First published: ESM'2014, 28th European Simulation and Modelling Conference, October 22-24, 2014, FEUP - University of Porto, Portugal

ANALYSIS BY STATE: AN ALTERNATIVE VIEW ON DISCRETE-EVENT TIME SERIES

Arne Koors and Bernd Page
Department of Informatics
University of Hamburg
Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
E-mail: {koors, page}@informatik.uni-hamburg.de

KEYWORDS
Simulation Output Analysis, Visualization, Discrete Event Simulation, Simulation Dynamics, Time Series Analysis.

ABSTRACT

In discrete event simulation experiments, state variables' values are recorded and further processed to explore the dynamics of the modelled system. This paper introduces a family of so-called Analysis by State methods for exploration of relationships between two discrete event simulation output time series. Here, state intervals of a primary time series are visually augmented with information gained by processing corresponding time intervals of a secondary time series, e.g. by displaying interval-wise correlation, distribution, sample aggregates or sample parameters in form of background histograms or heat maps. The desired benefit is to further support and comfortably enhance identification of characteristics and relationships in pairs of discrete-event time series.

INTRODUCTION

Discrete event simulation is a methodology that models dynamic systems and runs experiments on these models, in order to gain insights that can be re-transferred to the investigated original system (Page and Kreutzer 2005).
During simulation runs, time series of state variables' observational values are recorded for analysis after completion of experiments, to explore the dynamics of the modelled system. It is specific for discrete event simulation that these time series are *not equidistant*, because arbitrary time spans between discrete events may be observed, e.g. stochastic inter-arrival times.
The most common forms of analysing single discrete event simulation output time series are characterisation by descriptive statistics, testing for stationarity, identification of initial transient phases, and determining simulation run length resp. number of replications (Fishman 2001; Page and Kreutzer 2005; Banks 2010; Hoad et al. 2011; Law 2014). Fewer techniques can be found concerning analysis of time series *pairs* in discrete event simulation. Obviously, standard approaches for equidistant time series could be applied, like scatter plots, computation of correlation coefficients or simply plotting pairs of time series and visually inspecting them for conspicuous relationships (Law 2014). However, the non-equidistant nature of discrete-event observation series impedes resorting to methods developed for equidistant time series: to be exact, these methods had to be adapted for *non-synchronized time-weighted observation series*, before generally applicable to discrete event simulation output. Though feasible in principle, canonically extended time-weighted scatter plots resp. time-weighted correlation coefficients are rarely implemented in discrete-event simulation packages and are therefore not in wide-spread practical use.
This paper introduces a family of so-called *Analysis by State* methods, to support exploration of relationships between two discrete event simulation output time series: *State intervals* of a primary time series are visually augmented with information gained from processing corresponding *time intervals* of a secondary time series in different ways.
The Analysis by state approach is inspired by Volume at Price Charts from *technical analysis* (TA; e.g. Kamich 2003; Ochoa 2010; Coulling 2013), a subfield of financial engineering (fig. 1). Here, the price line graph in the upper diagram part is enhanced by horizontal bars representing the cumulated length of vertical volume bars from the lower part of the diagram. The purpose of Volume at Price Charts is to highlight past potentials of buyers' support and sellers' resistance at different price levels, in view of the TA notion that past observations may indicate future financial market prices. The Volume at Price approach has been generalised and comprehensively extended in this work, towards the Analysis by State concept described hereafter.

Figure 1: Volume at Price Chart from financial engineering

For further finance-inspired discrete event simulation analysis and visualisation techniques see (Koors and Page 2012; Koors 2013; Koors and Page 2013).
The remainder of this paper is structured as follows: The following section describes the basic procedure for Analysis by State. Next, an overview of the method family as a whole is given, and an illustrating example model is introduced. Afterwards, six subsections explain the method family members in more detail. The final section gives a summary, an outlook and concludes the paper.

THE ANALYSIS BY STATE METHOD FAMILY

Basic Procedure

The concept and basic terms of the Analysis by State approach are illustrated in fig. 2: During a simulation run, event time instants and corresponding state values are recorded for a pair of chosen *primary* and *secondary* observation variables. On simulation report generation, the two resulting observation series (in the following *time series*) are displayed in the upper resp. lower part of a common diagram (fig. 2 and 3). The range of observed states of the primary observation variable is divided into n adjacent, non-overlapping *primary state intervals* of equal size. Primary state intervals may be highlighted by different background colours.

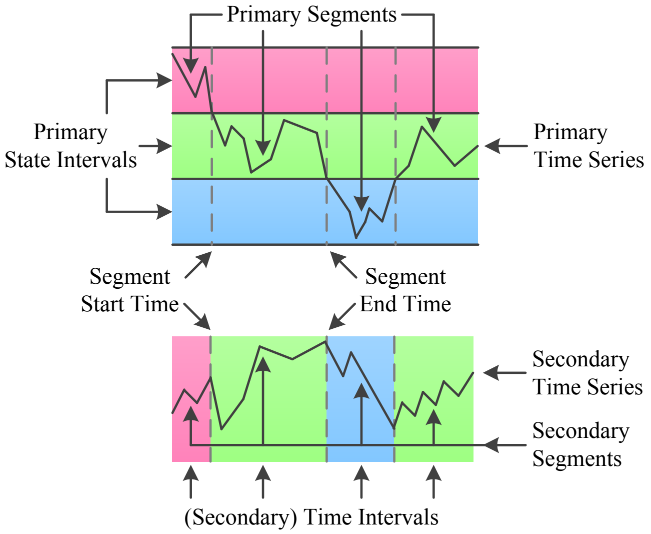

Figure 2: Concept and basic terms of Analysis by State

Next, the primary time series is divided into *primary segments*: Whenever observed states leave one primary state interval, a new primary segment is established. All following observations in the new state interval are incorporated in the newly established segment, until the interval is left again. On leaving the current state interval, the current segment is closed, and a new segment for the next state interval is opened. A primary segment thus contains subsequent observations that are all in the same primary state interval range.
A segment explicitly has a *start time* instant (event time instant of its first observation) and an *end time* instant (event time instant of the following's segment first observation). Thus, a segment corresponds to an uninterrupted *time interval*. All segments are adjacent and disjoint in time, like the original primary state intervals, and concatenation of all primary segments yields the original primary time series. One primary segment always belongs to exactly one primary state interval, but one primary state interval may refer to an arbitrary number of primary segments.
In this way, the original primary time series is mapped onto a (possibly large) number of time intervals, with each time interval corresponding to a primary state interval.

The primary segments' start and end time instants are now applied to the secondary time series, dividing it into *secondary segments*. In this manner, primary and secondary segments are associated with each other on a 1:1 time-instant defined basis. Because each associated primary segment corresponds to one primary state interval, all secondary segments now transitively map onto primary state intervals as well.
To illustrate this mapping, secondary time intervals may be coloured with the same background colours as their corresponding primary state intervals (see fig. 3): Whenever the primary observation variable has high values (here: pink primary state intervals), corresponding segments of the secondary time series receive a pink background, too. Conversely, the secondary time series has blue background, whenever state values of the primary time series are low.

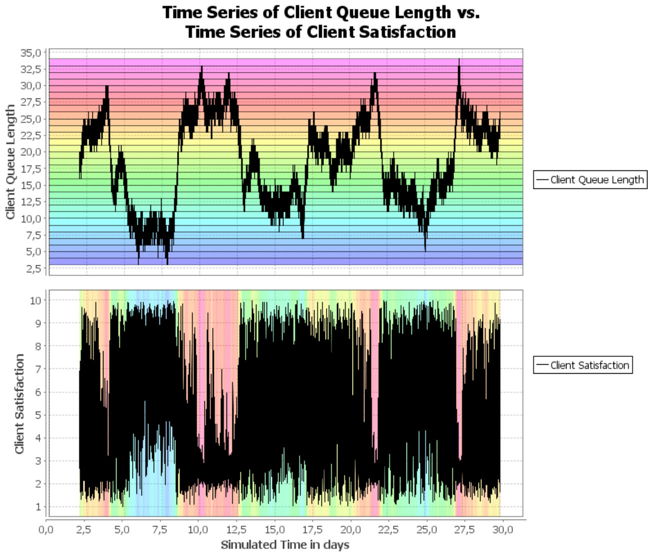

Figure 3: Basic Analysis by State diagram

In this process, secondary observations before the start of the primary time series are dropped, because they do not correspond to any defined primary state interval. Secondary observations behind the end of the primary time series are mapped to the last primary segment, because variable states are generally considered to last until simulation has ended. Note that in discrete event simulation, event time instants of primary and secondary time series need not be synchronised at all, and event density may differ both locally per series and overall across the two series, resulting in more complex algorithmic handling.

Overview

The family of *Analysis by State* methods (see fig. 4) differs in how secondary observations are related to primary state intervals:
In the *Sample by State* approach, all values of secondary segments which belong to the same primary state interval are incorporated into one *sample* per primary state interval.
Aggregation by State bases on *Sample by State* and applies an aggregate function on each sample of primary state intervals. Aggregate function values are displayed as horizontal histogram bars, based on the ordinate axis and drawn in the background of the primary time series.

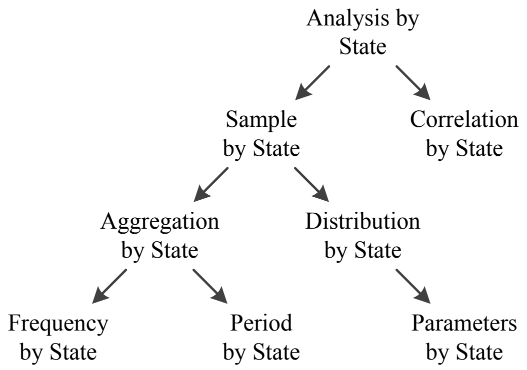

Figure 4: Analysis by State method hierarchy

Frequency by State is a specialisation of *Aggregation by State*: The secondary time series is a copy of the primary time series, and the aggregate function is fixed to *count*. The resulting frequency histogram of the primary time series is superimposed by the time series itself.
Period by State is similar to *Frequency by State*, but here a histogram of times spent per state interval is constructed, by implicitly deriving a secondary time series of inter-event time spans from the primary time series.
Distribution by State bases on *Sample by State*. Colour-coded histograms of the whole sample distribution are shown per primary state interval. Special aggregate values per sample are visualised as well. The chosen rows-of-histograms approach is equivalent to displaying an enhanced heat map in the background of the primary time series.
The *Parameters by State* method bases on *Distribution by State*. Here, all (thirteen) implemented aggregate functions are computed per sample. The aggregate values are shown in different series, as functions of primary state intervals. The result is a multi-aspect view of secondary state distribution per primary state interval, providing far more information than conventional scatter plots.
Last, *Correlation by State* computes partial correlation coefficients of corresponding primary and secondary observation segments. In doing so, total correlation is decomposable into partial correlation contributions per primary state interval.

The methods outlined above will be described in more detail in the following, illustrated by a simple example model.

Example model

A group of 10 servers with fixed service time distribution serves clients, which queue in front of the servers in a shared waiting queue. At the end of servicing one client, each server immediately services the next client from the waiting queue. The observed waiting queue length of a concrete simulation run is depicted in the upper part of fig. 3.
One of the servers leaves the group occasionally, to support other server groups. After some time, he returns and continues working. Likewise, an additional server from a different group arrives from time to time and temporarily works in the server group analysed here, returning to his own group some time later.
On leaving the system, clients are asked to rate their overall experience on a continuous scale from 1 (poor quality) to 10 (high satisfaction). The answer time (i.e. time clients need to decide on their rating) is recorded as well. Clients' rating usually will consider both total time spent in the system (processing time) and the actual quality of services rendered, summarised in one global mark. However, it is aim of the study to assess service quality only.
Recorded client satisfaction from a simulation run is depicted in the lower part of fig. 3. Obviously there is a high degree of over-plotting and comparatively low autocorrelation.
The system was modelled in DESMO-J, an open source discrete event simulation Java framework, which is developed and maintained by our Modelling and Simulation workgroup at the University of Hamburg (Göbel et al. 2013). The system was simulated for 30 days, with a statistics reset after 2 days of model time. The remaining four weeks of simulation contain more than 10,000 completed service operations.

Period by State

One might intuitively estimate that the upper primary time series of client queue length in fig. 3 is stationary and fluctuates around a mean level of approx. 20 waiting clients. The *Period by State* method helps to quantify how much observation time actually is spent in certain state intervals: All periods between subsequent events of the primary time series are determined and implicitly composed to a secondary, artificial "observation" series of time spans between primary events. Afterwards the primary and secondary time series are segmented by primary state intervals, and the (artificial) secondary "period observations" are mapped back to their corresponding primary state intervals (see section *Basic Procedure*). Finally, all collected periods per primary state interval are summed up, and horizontal histogram bars per state interval are drawn in the background of the primary time series (fig. 5).

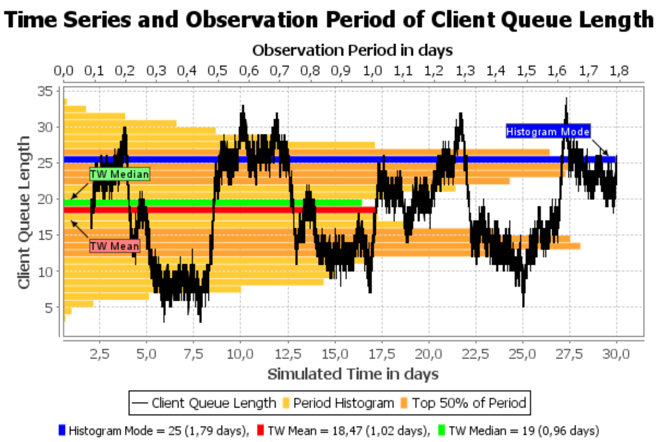

Figure 5: Period by State diagram

The length of each histogram bar is proportional to the sum of observation times spent in the corresponding state interval. A third axis is added at the top of the chart, indicating the observation period each histogram bar represents.
Additional information is given to enhance diagram interpretation:

- The histogram mode, i.e. the longest histogram bar, is highlighted in blue. If the histogram is unimodal, the mode bar can be considered as the centre of dynamics, around which state observations fluctuate. Because the mode bar spans from left to right through the whole charting area, it can be regarded as a second, implicit centred abscissa. The lower boundary of the state interval containing the mode bar and the mode bar's length are detailed at the bottom of the chart legend.
- The time-weighted mean and time-weighted median of the primary time series are computed, and the state intervals that contain their values are highlighted in red and green at the ordinate axis. The corresponding histogram bars are highlighted as well, and exact position and length of the mean and mode bars are detailed at the bottom of the chart legend, too.
- The colour of histogram bars is controlled by their accumulated length: the bars which contain the top 50% of total observation period are coloured in dark orange, the remaining bars in light orange. Thus, the (few) state intervals representing 50% (or a bit more) of total observation time can easily be spotted. The majority of dynamics happens in these state intervals.

The diagram layout – a horizontally rotated period histogram overlaid by its basic time series – is advantageous, compared to usual presentation of a vertical histogram next to a separate time series diagram. Beyond conventional analysis of histogram and time series on their own, Period by State diagrams facilitate the integrated analysis of relationships between their histogram and time series components: it can comfortably be seen when, in which sequence and how often histogram state intervals of interest were passed by the time series, and what happened beforehand and afterwards. Contributions of potential observational patterns and time intervals of interest to the period histogram bars become clearer, and might better explain which dynamic behaviour shaped specific histogram regions under investigation.
Issues like these could be examined by separate histogram and time series diagrams as well, but would involve permanently re-focusing back and forth between two diagram types, with additional rotation of histogram or time series by 90 degrees in one's mind's eye – a fatiguing and fault-prone process.

At a glance on the Period by State diagram, the original assumption of queue length stationarily varying around 20 clients can be rejected easily: the apparently "central" mean and median states (queue lengths 18 and 19) are mere transition states. The period distribution of client queue length in fact is bi-modal, with most frequent queue lengths at 25 and 13 clients.
This characteristic is attributable to model structure: In normal operation, the number of incoming and served clients balances out. When one of the servers is absent for support of a different group, the remaining servers cannot handle all incoming clients; thus client queue length grows. After return of the server, client queue length stabilises on the now higher level. On arrival of the additional server from another group, queue length shrinks again, since now more clients can be served than arrive. When the additional server leaves the group, queue length stabilises again on the now lower level. Thus in fact, queue length periodically alternates between two different levels. The first impression of spotting a stationary process is incorrect and a mere result of stochastic variance in client inter-arrival and service times.

Frequency by State

Period by State diagrams are meant for observation series of variables that should be time-weighted, like queue length or server utilisation. However, other variables exist where time-weighting makes no sense, for example client processing times or client satisfaction. These variables can be analysed by the Frequency by State method, which is basically identical to Period by State, with two exceptions:

- The implicitly constructed secondary "observation" series now is a mere copy of the primary time series, or – even more simple – a series that has a constant 1 (or any arbitrary value) at exactly the same time instants as the primary time series. As described in the *Basic Procedure* section, secondary segments (now containing arbitrary values) are constructed and mapped back to primary state intervals.
- Instead of adding these "observed" secondary values per primary state interval, they are just *counted* and visualised as horizontal histogram bars per state interval.

In a nutshell, Frequency by State diagrams show *the number of events* per primary state interval, whereas Period by State diagrams visualise *the sum of inter-event periods* per primary state interval.
As an example, the Frequency by State diagram of client processing time is shown in fig. 6.

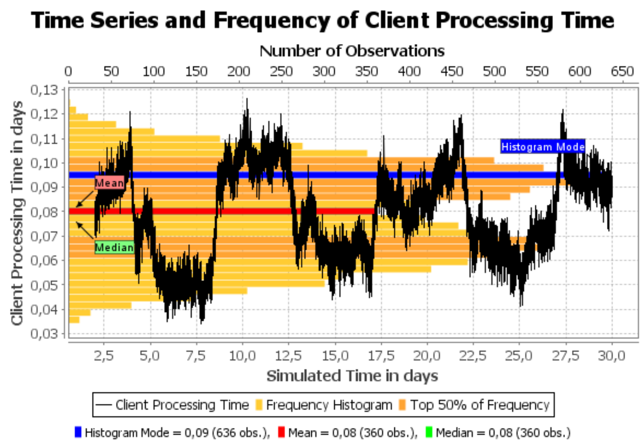

Figure 6: Frequency by State diagram

Apart from the construction process, the only visual difference to Period by State diagrams is the label of the third axis (top of the chart), which now quantifies the number of observations per state interval, instead of the observed period.
Unlike fig. 3, both Period by State and Frequency by State diagrams hide their secondary, implicitly created time series, because it does not contain genuine experiment observations: it was only constructed for intermediary reasons and thus should not confuse the experimenter. Likewise, there are no background colours indicating primary state intervals, because state intervals are clearly denoted by histogram bars. However, if of interest, secondary time series and state intervals may be displayed by setting respective parameters.

Unsurprisingly, the Frequency by State diagram of client processing time closely resembles the Period by State diagram of client queue length: total client processing time is the sum of time spent in the waiting queue (approximately proportional to client queue length) and service time. Since service time is distributed independently and identically, its variations will balance out in the long run (here: > 10,000 observations). Thus, clients' service in the average only adds a constant span to total processing time. For this reason, frequencies of total processing time are predominantly determined by periods of queue length, resulting in similar diagrams.

Sample by State and Aggregation by State

Both Period by State and Frequency by State are specialisations of the more general *Sample by State* approach. Here, all values of secondary segments that belong to the same primary state interval are collected in a special data structure, a *sample* (basically a multiset of observed secondary states).

Its specialisation *Aggregation by State* defines aggregate functions on these samples, in order to map every primary state interval's sample to one unique function value. Thirteen pre-defined aggregate functions have been implemented: first, last, count, sum, minimum, maximum, median, first mode, mean (=average), unbiased (=empirical) standard deviation, coefficient of variation, unbiased skewness and unbiased excess kurtosis. The modeller is free to add further aggregate functions as needed.

Apart from conceptionally offering arbitrary aggregate functions, Aggregation by State is a more general concept than Period by State or Frequency by State: here, the secondary time series can be selected freely from any observation series of the simulation experiment; it is not computed implicitly.

Nevertheless, visualisation of Aggregation by State, Period by State and Frequency by State follows the same concept: The primary time series is drawn onto a background histogram, whose bar lengths are determined by the chosen aggregate function. The secondary time series and background markers for primary state intervals resp. secondary time intervals may be displayed (or not).

Fig. 7 shows an Aggregation by State diagram, where client queue length has been chosen as primary time series and client satisfaction as secondary time series (cf. fig. 3, upper and lower part). The aggregate function is set to *coefficient of variation* (CV; ratio of empirical standard deviation to sample mean; relative standard deviation). Secondary time series and background markers for state intervals are hidden. Note that the third axis (top of the chart), quantifies the value of the secondary CV per primary state interval (i.e. client satisfaction CV per client queue length).

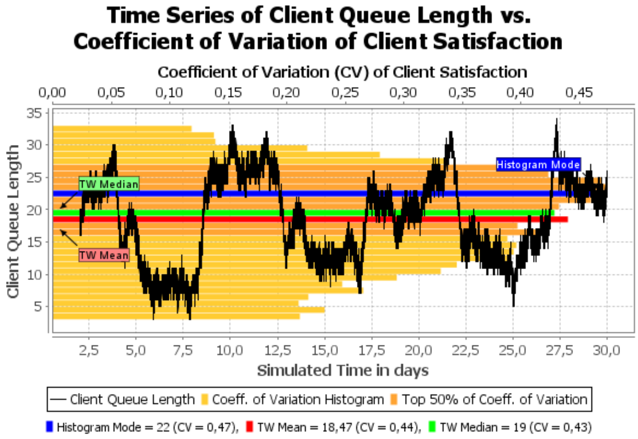

Figure 7: Aggregation by State diagram

The lower part of fig. 3 shows changing "cluster" ranges for the client satisfaction time series, implying that standard deviation of client satisfaction is not constant.

The Aggregation by State diagram in fig. 7 reveals more precisely, that variation of client satisfaction has a functional dependency on client queue length: When queue length is very low, clients' rating does not vary much. At low to high queue lengths, clients' satisfaction is in a wide range. At very high queue lengths, the rating range narrows considerably.

Also note that the CV is not symmetric, but skewed towards high client queue lengths.

If variation of satisfaction and client queue length were independent of each other, all histogram bars would have approximately equal length, apart from smaller stochastic deviations.

The observed phenomenon will be analysed further by applying additional Analysis by State family members.

Distribution by State

Like Aggregation by State, the *Distribution by State* method bases on Sampling by State. However here, each sample is visualised *in its entirety* in the primary state interval it belongs to.

For this purpose, the value range of every sample is divided into sub-intervals of equal span. Then, the number of secondary observations per sub-interval is counted, as representative for its subintervals' population density. This process is equivalent to the process of *binning* in the context of histogram construction. Finally, each sub-interval is colour-coded by population density and drawn as a rectangular cell into the Distribution by State diagram (fig. 8). Top and bottom cell boundaries are determined by the range of the corresponding primary state interval, and left and right cell boundaries are identical to the aforementioned sub-interval boundaries of samples.

The described construction process is repeated for every primary state interval resp. its corresponding sample of secondary observation values, resulting in an array of colour-coded "histogram rows" from top to bottom, in the upper diagram part.

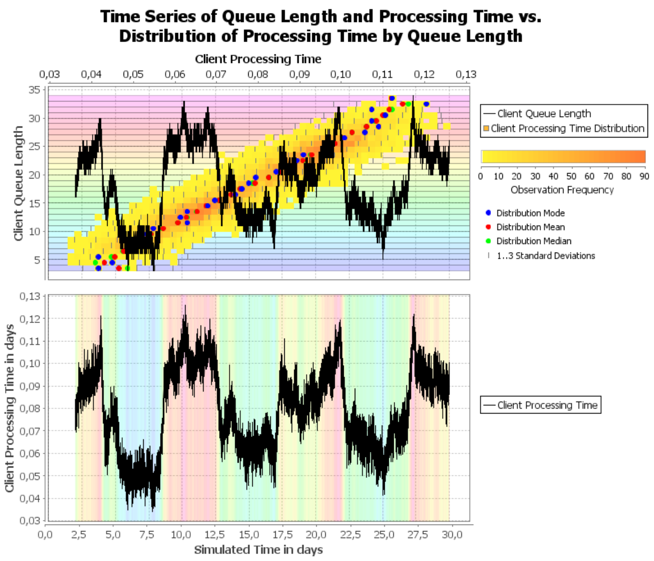

Figure 8: Distribution by State diagram

Visually, these continuous rows of histograms resemble an integral heat map (graphic representation of a data cell matrix), with colour coding the population density of heat map cells. Note however, that the graph still is a vertical array of horizontally laid out colour-coded sample distribution histograms. Adhering to this view, additional sample characteristics can be determined and highlighted per state interval (resp. horizontal sample histogram):

- The cell with the highest population (mode cell) is highlighted by a blue dot in the centre of the cell.
- The sample mean and the sample median are highlighted by red resp. green dots.
- Small vertical grey lines indicate the distance of one sample standard deviation from the mean dot. There are up to three standard deviation indicators left and right of the sample mean, to give an impression of sample variance and sample outliers.

Colour-coding of distribution histogram cells is performed on a global basis, i.e. the minimum and maximum of all cells (throughout all histograms) determine the total colour range. In this way, histograms of different state intervals become comparable (same colours code the same population density). Analogously, the sample subinterval boundaries are determined globally, hence cells of different horizontal histograms have same sizes and are located exactly one below the other, allowing for the impression of a "virtual", integral heat map behind the primary time series.
In support of this, the third axis at the top of the chart is scaled to fit the full range of all sub-intervals' sample minima and maxima, i.e. the virtual heat map always will stretch onto the full background of the primary time series.
In fig. 8, the secondary time series is displayed, and background markers for primary state intervals resp. secondary time intervals are switched on. The benefit of displaying both time series and the virtual heat map within one Distribution by State diagram is analogous to the overlaid histogram concept of Period, Frequency and Aggregation by State: The state distribution of the secondary time series can be seen at a glance, when analysing the primary time series. Additionally, by regarding background colours, it can easily be seen when, in which sequence and how often states of primary distribution histograms were passed by the *secondary* time series, and what happened beforehand or afterwards.

Fig. 8 confirms what was already made plausible in the *Frequency by State* section: Client queue length and client processing time are highly positively cross-correlated. Both time series displayed one beneath the other show similar details and background colour coding; therefore it is not surprising, that the upper virtual heat map is located closely to the bisector.
Of more interest is the relationship between client processing time and client satisfaction, shown in fig. 9.

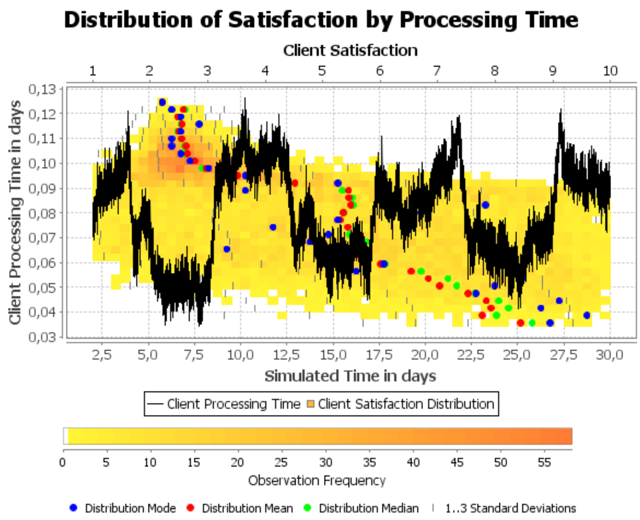

Figure 9: Distribution by State diagram

The Distribution by State diagram in fig. 9 quickly generates three insights:

- The mean and median of client satisfaction sample histograms are negatively cross-correlated to processing time, at high processing times > 0.09 days (approx. 2 hours and 10 minutes) and at low processing times < 0.06 days (approx. 1 hour and 25 minutes): there are descending sequences of red and green dots in these state intervals.
- However, this seems not so clear with the blue mode cells (indicating most frequent satisfaction per queue length) and medium processing times between 0.06 and 0.09 days.
- There is one major "frequency centre" in the virtual heat map at processing times > 0.09 days, yielding low rating from 2 to 5. Without further analysis one might wrongly conclude that mostly client satisfaction is low (which is supported by a frequency histogram of client satisfaction, not shown here), albeit it is unclear whether low ratings are really caused by poor service quality.

Parameters by State

Distribution by State diagrams visualise the whole distribution of primary state intervals' samples as colour-coded histograms, plus four aggregate functions (mode, mean, median, 1-3 standard deviations). The Parameters by State approach takes the next step and consequently

visualises *all* aggregate functions on primary state interval samples in one diagram.
In order to show all sample aggregates of the secondary time series as functions of the primary observation variable, the basic Distribution by State diagram is reflected over the bisector (fig. 10). Thus, primary state intervals are located at the abscissa (top and bottom axes of the diagram) and aggregate values of samples on the ordinates (left and right axes of the diagram). The underlying coloured histograms (resp. virtual heat map) are reflected as well. The original primary and secondary time series are not displayed; because of the change in diagram orientation they had to run from the bottom to the top of the chart, which is counter-intuitive and could confuse. Hiding the original time series visually clears space for a) connecting the dots of mode, mean and median by lines, appearing now like continuous mode, mean and median "functions" of the primary variable; and b) adding two more aggregate functions: minimum and maximum (see upper part of fig. 10).

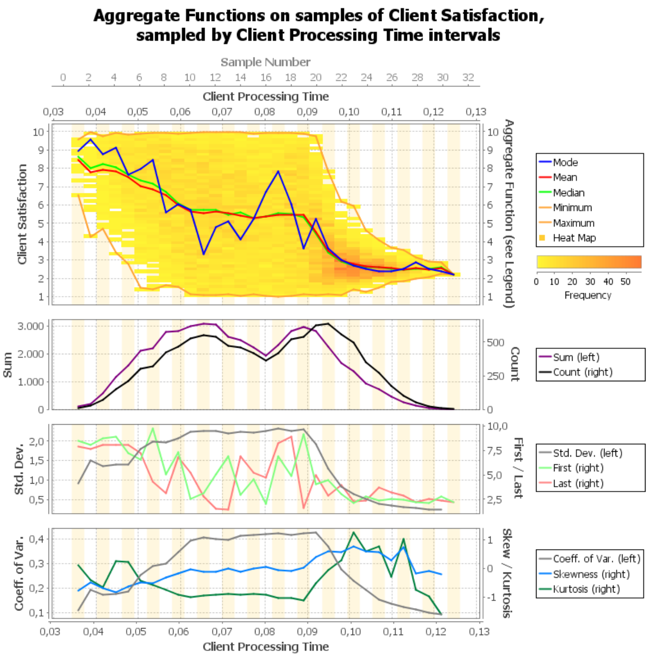

Figure 10: Parameters by State diagram

The above-mentioned aggregate functions have values in the same range as the secondary time series itself; therefore they can be superimposed with each other and with the virtual heat map in the top charting area.
However, aggregate functions like sum, count or standard deviation may be on different scales; therefore they are visualised in separate diagram sections below the main charting area. Every sub-diagram can refer to two scale axes at the left and right, hence two (or three) aggregate functions are displayed per sub-diagram, with mapping of aggregate functions explained by the right hand side legend.
Sometimes not all aggregate functions are of equally high interest: for instance, the *sum* of secondary observations will not always have an interpretation; *first* and *last* secondary observations per primary state interval may be consequences of stochastic processes and may be neglected sometimes.
The bottom diagram section contains the coefficient of variation (below the standard deviation section) and skewness and kurtosis, all in unbiased form. Deviation / variation, skewness and kurtosis give a fair impression of dispersion in vertical histogram columns at the top charting area, supporting interpretation better than estimation of histogram colour gradients with the naked eye.

If the secondary time series is distributed identically and independently of the primary observation variable, almost all aggregate functions should approximate horizontal lines (except for smaller statistical variations), and the top heat map should homogenously show horizontal stripes.
However, this is not the case in the example model. Fig. 10 visualises all aggregation functions on client satisfaction samples by processing time state intervals. Analysis of the Parameters by State diagram suggests the following findings:

- Processing times of less than 0.06 days lead to high client satisfaction with low coefficients of variation.
- Processing times of more than 0.09 days result in low client satisfaction with low coefficients of variation.
- For both cases above, negative correlation was already found in Distribution by State analysis (cf. fig. 9). The high positive correlation of queue length and processing time (fig. 8) suggests that client queue length may be a determining factor on rating outside the interval from 0.06 to 0.09 days, via its impact on processing time.
- At processing times *between* 0.06 and 0.09 days, a wide rating range is observed, and client satisfaction has high standard deviation resp. a high variation coefficient.
- Moreover, client satisfaction seems to be independent of total processing time in the range from 0.06 to 0.09 days: here, mean, median, standard deviation, CV, skewness and kurtosis are almost constant.
- If interested in the isolated evaluation of service quality, it may be hypothesised that clients' rating for service quality is more reliable at medium processing times from 0.06 to 0.09 days, because here variation of waiting times (i.e. the queue length component) apparently has no influence.
- Abstracting the "outer zones" of the virtual heat map, an average rating of 5.5 and standard deviation of approx. 2.2 are observed.

In fig. 11 the most popular means to examine two observation variables in discrete event simulation is shown, the scatter plot (here: for client processing time versus client satisfaction).

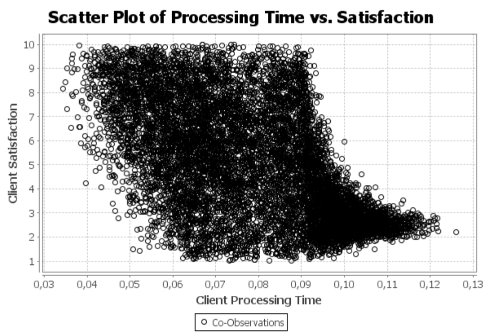

Figure 11: Scatter Plot of Co-Observations

Though the scatter plot's shape is basically similar to the upper virtual heat map in the Parameters by State diagram (cf. fig. 10), it suffers from heavy over-plotting, meaning that frequencies in black zones can hardly be estimated (compared to colour-coded heat maps). Because scatter plots are not sub-divided into state intervals (in contrast to the Analysis by State family's members), no indication "lines" for mean, median or mode can be drawn, and no local standard deviation, skewness or kurtosis per state interval can be determined. If only relying on scatter plots, the discrete event modeller might miss important information that Analysis by State could provide at low additional effort.

Correlation by State

The Sample by State approach and all its specialisations disregard time and sequence information of secondary time series: Samples merely contain observed values, but lack information, *when* and in which sequence values were observed.
However, it is of interest to relate primary segments and secondary segments to each other on a time basis, in terms of cross-correlation.
Since primary and secondary segments have the same start and end time instants, the total correlation coefficient of primary and secondary time series can be split up into *partial correlation coefficients* (PCC): The total correlation coefficient is computed as usual (e.g. on basis of "global" means and standard deviations per time series), but source observations are restricted to corresponding pairs of primary and secondary segments, per primary state interval. By this procedure, the amount that every primary state interval contributes to the total correlation coefficient becomes quantifiable.
Fig. 12 shows the result of this concept, the Correlation by State diagram: The primary time series is drawn on a background histogram of partial correlation coefficients (dark orange). The sum of histogram bar lengths is equal to the total correlation coefficient between both time series. In this respect, the Correlation by State diagram visualises the share each state interval has in overall cross-correlation.

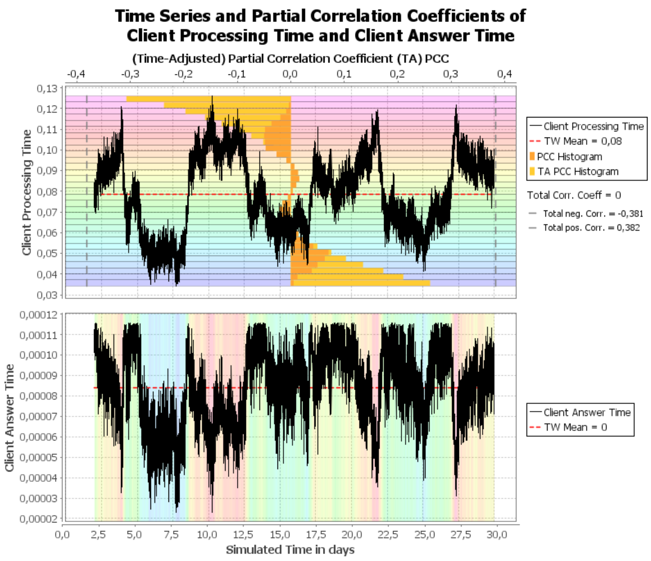

Figure 12: Correlation by State diagram

Long dark orange bars result from any (or a combination) of two factors: a) In these state intervals both time series have long or many segments in common; and/or b) cross-correlation between both time series is high in these state intervals. Note that due to multiplication of these two factors, state intervals with long common segments but low cross-correlation might yield the same PCC as state intervals with only few common segments but high cross-correlation. To correct for the time factor, each PCC bar's length is divided by the sum of segment periods per corresponding state interval. The result is shown in a light orange histogram of *time-adjusted* partial correlation coefficients (TA PCC). The TA PCC histogram is drawn behind the PCC histogram and has the same scale, indicated on the third top axis.
The TA PCC histogram visualises the degree of cross-correlation between both time series, independent of observation period and thus helps to identify where "original" cross-correlation is high.
The standard correlation coefficient of both time series is given in the diagram legend (here: 0), and further vertical dashed grey lines indicate the sums of all positive resp. all negative partial correlation coefficients.

The time series of client answer time (i.e. how long clients needed to decide on their rating) is graphed in the lower part of fig. 12.
Unfortunately the correlation coefficient between client processing time and client answer time is 0, meaning that there is no overall linear relationship between processing times and answer times.
However, a closer inspection of the Correlation by State diagram reveals high partial correlation coefficients per state intervals: At high processing times > 0.09 days, there is strong negative correlation to answer time (fig. 12: top histogram "bulge" to the left), meaning when overall service was long, clients gave their low rating (cf. fig. 10) within short time, presumably out of frustration. Conversely, low processing times < 0.06 days are strongly positively correlated to answer times (bottom histogram bulge to the right): When overall service was short, clients gave their high rating (cf. fig. 10) within short time as well, presumably desiring not to stay longer than necessary. All in all, rating from clients with both low and high processing times was given rather hastily. Therefore, rating from these two groups should be handled with care.
Time-adjusted partial correlation coefficients at medium processing times between 0.06 and 0.09 days are comparatively low, meaning all answer times (short to long) were observed independently of processing times. Here, clients took more time to come up with final ratings. Chances may be higher that overall experience is better reflected in these ratings, because not only the waiting period component is considered.
As a conclusion, based on Correlation by State and Parameters by State analysis, clients with medium processing times a) took more time for a supposedly thorough answer, assigning b) wide-ranging ratings, which are c) identically distributed and independent of processing time. Therefore chances are higher that this group's rating provides more significant indication to true service quality, compared to other groups, where the effect of short or long queue lengths distorts overall rating.

Apart from analysis of "local" correlation, the Correlation by State method lends itself for consistency checking of total correlation against partial correlation coefficients: If total correlation is highly positive or highly negative, no conspicuous partial correlations with reversed sign should be observable. Analogously, if total correlation is around zero, all partial correlation coefficients should amount to approx. zero as well, without remarkable single or systematic aberrations. When total and partial correlation coefficients are inconsistent (cf. fig. 12), differing local correlation should be explained. In that case the "traditional" total correlation coefficient loses significance for this variable pair and should be treated carefully in the further course of simulation output analysis.

SUMMARY, OUTLOOK AND CONCLUSION

The Analysis by State approach relates discrete-event time series on basis of state intervals and series' segments. A family of six specialised methods has been presented:
Period by State and Frequency by State aim at identification of important state intervals for a primary time series.
Aggregation by State, Distribution by State and Parameters by State target at analysis of sample characteristics for a freely selectable secondary time series, in relation to its primary time series' state intervals.
Correlation by State is concerned with state-local correlation and inner consistency checking of correlation coefficients.
All above-mentioned concepts are supported by corresponding diagrams. Since the six methods and their visualisations have complementary focus, they should be used in conjunction with each other.

All Analysis by State methods have been implemented in Java, as extensions of DESMO-J (www.desmo-j.de), an open source discrete event simulation framework, which is developed at the University of Hamburg. The implementation makes use of the JFreeChart library for visualisation purposes and is part of the more comprehensive software package FAVOR (Framework for Analysis and Visualization Of simulation Results).

Not all pairs of time series will yield additional insight by applying Analysis by State methods. Often, relationships can already be clarified by standard analysis, and/or no additional information may be contained in observations. In other cases, the nature of hidden information may be of different type, which the methods discussed here do not focus on, e.g. phenomena of periodicity.

In the future, the Analysis by State concept will be extended to *Autocorrelation by State*, enabling the modeller to check whether any nth degree auto-correlation of secondary time series depends on state intervals of a primary time series. Another specialisation, *Kernel Density Estimation by State*, is under consideration as well.

The purpose for introducing the Analysis by State method family is to further contribute to the statistical analysis of discrete event simulation dynamics. This is realised by visually augmenting original time series with additional information, or by series transformation. The desired benefit is to support and enhance identification of characteristics and relationships in discrete-event time series, in an easily to handle and comfortable way.

REFERENCES

Banks, J. 2010. *Discrete-event system simulation.* Pearson, Upper Saddle River, N.J, Singapore.

Coulling, A. 2013. *A Complete Guide to Volume Price Analysis.* CreateSpace Independent Publishing Platform.

Fishman, G.S. 2001. *Discrete-event simulation. Modeling, programming, and analysis.* Springer, New York.

Göbel, J.; P. Joschko; A. Koors; and B. Page. 2013. "The Discrete Event Simulation Framework DESMO-J: Review, Comparison to other Frameworks and Latest Development". In *Proceedings of the 27th European Conference on Modelling and Simulation,* European Council for Modelling and Simulation, W. Rekdalsbakken, R.T. Bye; and H. Zhang (Eds.) (Aalesund - Norway, 27th-30th May 2013), 100–109.

Hoad, K.; S. Robinson; and R. Davies. 2011. "AutoSimOA: A Framework for Automated Analysis of Simulation Output". *Journal of Simulation* 5, 9–24.

Kamich, B.M. 2003. *How Technical Analysis Works.* New York Institute of Finance, New York.

Koors, A. 2013. "Assessing Risk in Discrete Event Simulation by Generalized Deviation". In *Proceedings of the 8th EUROSIM Congress on Modelling and Simulation,* K. Al-Begain, D. Al-Dabass, A. Orsoni, R. Cant; and R. Zobel (Eds.) (Cardiff, Wales - UK, 10th-13th September 2013), 336–344.

Koors, A. and B. Page. 2012. "Transfer and Generalisation of Financial Risk Metrics to Discrete Event Simulation". In *Proceedings of The International Workshop on Applied Modeling and Simulation 2012,* The International Workshop on Applied Modeling and Simulation, A. Bruzzone, W. Buck, E. Cayirci; and F. Longo (Eds.) (Rome - Italy, 24th-27th September 2012). Genoa, Italy, 100–107.

Koors, A. and B. Page. 2013. "Application and Visualization of Financial Risk Metrics in Discrete Event Simulation - Concepts and Implementation". In *Proceedings of The International Workshop on Applied Modeling and Simulation 2013,* A. Bruzzone, C. Frydman, S. Junco, E. Cayirci; and C. Zanni-Merk (Eds.) (Buenos Aires - Argentina, 25th-26th November 2013), 118–130.

Law, A.M. 2014. *Simulation modeling and analysis,* Boston, Mass.

Ochoa, F.O. 2010. *Secrets of a Pivot Boss.* 311 Publishing, Washington, D.C.

Page, B. and W. Kreutzer. 2005. *The Java simulation handbook. Simulating discrete event systems with UML and Java.* Shaker, Aachen.

AUTHOR BIOGRAPHIES

ARNE KOORS obtained his master degree in Computer Science from University of Hamburg, Germany. Since then he has been working as a software developer and management consultant in the manufacturing industry, primarily in the field of demand forecasting and planning. Furthermore, he works as a research associate and on his PhD thesis on analysis and visualisation of discrete event simulation dynamics in the Modelling & Simulation research group led by Prof. Dr. Page.

BERND PAGE holds degrees in Applied Computer Science from the Technical University of Berlin, Germany and from Stanford University, USA. As professor for Modelling & Simulation at the University of Hamburg he researches and teaches in Computer Simulation and Environmental Informatics. He is the head of the workgroup which developed DESMO-J and the author of several simulation books.