ProFiDo – The Processes Fitting Toolkit Dortmund

Falko Bause, Peter Buchholz, Jan Kriege
Informatik IV, TU Dortmund, D-44221 Dortmund
{falko.bause,peter.buchholz,jan.kriege}@udo.edu

Author prepared version of a paper published in
Proc. of the 7th International Conference on Quantitative Evaluation of SysTems (QEST 2010), IEEE Computer Society, 2010.

http://doi.ieeecomputersociety.org/10.1109/QEST.2010.20

ProFiDo – The Processes Fitting Toolkit Dortmund

Falko Bause, Peter Buchholz, Jan Kriege
Informatik IV, TU Dortmund, D-44221 Dortmund
{falko.bause,peter.buchholz,jan.kriege}@udo.edu

Abstract

This paper describes the Java-based toolkit ProFiDo which integrates several tools for fitting input models. Currently supported are command line tools for fitting probability distributions, ARIMA processes and Markovian arrival processes. The toolkit provides a graphical user interface which allows for the specification of workflows that describe the different steps of data preprocessing, parameter fitting and result visualization.

The basis for the interoperability of the different tools is an XML based interchange format for the specification of various types of processes. An XML based configuration file supports the extension of the toolkit by integrating additional fitting methods or analysis approaches.

1 Introduction

In stochastic modeling the appropriate representation of quantitative information that describes arrivals, services, failures, repairs or similar aspects is of outstanding importance. Often measurements from a real system or a detailed simulation model are available that have to be modeled by some stochastic model. The measurements are denoted as traces and the process of setting the parameters of the model in question is denoted as parameter fitting.

Parameter fitting is usually restricted to distributions. In simulation different tools are commercially available that generate a distribution modeling the empirical distribution of the trace which can then be directly used in different simulation environments. Examples for those tools are the Arena input analyzer [14] or ExpertFit [16]. Maximum likelihood (ML) estimates for the parameters of different distributions are known [13]. However, general distributions cannot be used in analytically solvable models which usually require some Markov model to represent the distribution. The most general class of Markovian models are phase type (PH) distributions [18]. For the fitting of the parameters of a PH distribution different methods exist which have been implemented in prototype programs [1, 11, 27]. However, a general tool incorporating different fitting methods is missing to the best of our knowledge.

Additionally, it is known that in practice the assumption of independent and identically distributed data (*iid*) is not satisfied in many application areas [23]. Often inter-event times in a trace are correlated and the negligence of these correlations in stochastic models results in models insufficient to describe real systems. For modeling time dependent stationary processes the family of AR (Auto Regressive), ARMA (Auto Regressive Moving Average), ARIMA (Auto Regressive Integrated Moving Average) and ARTA (Auto Regressive To Anything [6]) models are known since the pioneering work of Box and Jenkins [3] in the late sixties. However, these methods are only rarely used in simulation although several of these models are nowadays incorporated in standard statistical software [26] or are supported by specific tools [2, 7]. Like general distributions, AR processes cannot be used in analytically solvable models. In these models again Markovian descriptions need to be used which are available in form of Markovian Arrival Processes (*MAPs*) [17]. The parameter fitting problem for MAPs is inherently complex and only recently first prototype implementations of parameter fitting tools became available [4, 8, 22].

In summary, the situation with fitting AR processes or MAPs is not really satisfactory since many different approaches are available but are not accessible in a coherent way. This makes the use of the methods and their comparison very cumbersome. Consequently, it would be important to have a tool that contains several of the available methods and allows one to use them in a unified way to model traces. Apart from this the resulting models should be easily usable in analysis tools, let it be simulation tools or tools for analytically solvable models like extended queuing networks or SPNs. Furthermore, the comparison of different processes by comparing the resulting distributions and the correlation structure should be supported.

This paper presents the tool ProFiDo [24] which is a flexible environment that allows the integration of command line tools realizing different steps of data preprocessing, parameter fitting and analyzing the resulting processes. The idea of the tool is to provide an XML based interchange format for the interoperability of the different fitting tools and a graphical interface

to define workflows that specify the different steps. For each step a command line tool has to be available. In the current version, ProFiDo includes several of our own fitting methods and some open source tools. The available framework allows an easy integration of further methods and tools. Nevertheless, even in its current state it is to the best of our knowledge the first tool that includes methods for generating PH distributions, MAPs and AR models from trace data.

The paper is structured as follows. In the next section we give a very brief overview of the different processes and the available fitting methods. Section 3 introduces the basic ideas underlying the tool. Then, in Section 4, we introduce the configuration of the tool and give afterwards an overview of the XML format that is used to interchange information between modules. In Section 5 an example is presented. The paper ends with the conclusions and a short outlook on future extensions of the tool.

2 Theoretical Background

For the lack of space it is not possible to give a detailed description of the different fitting methods that are currently integrated in ProFiDo. Thus, we only present a brief overview of the process types that are supported and introduce some pointers to the literature were to find the algorithms currently included in the tool. ProFiDo considers two general classes of processes, namely AR processes to be used in conjunction with simulation and MAPs usable for simulation and numerical analysis approaches.

We begin with the former class and briefly introduce processes of the AR type, for further details we refer to [3]. Let $z_k = \tilde{z}_k + \mu$ be the inter-event time of the kth event with

$$\tilde{z}_k = \sum_{i=1}^{p} \phi_i \tilde{z}_{k-i} + a_k + \sum_{j=1}^{q} \theta_j a_{k-j} \tag{1}$$

where μ is the mean of the inter-event time, a_k is a random variable with normal distribution, mean 0 and variance σ_a. $\theta_1, \ldots, \theta_q$ are the moving average parameters and $\phi_1, \ldots, \phi_p$ are the autoregressive parameters. The above process is denoted as ARMA model, choosing $p = 0$ we obtain an MA process and choosing $q = 0$ results in an AR process. An extension of ARMA models are ARIMA models which even allow for the fitting of homogeneous non-stationary behavior [3].

Parameter fitting for ARIMA models usually implies that the above parameters are estimated with a regression approach that is available in several statistical tools including the software R [26]. It should be noted that (1) may yield negative values for z_k which are not usable, if realizations are interpreted as time steps. If this problem occurs, negative values have to be neglected or the values of the trace have to be logarithmically scaled such that e^{z_k} becomes the realization of the kth time.

PH distributions and MAPs are described by Markov processes. Let n be the number of states, then a PH distribution is given by $(\pi, \mathbf{D_0})$ where π is the initial probability vector and $\mathbf{D_0}$ is an $n \times n$ generator matrix of an absorbing Markov process. The representation $(\pi, \mathbf{D_0})$ is redundant since $2n - 1$ parameters are sufficient to characterize a PH distribution with n states (cf. [19, 20, 25]). Unfortunately, for the general class of PH distributions no canonical representation of the parameters is known yet, only for the subclass of acyclic PH distributions (i.e., matrix $\mathbf{D_0}$ can be transformed into an upper triangular matrix by permutations of rows and columns) a canonical form with a minimal number of parameters is available.

Consequently, many fitting methods fit the parameters of acyclic PH distributions or even subclasses of this class. For the maximization of the likelihood value of PH distributions, often expectation maximization (EM) algorithms are used. EM algorithms are local optimization methods with a monotonic but often slow convergence towards a local optimum. The use of EM algorithms for parameter fitting of PH distributions has been proposed in [1]. The basic approach becomes impractical if the trace is long or the number of states of the PH distribution becomes larger. More efficient is the use of EM algorithms for specific subclasses of PH distributions like in the tool G-FIT which uses an EM algorithm to fit the parameters of hyper-Erlang distributions [27]. Other approaches use general optimization algorithms to fit the parameters of acyclic PH distributions in order to maximize the likelihood according to the trace or approximate the moments of the trace [5, 11].

To include correlations, MAPs rather than PH distributions are used. A MAP is characterized by two $n \times n$ matrices $(\mathbf{D_0}, \mathbf{D_1})$ such that $\mathbf{D_0} + \mathbf{D_1}$ is a generator of an irreducible Markov process, $\mathbf{D_0}$ is non-singular and contains non-negative values outside the diagonal and $\mathbf{D_1}$ contains only positive values. Every PH distributions can be expanded into a MAP by defining $\mathbf{D_1} = (-\mathbf{D_0}\mathbb{1})\pi$ where $\mathbb{1}$ is a vector of ones of length n.

Parameter fitting of MAPs can also be done with EM algorithms as extensions of [1]. In [4] one specific realization is described. However, when the approach is used without any preprocessing, it is inefficient and not usable for larger traces or MAPs of a larger order. But due to its monotonic convergence, an EM algorithm may be used to improve a MAP which has

been computed with some other method. Alternative and more efficient fitting methods for MAPs fit derived quantities like joint moments or lag k autocorrelation coefficients and are presented for example in [5, 8, 12, 22].

It is often more efficient to fit the parameters of a MAP in two steps. First, a PH distribution is generated which is afterwards expanded to a MAP. This approach is used in [5, 12, 22] and allows the combination of different fitting steps. E.g., one may start with a moment fitting as in [5], improve the resulting PH distribution by an EM algorithm as in [1] or [4]. The resulting PH distribution is then expanded into a MAP by fitting the lag k autocorrelation [12] or the joint moments [5]. Finally, the MAP is improved by some additional iterations of an EM algorithm as in [4]. To realize such an approach, one needs a very flexible software environment that incorporates modules performing the different steps. Of course, to combine the steps, an interchange format for traces, distributions and MAPs is needed too. The tool ProFiDo has been developed with the mentioned workflow in mind.

3 ProFiDo

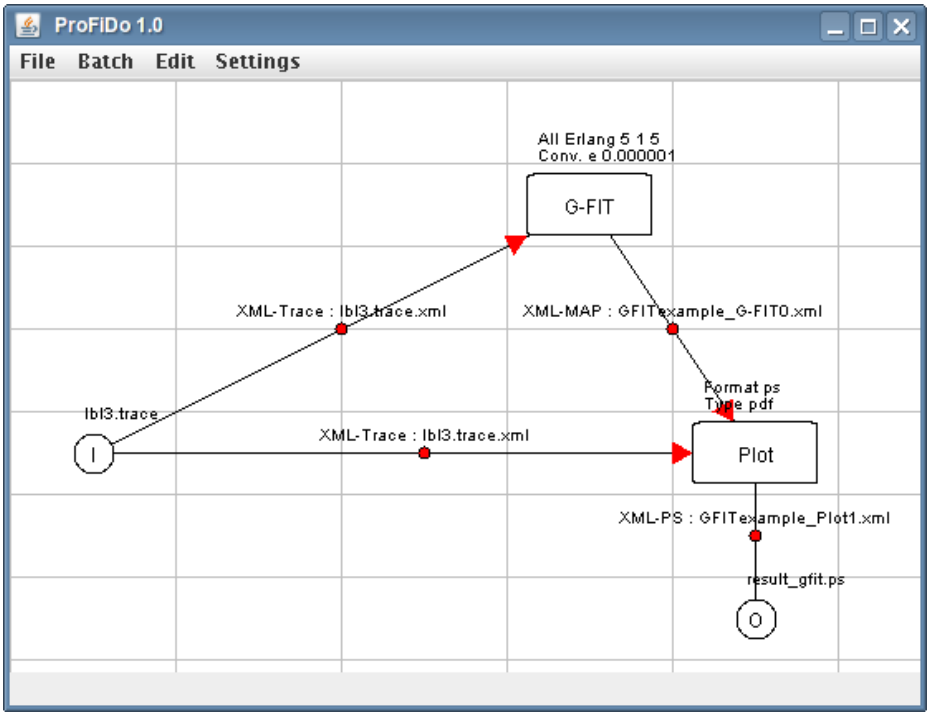


Figure 1: Example of a workflow

ProFiDo is a Java-based toolkit which integrates several command-line based fitting tools into a consistent user interface. The graphical user interface provided by ProFiDo enables the user to specify custom workflows of program executions and result propagation. The tool allows one to define by means of some workflow stepwise approaches to fit parameters of distributions or processes according to some trace. The resulting distributions and processes can then be compared with one another and with the original trace. Thus, ProFiDo is a framework for parameter fitting and for evaluating different fitting methods. The basic concept of the tool is a workflow that will be presented first.

An example of such a workflow is shown in Fig. 1. The intention of this workflow is to evaluate the fitting result received from a call to the tool G-FIT [27], which fits a hyper-Erlang distribution to a given trace, by comparing the probability density function (pdf) of the fitted hyper-Erlang distribution with the one of the original trace.

Workflows in ProFiDo consist of two types of nodes: nodes indicating the call of a tool, so-called job nodes in the following, and nodes which indicate inputs into and outputs from the workflow. In Fig. 1 two job nodes are shown, `G-FIT` and `Plot`, and two nodes for input/output, `I` and `O`. As mentioned, most fitting tools use different input/output formats, making their use in such workflows difficult. Therefore we developed an XML based interchange format (cf. Sect. 5) enabling a consistent data flow between job nodes.

The workflow of Fig. 1 operates as follows: At input node `I` the file `lbl3.trace` is read and converted to a trace description within the interchange format, called `XML-Trace`. This kind of trace serves as input for the job nodes `G-FIT` and `Plot` which is indicated by the directed arcs starting at input node `I`. The output of job node `G-FIT` is the result of a call to the tool G-FIT for the given trace. In the interchange format this result, originally a hyper-Erlang distribution, is represented by the description of a Markovian Arrival Process (MAP), called `XML-MAP`. The result of G-FIT, i.e. the `XML-MAP`, and the original `XML-Trace` are used by job node `Plot` to visualize both pdfs as shown in Fig. 2. `Plot` is a special node offered by ProFiDo. It is able to read traces and descriptions of distributions, MAPs, ARIMA and ARTA processes and optionally generates a plot of probability density functions, (complementary) cumulative distribution functions, (joint) moments, or autocorrelation functions of the inputs within a single diagram.

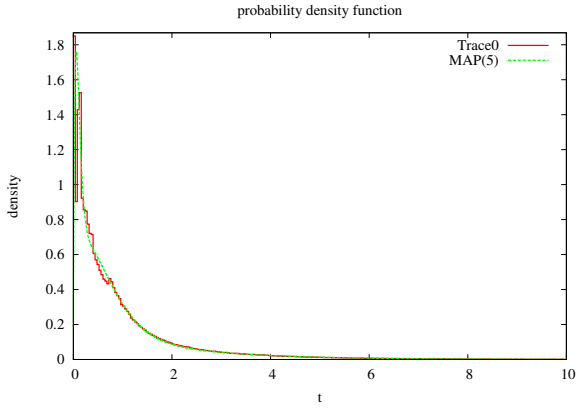


Figure 2: Pdfs of the LBL3 trace and the fitted MAP(5) (according to the workflow shown in Fig. 1).

The output of job node `Plot` is, as for all other job nodes, in interchange format. In Fig. 1 the output of `Plot` is directed to output node `O` where the result in interchange format is converted into a postscript file named `result_gfit.ps`.

Job nodes usually represent the call of a command-line based tool which cannot handle input or produce output in the interchange format directly. Thus, technically, a call indicated by a job node includes calls to appropriate converter scripts, which transform data in interchange format into tool specific input or the output of tools into the interchange format. More details will be given in Sects. 4 and 5.

Apart from the specification of job nodes, the user needs to define directed arcs in order to make the workflow description complete. The only attribute of arcs in ProFiDo is the name of a file storing the data in interchange format. All these files are stored in a separate result directory. By default the arc attributes are set such that no conflicts can occur.

After the workflow description has been completed the user can generate a shell script at the push of a button which executes the entire workflow. The shell script of the workflow in Fig. 1 is shown in Fig. 3. The initial part covers the conversion of the input in file `lbl3.trace`. Afterwards scripts for the job nodes `G-FIT` and `Plot` are called followed by a conversion of the result from the interchange format into postscript.

The script `gfit.sh` being called in the code of Fig. 3 is displayed in Fig. 4. It is a typical example of a script associated with a command-line based fitting tool. The first part deals with parameter handling and the conversion of data in interchange format into tool specific input. Due to performance reasons the interchange format for a trace contains mainly a link to the original trace file (cf. Sect. 5). Afterwards the tool G-FIT is called followed by a conversion of its output into the interchange format.

Currently the following fitting tools are supported by ProFiDo:

- G-FIT [27]: An EM algorithm is used to fit the parameters of hyper-Erlang distributions maximizing the likelihood value. Due to the restricted structure of the distributions, the algorithm is efficient and can be applied to large traces. As shown by many examples fitting results are usually as good as fitting results for more general PH distributions.

- MomFit [5]: A non linear optimization algorithm is applied to fit the parameters of an APH distribution to moments of the trace. The optimization approach minimizes the least squares difference between the weighted moments of the trace and the fitted distribution. Since moments rather than the likelihood are optimized, fitting becomes independent of the length of the trace.

```
#!/bin/sh
# ------------------------------
# Export of workflow: GFITexample
# ------------------------------

export BASEDIR="/home/fred/FT"

# Derived folders:
export RESULTDIR="${BASEDIR}/Results"
export SOFTWAREDIR="${BASEDIR}/Software"

# ------ Convert non-XML files ------

${SOFTWAREDIR}/scripts/TraceToXMLTraceWithValueCount.sh \
    "/home/fred/FT/lb13.trace" \
    "${RESULTDIR}/lb13.trace.xml"

# ------ Execute Jobs ------
# Execute G-FIT
cd "${SOFTWAREDIR}/Gfit/"

./gfit.sh "${RESULTDIR}/lb13.trace.xml" \
"${RESULTDIR}/GFITexample_G-FIT0.xml" \
-a 5 1 5 -e 0.000001 -par

# Execute Plot
cd "${SOFTWAREDIR}/Plot/"
${SOFTWAREDIR}/scripts/XMLMapToModgen.tcl
"${RESULTDIR}/GFITexample_G-FIT0.xml" \
"${RESULTDIR}/GFITexample_G-FIT0.xml.conv"

${SOFTWAREDIR}/scripts/CreateLinkToRefValueTarget.tcl \
"${RESULTDIR}/lb13.trace.xml" \
"${RESULTDIR}/lb13.trace.xml.conv"

./plot.exe -format=ps -plot=pdf \
-output="${RESULTDIR}/GFITexample_Plot1.xml.org" \
-trace "${RESULTDIR}/lb13.trace.xml.conv" \
-map "${RESULTDIR}/GFITexample_G-FIT0.xml.conv"

${SOFTWAREDIR}/scripts/ModgenPSToXMLPS.sh \
"${RESULTDIR}/GFITexample_Plot1.xml.org" \
"${RESULTDIR}/GFITexample_Plot1.xml"


# ------ Convert XML-Files to external outputs ------

${SOFTWAREDIR}/scripts/CreateCopyOffRefValueTarget.tcl \
"${RESULTDIR}/GFITexample_Plot1.xml" \
"/home/fred/FT/result_gfit.ps"
```

Figure 3: shell script generated for the workflow of Fig. 1

- MAP EM [4]: An EM approach is applied to the whole trace. The approach is time consuming such that it cannot be applied to fit MAPs with a larger state space to long traces. However, since EM algorithms have a monotonic convergence, the algorithm can be used to improve the likelihood of a MAP that has been fitted by some other more efficient approach.
- JMomFit [5]: The algorithm performs a least squares fitting of the weighted joint moments of the trace and the fitted MAP. It is started with a PH distribution that is expanded to a MAP. Since only linear least squares problems have to be solved and joint moments are fitted, the approach is very efficient and the effort is independent of the length of the trace. The quality of the fitting depends on the parameters of the initial PH distribution.
- Autocorrelation Fitting [12, 15]: The approach is similar to the previous one but instead of joint moments lag-k autocorrelations are fitted. This implies the use of a non-linear optimization method such that the effort becomes slightly higher than for joint moment fitting but the effort is still independent of the trace length.
- ARIMA model fitting with R [9, 26]: The free software R contains a large number of statistical methods including method to determine the parameters of ARIMA models. This function of R is used in the tool to generate ARIMA processes from trace data.

```
#!/bin/sh
# First Parameter = Input File
# Second Parameter = Output Filename
# Other Parameters = Other GFit Parameters

# This script can be seen as a black-box
# capsulating the execution of GFit.
# First the Input Tracefile is converted,
# then GFit is executed, finally the
# result is converted into the standard
# XML-Map format.

# ATTENTION: script must be executed
#            directly from its directory

INPUT=$1
OUTPUT=$2
shift
shift

# Read directories and filenames
SCRIPTDIR="../scripts/"
TRACEFILE=`${SCRIPTDIR}EchoRefFileNameFromXML.tcl "$INPUT"`

# Create link to trace file in G-Fit directory
\rm -f "local.trace"
\ln -f -s "$TRACEFILE" "local.trace"

# Execute G-Fit
./gfit "local.trace" $@

# Create XML Output
`${SCRIPTDIR}GFitToXMLMap.tcl "gfit.result" "${OUTPUT}"`

# Remove local output
\rm -f local.trace
\rm -f gfit.result
```

Figure 4: `gfit.sh` being called in the script of Fig. 3

An XML-based configuration file facilitates the extension of ProFiDo.

4 Tool Configuration

ProFiDo itself provides only a core functionality to handle graphs including support for storing and loading graphs and the parameter specification of graph elements. The fitting tools being supported by ProFiDo are not hard-coded into ProFiDo's Java code. On each start of ProFiDo an XML-based configuration file is parsed which completes the GUI's functionality.

The configuration file mainly consists of a list of node types (see `<program>` in Fig. 5) together with corresponding links to scripts and lists of parameters which are needed for a call of the corresponding tool.

Part of the parameter list for the job type `G-FIT` is shown in Fig. 6. From this specification ProFiDo's GUI generates a tool specific property window in which the user can enter parameters for the corresponding call. The property window for `G-FIT` being generated from the configuration file of Fig. 6 is shown in Fig. 7. Parameter groups are listed sequentially in the property window. Each parameter group is identified by a name attribute, see e.g. `All Erlang` in Figs. 6 and 7, together with a tool specific key determining the option identifier for a call. E.g., one possible call to G-FIT is `gfit -a 5 1 5` which instructs G-FIT to "fit all settings of a hyper-Erlang distribution and return the best fitted distribution", where the first parameter sets "the overall number of (internal) states used by the hyper-Erlang distribution" and the second and third parameters set the minimal and maximal number of Erlang branches being considered. Depending on the user input ProFiDo displays the resulting call in the property window. By default ProFiDo offers possibilities to enable and visualize parameter groups. Disabling a parameter group implies that the whole parameter group is neglected when generating the tool specific call, a visible parameter group is shown in the workflow (cf. parameter group `All Erlang` in Fig. 1). For each parameter group ProFiDo's GUI is able to display a short description (by clicking on the question marks in the property window) which results in a pop-up window presenting the information of the `name` and `description` tags, cf. Fig. 6.

Finally we want to mention some additional configuration features supported by ProFiDo:

```
[...]
<program>
        <general>
                <name>G-FIT</name>
                 <binary>Gfit/gfit.sh</binary>
        </general>
        <input>
                <type>XML-Trace</type>
                <parameter>Input Filename</parameter>
        </input>
        <output>
                <type>XML-MAP</type>
                <parameter>Output Filename</parameter>
        </output>
        <parameterlist>
          [...]
        </parameterlist>
</program>
<program>
        <general>
                <name>MAP_EM</name>
                <binary>EM/EM.sh</binary>
        </general>
        <input>
          [...]
</program>
[...]
```

Figure 5: Structure of ProFiDo's configuration file

The part of ProFiDo's configuration file in Fig. 6 shows an example of a static parameter group, i.e. where the number of parameters is fixed (as in the call `gfit -a 5 1 5`). For configuration one also has the possibility to define dynamic parameter groups where the number of parameters depends on some other parameter. E.g. another possible call to G-FIT is `gfit -f 3 2 5 7` which instructs G-Fit to determine a hyper-Erlang distribution with 3 branches and 2 states in the first, 5 states in the second and 7 states in the third branch.

The `successor` tag shown in Fig. 6 is used to define a call sequence of parameters, if several parameter groups are enabled, since some tools might accept parameters only in specific sequences.

5 Internal XML-based Interchange Format

As already mentioned ProFiDo uses an XML based interchange format for process descriptions and result propagation. As one can see from Fig. 1 ProFiDo has the policy that only XML files are interchanged between the different nodes of a workflow. Consequently, the XML interchange format must be able to describe all types of stochastic processes and distributions that the supported tools expect as input or output. Since the usual input of a workflow for fitting is a trace file and possible outputs include plots of various characteristics of the fitted models, also these types of files should be supported by the XML interchange format. In the following we will outline the general structure of these XML documents, introduce the notation for different process types and provide some demonstrative examples.

The XML description of a process in ProFiDo always starts with the tag <profido> and is ended by </profido>. The process specification itself is expected between these tags. Usually the specification contains the description of a single model, e.g. a MAP or a PH distribution. Additionally, it is possible to provide alternative descriptions of the same model, so that a tool can select the most appropriate description. An example for an alternative description is a PH distribution that can as well be described as a MAP.

The XML description of a MAP contains the two matrices $\mathbf{D_0}$ and $\mathbf{D_1}$ and the order, i.e. the number of states, of the MAP. For a PH distribution the number of states, matrix $\mathbf{D_0}$ and the initial probability vector π are required. An example for a MAP is given in Fig. 8 that shows the graphical representation of a MAP(2) and the corresponding XML description. Solid edges denote transitions from $\mathbf{D_0}$ and dashed edges transitions from $\mathbf{D_1}$.

As one can see the MAP description is enclosed by <map> ... </map>. The entries of the two matrices have to be specified in row-major form separated by blanks between the tags <d0> ... </d0> and <d1> ... </d1>, respectively.

The XML specification of a PH distribution looks similar to that of a MAP. An example for a PH distribution with 2 transient and 1 absorbing state is shown in Fig. 9. The description of the PH distribution is started and ended by <ph> and

```
<program>
  <general>
    <name>G-FIT</name>
    [...]
  </general>
    [...]
  <parameterlist>
    [...]
    <parametergroup visible="true" type="text">
      <name>All Erlang</name>
      <description>Fit all settings of a
            Hyper-Erlang distribution. </description>
      <key>-a</key>
      <default>5 1 5</default>
      <parameter type="static">
        <name>N</name>
        <description>Overall number of states.</description>
        <successor>m_min</successor>
      </parameter>
      <parameter type="static">
        <name>m_min</name>
        <description>Minimal number of branches.</description>
        <successor>m_max</successor>
      </parameter>
      <parameter type="static">
         <name>m_max</name>
         <description>Maximal number of branches.</description>
         <successor></successor>
      </parameter>
       <successor>Conv. e</successor>
    </parametergroup>
    [...]
  </parameterlist>
</program>
```

Figure 6: Part of ProFiDo's configuration file

</ph>, respectively, and contains the matrix $\mathbf{D_0}$ and the initial probability vector π in XML format. As mentioned, the XML interchange format allows for the description of equivalent representations of the same process or distribution. E.g., in addition to the PH specification, the XML description in Fig. 9 comprises a second part that contains the representation of the PH distribution as a MAP. Note that, of course the description would still be valid if either the PH part or the MAP part is omitted, but using alternative representations enables tools to choose the most appropriate description.

Beside MAPs and the related PH distributions ProFiDo's XML interchange format can be applied for the specification of other stochastic processes. An example for this are $ARIMA(p, d, q)$ processes that can be defined within the tags <arima> ... </arima>.

An example for this is given in Fig. 10. The description consists of the number of AR and MA coefficients, two lists with the vectors of AR and MA coefficients, the degree of differencing d, the variance of the white noise and the mean that the time series should fluctuate around.

Since the usual input of a workflow is a trace file and the possible output includes images with various plots, the XML interchange format has to account for these types of files as well. Hence, the format includes $<$file$>$ tags which may contain a reference to the actual data file and a type specification.

6 Examples

In the following we will emphasize the possibilities for workflow creation in ProFiDo by two demonstrative examples.

Fig. 11 shows a workflow consisting of several fitting tools that all work on the same input trace and propagate their results to the included visualization tool for a comparison of the fitting quality.

The common input trace for all tools is *LBL-TCP-3* [23] that is often used as a benchmark trace for assessing the quality of fitting tools. As one can see from Fig. 11 the workflow consists of three different branches. The first branch consists of two steps. In the first step `G-FIT` [27] is used to fit a hyper-Erlang distribution. This result is used as an initial MAP for `MAP_EM` from [4] to provide a better starting point such that the EM algorithm reaches convergence faster than with a random initial MAP. The tools from the second branch first fit a PH distribution according to the moments of the trace and

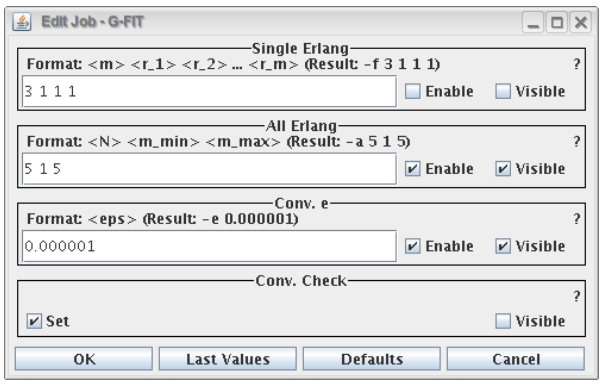
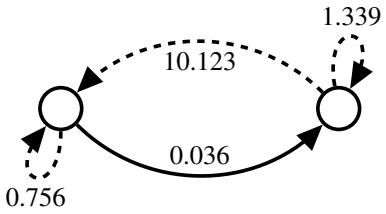


Figure 7: Properties window of G-FIT Job of Fig. 1



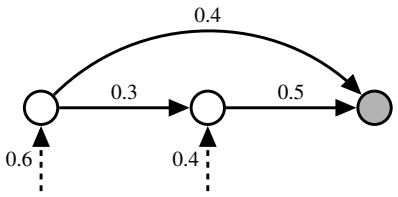

```
<profido>
  <map>
    <states> 2 </states>
    <d0>
      −0.792    0.036
       0.000  −11.462
    </d0>
    <d1>
       0.756    0.000
      10.123    1.339
    </d1>
  </map>
</profido>
```

Figure 8: $MAP(2)$ with XML description

then expand this distribution into a MAP considering the joint moments as described in [5]. Finally, the last branch fits an $ARMA(p, q)$ model using an approach from the statistical software R [9]. The trace and all the resulting models are fed into the integrated visualization tool that creates two plots in this case, one for the autocorrelation structure and one for the cumulative distribution function as shown in Fig. 12.

The second example in Fig. 13 shows an excerpt from the empirical study that has been conducted in [15] to compare different MAP fitting algorithms. The workflow from Fig. 13 consists of the joint moment fitting approach [5] that has already been used in the previous example and an autocorrelation fitting algorithm [15]. Both approaches use the same acyclic phase-type distribution which results from a moment fitting as input and expand it into a MAP. The autocorrelations of the input trace and the MAPs are plotted into an image file. Additionally, the trace and the MAPs are fed into a node labeled `Print` which is the counterpart to the plotter for data that should not be visualized by an image but printed in textual format. In this case the first five joint moments $\nu_{i,i}$ are output as a LaTeX table.

The results are shown in Table 1 and Fig. 14 and indicate that MAPs resulting from joint moment fitting provide a good approximation for joint moments but a bad approximation for autocorrelations while the characteristics of MAPs resulting from autocorrelation fitting are the other way around.




```
<profido>
  <ph>
    <states> 2 </states>
    <d0>
      −0.7   0.3
       0.0  −0.5
    </d0>
    <pi> 0.6 0.4 </pi>
  </ph>
  <map>
    <states> 2 </states>
    <d0>
      −0.7   0.3
       0.0  −0.5
    </d0>
    <d1>
      0.24 0.16
      0.3  0.2
    </d1>
  </map>
</profido>
```

Figure 9: $PH(2)$ distribution with XML description

```
<profido>
  <arima>
    <arcount> 3 </arcount>
    <macount> 2 </macount>
    <ar> 0.0402 0.4567 −0.4164 </ar>
    <ma> 0.6602 −0.1109 </ma>
    <d> 0.0 </d>
    <variance> 0.1711 </variance>
    <mean> 0.0 </mean>
  </arima>
</profido>
```

Figure 10: XML description of an $ARMA(3, 2)$ process

7 Conclusions and Future Work

We presented ProFiDo [24], a Java-based toolkit which integrates a variety of tools for fitting input models. ProFiDo currently supports several command line fitting tools as listed at the end of Sect. 3. The results of fitting tools can be compared by plotting density, distribution function and lag-k autocorrelations. Additionally moments and joint moments can be printed in textual form but are not graphically visualized.

ProFiDo is accompanied with an XML based interchange format for process descriptions enabling the interoperability of fitting tools with different interfaces. In order to integrate further tools the user only has to change an XML based tool configuration file and to provide tool specific converter scripts which transform the tools' output to the interchange format and vice versa.

To use the different process types in simulation models or in Markov models, the description in the interchange format has to be translated into tool specific descriptions. We plan to export MAPs, ARMA and ARTA processes to a module for the simulation tool OMNeT++ [10, 21] supporting the integration of complex traffic sources into simulation models. For

numerical analysis techniques only MAPs and PH distributions are usable. As a next step the matrices from these processes will be translated into a sparse matrix format which can be read by the tool Nsolve that integrates a large number of numerical solvers for steady state analysis using either a Kronecker representation or a sparse matrix representation of the generator matrix. MAPs and PH distributions can thus be used as input processes for Markovian systems or as firing time distributions for transitions of SPNs.

In the future we are going to integrate additional tools for fitting distribution functions and some non-Unix tools, like e.g. a tool for fitting ARTA processes (cf. [2]). Furthermore we plan to optimize tool handling, concerning the definition of experiment series, and performance concerning workflow execution, so that the change of a few parameters does not necessarily result in the re-execution of the whole workflow.

Table 1: Joint moments for LBL3 trace and the fitted MAPs (according to the workflow from Fig. 13)

Joint Moment	**lbl3**	**MAP3_JM**	**MAP3_AC**
1, 1	1.30	1.29	1.30
2, 2	17.42	17.65	15.97
3, 3	890.5	890.5	673.9
4, 4	107474	105484	70378
5, 5	21417760	22268852	14002609

References

[1] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase type distributions via the EM algorithm. *Scand. J. Statist.*, 23:419–441, 1996.

[2] Bahar Biller and Barry L. Nelson. Fitting Time-Series Input Processes for Simulation. *Oper. Res.*, 53(3):549–559, 2005.

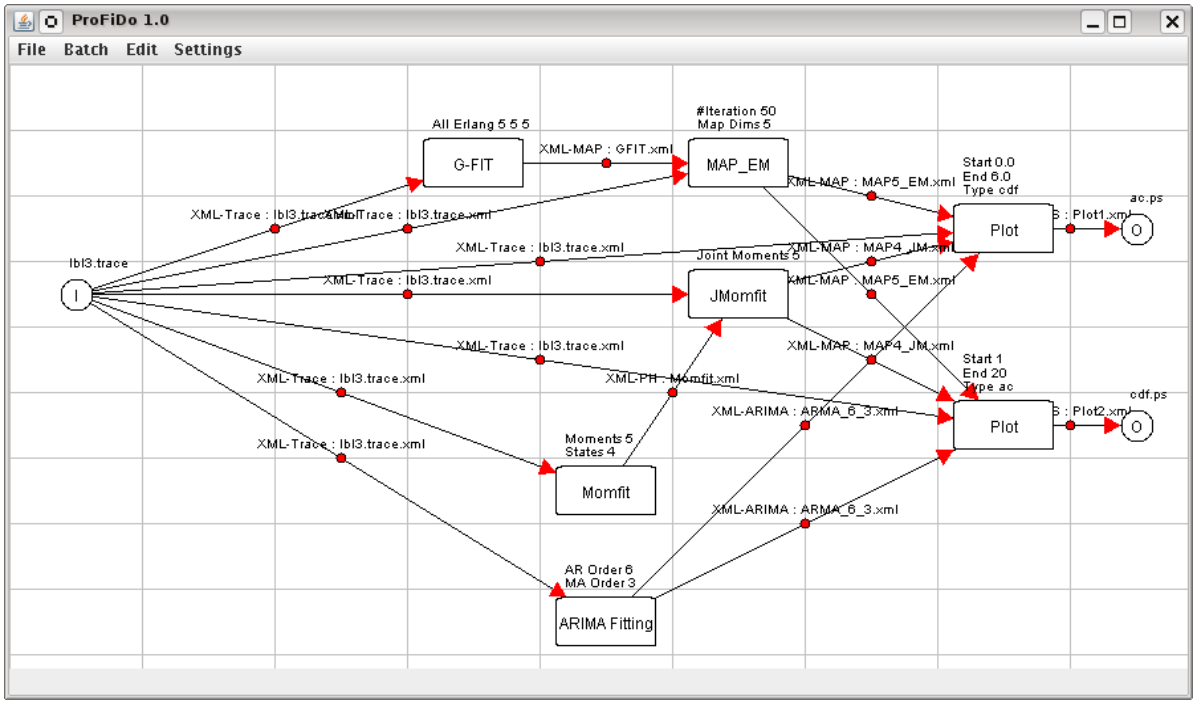


Figure 11: ProFiDo workflow using several fitting tools and comparing their results

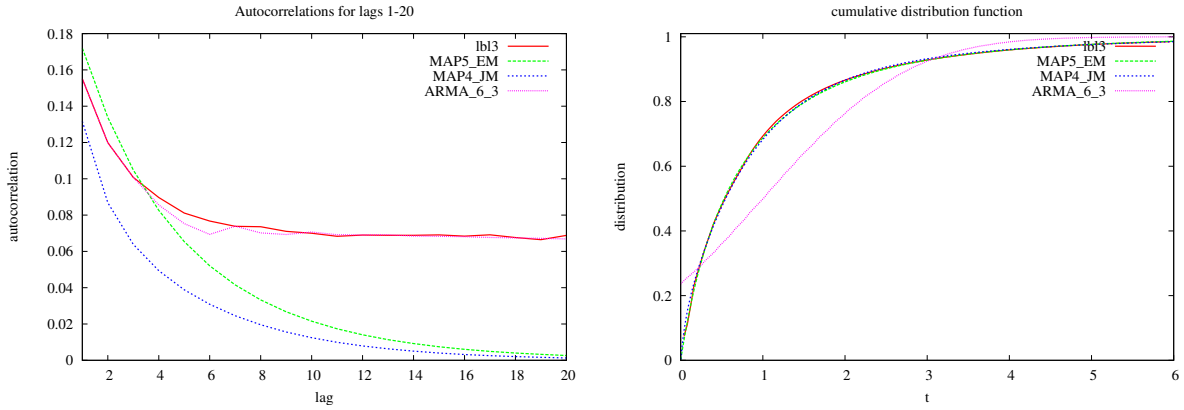


Figure 12: Autocorrelations and cdfs of the LBL3 trace and the fitted processes (according to the workflow from Fig. 11)

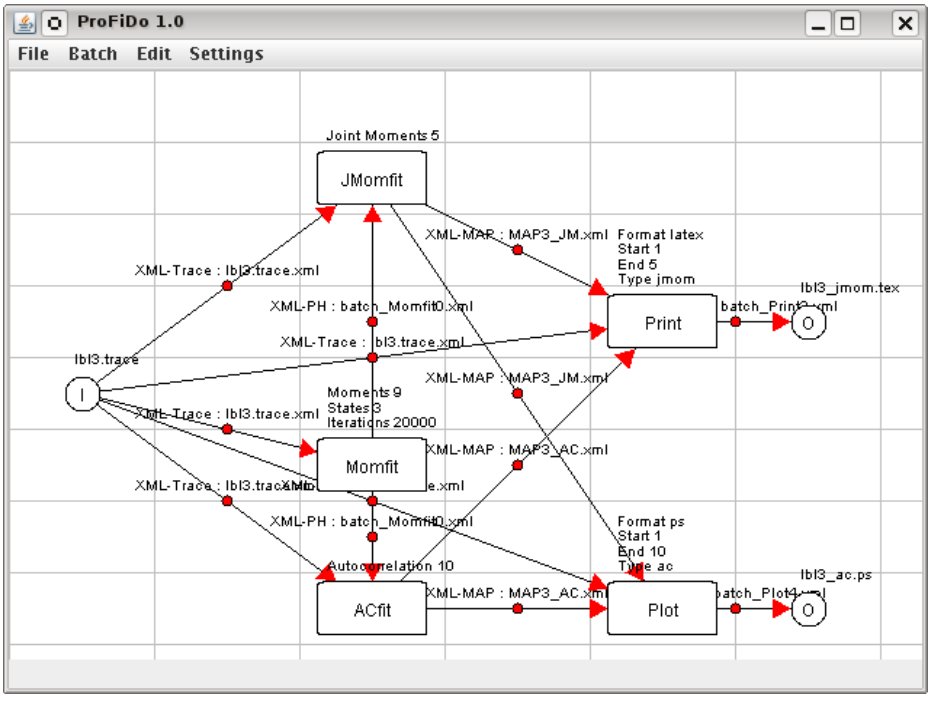


Figure 13: ProFiDo workflow with two MAP fitting algorithms using the same trace and phase-type distribution as input

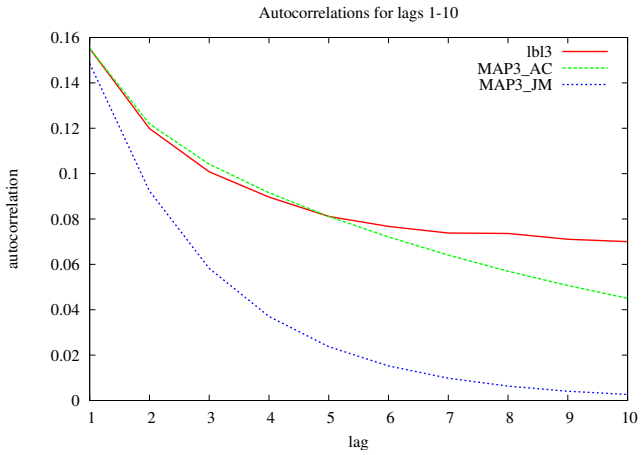


Figure 14: Autocorrelations for LBL3 trace and the fitted MAPs (according to the workflow from Fig. 13)

[3] G.E.P. Box and G.M. Jenkins. *Time Series Analysis - forecasting and control*. Holden-Day, 1970.

[4] Peter Buchholz. An EM-Algorithm for MAP Fitting from Real Traffic Data. In Peter Kemper and William H. Sanders, editors, *Computer Performance Evaluation / TOOLS*, volume 2794 of *Lecture Notes in Computer Science*, pages 218–236. Springer, 2003.

[5] Peter Buchholz and Jan Kriege. A Heuristic Approach for Fitting MAPs to Moments and Joint Moments. In *Proc. of the 6th International Conference on Quantitative Evaluation of SysTems (QEST 2009)*, pages 53–62, Los Alamitos, CA, USA, 2009. IEEE Computer Society.

[6] Marne C. Cario and Barry L. Nelson. Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters*, 19(2):51–58, 1996.

[7] Marne C. Cario and Barry L. Nelson. Numerical Methods for Fitting and Simulating Autoregressive-To-Anything Processes. *INFORMS J. on Computing*, 10(1):72–81, 1998.

[8] Giuliano Casale, Eddy Z. Zhang, and Evgenia Smirni. KPC-toolbox: Simple yet effective trace fitting using markovian arrival processes. In *QEST*, pages 83–92, 2008.

[9] John M. Chambers. *Software for Data Analysis: Programming with R*. Springer, New York, 2008. ISBN 978-0-387-75935-7.

[10] R. Hornig and A. Varga. An Overview of the OMNeT++ Simulation Environment. In *Proc. of 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools)*, 2008.

[11] András Horváth and Miklós Telek. PhFit: A General Phase-Type Fitting Tool. In *TOOLS '02: Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools*, pages 82–91, London, UK, 2002. Springer-Verlag.

[12] G. Horvath, M. Telek, and P. Buchholz. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag-correlation. In *QEST*, pages 124–133. IEEE CS Press, 2005.

[13] W. D. Kelton and A. Law. *Simulation Modeling and Analysis*. McGraw Hill, 2000.

[14] W. D. Kelton, R. P. Sadowski, and D. A Sadowski. *Simulation with Arena*. McGraw-Hill, 4 edition, 2007.

[15] Jan Kriege and Peter Buchholz. An empirical comparison of MAP fitting algorithms. In *MMB 2010, 15th International GI/ITG Conference on Measurement, Modelling and Evaluation of Computing Systems and Dependability and Fault Tolerance*, 2010.

[16] Averill M. Law and Michael G. McComas. ExpertFit distribution-fitting software: how the ExpertFit distribution-fitting software can make your simulation models more valid. In *Winter Simulation Conference*, pages 169–174, 2003.

[17] M.F. Neuts. A versatile Markovian point process. *Journal of Applied Probability*, 16:764–779, 1979.

[18] M.F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. John Hopkins University Press, 1981.

[19] Colm Art O'Cinneide. On non-uniqueness of representations of phase-type distributions. *Stochastic Models*, 5(2):247–259, 1989.

[20] Colm Art O'Cinneide. Phase-type distributions: open problems and a few properties. *Stochastic Models*, 15(4):731–757, 1999.

[21] OMNeT++ Community Site. URL:http://www.omnetpp.org/.

[22] A. Panchenko and P. Buchholz. A Hybrid Algorithm for Parameter Fitting of Markovian Arrival Processes. In *Proc. of 14th Int. Conf. on Analytical and Stochastic Modelling Techniques and Applications*, pages 7–12. SCS Press, 2007.

[23] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions in Networking*, 3:226–244, 1995.

[24] ProFiDo - Processes Fitting Toolkit Dortmund. http://www4.cs.uni-dortmund.de/profido.

[25] M. Telek and G. Horváth. A minimal representation of Markov arrival processes and a moments matching method. *Performance Evaluation*, 64(9-12):1153–1168, Aug. 2007.

[26] The R Project for Statistical Computing. http://www.r-project.org/.

[27] Axel Thümmler, Peter Buchholz, and Miklós Telek. A Novel Approach for Phase-Type Fitting with the EM Algorithm. *IEEE Trans. Dependable Sec. Comput.*, 3(3):245–258, 2006.