GENERIC BUILDING BLOCKS FOR SIMULATION MODELLING OF STOCHASTIC CONTINUOUS SYSTEMS♣

M. Albertyn[1] and P.S. Kruger[2]

[1]The Defence Institute, a Division of Armscor Business (Pty.) Ltd.
Pretoria, South Africa
martina@armscor.co.za

[2]Department of Industrial and Systems Engineering
University of Pretoria, South Africa
paul.kruger@eng.up.ac.za

ABSTRACT

The key objective is to present the generic building blocks of a methodology that can be used to model stochastic continuous systems efficiently. The original simulation model of a real-world system is used as the basis for the development of a generic modelling methodology. The generic building blocks of the methodology are used to construct two new simulation models using two different simulation software packages (Arena and Simul8). The evaluation method, the determination of adequate sample sizes and the verification and validation of the models are discussed. The models and software packages are compared and conclusions are presented.

OPSOMMING

Die hoofdoelwit is om die generiese boublokke van ‘n metodiek voor te hou wat gebruik kan word om stogastiese kontinue stelsels doeltreffend te modelleer. Die oorspronklike simulasiemodel van ‘n werklike-wêreld-stelsel word gebruik as die basis vir die ontwikkeling van ‘n generiese modelleringsmetodiek. Die generiese boublokke van die metodiek word gebruik om twee nuwe simulasiemodelle te konstrueer met twee verskillende simulasiesagtewarepakkette (Arena en Simul8). Die evaluasiemetode, die vasstelling van voldoende monstergroottes en die verifikasie en validering van die modelle word bespreek. Die modelle en sagtewarepakkette word vergelyk en gevolgtrekkings word voorgehou.

♣ This paper is a more detailed version of a paper titled: “*Generic Modelling of a Stochastic Continuous System*” that was presented at the 16th European Simulation Multiconference (ESM’2002) that was held from 3 to 5 June 2002 in Darmstadt, Germany.

[1] The author was enrolled for the PhD (Industrial Engineering) at the Department of Industrial and Systems Engineering, University of Pretoria

1. INTRODUCTION

The system that is used as the basis for the simulation modelling methodology illustrated in this article is a continuous process plant. The plant produces chemical products from coal. A simplified schematic representation of the plant is shown in Figure 1. The plant is representative of a real continuous process plant (and also of any stochastic continuous

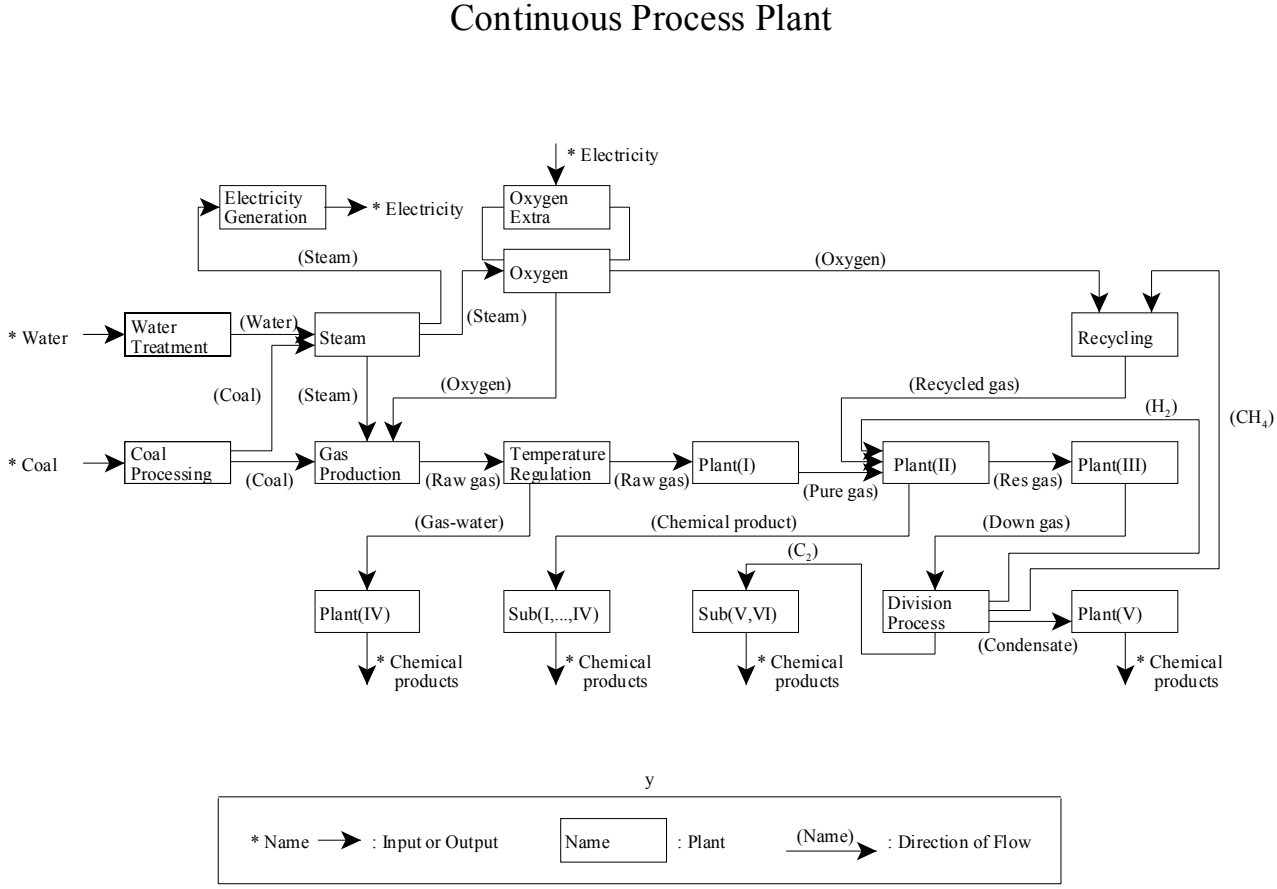

Figure 1: Continuous Process Plant

The need for a simulation model of the plant originally arose because the plant management identified the necessity of a decision support tool at a strategic level (Owen [1]). In a plant of this size and complexity it is extremely difficult to predict what the effect of a proposed change is going to be on the operation of the plant. The complex interrelationships that define the operation of the plant, as well as random events such as failures of equipment, can be dealt with in a simulation model. The main objectives of the model are to identify problem areas ("bottlenecks") in the system and to study the effects of projected scenarios on the system. Projected scenarios may include added capacity at bottlenecks, changes in the support philosophy, etc.

The original simulation model of the plant was developed, refined, expanded and maintained over a three-year period from 1994 to 1996. This relates closely to a comment from Crowe et al. [2] to the effect that it may take two to four man-years to supply answers to technically

sophisticated problems with a simulation model. From 1996 to 1999 the model was in continuous use as a decision support tool. It was used for the evaluation of several different proposed scenarios. During 1999, a concern developed that the original model (constructed according to a model definition that reflected the 1996 plant configuration) might no longer accurately reflect the current status of the plant configuration. It was decided to explore the feasibility of updating the model to the current status of the plant configuration. This presented a unique opportunity to do something more than just updating the original model. It afforded the chance to use the original model as a point of departure for the development of a new modelling methodology that would be more generic and at the same time address the shortcomings of the original model.

The objective of this article is to present the generic building blocks of a methodology that can be used to model any generic equivalent of a stochastic continuous system very efficiently. The original simulation model of the real continuous process plant is used as the prototype for the development of the generic modelling methodology. The methodology is used to construct two new simulation models of the continuous process plant. The new models are constructed in two different simulation software packages to clearly demonstrate the use of the methodology.

2. MODEL DEFINITION AND CONCEPTUALISATION

The model definition and conceptualisation of the original simulation model give a perspective of the size, complexity and level of detail of stochastic continuous systems that can be accommodated by the generic modelling methodology. It should be noted that the methodology renders a model that can be used as a decision support tool at a strategic level (see previous section). The methodology does not make provision for the level of detail necessary to model the transient behaviour of chemical processes.

For the purpose of this article, the original simulation model is not addressed in detail. Only a brief overview of the most important concepts is presented. Albertyn [3] provides a detailed explanation of the development of the original simulation model.

To demarcate the model, it is necessary to define the system being considered. The composition and process logic of the plant can be derived from Figure 1. For the purpose of the model, the total plant is considered to consist of 20 smaller plants and 147 modules. A module can be seen as a group of equipment that has a specific function. Some plants consist of groups of different types of module. Parallel lines of modules in a plant are referred to as "trains". The plant has a complex switching capability. This implies that if one of the modules of a train goes down, the whole train is not necessarily rendered inoperative.

The following key characteristics of the plant can be identified:

a) It is a "continuous flow" plant.

b) The plant is subject to two types of discrete events, namely services (chronological events) and failures (stochastic events).

c) The plant has complex interrelationships.

In the real world, the "time between failures" and the "time to repair" are stochastic events. This is in contrast to services which start at predetermined times by the decommissioning of the module for a predetermined time interval. The service schedules and service durations are strictly chronological events and therefore can be handled in a deterministic manner. Complex interrelationships are manifested in both the composition and the operation of the plant. The plant has feedback-loops and does not have a simple serial or parallel configuration. Crowe et al.[2] give some insights into the complexities that follow from the problem of the recycling (feedback-loops) of either matter or heat in chemical plants. The switched-on or -off logic of the different modules also presents complex interrelationships from an operational point of view. The continuous nature of the plant implies that all 147 modules are intrinsically interlinked in a way. If a breakdown in production (caused by service or failure) occurs at one point, it impacts immediately on upstream and downstream operations.

The throughput of the total plant depends on the capacities and the availabilities of the different modules. The availability of a module depends on the service schedule and the incidence of failures. To model the throughput at any given time, the status of each individual module must be known. The model also needs to identify the module group that represents the momentary "bottleneck" in the system at that specific time. This point is the momentary "driver" that will determine the maximum possible throughput. From this point, the status of the modules (whether switched on or off), for upstream and downstream module groups, can be determined. The philosophy of maximising the possible throughput will lead to all the modules in the identified "bottleneck" module group being switched on. In contrast to this, the upstream and downstream module groups could have some modules that are not switched on. Due to the complex interrelationship characteristic of the plant, the identification of the momentary "bottleneck" and the status determination of the associated switching logic require complex deterministic calculations. With the module status known and the momentary "bottleneck" identified, the throughput can be calculated and the status of the switching logic can be determined.

If all the previously mentioned procedures are included in a model and evaluated at predetermined time intervals, the results should approximate the operation of the real plant.

3. ORIGINAL MODEL DETAIL

The original simulation model was developed in Arena and incorporated a Watcom FORTRAN subroutine.[2]

The service schedules and the occurrence of failures of the modules are handled by the Arena part of the model. The theoretical probability distributions provided in Arena are used to model the stochastic phenomena (Pegden et al. [4]). The identification of the momentary "bottleneck", the calculation of the throughput and the determination of the status of the switching logic are handled by the FORTRAN subroutine. The Arena part of the model also collects statistics about the behaviour of the model.

[2] Arena is a simulation software package from what was formerly known as Systems Modeling Corporation and that now forms part of Rockwell Software Inc. and Watcom FORTRAN is a software package from the Watcom International Corporation. It should be noted that Arena is a registered trademark and usually denoted by Arena®. However, for the sake of simplicity it will be written simply as Arena in this paper.

The original simulation model uses the iteration time interval method of evaluation - see next section for a detailed explanation. A one hour iteration time interval is used (Albertyn[3]). The required sample size for appropriate output parameter estimation is determined and the model is verified and validated (Albertyn [3]).

The Arena part of the original simulation model is strictly generic and makes provision for the inclusion of stochastic phenomena. The FORTRAN subroutine is a double-edged sword however, leading to both advantages and disadvantages as far as the modelling method of the original simulation model is concerned.

The advantages of using a FORTRAN subroutine are:

a) The method allows the modeller to incorporate complex decision-making processes into the model.
b) The incorporation of a FORTRAN subroutine into the model to handle the complex mathematical calculations that are required helps to keep simulation runtimes within acceptable limits.

The disadvantages of using a FORTRAN subroutine are:

a) The FORTRAN subroutine has extremely complex structures and to a large extent, it is not generic. (In fact, a small change in the model definition can possibly lead to major changes in the subroutine.)
b) The method gives rise to a very complicated model structure, involving two different software packages and complex interfacing, compiling and linking.
c) The complex structure of the model complicates debugging. (It is sometimes difficult to assess whether a faulty event occurs in the Arena part of the model, or in the FORTRAN subroutine.)

The user-friendliness of simulation models is an important factor in their acceptance and continued usage (Bonnet [5]; Elder [6]; Simulation Fax Survey Results [7]). The user-friendliness of the original simulation model is enhanced by the use of input files for the manipulation of model parameters whenever possible. The use of animation and graphics can also enhance user-friendliness and help with model debugging (Pegden et al. [4]; Elder [6]). The original model only includes limited use of animation and graphics because over-elaboration can have an adverse effect on runtimes.

4. GENERIC MODEL CONCEPTUALISATION

The original simulation model of the real continuous process plant is used as the prototype for the development of the generic modelling methodology. The methodology presents a "toolbox" of methods and techniques that can be used to model a stochastic continuous system, or any generic equivalent of such system, very efficiently. The methodology is able to accommodate a model of a stochastic continuous system of approximately the same size, complexity and level of detail as the model presented in the previous sections. The methodology renders a model that can be used as a decision support tool at a strategic level.

The generic modelling methodology presents a structured approach with the following characteristics:

a) Short development time for the model.
b) Short turnaround times for model maintenance.
c) User-friendliness as perceived from the development, maintenance and usage perspectives.
d) Short simulation runtimes.
e) Compact model size.
f) Robust modelling ability.
g) Accurate modelling ability.
h) Single software application.

The characteristics follow from the design criteria and result in the efficiency of the methodology.

The generic modelling methodology consists of eight methods and techniques that were identified and developed to accommodate the characteristics of the continuous process plant in a simulation model. Table 1 gives an overview of the characteristics of the plant, the appropriate method or technique and the purpose of each method or technique.

Van Dyk [8] postulates that the hierarchy of terminologies used in industrial engineering proceeds along a continuum. The following hierarchy is suggested: tool, technique, method, approach and philosophy (arranged from lower to higher order). It is suggested that the transition within this hierarchy should occur continually. Even though Van Dyk does not distinguish between the terms “method” and “methodology”, in the context of this article the term “method” is perceived to be indicative of a lower-order terminology while the term “methodology” is perceived to be indicative of a higher-order terminology. Table 1 adheres to the hierarchy suggested by Van Dyk with the term “technique” perceived to be indicative of a lower-order terminology and the term “method” to be indicative of a higher-order terminology.

A detailed discussion of the eight methods and techniques that make up the generic modelling methodology is unfortunately impossible due to limitations of space (with the exception of the two simulation model evaluation methods discussed in Section 6). The rest of this article focuses on the generic “high-level” building blocks that were developed to accommodate the methods and techniques of the generic modelling methodology and the two new simulation models of the continuous process plant constructed with the “high-level” building blocks.

Plant Characteristic	Method or Technique	Purpose
"Continuous flow"	Variables technique	Uses variables to represent continuous process flows, such as the throughput values, as real numbers
	Iteration time interval simulation model evaluation method (see Section 6)	Uses an iteration time interval evaluation method to advance the simulation model in simulated time
	Event-driven simulation model evaluation method (see Section 6)	Uses an event-driven method to advance the simulation model in simulated time
Discrete events (services and failures)	Entity-represents-module (ERM) method	Uses entities to represent the modules in the simulation model and determines the status of the modules at any given moment in time
Complex interrelationships	Fraction-comparison (FC) method	Identifies the momentary "bottleneck" at any given time
	Iterative-loop technique	Determines the governing parameters of the FC method
	Time "bottleneck" identification technique	Identifies "bottlenecks" based on the time that each smaller plant is the "bottleneck"
	Production lost "bottleneck" identification technique	Identifies "bottlenecks" based on the possible production that is lost due to each smaller plant

Table 1: Plant Characteristics and Appropriate Methods and Techniques

The generic modelling methodology uses generic "high-level" building blocks to build up the required model according to a model definition. Five distinct generic "high-level" building blocks can be identified:

a) Plant with multiple service cycles and failures of modules.
b) Plant with a service cycle and failures of modules.
c) Plant with a service cycle of modules.
d) Plant with failures of modules.
e) Logic engine.

For example, every one of the 20 plants that make up the original simulation model can be constructed by using one of the generic building blocks a) to d). The distinction between building blocks a) and b) is necessary because some of the modules have more than one service cycle. For example, the coal-processing plant has three service cycles superimposed on one another and hence provision is made in building block a) for three superimposed service cycles. Naturally, every one of the 20 plants can be represented by building block a) because it makes provision for the most complex plant configuration encountered in this context. Building blocks b), c) and d) are only included to ensure that a compact model size

is achieved (by eliminating all unnecessary modelling components). These four building blocks represent the modelling components of the Arena part of the original simulation model. Therefore, the services and failures of the modules are handled by these building blocks.

The logic engine building block represents the FORTRAN subroutine of the original simulation model. Consequently, the identification of the momentary "bottleneck", the calculation of the throughput and the determination of the status of the switching logic are handled by the logic engine building block. A subtle change is that the collection of statistics about the behaviour of the model is now handled by the logic engine building block where previously it was handled by the Arena part of the original simulation model. One of the key innovations of the generic modelling methodology is the development of a more streamlined method to deal with the aforementioned tasks of the logic engine building block. In the original simulation model, these tasks are handled in a linear, sequential, step-through fashion by approximately 2 000 lines of FORTRAN programming code. The generic modelling methodology uses the so-called fraction-comparison (FC) method to handle these tasks with a matrix-based method of evaluation in less than 100 lines of programming code. A detailed explanation of the principles and functioning of the FC method does not fall within the scope of this article.

The huge reduction in the number of lines of programming code necessary to perform the tasks of the logic engine building block opens up the possibility to construct, debug and change it more easily in a strict programming language such as FORTRAN. Alternatively, the logic engine building block can now also be constructed from the standard simulation building blocks within a simulation software package without incurring the previous penalties of large model size, excessive runtimes and difficult debugging.

The model evaluation is also handled by the logic engine building block. In one of the previous sections it is indicated that the simulation model should be evaluated at predetermined time intervals. Two possible ways to handle the model evaluation are to use an iteration time interval evaluation method or an event-driven evaluation method. The basic difference between the two methods is that the iteration time interval method evaluates the model with a time interval of constant length, while the event-driven method evaluates the model with a time interval of variable length, depending on the events that take place. The logic engine building block allows the modeller to use either of the two evaluation methods, depending on the requirements of the specific application. Several factors such as accuracy, ease of use, simplicity of construction, runtimes, etc. may determine which of the two evaluation methods is best suited to a specific modelling problem. Albertyn [9] provides a comparison of the two methods and indicates their respective strengths and weaknesses.

5. NEW MODEL CONSTRUCTION

The generic "high-level" building blocks of the generic modelling methodology is used to construct two new simulation models of the continuous process plant. The new models are constructed in two different simulation software packages to clearly demonstrate the use of the methodology. The models are constructed in the Arena Standard Edition and Simul8

Standard simulation software packages respectively.[3]

The use of Arena follows logically from the fact that the original simulation model provides the point of departure for the development of the generic modelling methodology. The use of Simul8 presents a good perspective of the use of the methodology in a different simulation software package than the one in which the methodology originally evolved.

In both the Arena and the Simul8 simulation modelling environments, the first step is to develop the five generic "high-level" building blocks described in the previous section. Each "high-level" building block is constructed from several of the standard simulation building blocks or constructs available in the two simulation software packages. The ways in which the generic "high-level" building blocks manifest themselves in the two different simulation software packages differ because each software package has its own unique philosophy, conventions, logic, nomenclature, etc. This is especially true for the logic engine building block that is constructed mainly from standard simulation building blocks in the Arena environment but in the Simul8 environment, it consists primarily of a block of Visual Logic code. (Visual Logic is the logic building environment of Simul8 that allows the modeller to add very detailed logic to control the operation of the model.)

The generic "high-level" building blocks are then used to construct two identical simulation models in the two simulation software packages. The models are identical in the sense of conforming to exactly the same model definition but not in the construction of their "high-level" building blocks. Theoretical probability distributions are used to model the stochastic phenomena (Pegden *et al*. [4]). User-friendliness is enhanced by using spreadsheets to manipulate the inputs to and the outputs from both the Arena and the Simul8 models. Spreadsheets are perceived to be more accessible to modern users of computers than the input files that are used by the original simulation model.

6. EVALUATION METHOD

The original simulation model uses the iteration time interval method of evaluation with a one-hour iteration time interval. The generic modelling methodology makes provision for using either the iteration time interval or the event-driven method of evaluation. The event-driven method of evaluation is used in both the Arena and Simul8 models.

Several factors may determine which of the two evaluation methods is best suited for a specific modelling problem (Albertyn [9]). In order to give a better understanding of the main reason why the event-driven method of evaluation is used in this instance, it is necessary to introduce the concept of event density. In this context, event density may be defined as the number of events per time unit.

$$Density_{(Event)} = \frac{Number_{(Event)}}{Time} \qquad (events / hour)$$

[3] Simul8 is a simulation software package from the Simul8 Corporation. It should be noted that Simul8 is a registered trademark and usually denoted by Simul8®. However, for the sake of simplicity it will be written simply as Simul8 in this paper.

For the model definition used, there are on average 5044,6 and 5032,8 events per replication that warrant model evaluation in a simulated period of 18 months for the Arena and Simul8 models respectively. Twenty replications of a simulated period of 18 months were completed for both models. From these, the mean number of events per replication is calculated - see next section for a detailed explanation. It follows that the event density values are 0,39 events per hour for both the Arena and Simul8 models.

That is approximately one event every two hours. It implies that in this instance, an event-driven evaluation method does approximately half the number of evaluations if compared to an iteration time interval evaluation method that uses a one-hour iteration time interval. This leads to simulation runtimes for the event-driven evaluation method that are approximately half of those that will be achieved with the iteration time interval method of evaluation (Albertyn [9]). Of course, the event density cannot be calculated before a simulation run (comprising a number of replications) has been completed to supply a value for the mean number of events that warrant scrutiny. Paradoxically, this implies that the model should already exist in order to find out which is the better method of model evaluation to use in the construction of the model. This problem is overcome by making a first-order estimate of the number of events that will occur. Another major benefit of using the event-driven method of model evaluation is that it is not necessary to determine a bandwidth of iteration time intervals that would render valid results, as is the case if the iteration time interval method of evaluation is used (Albertyn [3] [9]).

7. DETERMINATION OF ADEQUATE SAMPLE SIZE

The results from a stochastic model may show variation among different replications due to the randomness of chance events such as breakage. This implies that, during the simulation run of a scenario, more than one replication has to be completed in order to find responses that are "stabilised" and representative of the simulated scenario. Kelton et al. [10] define replications as identical, independent simulation runs.

> *"Each run starts and stops the same way and uses the same input-parameter settings (that's the "identical" part), but uses separate input random numbers (that's the "independent" part) to generate* [the stochastic events like] *the inter-arrival and service times."*

The minimum sample size needed to get a mean value that is representative of the simulated scenario can be calculated with the help of an equation from Crow et al. [11] The minimum sufficient sample sizes needed are nine and ten replications for the Arena and Simul8 models respectively. Twenty replications of a simulated period of 18 months were completed for both models. From these, the mean and the standard deviation from the mean are calculated. (The output from gas production is used as the variable of comparison.) Using the standard deviation, the corresponding minimum sufficient sample size is calculated. The sample size is calculated with an allowance for a 0,5% deviation from the real-world mean and a 99% confidence interval. That corresponds to a maximum error of the estimate of 0,5% from the real-world mean or a bandwidth of the answers from the 20 replications of no more than 1% of the real-world mean value. The number of replications completed in both instances should be more than, or equal to, the minimum sufficient sample sizes calculated for the answers to be taken as representative of the simulated scenario. The required values of nine and ten

replications for the Arena and Simul8 models respectively indicate that this constraint is adhered to in both instances.

8. MODEL VERIFICATION AND VALIDATION

Various authors and manuals stress the need for comprehensive model verification and validation (Pegden et al. [4]; Kelton et al. [10]; Simul8 [12]).

The verification involves the checking of the operation of the logic in terms of the chemical processes involved and the working of the other model constructs in terms of the services and breakages. The services and breakages generated are counted and evaluated at their real-world values.

The models are validated by determining the deviation of model results from the real-world values in a known scenario. Twenty replications of a simulated period of 18 months were completed for both models. The simulation results for the Arena and Simul8 models deviate by only 0,93% and 0,94% respectively from the real-world mean value. (The output from gas production is used as the variable of comparison.)

9. MODEL COMPARISON

Table 2 compares the Arena and Simul8 models in terms of several attributes. The mean number of events per replication and the corresponding event density value are calculated from the 20 replications made. The mean gas production output and the standard deviation from the mean are also calculated. Using the standard deviation, the corresponding minimum sufficient sample size (with an allowance for a 0,5% deviation from the real-world mean value and a 99% confidence interval) is calculated. The deviation of the mean gas production output from the real-world mean value is calculated and this gives an indication of the validity of the model. The model runtime is important when a series of simulation experiments needs to be conducted and the model size can play a role in both the runtime and the transportability of the model.

Attribute	Unit	Arena Model	Simul8 Model
$n_{(Replicate)}$	-	20	20
$Time_{(Period)}$	(month)	18	18
$n_{(Event)}$	-	5044,6	5032,8
$Density_{(Event)}$	(event/h)	0,39	0,39
$Output_{(Gaspro)}$	(nm^3/h)	1354488	1354460
$StdDev_{(Gaspro)}$	(nm^3/h)	5454,9	6077,5
$n_{(Sample\ Size)}$	-	9	10
Deviation	(%)	0,93	0,94
Runtime	(min)	10,3	7,8
Size	(KB)	1785	610

Table 2: Simulation Model Comparison

Where:

$n_{(Replicate)}$	:	Number of replications in simulation run
$Time_{(Period)}$	:	Simulated period of every replication (month)
$n_{(Event)}$	:	Mean number of events per replication
$Density_{(Event)}$	:	Event density value (events per hour)
$Output_{(Gaspro)}$	:	Mean gas production output (normalised cubic metres per hour)
$StdDev_{(Gaspro)}$	:	Standard deviation of mean gas production outputs (normalised cubic metres per hour)
$n_{(Sample\ Size)}$	:	Minimum sufficient sample size
Deviation	:	Deviation from real-world mean value of gas production output(%)
Runtime	:	Runtime for 20 replications (minute)
Size	:	File size of model (kilobytes)

Table 3 compares the Arena Standard Edition and Simul8 Standard simulation software packages as used in this context. Some of the statements presented in Table 3 are subjective perceptions that follow from the use of the packages during the development of the two models.

Attribute	**Arena**	**Simul8**
Acquisition cost	13,6	1
Annual licencing fees	2,0	None
Graphics capability	More advanced	More basic
Complexity of modelling environment with respect to familiarisation and use	More complex	More simplistic
Simulation modelling capability	More capable	Adequate
Ease of use	More difficult	More easy
Accuracy	15 decimal digits	10 decimal digits
Logic programming language accessibility	Less accessible (VBA is accessible but not integral part of software)	More accessible (Visual Logic is integral part of software)
Model size	Larger	Smaller
Runtime	Longer	Shorter

Table 3: Software Comparison

Where:

VBA : Visual Basic® for Applications[4]

[4] Visual Basic® for Applications is a registered trademark of the Microsoft® Corporation.

It should be noted that the Arena acquisition cost and annual licensing fees are given as values normalised to the acquisition cost of Simul8. It should also be noted that a less expensive version of Arena, called Arena Basic Edition, is available. The acquisition cost of Arena Basic Edition is about a third of that of Simul8 and it has no annual licensing fee. It does however only allow modelling with the "*Basic Process*" template. The "*Basic Process*" template contains only the most basic building blocks and a vital omission is the ability to read data from or write data to an external file. The "*ReadWrite*" building block of Arena is contained in the "*Advanced Process*" template that is not available in Arena Basic Edition. A basic design philosophy of the generic modelling methodology is to use the most basic of the standard simulation building blocks (in the respective simulation software packages) whenever possible. This approach supports the design criteria of compact model size and short simulation runtimes. The ability to read data from or write data to an external file is seen as a basic capability that is needed to support the user-friendliness design criterion of the generic modelling methodology. Apparently the capability to read data from or write data to an external file can be achieved in Arena Basic Edition through the use of VBA code. This possibility, however, violates the single software application design criterion of the generic modelling methodology and hence it was not considered a viable option.

A variable in Arena is accurate to 15 decimal digits (that is comparable to a Double Precision or Real*8 variable defined in FORTRAN) and in Simul8, a variable is accurate to 10 decimal digits. This difference should not be of concern to a modeller in the normal applications of this type of simulation software package. However, operations where floating-point errors tend to accumulate will need extra consideration.

In a previous section, it is stated that the generic modelling methodology presents a structured approach with the following characteristics: short development time, short maintenance time, user-friendliness, short runtimes, compact size, robustness, accuracy and preferably a single software application. Both Arena and Simul8 conform to all these characteristics with only minor differences between them. In both packages, short development and maintenance times are achieved by using the "high-level" building blocks. Both packages allow hierarchical modelling (by using sub-models) and allow easy manipulation of inputs and outputs with spreadsheet files (user-friendliness). Acceptable runtimes and model sizes are achievable with both packages. The robustness of the generic modelling methodology, as well as both packages, is proven by the ease of model construction in both instances. Both packages produce accurate models (proved through verification and validation) and allow the whole model to be accommodated in a single software application.

From Tables 2 and 3, the strengths and weaknesses of the Arena and Simul8 models can be derived.

The **strengths** of the Arena model are a more advanced graphics capability and additional modelling capabilities, such as transporters, conveyors, etc. (These additional capabilities do not feature in the generic modelling methodology but could be important for users when seen in the broader perspective of general simulation modelling applications.) Arena is also more widely accepted as an "industry standard" among simulation software packages. (According to marketing material of Arena more than 75% of the top 30 companies in Fortune's Global 500 use Arena.)

The **weaknesses** of the Arena model are higher acquisition cost, annual licensing fees, a more complex modelling environment (and therefore more difficult to learn and use), no internal logic programming language, larger model size and longer runtime.

The **strengths** of the Simul8 model are lower acquisition cost, no annual licencing fees, more simplistic modelling environment (and therefore easier to learn and use), inclusion of an internal logic programming language, smaller model size and shorter runtime.

The **weaknesses** of the Simul8 model are a more basic graphics capability and less modelling capabilities. The Simul8 Standard package only provides five building blocks, but the inclusion of Visual Logic allows great modelling freedom and creativity.

10. CONCLUSIONS

It can be accepted with a high level of confidence (proved through verification and validation) that both the Arena and Simul8 models are valid representations of the real-world continuous process plant. The generic modelling methodology that is introduced and the "high-level" building blocks that are detailed in this article to model a stochastic continuous system, can thus be accepted as authentic and valid enhancements of the simulation modelling "toolbox" (to the extent that the authenticity and validity of the generic modelling methodology and the "high-level" building blocks were established by the construction of two instances of the same simulation model in two different software packages). Any one of the models (or their generic variants) can be implemented with confidence in any further projected scenario analysis and "what if" studies.

The generic modelling methodology presents a structured approach that renders high-quality models with the following characteristics: short development time, short maintenance time, user-friendliness, short runtimes, compact size, robustness, accuracy and a single software application. The methodology can be used as a point of departure for the development of simulation models in different simulation software environments.

There is no clear, distinct "better" option when the Arena and Simul8 models are compared. Both models conform to the characteristics of the generic modelling methodology stated in the previous paragraph with only minor differences between them.

The Arena Standard Edition and Simul8 Standard software packages can be compared in terms of their main strengths. The most important strengths of Arena are a more accomplished modelling capability and a wider acceptance and use in the simulation industry. These should be weighed against the strengths of Simul8, namely lower cost of ownership, simplicity of use and the inclusion of an internal logic programming language. This inclusion greatly enhances modelling freedom and creativity and together with the lower cost of ownership may just tip the scales in favour of Simul8.

It should be noted that the results presented in this article do not necessarily reflect the behaviour of systems that seem similar, but in fact have different response characteristics. Prospective modellers are cautioned against blindly accepting these results as indicative of how other systems will behave. It should be stressed that simulation modelling results can

only be accepted with a high degree of confidence after successful verification and validation of that specific model.

11. REFERENCES

[1] **Owen, R.** 1994. "Strategic Maintenance is Essential". *Mechanical Technology*, May 1994, 15-17.

[2] **Crowe, C.M**.; A.E. Hamielec; T.W. Hoffman; A.I. Johnson; D.R. Woods; and P.T. Shannon. 1971. *Chemical Plant Simulation - An Introduction to Computer-Aided Steady-State Process Analysis*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1-2, 5, 14.

[3] **Albertyn, M.** 1995. *Modelling of a Stochastic Continuous System*. *Magister* Dissertation. Department of Industrial and Systems Engineering, University of Pretoria, Pretoria, 1-78.

[4] **Pegden, C.D.; R.E. Shannon; and R.P. Sadowski.** 1990. *Introduction to Simulation Using SIMAN*. McGraw-Hill Inc., New York, 44-56, 305-308.

[5] **Bonnet, W.J.** 1991. *A Microcomputer Program for the Steady State Simulation of a Hydrogen Production Plant*. *Magister* Dissertation. Faculty of Engineering, University of Pretoria, Pretoria, 12.

[6] **Elder, M.D.** 1992. *Visual Interactive Modelling: Some guidelines for its Implementation and some Aspects of its Potential Impact on Operational Research*. *Philosopiae Doctor* Thesis. Department of Management Science, University of Strathclyde, Glasgow, 3-4, 14, 54, 242.

[7] "Simulation Fax Survey Results". 1993. *Industrial Engineering*, May 1993, vol.25, no.5, 10.

[8] **Van Dyk, L.** 2001. "The Philosophy – Tool Continuum: Providing Structure to Industrial Engineering Concepts. *South African Journal of Industrial Engineering*, May 2001, vol.12, no.2, 1-14.

[9] **Albertyn, M.** 2000. "Iteration Time Interval versus Event Driven Evaluation of a Stochastic Continuous System". In *The Proceedings of the 2000 Summer Computer Simulation Conference* (Vancouver, British Columbia, July 16-20). The Society for Computer Simulation International, San Diego, C.A., 129-134.

[10] **Kelton, W.D.; R.P. Sadowski; and D.A. Sadowski.** 1998. *Simulation with Arena*. WCB McGraw-Hill, Boston, Massachusetts, 36.

[11] **Crow, E.L.; F.A. Davis; and M.W. Maxfield.** 1960. *Statistics Manual*. Dover Publications Inc., New York, 48.

[12] *Simul8 Manual and Simulation Guide*. 1999. Simul8 Corporation, Herndon, V.A., 34.