ON THE REQUIREMENTS OF SIMULATION OUTPUT ANALYSIS SOFTWARE

(Presented at the ESS'99 – European Simulation Symposium, Erlangen, Germany, October 99)

Guilherme Luís Roëhe Vaccaro[1,2], Flávio Rech Wagner[2]

[1]Department of Mathematics – Pontific Catholic University of Rio Grande do Sul
Porto Alegre, RS, Brazil
[2]Computer Science Institute – Federal University of Rio Grande do Sul
Porto Alegre, RS, Brazil
E-mail: vaccaro@pucrs.br, flavio@inf.ufrgs.br

KEYWORDS

Discrete simulation, output analysis, analysis software.

ABSTRACT

The main purpose of this paper is to discuss the requirements of simulation output analysis software from different viewpoints: statistics, computational and user interface. The impact of lack of software support for simulation output analysis on the use of statistical techniques is discussed and confronted with those viewpoints. Finally, suggestions are made concerning the most relevant characteristics of simulation output analysis software.

1 INTRODUCTION

Simulation is a statistical sampling experiment. Therefore, the analysis of the output data generated by experiments is one of the main steps to be performed in a simulation project. Only by means of statistical analysis of output data obtained from a model it is possible to obtain scientific support for the decision making based on this model (Law and Kelton, 1991).

Experience has shown that the majority of simulation users pay little attention to output data analysis. Users calculate considerably fewer statistics than would be needed to ensure the correct understanding of the data and thus of the system as a whole. This lack of attention has surely many roots, but the main one is probably the fact that analysis software lacks adequate support for typical simulation users, who are not usually experts in Statistics.

This paper discusses which are the main needs of these typical simulation users, which factors contribute for the continuity of the current situation, and which aspects should be taken into account in the development of a tool for simulation output data analysis.

The remaining of this paper is organized as follows. Section 2 briefly discusses the role of output analysis in simulation methodologies. The main sources of difficulty for the achievement of adequate output data analysis are identified and discussed in Section 3. The most desirable characteristics of software for simulation output data analysis are then derived in Section 4. Section 5 briefly discusses relevant issues regarding the presence of output analysis facilities in simulation environments. Important suggestions for improving output analysis are given in Section 6. Finally, Section 7 draws conclusions and discusses future work.

2 THE ROLE OF OUTPUT ANALYSIS IN SIMULATION

Simulation is a tool for exploration and analysis that may be applied in the design and operation of complex systems and processes. It is based on the development and analysis of models that are representative regarding the real system and the goals of the study. Words "exploration" and "analysis" are related to the concept of "understanding": understanding the behavior and performance of the system, understanding the interactions between the system and the outer environment, and so on. The necessity of understanding is derived from a non-acceptance of a current situation and from the consequent necessity of modification or improvement. No simulation project can be useful if it does not lead to decision making.

On the other side, the literature strongly recommends that decision making must be based upon scientific evidence, which means information obtained with scientific basis. The analysis of simulation output data consists in the design of adequate simulation experiments and in obtaining appropriate measures for interpreting the results of such experiments (White Jr. 1997). The transformation of raw data into information is a result of data analysis. In the specific case of simulation, this depends on the utilization of statistical analysis. Simulation is essentially a sampling process and must resort to all available statistical tools.

Despite the fact that the above arguments are fully accepted in theory – and many international organizations and conferences foster discussion forums for research advances in this field – the operational reality is very different. It is not difficult to identify in the literature a great worry with regard:

– to the lack of scientific basis in decision making following simulation projects; and
– to the lack of adequate analysis tools.

Since the beginning of this decade, simulation specialists (Shannon 1992; Gogg and Mott 1995) suggest that the amount of time allocated to the three major tasks to be performed in a simulation project should be distributed as shown in Figure 1. This means that only 20% of the total time of the project should be "wasted" with system modeling, while the remaining time must be equally divided between data collection and data analysis and system exploration. This means that if, for example, two weeks are necessary for modeling a system, then it will be correct to invest about four weeks in each of the other major tasks of data collection and data analysis.

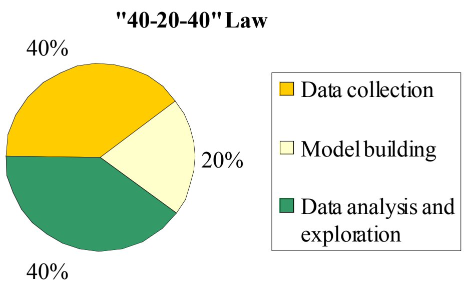

Figure 1: "40-20-40" Law. Source: Shannon 1992.

The development and availability of computational resources turned simulation into a powerful tool for planning, design and control of complex systems. Many advances have been achieved regarding the development of adequate environments for simulation modeling and execution of experiments. Integrated environments based on the VIM (Visual Interactive Modeling) and VIS (Visual Interactive Simulation) paradigms made easier to any user to create and experiment with a simulation model. Several initiatives have been taken to facilitate simulation input data modeling, most of them leveraged by the GIGO (Garbage In, Garbage Out) paradigm of the computational programming. However, the analysis of output data is still relegated to a secondary place, receiving few attention from the simulation user, who regards it as that phase between the "end of the simulation" and the final documentation.

3 USUAL PROBLEMS IN SIMULATION OUTPUT ANALYSIS

Several factors are presented in the literature (Charnes et al. 1994; Sanchez et al. 1994; Gregor and Kosturiak 1997) as contributing to a weak or incorrect utilization of simulation output data analysis. By analyzing the appointed reasons, we suggest their classification into four major characteristic groups:

- **Lack of statistical basis**: The problem is not due to the techniques used, but to their incorrect application. The adequate application of a statistical technique requires an understanding of the assumed hypotheses. Simulation is a tool for exploring and comparing alternatives, but only adequate statistical analysis can lead to consistent conclusions.

  The profile of the simulation user shows individuals whose contact with methodologies for statistical analysis occurred only once during a course in their undergraduate or graduate programs (Charnes et al. 1994). Very few users had the opportunity to make a deeper study and application of these methodologies.
- **Lack of adequate software:** Available software has a slave role. Most of them offer several data analysis alternatives in order to free users from a time consuming analysis work, but do not worry about hypotheses and criteria for a correct applicability of the available analysis techniques. Furthermore, very often the analysis task is simply not performed, because the user either does not have enough time or do not want to learn how to use a complex multipurpose statistical analysis software.

  The user needs an analysis software which is capable of giving him or her measures and information about the validity of the hypotheses supporting a given analysis technique or result.
- **Errors in the concept of simulation:** in the past, university courses have often erroneously diffused simulation as a set of mechanical methods for the computational modeling of problems. The result has been the dissemination of the wrong concept that simulation is a programming exercise, which looses interest after model verification and validation.

  The use of non-standard languages or even languages that have not been developed for simulation purposes strongly contributed for this wrong emphasis on training courses. As a consequence, many university courses relegated to a secondary role the necessity of adequate statistical analysis for a correct understanding of the simulation output data. Despite the emergence of new simulation technologies and more functional and simpler computational environments, few advances have been noticed in order to turn simulation users into good analysts.
- **Historical reasons**: The mentality of many simulation users still has its roots on former times when highly efficient programs were needed because of extreme computational costs. In this context, experiment replication required a very convincing justification, and intuition and experience of highly specialized teams

compensated the lack of output data. Technological evolution and integrated simulation environments led to smaller project teams, with fewer specialists. As a consequence, the discussion space and the technical quality of projects shrunk.

A schematic representation of the previous considerations is the "current reality tree" shown in Figure 2. This instrument, conceived by Goldratt (1994), allows the visualization of conflicts and the understanding of associated cause-effect relationships.

These relationships are depicted by arrows pointing from cause to effect, in the form "cause → effect". The perceived problem is located at the top of the tree and is handled as the termination of a process composed by many consecutive steps. The lower nodes of the tree present facts that have been appointed as causing a given effect. Many arrows arriving at the same node indicate that the corresponding causes are alternative, i.e., that any of these causes is sufficient for the occurrence of the effect (logical "or"). For the representation of simultaneous causes (logical "and"), a ring circling the arrows is used. The lowest nodes of the tree correspond to factors that are closer to the primary causes of the problem under analysis. This suggests that factors closer to the tree root contribute to a larger number of effects, but also that these factors are more generic and thus more difficult to solve.

Factors suggested in the tree in Figure 2 have been identified by the authors' experience on simulation studies and by the analysis of the literature. The set of problems depicted in this tree should not be considered as exhaustive, but highlights the process of information organization made possible by the current reality tree. Additional characteristics could be inserted into the tree according to the analysis of more specific contexts.

An analysis of the tree in Figure 2 identifies the lack of adequate software for typical simulation users as one of the main causes of delays and negligence in the output data analysis. The availability of a more friendly analysis tool, focused on the typical

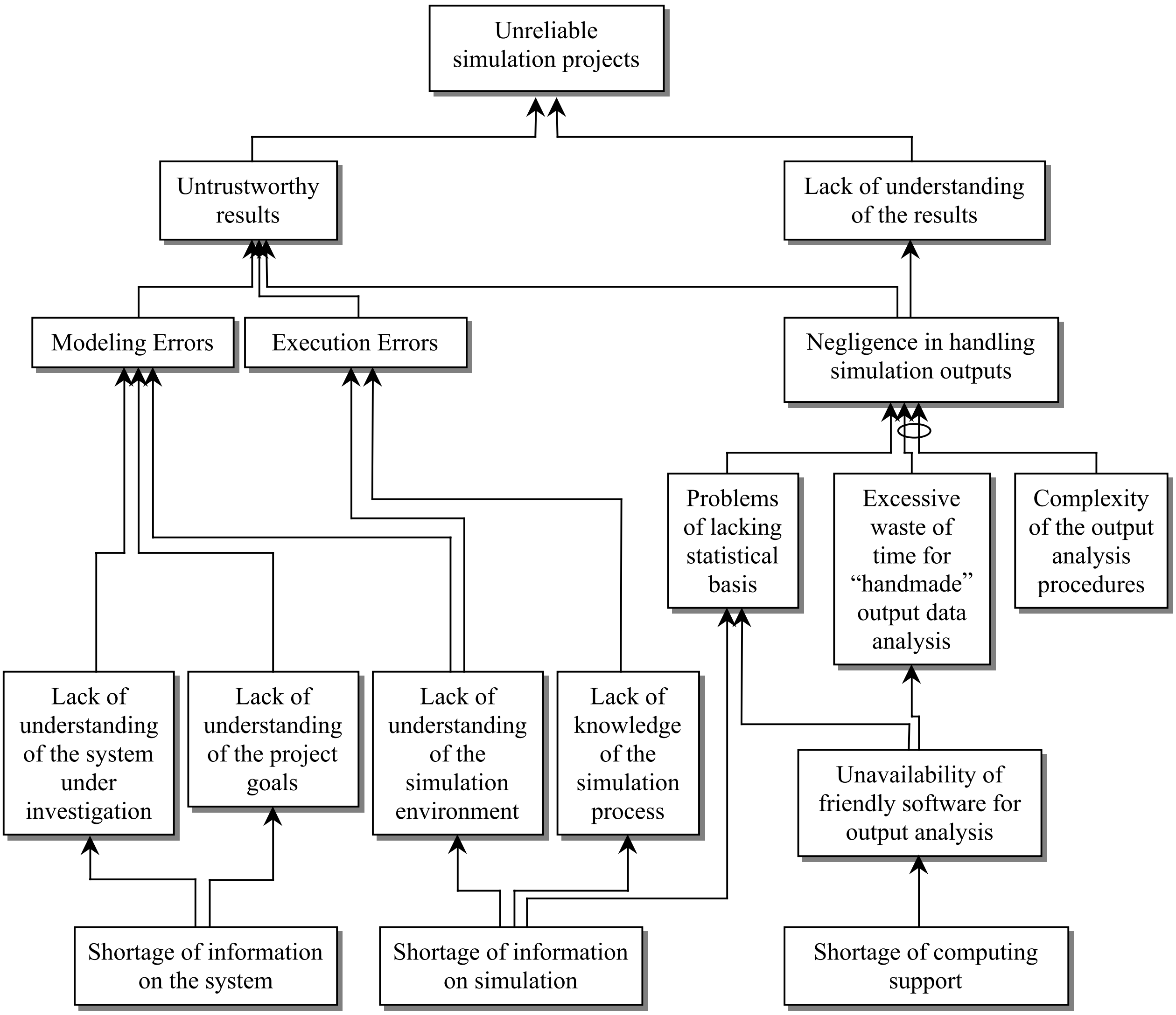

Figure 2: Current reality tree for inefficient simulation projects.

simulation user, would remove this factor from the tree and certainly minimize the time consumption and the complexity that are associated to this task. Adding some intelligence to the user interface and to the analysis routines, so that the less experienced user could be guided through the maze of available statistical methods, would also act positively on the solution of problems caused by the lack of statistical basis.

Finally, the lack of information, in its widest sense, is directly reflected by several factors, such as poor knowledge of the simulation process and of the adequate analysis techniques, lack of understanding of relevant characteristics of the simulation tool and of the system under modeling, and lack of understanding of the right questions that the simulation study must answer.

4 DESIRABLE CHARACTERISTICS OF SIMULATION OUTPUT ANALYSIS SOFTWARE

The literature reports on several studies about statistical support for simulation output data analysis (Banks, 1996; Charnes et al. 1994; Sanchez et al. 1994; Mollamustafaoglu et al. 1993). These studies, together with our own considerations about difficulties experienced by simulation users, allowed us to identify several desirable characteristics for simulation output analysis software. We discuss these characteristics according to three different viewpoints: the statistical, the computational and the user viewpoints.

4.1 STATISTICAL VIEWPOINT

Although simulation is a sampling procedure, it is not necessary that all statistical procedures are included in software for simulation output data analysis. A good analysis software can be produced by taking advantage of the most used statistical tools. More sophisticated tools can be left for more experienced users that are aware of state-of-the-art statistical packages.

Output data analysis software should include procedures for descriptive analysis, inference and, mainly, for giving the user a feedback with regard to the quality of input data and of produced results. As an example, the detection of the data variability should generate information on the minimum data volume that is acceptable for the realization of satisfactory analyses according to previously fixed criteria.

Explicit information regarding hypotheses that must be satisfied for the application of each method is another point that should be more strongly explored. A clear way of determining which methods should not be used would be highly recommended. This would give confidence to the user and would lead to more robustness of the results from a technical viewpoint. This may be difficult to implement as a generic resource, but the effort may be rewarding.

Graphical representation of the results is a primary necessity. The identification of the results along the experiment, as well as their correlation structure, should be a default option for the analyst.

One of the main statistical analysis tools in simulation studies is the calculation of confidence intervals for the observed measures. Analysts more easily accept these intervals than hypothesis tests because they give more information (Kelton 1994). The use of point estimates is a highly risky practice in simulation. The variability of output data is inherent to most natural processes, in particular those modeled by stochastic simulations. It is thus much more appropriate to give confidence intervals as a minimal form of result analysis (Banks 1996). Therefore, confidence intervals should be presented automatically by analysis reports, together with an indication on the methods and values that have been used for their generation.

Although it is not directly related to the output analysis itself, the design of experiments should also be supported and even at least partially automated, by the simulation environment. Without support, this task becomes time consuming and boring and does not stimulate the realization of tests that would make the results of the study more robust and reliable.

Methods that are specific for handling data of steady state and cyclic simulations should also be provided. When applying the method of batch means, for instance, it would be interesting that the software could suggest the batch sizes.

These are some examples of techniques that, together, could produce good software for simulation output data analysis. Other methods could and should be considered, but an excessive specialization would tend to make the software similar to many others, whose complexity is prejudicial to the goal of eliminating existing barriers to the utilization of output data analysis techniques in simulation studies.

4.2 COMPUTATIONAL VIEWPOINT

Under a structural point-of-view of the software, the use of object-orientation is highly recommended (Roberts and Dessouky 1998; Mollamustafaoglu et al. 1993). The qualities of modularity, hierarchical inheritance and structural familiarity provided by object-orientation induce several desirable features:

- Extensibility and maintainability: The capability of easily aggregating new methods or implementing more efficient algorithms result in higher reliability (i.e., quality that endures) to the software;
- Organization: The strongly hierarchical structure of object-oriented software allows methods to be more efficiently allocated, according to their

goals. Mollamustafaoglu et al. (1993) suggest structuring analysis methods by classes that are appropriate for:

- Basic statistical analysis;
- Analysis of terminating simulations;
- Analysis of non-terminating simulations; and
- Analysis of cyclic data.

This organization is directly reflected by an easier software operation, which is one of the main indications of user friendliness;

- Independence: Object-oriented software designs represent the abstraction of concepts and associations that are needed for the computational solution of a problem. This way, they make the conceptual structuring of the solution independent from implementation aspects, and the software becomes usable in a wider range of problems.

As stated by Roberts & Dessouky (1998), object-orientation brings a new perspective concerning statistical analysis tools, mainly on the sense of its structuring and development. Object-orientation allows the construction of highly reusable code and, more important, allows easy replacement and upgrade of analysis modules without modifications on other parts of the model.

Robustness is another crucial aspect. Analysis software should not present results that are inconsistent due to implementation problems or because a theoretical premise is not correctly considered. The information that is generated by such software is directly used in decision making processes and is thus critical. Therefore, a careful programming of the software is highly recommended, and object-orientation is an implementation paradigm that enforces a good programming practice.

4.3 USER´S VIEWPOINT

The output data analysis software must be friendly. Simulation is performed by professionals who come from several areas and have different backgrounds, especially regarding knowledge of statistical aspects and techniques. The lack of a deeper understanding of probability and statistics leads them to use only very simple analysis procedures. The consequent losses regarding the quality and correctness of decisions based on simulation outputs are considerable. Tool developers must do a great effort in order to make easier the access to more elaborate techniques. In case of doubt while taking a decision concerning tool development and especially user interface, the goal should always be to give support to the less experienced user (Sanchez et al. 1994).

An example of user friendliness may be a more flexible user interface format according to the degree of user expertise:

- For novice users, **interrogative** interfaces would be more interesting and easier to use. By means of a sequence of questions, the software could lead the user through the possible choices;
- Intermediate users could feel more comfortable with **declarative** interfaces, through which they give to the software indications on *what* they want to do, but not on *how* to do it;
- Advanced users could make use of **imperative** interfaces, which would allow them to indicate not only *what* to do, but also *how* to do it.

The utilization of a context-sensitive on-line help and of assistants could fill the lack of user's statistical knowledge in certain situations. In the case of graphical outputs, certain tests and even confidence intervals, examples and short texts can help the user to make a correct reading and interpretation of the results.

4.4 CONSIDERATIONS ON SOFTWARE QUALITY

Software is a tool: It is an extension of arms, legs and mind of its user. As with any other tool, the quality of software is measured by the "number" of activities that the user considers as well succeeded thanks to its utilization. The case for simulation analysis software is not different. As stated by Ursey and Dooley (1996), the main characteristics of software quality that are perceived by the user are:

- **Robustness** – capability of answering correctly even in adverse situations;
- **Friendliness** – easiness of access, execution and interpretation of interface and results; and
- **Capability** of answering to different types of questions formulated by the user.

The trade-off between responding to the needs of very different user profiles, as is the case of simulation users, and maintaining the friendliness may require a great design effort. In fact, this discussion does not seem irrelevant, since the main difficulty to be faced is cultural: to make the user *perceive* the quality of the analysis software. This means to understand the *benefits* of scientifically supporting conclusions by means of the software and, as a consequence, to become motivated to use it.

4.5 CONSIDERATIONS ON USING AN INTEGRATED SIMULATION PACKAGE

The use of integrated environments for simulation modeling, execution and analysis may enhance the quality of the conclusions driven from simulation studies. This happens because the availability of the analysis tools in the same simulation environment stimulates the user to make the necessary output analysis. The user will not need to manipulate the contents of various files or make complex data format transformations in order to feed another analysis software.

SIMOO (Copstein et al. 1997) is an example of environment that presents several features that make easy the integration of analysis modules:

- It is totally object-oriented (both from the modeling viewpoint as from the implementation perspective);
- It presents a separation between "autonomous objects", which implement the model logic, and "interface elements", which may for instance implement statistical methods for data analysis that are logically independent from particular models;
- It implements a special object called "monitor", which function is to observe messages exchanged between objects and may help collect data for a later statistical analysis.

Because of those characteristics, SIMOO is an interesting and desirable platform for the implementation of a module for analyzing output data that is tightly integrated into the simulation environment. More information on the SIMOO environment may be obtained in the site at http://www.inf.ufrgs.br/gpesquisa/simoo/.

5 SIMULATION OUTPUT ANALYSIS SOFTWARE

In the following, some of the most commonly used simulation environments are analyzed with regard to the previous considerations. This section, as this paper, does not intend to exhaust all possibilities in terms of existing environments and analysis tools, but rather to show the most common solutions offered to the typical users of simulation.

The most common types of output analysis software are independent packages and modules integrated into simulation environments. Although the purpose of these software types is the same, they can not be compared directly, because they have many significant differences. Particularly, independent packages intend to communicate with many simulation environments and to provide several kinds of analysis tools. They do not aim at a typical user profile and are not implemented with an integrated philosophy of operation, as is the case for analysis modules found in simulation environments.

In the following, five integrated simulation environments will be analyzed with regard to their output data analysis capabilities:

– AutoMod / AutoStat, from AutoSimulations, Inc.;
– ARENA / Output Analyzer, from Systems Modeling Group;
– ProModel, from ProModel Corporation;
– Taylor II, from F & H Simulations; and
– Micro Saint, from Micro Analysis and Design, Inc.

Relevant features of the SimStat environment, from Kovach Computing Systems, which is an independent package, will be also discussed.

AutoStat is a module that can be integrated into the AutoMod environment and provides support to the design of experiments, with interactive or batch execution. It also supports common random numbers. Techniques such as classical batch means, Welch's method and autocorrelation analysis provide detection of initial bias and warm-up period. Finally, some descriptive statistics, including confidence intervals, and statistical graphics are provided.

ARENA incorporates a module, named Output Analyzer, to perform "offline" output data analysis. Communication of data between this module and the simulation environment is done by means of files and introduction of gathering directives into simulation models. Data can also be imported from external files, with the purpose of comparison with real data. The most remarkable features provided by this module are:

- Bar graphs and histograms;
- Step graphics and linear interpolation graphics;
- Correlograms;
- Frequency tables; and
- Confidence intervals for the statistical mean and the standard deviation.

For the batch means technique, the Output Analyzer also provides the definition of batches by their size or time intervals, which are tested by an independence test with a fixed level of 5% of significance. If rejected, a warning message is presented to the user. After grouping, data can be smoothed by moving averages or can be analyzed against another data set for comparison purposes. Analysis of variance is also provided at this step.

ProModel is typically an industrial simulator. So, it automatically presents pie charts of utilization for each resource in a model. Its statistical analysis tools are mixed with the simulation environment, providing:

- minimum, maximum and mean processing time;
- minimum, maximum and mean size of queues; and
- minimum, maximum and mean waiting time on queues.

Confidence intervals can also be calculated for numerical variables and there is a reasonable support for the design of experiments, including the possibility of automatically executing multiple replications and scenarios. Welch's method is provided for determining warm-up periods. Classical batch means are presented for autocorrelation treatment, but the user must provide the size or the duration of each batch.

As a simulation environment that highlights a simulation language, Taylor II can provide a relatively complete support for simulation output analysis, especially based on graphics. Its main facilities are:

- bar graphs;
- step graphics for queues;

- histograms for the queue size and waiting times on queues;
- confidence interval graphics;
- autocorrelograms;
- pie charts; and
- Gantt charts.

Taylor II also provides interactive histograms that allow the user to graphically obtain central tendency measures. The use of common random numbers and antithetic variables is also supported.

Micro Saint does not present many alternatives for output analysis, which is performed “offline”. Data representation is always numerical, and the software automatically presents a table with the statistical average, the standard deviation, the minimum and the maximum of the observed values for each selected variable. It can also present scatter plots, step graphics, bar graphs, and histograms. For each queue defined in the model, Micro Saint automatically saves the maximum, minimum, mean and standard deviation of its size and waiting time of parts.

The SimStat package slightly differs from the above environments because it is independent of any simulator. Using a functional paradigm of operation, SimStat can communicate with many different databases and calculate several descriptive statistics and confidence intervals. Hypothesis tests can also be performed to compare means and variances of different data sets. Multiple replications are supported, as also common random numbers and antithetic variables. The Welch’s method is provided for warm-up determination. Histograms, pie charts, step graphics and charts of probability distribution functions complete the facilities provided by this tool.

The analysis of these environments shows that there is no consensus on what type of environment is best suited for the typical simulation user. The mixing of both the simulation and the output analysis environments can provide flexibility in terms of data manipulation and “mechanization” of analysis routines, allowing the presentation of “online” analysis. On the other hand, these analysis modules generally do not implement techniques that really take advantage of their coupling to the simulation environment. As a consequence, independent software, such as SimStat, becomes more interesting because it can both present the same analysis facilities as integrated modules and communicate with different simulation environments. As a negative point, independent software suffers some resistance from simulation users, especially when it is not integrated into the same user interface.

6 SUGGESTIONS ON IMPROVING SIMULATION OUTPUT ANALYSIS

In order to enhance and qualify the utilization of simulation output data analysis, the first goal is maybe to convince the user that statistical analyses are essential to the decision making. In sequence, training on the application of a minimal set of methods that are capable of solving a sufficiently large number of problems must be proposed.

On the other hand, no set of methods will be effective without training on the simulation methodology. At least for several years from now, no software will be capable of making analyses by itself and verifying the validity of information obtained from the simulation study. This task still requires the continuous intervention of a human user, who needs to know not only the system under investigation, but also how to interpret data that are presented by the software. Adequate software may ease the task, but will not exempt him or her to think! More opportunity for simulation training and use of statistical analysis tools in university courses is an easy and efficient way of minimizing much of the shortages we discussed.

The notations, the language and the approach followed by output analysis software must be specific for the simulation users. The software design must consider that the user is only interested in Statistics as long as he or she obtains correct numbers that can be easily interpreted. This aspect is not correctly explored by current software and could bring considerable advantage with a minimal effort. Currently, when the phase of interpreting results arrives, the simulation user is confronted with a crossroad: either he or she chooses a statistical package, with general-purpose language and approach (i.e. not oriented towards simulation output data), or only an elementary, but well known, analysis methodology will be applied. If a statistical package is chosen, the user will have to “dig” into the package, searching for adequate methods among a large number of available methods. As a consequence, most of the users do not perform a more complete analysis. The use of expert systems associated to those packages could also help minimize this problem, but the correct solution would be to offer a set of methods and techniques for statistical analysis that are appropriate for the problem at hand and for the user expertise.

7 CONCLUSIONS AND FUTURE WORK

The analysis of current simulation environments reveals a considerable shortage on software that may facilitate the process of analyzing output data. The strong emphasis on computational modeling and the gap when this is compared to the emphasis that would also be needed on statistical techniques for output data analysis created a dangerous separation between simulation and analysis. The result is that the output data analysis, one of the most important phases of the simulation process, as it directly affects decision making, is often neglected and decisions are taken based either on the "appearance" of data or on the result of inadequate tests.

Recovering the importance of this task in the simulation methodology depends on several actions, going from the education on the analysis techniques to the development of tools that give adequate support to the analysis.

Simulation users need software with friendly interfaces and focused on the questions that are most relevant to the data analysis. A minimal set of techniques with broad application for the types of analyses that are usually found in simulation studies must be determined. These techniques must also be helped by computational mechanisms that facilitate the interpretation of analysis results.

This paper discussed the results of the first step of a study on simulation output data analysis. The next planned steps involve the complete specification of the recommended techniques for integration into an analysis software, as well as the specification of the structural and interfacing features of this software. Afterwards, the software will be implemented in tight integration with an existing object-oriented simulation environment. As a result, we hope to contribute to diffuse the importance of the output data analysis as a fundamental task of a simulation methodology, in order to give a better theoretical and scientific background to decisions based on simulation studies. We also hope to demonstrate the usefulness of a tight integration of adequate analysis techniques within a simulation environment.

REFERENCES

Banks, J. 1996. "Output Analysis Capabilities of Simulation Software". In: *Simulation*, vol. 66, nº 1, January. p. 23 – 30.

Charnes, J.; Carson, J.; Dewsnup, M.; Seila, A.; Tew, J. & Sadowski, R. 1994. "Output Analysis Research: Why Bother? A Panel Discussion". In: *Proceedings of the 1994 Winter Simulation Conference*. San Diego, USA. p. 399 – 405.

Copstein, B.; Wagner, F. & Pereira, C. 1997. "SIMOO: An Environment for the Object-Oriented Discrete Simulation". In: *Proceedings of the ESS'97 – Simulation in Industry. 9th European Simulation Symposium*. Passau, Germany. p. 21 - 25.

Gregor, M. & Kosturiak, J. 1997. "Simulation: Strategic Technique for the Factory's Future". In: *Simulation*, vol. 69, n° 5, November. p. 291 – 305.

Gogg, T. & Mott, J. 1996. *Improve Quality & Productivity with Simulation*. 3rd ed. JMI Consulting Group. USA. 300 p.

Goldratt, E. 1994. *It'$ Not Luck.* 1st ed. Educator. USA. 283 p.

Kelton, W. D. 1994. "Analysis of Output Data". In: *Proceedings of the 1994 Winter Simulation Conference*. San Diego, USA. p. 62 – 68.

Law, A. & Kelton, W. D. 1991. *Simulation Modeling and Analysis*. 2nd ed. McGraw-Hill, Inc. New York, USA. 739 p.

Mollamustafaoglu, L.; Gurkan, G. & Ozge, A.Y. 1993. "Object-Oriented Design of Output Analysis Tools for Simulation Languages". In: *Simulation*, vol. 60, nº 1, January. p. 6 – 16.

Roberts, C. & Dessouky, Y. 1998. "An Overview of Object-Oriented Simulation". In: *Simulation*, vol. 70, nº 6, June. p. 359 – 368.

Sanchez, P.; Chance, F.; Healy, K.; Henriksen, J.; Kelton, W. & Vincent, S. 1994. "Simulation Statistical Software: An Introspective Appraisal". In: *Proceedings of the 1994 Winter Simulation Conference*. San Diego, USA. p. 1311 – 1315.

Shannon, R. E. 1992. "Introduction to Simulation". In: *Proceedings of the 1992 Winter Simulation Conference*. San Diego, USA. p. 65 – 73.

Ursey, M. & Dooley, K. 1996. "The Dimensions of Software Quality". In: *Quality Management Journal*, vol. 3, n° 3. p. 67 – 86.

White Jr., K. 1997. "An Effective Truncation Heuristic for Bias Reduction in Simulation Output". In: *Simulation*, vol. 69, n° 6, December. p. 323 – 334.

BIOGRAPHIES

Guilherme Luís Roëhe Vaccaro is assistant professor at PUCRS – Pontific Catholic University of Rio Grande do Sul and is currently performing his Ph.D. studies at UFRGS – Federal University of Rio Grande do Sul, Brazil. He received both his M.Sc. degree in Industrial Engineering and his B.Sc. degree in Applied and Computational Mathematics from UFRGS.

Flávio Rech Wagner is full professor at the Computer Science Institute of UFRGS. He received his Dr.-Ing. degree in Computer Science from the University of Kaiserslautern, Germany. He is a member of the SCS, a researcher of the Brazilian National Research Council (CNPq) and President of SBC, the Brazilian Computer Society.