Simulation in Production
and Logistics 2015
Markus Rabe & Uwe Clausen (eds.)
Fraunhofer IRB Verlag, Stuttgart 2015

Correlated Random Number Generation for Simulation Experiments

Generierung korrelierter Zufallszahlen für Simulationsexperimente

Falko Bause, Jan Kriege, TU Dortmund, Dortmund (Germany),
falko.bause@tu-dortmund.de, jan.kriege@tu-dortmund.de

Abstract: The design of adequate input models is crucial for the validity of simulation experiments. Nowadays it is common to analyse trace data and to determine appropriate probability distributions for the generation of identically and independently distributed random numbers. Although it is well known that ignoring the correlation structure might give false results, modelling and generation of correlated random numbers is not supported by most simulation tools. In this paper we present an approach for generating correlated random numbers within the well-known simulator Arena. Fitting of input processes is done using the free available toolkit ProFiDo. Resultant process definitions are loaded into an Arena module with specific Visual Basic code for the generation of correlated random numbers. The effect of correlated versus uncorrelated input data is shown for the simulation results of an Arena assembly line model.

1 Introduction

As for any system analysis the GIGO (garbage in, garbage out) principle also holds for simulation experiments and often huge effort is invested to collect and analyse input data from real world. It is common practice to inspect trace data and to derive appropriate probability distributions (Law 2012). Nowadays this is supported by several tools like ExpertFit (Law and McComas 2003) or the Arena Input Analyser (Rathmell and Sturrock 2002) and resultant input models are used for the generation of identically and independently distributed random numbers. Even though it is well known that ignoring the correlation structure of input data might give false results (Livny et al. 1993) and modelling of correlation is considered as one of the most important issues in simulation (Biller and Nelson 2002), modelling and generation of correlated random numbers is not supported by most simulation tools. Since the work of Box and Jenkins (1970) well-known approaches to model correlation are time series models like AR (Auto Regressive), ARMA (Auto Regressive Moving Average) or ARIMA (Auto Regressive Integrated Moving Average) models or multivariate normal or Johnson distributions (Biller and Gunes 2010; Law 2014). Most of these input models are able to capture the correlation well, but have a normal distribution, which is presumably one of the reasons why corresponding support is usually missing in simulation tools. An often applied workaround to capture correlation of input data is performing a trace driven simulation which has several disadvantages. E.g., traces may have certain irregularities or might be too short for simulation runs, so that theoretical input models should normally be preferred (Law 2014).

In recent years and decades research concentrated on the development of more general process definitions and appropriate fitting methods as e.g. for Markovian Arrival Processes (MAPs) (Neuts 1979) and Auto-Regressive-to-Anything (ARTA) processes (Cario and Nelson 1996). In most cases only prototype tools giving a proof of concept were developed, many with their own formats and user interfaces which makes comparisons and their use in practice difficult. Even if an appropriate process type and corresponding parameterisation has been found their use in simulation is not straightforward and often extra programming is required.

A multi-year project (DFG-project 2014) supported by the Deutsche Forschungsgemeinschaft (DFG) addressed these problems and resulted in the toolkit ProFiDo (Processes Fitting Toolkit Dortmund) (Bause et al. 2010; ProFiDo 2014), a free software, released under the terms of the GNU General Public License. ProFiDo provides a graphical user interface supporting the use of a variety of tools for the fitting and modelling of processes. Included are tools for fitting common distributions, phase-type distributions, MAPs, AR/ARIMA and ARTA processes. ProFiDo comes along with a well-defined interchange format such that its output can be easily used as input for simulators (Kriege and Buchholz 2011). In this paper we show how process descriptions resulting from ProFiDo's fitting tools can be used for the generation of correlated random numbers in the simulator Arena.

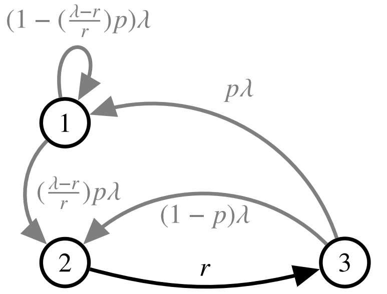
The structure of the paper is as follows. In Section 2 we present theoretical foundations and basics of the toolkit ProFiDo. Section 3 describes the integration of ProFiDo's output into Arena and the generation of correlated random numbers. Section 4 shows the effects of ignoring correlations for an assembly line model from Arena's standard example library. The paper ends with the conclusions in Section 5.

2 Modelling and Fitting of Input Data

2.1 Theoretical Background

This section concentrates on Markovian Arrival Processes (MAPs), since this class of processes is used in the example of Section 4. For details on other types of processes we refer to the mentioned literature.

A MAP is described by two matrices D_0 and D_1 such that $D_0 + D_1$ is the generator of an irreducible Markov process (see Fig. 1). D_0 is non-singular and contains non-negative values outside the diagonal and D_1 contains only non-negative values. D_0 describes internal transitions of the Markov process whereas transitions of D_1 specify arrivals. The arrival instances generated by a MAP can be described by a discrete time Markov chain (DTMC) with transition probability matrix $P = (-D_0)^{-1}D_1$. For a valid MAP P is irreducible and the unique steady state vector π given by $\pi P = \pi$, $\pi e = 1$ exists with e being a column vector of ones of appropriate dimension. In steady state the probability density function (PDF) f and the cumu-


$(1-(\frac{\lambda-r}{r})p)\lambda$
$p\lambda$
1
$(\frac{\lambda-r}{r})p\lambda$
$(1-p)\lambda$
2
r
3

$$\boldsymbol{D}_0 = \begin{bmatrix} -\lambda & 0 & 0 \\ 0 & -r & r \\ 0 & 0 & -\lambda \end{bmatrix}$$

$$\boldsymbol{D}_1 = \begin{bmatrix} (1-(\frac{\lambda-r}{r})p)\lambda & (\frac{\lambda-r}{r})p\lambda & 0 \\ 0 & 0 & 0 \\ p\lambda & (1-p)\lambda & 0 \end{bmatrix}$$

$$\boldsymbol{\pi} = \left(\frac{r}{\lambda}, \frac{\lambda-r}{\lambda}, 0\right)$$

***Figure 1:** Example of a MAP ($p \in]0,1[$, $\lambda > r > p\lambda/(1+p)$)*

lative distribution function (CDF) F of the random variable X specifying the interarrival times of a MAP are given by

$$f(x) = \pi \exp(\mathrm{D}_\mathbf{0}\, x)(-\mathrm{D}_\mathbf{0})e \qquad (1)$$

$$F(x) = 1 - \pi \exp(\mathrm{D}_\mathbf{0}\, x)e \qquad (2)$$

resulting in the moments $\mu_i := E[X^i] = i!\pi(-D_0)^{-i}e$. Consecutive interarrival times x_0, x_1, … , x_k ($k \in N$) are usually not independent and their joint distribution is given by

$$\mathrm{f}(\mathrm{x}_0, \mathrm{x}_1, ..., \mathrm{x}_k) = \pi \exp(\mathrm{D}_0 \mathrm{x}_0)\,\mathrm{D}_1 \exp(\mathrm{D}_0 \mathrm{x}_1)\,\mathrm{D}_1 ... \exp(\mathrm{D}_0 \mathrm{x}_k)\,\mathrm{D}_1\, e \qquad (3)$$

giving the basis for the calculation of several characteristical values to express dependency of interarrival times. E.g., the lag-k autocorrelations ρ_k are given by (Buchholz et al. 2014):

$$\rho_k = \frac{\mu_1^{-2}\pi(-\mathrm{D}_0)^{-1}\,\mathrm{P}^k(-\mathrm{D}_0)^{-1}e - 1}{2\mu_1^{-2}\pi(-\mathrm{D}_0)^{-1}(-\mathrm{D}_0)^{-1}e - 1} \qquad (4)$$

Figure 1 presents an example of a MAP with three parameters: λ and p can be used to define different "degrees" of correlation and r specifies the rate of interarrivals. Interarrival times of this MAP are exponentially distributed with mean 1/r for all parameter values ($p \in]0, 1[$, $\lambda > r > p\lambda/(1 + p)$) and the lag-1 autocorrelation is given by $\rho_1 = (p + 1)\, r/\lambda - p - (r/\lambda)^2$ and is non-zero if $p \neq r/\lambda$. For $p = r/\lambda$ the MAP coincides with an exponential distribution. Figure 2 shows the values of the autocorrelation function for the first lags for parameters r = 5/3, p = 0.05 and λ = 3.

The process of determining appropriate matrices D_0, D_1 for given statistical characteristics, as e.g. moments and autocorrelations derived from traces, is referred to as MAP fitting. In the last decades several fitting techniques have been developed with specific pros and cons (Buchholz et al. 2014), so that comparison of different fitting results is advisable. Unfortunately most fitting techniques come along, if at all, with their own prototype tools having their own formats and interfaces. This does not only makes comparisons difficult, but also impedes the integration of fitting results

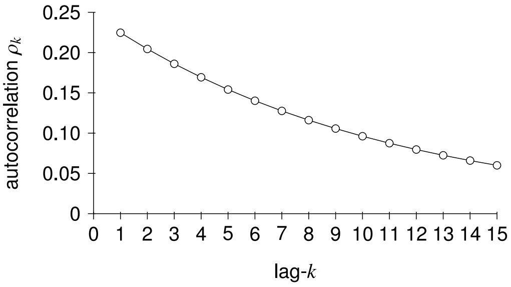
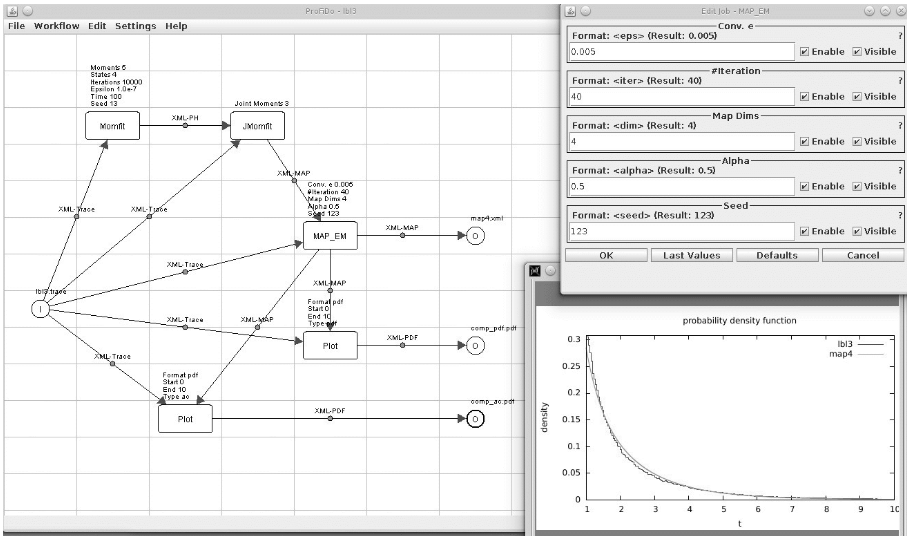


***Figure 2:** Autocorrelation function of example MAP (r=5/3, p=0.05, λ=3)*



Moments 5
States 4
Iterations 10000
Epsilon 1.0e-7
Time 100
Seed 13

Joint Moments 3

Conv. e 0.005
#Iteration 40
Map Dims 4
Alpha 0.5
Seed 123

Format pdf
Start 0
End 10
Type pdf

Format pdf
Start 0
End 10
Type ac

Conv. e
Format: <eps> (Result: 0.005) ?
0.005 ☑ Enable ☑ Visible

#Iteration
Format: <iter> (Result: 40) ?
40 ☑ Enable ☑ Visible

Map Dims
Format: <dim> (Result: 4) ?
4 ☑ Enable ☑ Visible

Alpha
Format: <alpha> (Result: 0.5) ?
0.5 ☑ Enable ☑ Visible

Seed
Format: <seed> (Result: 123) ?
123 ☑ Enable ☑ Visible

***Figure 3:** Example workflow in ProFiDo*

into simulation tools. The freely available toolkit ProFiDo, described in the next subsection, intends to reduce these deficits.

2.2 ProFiDo - A Toolkit for Fitting Processes

ProFiDo (Bause et al. 2010; ProFiDo 2014) is a Java-based toolkit which integrates several tools for fitting the parameters of distributions or processes to be used as traffic sources or service processes in stochastic modelling. The basic idea of the toolkit is to provide an XML based interchange format for the interoperability of different fitting tools and a graphical interface to define workflows that specify the different steps of data preprocessing, parameter fitting and analysing the resulting processes. Figure 3 shows an example of such a workflow. The workflow starts with an input node named *I* which is associated with a trace file (*lbl3.trace*). Its content is used for three fitting tools named *Momfit*, *JMomfit* and *MAP_EM*. *Momfit* essentially fits the distribution of a MAP, i.e. matrix D_0. *JMomfit* uses the joint moments of a trace and the distribution of a MAP in order to determine a suitable matrix D_1. *MAP_EM* uses an EM algorithm to determine a MAP from a trace and usually profits from an initial guess which is determined by the tool *JMomfit* in this workflow. ProFiDo offers additional tools which can also fit a complete MAP description in one step, but are not used in the example. In figure 3 the result of the tool *MAP_EM* is used for a special node *Plot* which allows the comparison of several statistical characteristics of fitted processes and original trace. Parameters of fitting tools can also be set within ProFiDo's graphical user interface to influence the quality of fitting results.

All fitting tools in ProFiDo are encapsulated by translator modules which transform output of tools into ProFiDo's XML based interchange format and data in interchange format into the tool specific input format. The interchange format allows for the description of a variety of probability distributions and stochastic processes and is the basis for the integration of fitting results into simulation tools. Kriege and Buchholz (2011) describe the use of such input models in the open source simulation framework OMNeT++. In the next section we present their integration into the simulator Arena.

3 Integrating Correlated Random Number Generation into Arena

The well-known simulator Arena (Kelton et al. 2015; Rathmell and Sturrock 2002) is widely used for modelling manufacturing and logistics systems. Models in Arena consist of modules that encapsulate a specific behaviour or perform specific tasks and are organised in templates. As mentioned, we have developed a template for Arena that allows for an easy integration of stochastic processes fitted with ProFiDo, and thus enables Arena to generate correlated random numbers from those processes.

In the following we will present one of the modules from the template in detail that can be used to sample random numbers from a MAP. The template also contains a module to generate random numbers from ARTA processes (Cario and Nelson 1996), but we will focus on the module for MAPs in this work. We first outline the module internals and some implementation details to give an overview how the module actually works. For a user who just wants to use the module in his model these details are of course hidden, but there are some parameters that can be used to influence the module's behaviour which will be explained at the end of this section.

Figure 4 shows the logic view of the MAPArrivals module in Arena. The module is composed of seven standard modules. The behaviour can be split into three parts. The first two modules in figure 4 are only executed once at the beginning of the simulation and are used to load a file with ProFiDo's XML description and initialise variables accordingly. Most of this work is done in the first *VBA* block in figure 4. *VBA* blocks can be used in Arena to integrate Visual Basic for Applications code into the model. The code is executed every time the process flow reaches the corresponding module. After everything is set up the main loop is entered that consists of three modules. *VBA* module 2 is used to generate the next interevent time from the MAP. The module contains code that simulates the underlying Markov chain of the MAP. The algorithm in figure 5 shows a pseudocode version of what is computed in this module. The *Delay* module is actually the only module that consumes model time in figure 4. Here the process flow is delayed until the next arrival is due. After that the process is split into two branches. The first branch will

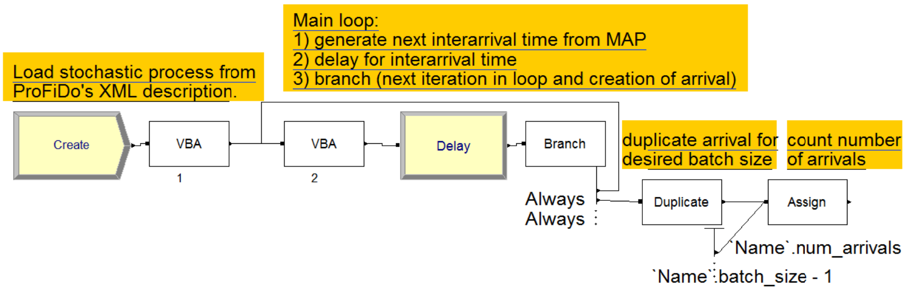

Main loop:
1) generate next interarrival time from MAP
2) delay for interarrival time
3) branch (next iteration in loop and creation of arrival)
Load stochastic process from
ProFiDo's XML description.
Create
VBA
1
VBA
2
Delay
Branch
duplicate arrival for
desired batch size
count number
of arrivals
Always
Always
Duplicate
Assign
`Name`.num_arrivals
`Name`.batch_size - 1

Figure 4: *Logic of the MAPArrivals Module*

```
Input: $\lambda_1, \cdots, \lambda_n, \boldsymbol{P}_0, \boldsymbol{p}_0, \boldsymbol{P}_1$, current state $i$;
Output: Time to the next event $x$ and new state $i$;
 1: initialise $x = 0$ and $found = false$ ;
 2: repeat
 3:   $x = x + exponential(\lambda_i)$;
 4:   $u = uniform(0, 1)$;
 5:   if $u \le \boldsymbol{p}_0(i)$ then
 6:     $sum = 0$;
 7:     $j = 1$;
 8:     repeat
 9:       $sum = sum + \boldsymbol{P}_0(i, j)$;
10:       $j = j + 1$ ;
11:     until $sum > u$ ;
12:   else
13:     $sum = \boldsymbol{p}_0(i)$;
14:     $j = 1$;
15:     repeat
16:       $sum = sum + \boldsymbol{P}_1(i, j)$;
17:       $j = j + 1$ ;
18:     until $sum > u$ ;
19:     $found = true$ ;
20:   end if
21:   $i = j$ ;
22: until $found == true$ ;
23: return $x$ and $i$ ;
```

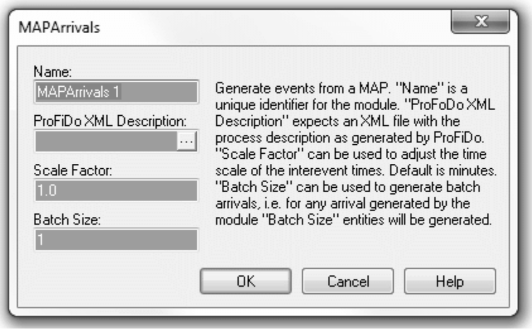
Figure 5: *Algorithm: Generate_Random_Number_from_MAP*

start the next iteration of the loop, while the second branch leaves the MAPArrivals module after a postprocessing step. The postprocessing step consists of two modules. In the first module the arrival is duplicated in case of batch arrivals and finally the number of generated arrivals is counted.

As already mentioned, figure 5 shows code to generate random numbers from a MAP (Buchholz et al. 2014) that is implemented in one of the VBA modules from figure 4. To simplify the notation of the algorithm we define

$$\lambda_i = -D_0(i,i) \qquad P_0(i,j) = \begin{cases} D_0(i,j)/\lambda_i & \text{for } i \neq j \\ 0 & \text{for } i = j \end{cases} \tag{5}$$
$$p_0 = P_0\, e \qquad P_1(i,j) = D_1(i,j)/\lambda_i$$

The algorithm draws random numbers from an exponential and a uniform distribution. The exponential random number is used to determine the time spent in the current state of the MAP (line 3). The uniform random number is used to determine the next state. A transition to another state can either occur according to D_0 (lines 5 to 11) or according to D_1 (lines 12 to 20). Transitions according to D_0 are internal transitions, while transitions according to D_1 trigger an event. Consequently, the



MAPArrivals

Name: MAPArrivals 1

ProFiDo XML Description:

Scale Factor: 1.0

Batch Size: 1

Generate events from a MAP. "Name" is a unique identifier for the module. "ProFoDo XML Description" expects an XML file with the process description as generated by ProFiDo. "Scale Factor" can be used to adjust the time scale of the interevent times. Default is minutes. "Batch Size" can be used to generate batch arrivals, i.e. for any arrival generated by the module "Batch Size" entities will be generated.

***Figure 6:** Attribute window of the MAPArrivals module*

main loop of the algorithm is repeated until a transition according to D_1 has taken place. The interevent time is then returned as the sum of all times spent in the different states of the MAP until that transition occurred.

The details of the implementation presented so far are usually hidden from the user. If the module is used within a larger model it can be completely parameterised using the attribute window shown in figure 6. Apart from a name and a file with a MAP description in ProFiDo's XML format, one can specify a scale factor and the batch size. The MAPArrivals module assumes that the model time is given in minutes. If another time scale should be used this can be accomplished by setting the scale factor. The batch size determines how many arrivals should be generated at every event generated by the MAP.

4 Example

To demonstrate the effect of correlation on simulation results we use an assembly line model shown in figure 7 that is based on one of Arena's example models. The assembly line consists of 4 multi-server stations each associated with a buffer. The first buffer is infinite, while the remaining three buffers have limited capacity. Arriving parts are first processed at station 1, then passed to station 2 and so on. The details on the number of machines at each station, buffer sizes and deterministic processing times are summarized in table 1. Due to the limited buffer sizes of stations 2, 3 and 4, processed parts may not be able to leave a machine until space in the next buffer is available. This leads to the situation that a machine might be idle although parts are waiting to be served in the corresponding queue. This blocking phenomenon is known as transfer blocking (Tempelmeier and Kuhn 1993).

The model was simulated with arrivals generated by two different MAPs that are constructed according to figure 1 with mean arrival rates of r = 1.3 and r = 1.6 ($p=0.05$, $\lambda=3$). For comparison we simulated the model with arrivals according to exponential distributions with the same rates. Recall, that the interarrival time distribution of the MAPs is exponential. Thus, differences in the results are only caused by the autocorrelation of the MAPs. Table 2 shows the mean queue length for the buffers and the mean number of busy servers for the stations. If we compare the results of the MAP and the distribution with rate 1.3 we see that the main differences can be observed for the mean number of busy servers at station 1 and the mean queue length for the first two buffers. While the first buffer is almost empty for the

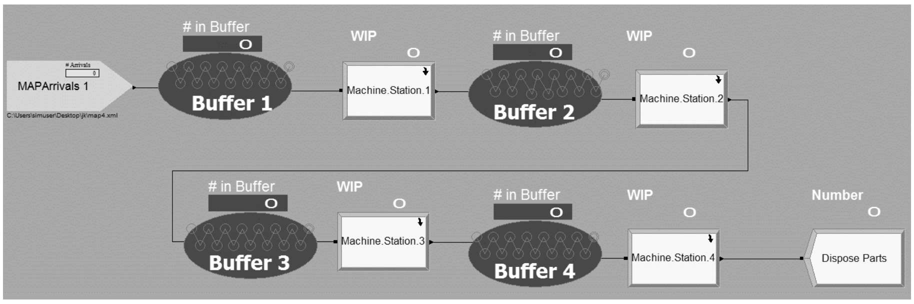


Figure 7: *Assembly line example model*

exponential distribution we have a mean queue length of 1.55 for the MAP. Of special interest are the values denoted as *Enter Buffer 2* in the table. These values indicate the mean number of parts that have been processed at station 1 but cannot proceed to buffer 2 because it is full, i.e. that are blocking a server at station 1. Note, that the values for the mean number of busy servers at station 1 include these values, implying that with correlated arrivals the servers are not really busy but are occupied due to transfer blocking for most of the time. With an increased arrival rate of 1.6 these effects become more significant. The mean number of parts that are blocked from entering buffer 2 increases to 1.67, implying that more than 50% of the server capacity at station 1 is lost due to transfer blocking.

Table 1: *Servers and buffers for assembly line model*

Station	Buffer size	Number of servers	Service time
Station 1	∞	3	3
Station 2	5	9	5
Station 3	6	15	8
Station 4	1	11	6

For the previous example we have chosen a very simple distribution to limit the effects on the difference between correlated and independent arrivals.

For our second example we used a trace with real data from Paxson and Floyd (1995) (LBL-TCP-3). The trace contains packet arrivals in a network and therefore, the data is of course not fully realistic for arrivals in an assembly line model. However, the data contains a complicated empirical distribution and correlation and serves well in demonstrating what data can be fitted if we do not limit ourselves to the standard distributions that are provided with simulation software. We adjusted the trace by a factor of approx. 150 in order to match the time scale of the model and used ProFiDo to generate a MAP with 4 states according to the workflow shown in figure 3. The best result obtained by the Arena Input Analyzer was 24 · beta(0.475, 17.6), i.e. a Beta distribution scaled to the interval [0, 24]. Again, we simulated the assembly line model with the MAP and the distribution. Some results are summarised in Table 3.

Table 2: *Simulation results (mean population in queues and servers with 95% confidence intervals)*

	MAP (rate 1.3)	Expo. (rate 1.3)	MAP (rate 1.6)	Expo. (rate 1.6)
Buffer 1	1.55 (±0.21)	0.005 (±0.002)	16.38 (±4.57)	0.42 (±0.06)
Buffer 2	1.52 (±0.07)	0.37 (±0.02)	3.30 (±0.17)	1.80 (±0.06)
Buffer 3	0.14 (±0.01)	0.03 (±0.002)	0.27 (±0.02)	0.13 (±0.004)
Buffer 4	0.11 (±0.01)	0.04 (±0.002)	0.21 (±0.01)	0.15 (±0.004)
Enter Buffer 2	0.59 (±0.04)	0.02 (±0.004)	1.67 (±0.12)	0.42 (±0.03)
Station 1	0.66 (±0.04)	0.09 (±0.004)	1.75 (±0.12)	0.50 (±0.03)
Station 2	6.50 (±0.07)	6.48 (±0.03)	8.02 (±0.11)	7.99 (±0.03)
Station 3	10.42 (±0.12)	10.38 (±0.04)	12.85 (±0.17)	12.80 (±0.05)
Station 4	7.80 (±0.09)	7.78 (±0.03)	9.62 (±0.13)	9.68 (±0.04)

Table 3: *Simulation results for fitted MAP and distribution (mean population in queues and servers with 95% confidence intervals)*

	MAP(4)	Beta distribution
Buffer 1	24.63 (±5.40)	2.98 (±0.55)
Station 1	1.96 (±0.10)	1.12 (±0.06)
Enter Buffer 2	1.88 (±0.10)	1.04 (±0.06)
Buffer 2	3.58 (±0.14)	2.73 (±0.09)

From the experiments we have seen, that autocorrelation might have a serious impact on the simulation results. Apparently, transfer blocking, which is common in assembly line models, even increases this effect. With correlated arrivals, sequences of short interarrival times tend to fall together, resulting in some periods with higher load. Due to the transfer blocking the servers cannot use their full capacity and therefore need considerable time to finish the load that has piled up, resulting in the effects shown in our examples.

5 Conclusion

The example of Section 4 demonstrates that correlations of arrival events might have a significant impact on simulation results. Modelling and fitting of a variety of corresponding input models came up in recent years, but their use has not found its way into most simulation tools. At present simulation experts often lack appropriate support which is partly caused by the diversity of input processes and associated tools and formats. The toolkit ProFiDo intends to reduce such deficits and offers fitting tools for several types of input models being unified under a precisely defined interface providing the basis for their use in simulation.

As mentioned, results of ProFiDo’s fitting tools can be directly used in the open source simulation framework OMNeT++ (Kriege and Buchholz 2011). In this paper we showed their direct use in the common simulator Arena. The Arena Arrival Module described in Section 3 is freely available (ProFiDo 2014) and might serve as a basis for the extension of other simulators.

References

Bause, F.; Buchholz, P.; Kriege, J.: ProFiDo - the processes fitting toolkit Dortmund. In: Proceedings of the 7th International Conference on Quantitative Evaluation of SysTems (QEST), Williamsburg (VA) 2010, pp. 87–96.

Biller, B.; Gunes, C.: Introduction to simulation input modeling. In: Proceedings of the Winter Simulation Conference, Baltimore (MD) 2010, pp. 49–58.

Biller, B.; Nelson, B.: Answers to the top ten input modeling questions. In: Proceedings of the Winter Simulation Conference, San Diego (CA) 2002, pp. 35-40.

Box, G.; Jenkins, G.: Time series analysis - forecasting and control. San Francisco: Holden-Day 1970.

Buchholz, P.; Kriege, J.; Felko, I.: Input modeling with phase-type distributions and Markov models - theory and applications. Cham et al.: Springer 2014.

Cario, M.; Nelson, B.: Autoregressive to anything: time-series input processes for simulation. Operations Research Letters 19 (1996) 2, pp. 51–58.

DFG-project: Markovian and rational arrival and service processes for performance and reliability analysis. Website http://ls4-www.cs.tu-dortmund.de/cms/de/forschung/projekte/dfg-maps/index.html 2014.

Kelton, W.; Sadowski, R.; Zupick, N.: Simulation with Arena, 6th edition. Bosten et al.: McGraw-Hill Higher Education 2015.

Kriege, J.; Buchholz, P.: Simulating stochastic processes with OMNeT++. In: Proceedings of the 4th International Workshop on OMNeT++, Barcelona 2011, pp. 367-374.

Law, A.: A tutorial on how to select simulation input probability distributions. In: Proceedings of the Winter Simulation Conference, Berlin 2012, pp. 1-15.

Law, A.: Simulation modeling and analysis. 5th edition. New York: McGraw Hill 2014.

Law, A.; McComas, M.: How the ExpertFit distribution-fitting software can make your simulation models more valid. In: Proceedings of the Winter Simulation Conference, San Diego (CA) 2002, pp. 199-204.

Livny, M.; Melamed, B.; Tsiolis, A.: The impact of autocorrelation on queuing systems. Management Science 39 (1993) 3, pp. 322-339.

Neuts, M.: A versatile Markovian point process. Journal of Applied Probability 16 (1979) 4, pp. 764-779.

Paxson, V.; Floyd, S.: Wide-area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking 3 (1995) 3, pp. 226-244.

ProFiDo: Website http://ls4-www.cs.tu-dortmund.de/profido 2014.

Rathmell, J.; Sturrock, D.: Arena: The Arena product family: enterprise modeling solutions. In: Proceedings of the Winter Simulation Conference, San Diego (CA) 2002, pp. 165-172.

Tempelmeier, H.; Kuhn, H.: Flexible manufacturing systems: decision support for design and operation. New York: Wiley 1993.