/cappfit.exe</binary>
 </general>
 <input>
  <type>XML-Trace</type>
  <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
  <parameter>Input Trace Filename</parameter>
 </input>
 <input>
  <type>XML-PH</type>
  <parameter>Input Filename</parameter>
 </input>
 <output>
  <type>XML-CAPP</type>
  <parameter>Output Filename</parameter>
 </output>
 <parameterlist>
  <parametergroup visible="true" type="filename">
   <name>Input Trace Filename</name>
   <description>Define Input Filename.</description>
   <key>–trace=</key>
```

```
     <default>trace.xml</default>
     <successor>Input Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Input Filename</name>
     <description>Define Input Filename.</description>
     <key>—dist=</key>
     <default>ph.xml</default>
     <successor>Output Filename</successor>
    </parametergroup>
    <parametergroup visible="true" type="filename">
     <name>Output Filename</name>
     <description>Define Output Filename.</description>
     <key>—output=</key>
     <default>capp.xml</default>
     <successor>Autocorrelations</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Autocorrelations</name>
     <description>Number of autocorrelation lags to match.</description>
     <key>—ac=</key>
     <default>30</default>
     <parameter type="static">
      <name>ac</name>
      <description>Number of autocorrelation lags.</description>
      <successor></successor>
     </parameter>
     <successor>min. AR order</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>min. AR order</name>
     <description>Minimal AR order of ARMA base process.</description>
     <key>—pmin=</key>
     <default>3</default>
     <parameter type="static">
      <name>pmin</name>
      <description>Minimal AR order.</description>
      <successor></successor>
     </parameter>
     <successor>max. AR order</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>max. AR order</name>
     <description>Maximal AR order of ARMA base process.</description>
     <key>—pmax=</key>
     <default>7</default>
     <parameter type="static">
      <name>pmax</name>
      <description>Maximal AR order.</description>
      <successor></successor>
     </parameter>
     <successor>min. MA order</successor>
    </parametergroup>
```

```
   <parametergroup visible="true" type="text">
    <name>min. MA order</name>
    <description>Minimal MA order of ARMA base process.</description>
    <key>–qmin=</key>
    <default>2</default>
     <parameter type="static">
     <name>qmin</name>
     <description>Minimal MA order.</description>
     <successor></successor>
    </parameter>
    <successor>max. MA order</successor>
   </parametergroup>
   <parametergroup visible="true" type="text">
    <name>max. MA order</name>
    <description>Maximal MA order of ARMA base process.</description>
    <key>–qmax=</key>
    <default>4</default>
    <parameter type="static">
     <name>qmax</name>
     <description>Maximal MA order.</description>
     <successor></successor>
    </parameter>
    <successor></successor>
   </parametergroup>
  </parameterlist>
 </program>
 <program>
  <general>
   <name>Modgen</name>
   <binary>modgen/modgen.sh</binary>
  </general>
  <input>
   <type>XML–CAPP</type>
   <converter></converter>
   <parameter>CAPP Input Filename</parameter>
  </input>
  <input>
   <type>XML–PH</type>
   <converter></converter>
   <parameter>PH Input Filename</parameter>
  </input>
  <input>
   <type>XML–MAP</type>
   <converter></converter>
   <parameter>Map Input Filename</parameter>
  </input>
  <input>
   <type>XML–ARIMA</type>
   <converter></converter>
   <parameter>ARIMA Input Filename</parameter>
  </input>
  <input>
   <type>XML–ARTA</type>
   <converter></converter>
   <parameter>ARTA Input Filename</parameter>
  </input>
  <output>
   <type>XML-Trace</type>
   <parameter>Output Filename</parameter>
  </output>
  <parameterlist>
   <parametergroup visible="true" type="filename">
    <name>CAPP Input Filename</name>
    <description>Define Input Filename.</description>
    <key></key>
    <default></default>
    <successor>PH Input Filename</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename">
    <name>PH Input Filename</name>
    <description>Define Input Filename.</description>
    <key></key>
    <default></default>
    <successor>Map Input Filename</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename">
    <name>Map Input Filename</name>
    <description>Define Input Filename.</description>
    <key></key>
    <default></default>
    <successor>ARIMA Input Filename</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename">
    <name>ARIMA Input Filename</name>
    <description>Define Input Filename.</description>
    <key></key>
    <default></default>
    <successor>ARTA Input Filename</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename">
    <name>ARTA Input Filename</name>
    <description>Define Input Filename.</description>
    <key></key>
    <default></default>
    <successor>Output Filename</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename">
    <name>Output Filename</name>
    <description>Define Output Filename.</description>
    <key></key>
    <default></default>
    <successor>TraceLength</successor>
   </parametergroup>
   <parametergroup visible="true" type="text">
    <name>TraceLength</name>
    <description>Define the length of the to be generated trace
    </description>
    <key>–observations=</key>
    <default>10000</default>
    <parameter type="static">
     <name>observations</name>
     <description></description>
     <successor></successor>
    </parameter>
    <successor></successor>
   </parametergroup>
  </parameterlist>
 </program>
 <program>
  <general>
   <name>Plot</name>
   <binary>Plot/plot.exe</binary>
  </general>
  <input>
   <type>XML–Trace</type>
   <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
   <parameter>Trace Input Filenames</parameter>
  </input>
  <input>
   <type>XML–PH</type>
   <converter></converter>
   <parameter>PH Input Filenames</parameter>
  </input>
  <input>
   <type>XML–MAP</type>
   <converter></converter>
   <parameter>Map Input Filenames</parameter>
  </input>
  <input>
   <type>XML–ARIMA</type>
   <converter></converter>
   <parameter>ARIMA Input Filenames</parameter>
  </input>
  <input>
   <type>XML–ARTA</type>
   <converter></converter>
   <parameter>ARTA Input Filenames</parameter>
  </input>
  <input>
   <type>XML–CAPP</type>
   <converter></converter>
   <parameter>CAPP Input Filenames</parameter>
  </input>
  <output>
   <type>XML–PS</type>
   <type>XML–PDF</type>
   <type>XML–PNG</type>
   <converter>scripts/FileNameToXMLReference.sh</converter>
   <parameter>Output Filename</parameter>
  </output>
  <parameterlist>
   <parametergroup visible="true" type="select">
    <name>Format</name>
    <description>Define the desired output format type.</description>
    <key>–format=</key>
    <default></default>
    <values>
     <value>ps</value>
     <value>pdf</value>
     <value>png</value>
    </values>
    <successor>Start</successor>
   </parametergroup>
   <parametergroup visible="true" type="text">
    <name>Start</name>
    <description>Define the start value.</description>
    <key>–start=</key>
    <default>0</default>
    <parameter type="static">
     <name>start</name>
     <description></description>
     <successor></successor>
    </parameter>
    <successor>End</successor>
   </parametergroup>
   <parametergroup visible="true" type="text">
    <name>End</name>
    <description>Define the end Value.</description>
    <key>–end=</key>
    <default>10</default>
    <parameter type="static">
     <name>end</name>
     <description></description>
     <successor></successor>
    </parameter>
    <successor>Type</successor>
   </parametergroup>
   <parametergroup visible="true" type="select">
    <name>Type</name>
    <description>Define what information to plot.</description>
    <key>–plot=</key>
    <default> </default>
    <values>
     <value>cdf</value>
     <value>pdf</value>
     <value>ac</value>
     <value>mom</value>
     <value>jmom</value>
    </values>
    <successor>Output Filename</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename">
    <name>Output Filename</name>
    <description>Define Output Filename.</description>
    <key>–output=</key>
    <default></default>
    <successor>Trace Input Filenames</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename" multi="true">
    <name>Trace Input Filenames</name>
    <description>Define Input Filenames.</description>
    <key>–trace </key>
    <default></default>
    <successor>PH Input Filenames</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename" multi="true">
    <name>PH Input Filenames</name>
    <description>Define Input Filenames.</description>
    <key>–cph </key>
    <default></default>
    <successor>Map Input Filenames</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename" multi="true">
    <name>Map Input Filenames</name>
    <description>Define Input Filenames.</description>
    <key>–map </key>
    <default></default>
    <successor>ARIMA Input Filenames</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename" multi="true">
    <name>ARIMA Input Filenames</name>
    <description>Define ARIMA Input Filenames</description>
    <key>–arima </key>
    <default></default>
    <successor>ARTA Input Filenames</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename" multi="true">
    <name>ARTA Input Filenames</name>
    <description>Define ARTA Input Filenames</description>
    <key>–arta </key>
    <default></default>
    <successor>CAPP Input Filenames</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename" multi="true">
    <name>CAPP Input Filenames</name>
    <description>Define Input Filenames.</description>
    <key>–model </key>
    <default></default>
    <successor></successor>
   </parametergroup>
  </parameterlist>
 </program>
 <program>
  <general>
   <name>Print</name>
   <binary>Plot/plot.exe</binary>
  </general>
  <input>
   <type>XML–Trace</type>
   <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
   <parameter>Trace Input Filenames</parameter>
  </input>
  <input>
   <type>XML–PH</type>
   <converter></converter>
   <parameter>PH Input Filenames</parameter>
  </input>
  <input>
   <type>XML–MAP</type>
   <converter></converter>
   <parameter>Map Input Filenames</parameter>
  </input>
  <input>
   <type>XML–ARIMA</type>
   <converter></converter>
   <parameter>ARIMA Input Filenames</parameter>
  </input>
  <input>
   <type>XML–ARTA</type>
   <converter></converter>
   <parameter>ARTA Input Filenames</parameter>
  </input>
  <input>
   <type>XML–CAPP</type>
   <converter></converter>
   <parameter>CAPP Input Filenames</parameter>
  </input>
  <output>
   <type>XML–Latex</type>
   <type>XML–Text</type>
   <converter>scripts/FileNameToXMLReference.sh</converter>
   <parameter>Output Filename</parameter>
  </output>
  <parameterlist>
   <parametergroup visible="true" type="select">
    <name>Format</name>
    <description>Define the desired output format type.</description>
    <key>–format=</key>
    <default>txt</default>
    <values>
     <value>txt</value>
     <value>latex</value>
    </values>
    <successor>Relative</successor>
   </parametergroup>
   <parametergroup visible="false" type="text">
    <name>Relative</name>
    <description>Print values relative to values of first input model.
```

```
   </description>
   <key>–relative=</key>
   <default>false</default>
   <parameter type="static">
    <name>relative</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Start</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>Start</name>
   <description>Define the start value.</description>
   <key>–start=</key>
   <default>0</default>
   <parameter type="static">
    <name>start</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>End</successor>
  </parametergroup>
  <parametergroup visible="true" type="text">
   <name>End</name>
   <description>Define the end Value.</description>
   <key>–end=</key>
   <default>10</default>
   <parameter type="static">
    <name>end</name>
    <description></description>
    <successor></successor>
   </parameter>
   <successor>Type</successor>
  </parametergroup>
  <parametergroup visible="true" type="select">
   <name>Type</name>
   <description>Define what information to print.</description>
   <key>–plot=</key>
   <default> </default>
   <values>
    <value>cdf</value>
    <value>pdf</value>
    <value>ac</value>
    <value>mom</value>
    <value>jmom</value>
   </values>
   <successor>Output Filename</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename">
   <name>Output Filename</name>
   <description>Define Output Filename.</description>
   <key>–output=</key>
   <default></default>
   <successor>Trace Input Filenames</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename" multi="true">
   <name>Trace Input Filenames</name>
   <description>Define Input Filenames.</description>
   <key>—trace </key>
   <default></default>
   <successor>PH Input Filenames</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename" multi="true">
   <name>PH Input Filenames</name>
   <description>Define Input Filenames.</description>
   <key>—cph </key>
   <default></default>
   <successor>Map Input Filenames</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename" multi="true">
   <name>Map Input Filenames</name>
   <description>Define Input Filenames.</description>
   <key>—map </key>
   <default></default>
   <successor>ARIMA Input Filenames</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename" multi="true">
   <name>ARIMA Input Filenames</name>
   <description>Define ARIMA Input Filenames</description>
   <key>—arima </key>
   <default></default>
   <successor>ARTA Input Filenames</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename" multi="true">
   <name>ARTA Input Filenames</name>
   <description>Define ARTA Input Filenames</description>
   <key>—arta </key>
   <default></default>
   <successor>CAPP Input Filenames</successor>
  </parametergroup>
  <parametergroup visible="true" type="filename" multi="true">
   <name>CAPP Input Filenames</name>
   <description>Define CAPP Input Filenames</description>
   <key>—model </key>
   <default></default>
   <successor></successor>
  </parametergroup>
 </parameterlist>
</program>
<program>
 <general>
  <name>Queue</name>
  <binary>queue/queue.sh</binary>
 </general>
 <input>
  <type>XML–Trace</type>
```

```
    <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
    <parameter>Trace Input Filenames</parameter>
   </input>
   <input>
    <type>XML-MAP</type>
    <converter></converter>
    <parameter>Map Input Filenames</parameter>
   </input>
   <input>
    <type>XML-ARIMA</type>
    <converter></converter>
    <parameter>ARIMA Input Filenames</parameter>
   </input>
   <input>
    <type>XML-ARTA</type>
    <converter></converter>
    <parameter>ARTA Input Filenames</parameter>
   </input>
   <input>
    <type>XML-CAPP</type>
    <converter></converter>
    <parameter>CAPP Input Filenames</parameter>
   </input>
   <output>
    <type>XML-PS</type>
    <converter>scripts/FileNameToXMLReference.sh</converter>
    <parameter>Output Filename</parameter>
   </output>
   <parameterlist>
    <parametergroup visible="true" type="text">
     <name>Service-Time</name>
     <description>Define the service time of the server.</description>
     <key></key>
     <default>"exponential(1.0s)"</default>
     <parameter type="static">
      <name>service time</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Buffer</successor>
    </parametergroup>
    <parametergroup visible="true" type="text">
     <name>Buffer</name>
     <description>Define the maximum buffer size</description>
     <key></key>
     <default>10</default>
     <parameter type="static">
      <name>buffer</name>
      <description></description>
      <successor></successor>
     </parameter>
     <successor>Sim. Time</successor>
    </parametergroup>
```

```
<parametergroup visible="true" type="text">
 <name>Sim. Time</name>
 <description>Define the simulation time.</description>
 <key></key>
 <default>175000s</default>
 <parameter type="static">
  <name>sim. time</name>
  <description></description>
  <successor></successor>
 </parameter>
 <successor>Output Filename</successor>
</parametergroup>
<parametergroup visible="true" type="filename">
 <name>Output Filename</name>
 <description>Define Output Filename.</description>
 <key></key>
 <default></default>
 <successor>Trace Input Filenames</successor>
</parametergroup>
<parametergroup visible="true" type="filename" multi="true">
 <name>Trace Input Filenames</name>
 <description>Define Input Filenames.</description>
 <key>–trace </key>
 <default></default>
 <successor>Map Input Filenames</successor>
</parametergroup>
<parametergroup visible="true" type="filename" multi="true">
 <name>Map Input Filenames</name>
 <description>Define Input Filenames.</description>
 <key>–map </key>
 <default></default>
 <successor>ARIMA Input Filenames</successor>
</parametergroup>
<parametergroup visible="true" type="filename" multi="true">
 <name>ARIMA Input Filenames</name>
 <description>Define ARIMA Input Filenames</description>
 <key>–arima </key>
 <default></default>
 <successor>ARTA Input Filenames</successor>
</parametergroup>
<parametergroup visible="true" type="filename" multi="true">
 <name>ARTA Input Filenames</name>
 <description>Define ARTA Input Filenames</description>
 <key>–arta </key>
 <default></default>
 <successor>CAPP Input Filenames</successor>
</parametergroup>
<parametergroup visible="true" type="filename" multi="true">
 <name>CAPP Input Filenames</name>
 <description>Define CAPP Input Filenames</description>
 <key>–capp </key>
 <default></default>
 <successor></successor>
```

```
   </parametergroup>
  </parameterlist>
 </program>
 <program>
  <general>
   <name>Dist_Tests</name>
   <article></article>
   <binary>disttest/disttest.r</binary>
   <requires>Rscript</requires>
  </general>
  <input>
   <type>XML-Trace</type>
   <converter>scripts/CreateLinkToXMLReferenceTarget.tcl</converter>
   <parameter>Trace Input Filenames</parameter>
  </input>
  <output>
   <type>XML-Latex</type>
   <converter>scripts/FileNameToXMLReference.sh</converter>
   <parameter>Output Filename</parameter>
  </output>
  <parameterlist>
   <parametergroup visible="true" type="filename" multi="true">
    <name>Trace Input Filenames</name>
    <description>Define Input Filename.</description>
    <key></key>
    <default>trace.xml</default>
    <successor>Output Filename</successor>
   </parametergroup>
   <parametergroup visible="true" type="filename">
    <name>Output Filename</name>
    <description>Define Output Filename.</description>
    <key></key>
    <default>test_results.xml</default>
    <successor></successor>
   </parametergroup>
  </parameterlist>
 </program>
</software>
<tools>
 <requires>sh, wish</requires>
 <converter type="input">
  <input>
   <type>Trace</type>
   <suffix>.trace</suffix>
  </input>
  <output>
   <type>XML-Trace</type>
   <suffix>.xml</suffix>
  </output>
  <binary>scripts/TraceToXMLTrace.sh</binary>
 </converter>
 <converter type="input">
  <input>
   <type>XML-MAP</type>
   <suffix>.xml</suffix>
  </input>
  <output>
   <type>XML-MAP</type>
   <suffix>.xml</suffix>
  </output>
  <binary>scripts/CopyFile.sh</binary>
 </converter>
 <converter type="input">
  <input>
   <type>XML-Dist</type>
   <suffix>.xml</suffix>
  </input>
  <output>
   <type>XML-Dist</type>
   <suffix>.xml</suffix>
  </output>
  <binary>scripts/CopyFile.sh</binary>
 </converter>
 <converter type="input">
  <input>
   <type>XML-ARTA</type>
   <suffix>.xml</suffix>
  </input>
  <output>
   <type>XML-ARTA</type>
   <suffix>.xml</suffix>
  </output>
  <binary>scripts/CopyFile.sh</binary>
 </converter>
 <converter type="input">
  <input>
   <type>XML-CAPP</type>
   <suffix>.xml</suffix>
  </input>
  <output>
   <type>XML-CAPP</type>
   <suffix>.xml</suffix>
  </output>
  <binary>scripts/CopyFile.sh</binary>
 </converter>
 <converter type="input">
  <input>
   <type>XML-PH</type>
   <suffix>.xml</suffix>
  </input>
  <output>
   <type>XML-PH</type>
   <suffix>.xml</suffix>
  </output>
  <binary>scripts/CopyFile.sh</binary>
 </converter>
```

```
<converter type="input">
 <input>
  <type>XML-ARIMA</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>XML-ARIMA</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/CopyFile.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-MAP</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Map</type>
  <suffix>.map</suffix>
 </output>
 <binary>scripts/XMLMapToMAPEM.tcl</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-Trace</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Trace</type>
  <suffix>.trace</suffix>
 </output>
 <binary>scripts/CreateCopyOfXMLReferenceTarget.tcl</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-PS</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>PS</type>
  <suffix>.ps</suffix>
 </output>
 <binary>scripts/CreateCopyOfXMLReferenceTarget.tcl</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-PDF</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>PDF</type>
  <suffix>.pdf</suffix>
 </output>
 <binary>scripts/CreateCopyOfXMLReferenceTarget.tcl</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-PNG</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>PNG</type>
  <suffix>.png</suffix>
 </output>
 <binary>scripts/CreateCopyOfXMLReferenceTarget.tcl</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-Latex</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Latex</type>
  <suffix>.tex</suffix>
 </output>
 <binary>scripts/CreateCopyOfXMLReferenceTarget.tcl</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-Text</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Txt</type>
  <suffix>.txt</suffix>
 </output>
 <binary>scripts/CreateCopyOfXMLReferenceTarget.tcl</binary>
</converter>
<converter type="output">
 <input>
  <type>Trace</type>
  <suffix>.trace</suffix>
 </input>
 <output>
  <type>XML-Trace</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/TraceToXMLTrace.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-MAP</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>XML-MAP</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/CopyFile.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-Dist</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>XML-Dist</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/CopyFile.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-ARTA</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>XML-ARTA</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/CopyFile.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-CAPP</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>XML-CAPP</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/CopyFile.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-PH</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>XML-PH</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/CopyFile.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-ARIMA</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>XML-ARIMA</type>
  <suffix>.xml</suffix>
 </output>
 <binary>scripts/CopyFile.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-Latex</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Report-Latex</type>
  <suffix>.tex</suffix>
 </output>
 <binary>scripts/XMLLatexToReport.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-PDF</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Report-Latex</type>
  <suffix>.tex</suffix>
 </output>
 <binary>scripts/XMLPDFToReport.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-MAP</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Report-Latex</type>
  <suffix>.tex</suffix>
 </output>
 <binary>scripts/XMLModelToReport.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML-Reference</type>
  <suffix>.xml</suffix>
 </input>
 <output>
  <type>Report-Latex</type>
  <suffix>.tex</suffix>
 </output>
 <binary>scripts/PlotToReport.sh</binary>
</converter>
<converter type="output">
 <input>
  <type>XML</type>
```

```
   <suffix>.xml</suffix>
  </input>
  <output>
   <type>Report-Latex</type>
   <suffix>.tex</suffix>
  </output>
  <binary>scripts/XMLToReport.sh</binary>
 </converter>
</tools>
<edgeviewer>
 <viewer>
  <type>XML-MAP</type>
  <type>XML-Trace</type>
  <type>XML-PH</type>
  <type>XML-PNG</type>
  <type>XML-PS</type>
  <type>XML-Latex</type>
  <name>Show XML-File</name>
  <intern>text</intern>
 </viewer>
 <viewer>
  <type>XML-MAP</type>
  <type>XML-Trace</type>
  <type>XML-PH</type>
  <type>XML-ARTA</type>
  <type>XML-ARIMA</type>
  <type>XML-CAPP</type>
  <name>Plot Details</name>
  <program>
   <name>Plot</name>
   <preset>
    <parametergroup fixed="true">
     <name>Format</name>
     <value>png</value>
    </parametergroup>
   </preset>
  </program>
  <converter>
   <input>XML-PNG</input>
   <output>PNG</output>
  </converter>
  <intern>image</intern>
 </viewer>
 <viewer>
  <type>XML-MAP</type>
  <type>XML-Trace</type>
  <type>XML-PH</type>
  <type>XML-ARTA</type>
  <type>XML-ARIMA</type>
  <type>XML-CAPP</type>
  <name>Print Details</name>
  <program>
   <name>Print</name>
   <preset>
    <parametergroup>
     <name>Format</name>
     <value>txt</value>
    </parametergroup>
   </preset>
  </program>
  <converter>
   <input>XML-Text</input>
   <output>Txt</output>
  </converter>
  <intern>text</intern>
 </viewer>
</edgeviewer>
<report>
 <viewer>
  <type>XML-MAP</type>
  <type>XML-Trace</type>
  <type>XML-PH</type>
  <type>XML-ARIMA</type>
  <type>XML-ARTA</type>
  <type>XML-CAPP</type>
  <name>Plot</name>
  <description>Creates a graphical representation of the model.
  </description>
  <program>
   <name>Plot</name>
   <preset>
    <parametergroup>
     <name>Format</name>
     <value>pdf</value>
    </parametergroup>
   </preset>
  </program>
  <converter>
   <input>XML-PDF</input>
   <output>Report-Latex</output>
  </converter>
 </viewer>
 <viewer>
  <type>XML-MAP</type>
  <type>XML-Trace</type>
  <type>XML-PH</type>
  <type>XML-ARIMA</type>
  <type>XML-ARTA</type>
  <type>XML-CAPP</type>
  <name>Print</name>
  <description>Creates a textual representation of the model.
  </description>
  <program>
   <name>Print</name>
   <preset>
    <parametergroup>
```

```
      <name>Format</name>
      <value>latex</value>
     </parametergroup>
     <!--
     -->
    </preset>
   </program>
   <converter>
    <input>XML-Latex</input>
    <output>Report-Latex</output>
   </converter>
  </viewer>
  <viewer>
   <type>XML-Text</type>
   <type>XML-Latex</type>
   <type>XML-PS</type>
   <type>XML-PDF</type>
   <type>XML-PNG</type>
   <name>PlotOutput</name>
   <description>Includes any created plot file into the report.
   </description>
   <converter>
    <input>XML-Reference</input>
    <output>Report-Latex</output>
   </converter>
  </viewer>
  <viewer>
   <type>XML-MAP</type>
   <type>XML-Trace</type>
   <type>XML-PH</type>
   <type>XML-Text</type>
   <type>XML-Latex</type>
   <type>XML-PS</type>
   <type>XML-PDF</type>
   <type>XML-PNG</type>
   <name>ModelPrint</name>
   <description>Encapsulates the XML-File into a Latex verbatim
    environment.
   </description>
   <converter>
    <input>XML</input>
    <output>Report-Latex</output>
   </converter>
  </viewer>
 </report>
</settings>
```