ProFiDo - A Toolkit for Fitting Input Models

Falko Bause, Philipp Gerloff and Jan Kriege
Informatik IV, TU Dortmund
D-44221 Dortmund, Germany
{falko.bause,philipp.gerloff,jan.kriege}@tu-dortmund.de

Author prepared version of a paper published in
Proc. of the 15th International GI/ITG Conference on Measurement, Modelling and Evaluation of Computing Systems and Dependability and Fault Tolerance (MMB & DFT 2010), Springer, 2010.
The original publication is available at www.springerlink.com
http://dx.doi.org/10.1007/978-3-642-12104-3_25

ProFiDo

A Toolkit for Fitting Input Models*

Falko Bause, Philipp Gerloff and Jan Kriege

Informatik IV, TU Dortmund

D-44221 Dortmund,

{falko.bause,philipp.gerloff,jan.kriege}@tu-dortmund.de

Abstract

The Processes Fitting Toolkit Dortmund (ProFiDo) provides a graphical user interface supporting the use of a variety of tools for the fitting and modelling of arrival processes. In this paper we present the first version of ProFiDo emphasising the fitting of Markovian Arrival Processes (MAPs).

1 Introduction

Very often arrival processes are specified by independent and identically distributed random variables and several tools and methods are available for fitting specific types of distributions [6, 7, 12]. In a variety of application areas, like computer networks, this type of specification has turned out as being insufficient (cf. [10]), since time-dependencies and correlations between arrival events are not captured. For the specification of time-dependent stationary input processes two directions are well-known: AR (Auto Regressive), ARMA (Auto Regressive Moving Average), ARIMA (Auto Regressive Integrated Moving Average) and ARTA (Auto Regressive To Anything [4]) models became prominent with the work of Box and Jenkins [2] and MAPs (Markovian Arrival Processes) whose intensive investigation started with the work of Neuts [8].

In recent years several of these input models have been incorporated into software for statistical computing, e.g. [11], or are supported by specific tools like ARTAFACTS [5], ARTAFIT [1], MAP EM [3], MAP MOEA [9]. Even though all these tools address the fitting of input models, there handling is different, since (command-line) interfaces and input/output formats differ. This makes the use and comparison of these tools and their corresponding fitting methods cumbersome.

The Processes Fitting Toolkit Dortmund (ProFiDo) described in the following section aims at reducing these deficits by providing a graphical user interface and an XML-based interchange format supporting the consistent use of tools for fitting input models.

*This research is supported by the Deutsche Forschungsgemeinschaft (DFG) within the project “Markovsche Ankunfts- und Bedienprozesse zur Leistungs- und Zuverlässigkeitsanalyse” (Markovian Arrival and Service Processes for Performance and Reliability Analysis).

2 The Processes Fitting Toolkit Dortmund

ProFiDo is a Java based software integrating different command-line based tools into a consistent interface allowing the user to specify a custom workflow of program execution and result propagation. To enable maximum flexibility we choose a graph based approach for displaying the workflow.

Within the graph each node, also called job in the following, represents one execution of a command-line tool. Currently a first set of jobs is supported, including fitting tools like G-FIT and MAP EM and additional utilities for input/output purposes. The latter type of jobs includes a tool Plot, which is able to visualise model characteristics, such as cumulative distribution functions and autocorrelation lags for given MAP descriptions or traces. Jobs can be placed arbitrarily on a grid-based canvas presented by the GUI's main window. Each job has its own properties window in which its parameters are defined. In order to avoid errors and to support the user, instant value checks and optional information texts for each parameter value are provided within the properties window.

The arcs of the graph represent the data flow between different jobs, with the output of a job being used as input for a subsequent job. Due to various origins and authors, most fitting tools use different input/output formats, leading to difficulties in exchanging results directly. To overcome these difficulties, we introduced an XML-based interchange format enabling a consistent data flow between different tools. In order to achieve this interchangeability we provide a set of tool specific converter scripts, which are used by the GUI to convert the input/output of each tool into the corresponding XML-description and vice versa. Since this conversion happens automatically and hidden from the user, no further consideration of different formats is needed when connecting different jobs. In this way a simple specification of workflows is possible. E.g., the output of a fitting tool can be used as input for a trace generator or a trace and the output of several fitting tools can be used as input for a plot. The GUI additionally supports the user in defining parameters for the jobs and manages default result filenames helping to avoid error-prone manual specifications. A sample workflow used to compare the fitting quality of two different tools by incorporating the above mentioned plot generator can be seen in Fig. 1. In addition to the main window's canvas the properties window of the G-FIT job is shown. The properties window allows for a check of parameters and to specify whether a parameter should be visible in the graph.

Since data flow within the graph is represented by arcs and only involves the XML-based interchange format, an additional type of node representing the conversion of external non XML-files (e. g. a trace file) into the corresponding XML-format (and accordingly vice versa) has been introduced. Those "file-nodes" (see nodes I and O in Fig. 1) can be directly connected to jobs and thus allow an easy import of data into the workflow and also enable a wide range of different outputs (Images, PS-files, etc.).

After creation of a workflow the GUI allows for the export of a folder containing all needed binaries and scripts. By analysing the data flow, the GUI is able to determine the execution order of the jobs and creates a bash script which executes the entire workflow. In order to allow a wide use of the workflow, several features for documentation purposes are implemented, including graphical export of the workflow displaying all needed information such as parameter details, result types and more.

The GUI supports the user with other helpful features like an unlimited undo history which can be saved and loaded with the

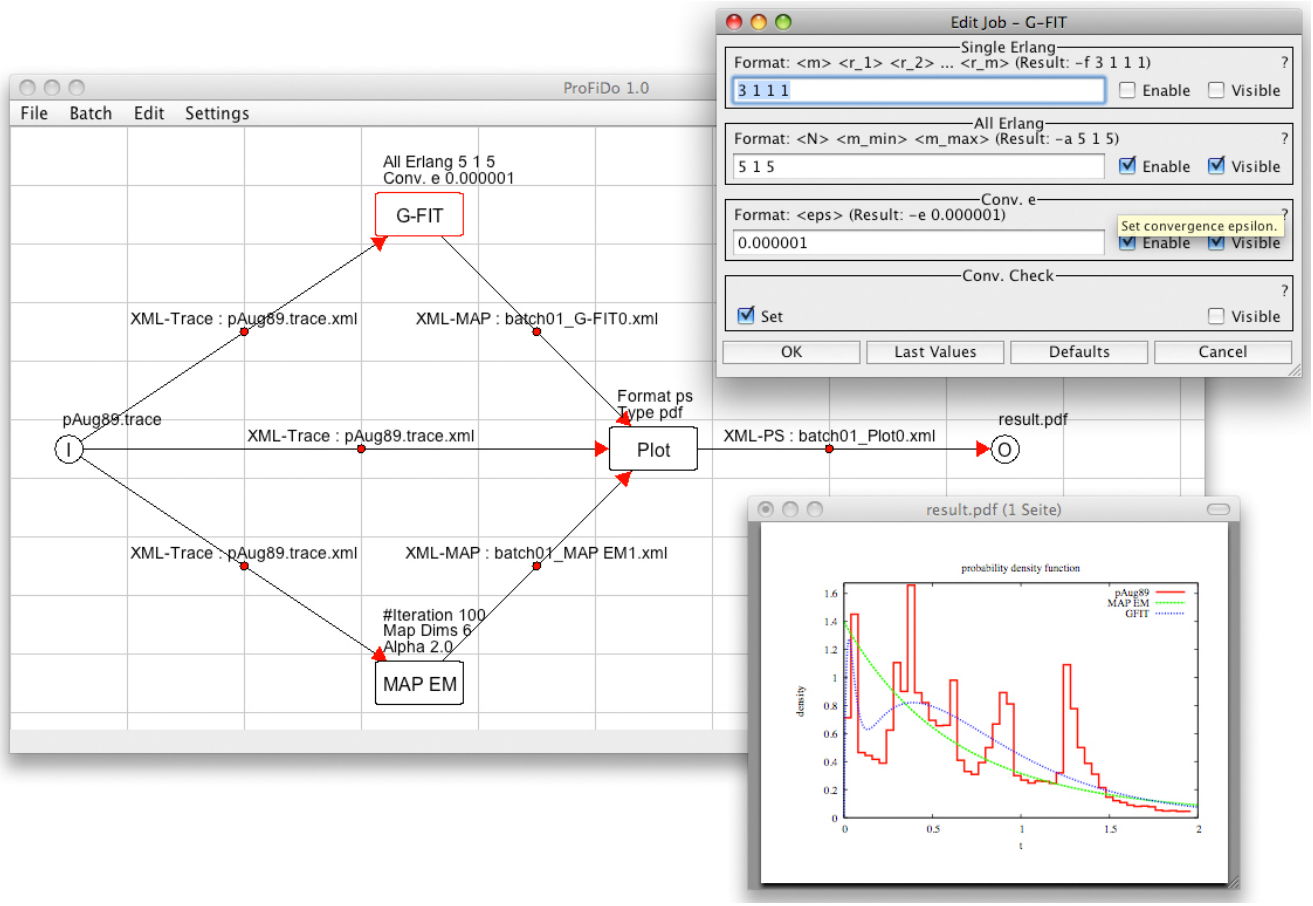

Figure 1: Sample Workflow

graph. ProFiDo can be easily extended to incorporate other command-line based tools, since the core functionality of the GUI is specified in an XML-based configuration file which is parsed on each startup. The different job types are determined by the different tools specified in this configuration file. Therefore the GUI can be extended easily by adding a tool's description and providing corresponding converter scripts, i.e. a converter for transforming data in the XML-based interchange format into the tool's input format and a converter which transforms the tool's output into the XML-based interchange format.

3 Conclusions

We presented ProFiDo, which provides an easy to extend graphical user interface for consistent use of a variety of tools for fitting input models and uses an XML-based interchange format for model descriptions.

This paper presented a first prototype of the GUI. Future work will be directed towards integrating additional, especially "AR-based" fitting tools that have been mentioned in Sec. 1. Since some of the fitting tools might require a long runtime we plan to add support for a parallel execution of tasks on different machines. In addition to the existing feature of exporting plots of traces and fitted models as graphics we intend to add further export functions like exporting properties of the models

into LaTeX tables. Furthermore, we plan to implement support for the specification of experiment series by providing a simple way to specify a series of fitting tasks for which only some of the parameters for the fitting tools are varied while other parameters are kept constant.

References

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

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

[3] 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.

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

[5] 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.

[6] 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.

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

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

[9] 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.

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

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

[12] 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.