Development of a Power Plant Simulation Model

Carlos Eduardo Vieira Batarda

Dissertação para obtenção do Grau de Mestre em

Engenharia e Gestão Industrial

Orientador: Profª Ana Paula Barbosa Póvoa

Versão Provisória

Setembro 2008

Acknowledgements

I would like to thank my supervisor, Prof. Ana Póvoa, for her advice throughout this research.

I would also like to thank the following persons:

a) Eng. João Furão at EDP, for all his contribution and kind support;

b) Dr. Martin Albertyn for his words of wisdom;

c) My family and friends for being always there.

Carlos Batarda

September 2008

Lisbon, Portugal

"There is a difference between knowing the path and walking the path."

Abstract: The system under scrutiny in this research is a thermal power plant in Sines, Portugal. The final power output of the plant is affected by random events such as equipment failures. Whenever such random events cause the power output to be below the demand of the electrical grid, it is said there is unplanned unavailability. Unplanned unavailability means shorter production and smaller profits and should therefore be minimised, which is not a trivial matter because the plant is a highly complex system. This problem can be handled by a simulation model, which may be used to identify the problem areas in the plant (bottlenecks) and to compare alternative scenarios. The simulation modelling approach has advantages when compared to other decision support tools, namely mathematical models. The generic simulation methodology that was used was developed by Albertyn (2004). A formal system description is first made and the conceptual development of the model is explained. The necessary data are collected for analysis and the model is implemented in a general programming language under the activity simulation perspective. The model is verified and validated. The model is then enhanced by including a variance-reduction technique and by using the event simulation perspective. Finally, the main results are discussed and the most significant conclusions are presented.

Keywords: Power plant, unplanned unavailability, bottleneck, generic methodology, simulation model.

Resumo: O sistema em estudo neste trabalho é uma central termoeléctrica em Sines, Portugal. A potência emitida pela central é afectada por acontecimentos aleatórios, nomeadamente falhas nos equipamentos. Quando esses acontecimentos aleatórios levam a que a potência emitida seja inferior ao pedido pela rede eléctrica, diz-se que há indisponibilidade não planeada. Indisponibilidade não planeada é sinónima de um défice de produção e portanto de lucros. A indisponibilidade não planeada deve assim ser minimizada, o que não é uma questão trivial dado que a central é um sistema altamente complexo. Este problema pode ser tratado recorrendo a um modelo de simulação, que poderá ser usado para identificar estrangulamentos no processo e para comparar alternativas. A abordagem da simulação tem vantagens relativamente a outras ferramentas de apoio à decisão, nomeadamente os modelos matemáticos. A metodologia genérica de simulação utilizada foi desenvolvida por Albertyn (2004). Começa-se por fazer uma descrição formal do sistema e explicar os principais conceitos do modelo. Os dados necessários são recolhidos e analisados, sendo o modelo implementado numa linguagem de programação genérica sob a perspectiva da actividade. O modelo é verificado e validado. Posteriormente o modelo é melhorado através da aplicação de uma técnica de redução da variância e da implementação da perspectiva do evento. Por último, discutem-se os principais resultados e apresentam-se as conclusões.

Palavras-chave: Central, indisponibilidade não planeada, estrangulamento, metodologia genérica, modelo de simulação.

TABLE OF CONTENTS

CONTENTS **PAGE**

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES AND FIGURES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The research presented in this document has its origins in the development of a simulation model of EDP's Sines power plant. The plant is operated by the national Portuguese electricity company, *Electricidade de Portugal* - EDP, and it uses thermal power obtained from coal. It is situated in Sines, Portugal.

The final electricity output throughput of the plant is affected by both chronological events – the preventive maintenance of the plant – and by random events such as equipment failures. These events compromise the ability of the plant to satisfy the demand of the electrical grid. Whenever the electricity output throughput of power plant is below the demand of the electrical grid, it is said there is unavailability. Unavailability should therefore be decreased as much as possible; however, this is not a straightforward matter since the power plant is a highly complex system. This problem can be handled by a simulation model that will allow for the description and analysis of the real system. The simulation modelling approach has some advantages when compared to other decision support tools, such as theoretical calculations and mathematical models. Simulation is better than theoretical calculations because it generally uses stochastic methods that incorporate the effect of random events into the calculations (Albertyn, 2004). On the other hand, when compared to mathematical models, simulation models offer greater flexibility in representing complex systems (Taha, 1987). Simulation also has disadvantages, namely lengthy runtimes and difficult interpretation of results, but it was concluded that it was the most suitable decision tool for this case. A further discussion of decision support tools in systems modelling is presented in chapter 2.

From a practical point of view, the simulation model can be used to identify problem areas in the plant or bottlenecks. The model also allows for proposed scenarios on the plant to be studied. Proposed scenarios could be, for instance, added capacity in the problem areas.

The work developed along this thesis follows the generic methodology proposed by Albertyn (2004), where the simulation of stochastic continuous systems was contemplated. The methodology was applied to the construction of a simulation model of a petrochemical plant. The Sines power plant also has a continuous process where stochastic aspects are present, which renders it eligible for the use of this methodology.

The model is going to be developed in a general programming language.

1.2 STRUCTURE OF THIS DOCUMENT

This dissertation report is organised in chapters. Each chapter deals with a specific part of the dissertation and is divided into smaller sections, which may or may not contain subsections. The first section of every chapter makes a short introduction to the subject of the chapter. The last section of every chapter sums up the most important conclusions reached in the chapter.

In the rest of this chapter relevant background information is presented and the problem is exposed. In Chapter 2 an investigation of decision support tools and simulation approaches is performed. Chapters 3 and 4 are the “core” of this dissertation, where all the work that was carried out is described. In Chapter 5, the most significant results of the research are discussed. In Chapter 6 the main conclusions of the dissertation are presented.

1.3 THE PROBLEM – BACKGROUND INFORMATION

The Sines power plant was built in the 1970’s, in a context of diversification of energy sources due to oil crisis. The plant is operated by the national Portuguese electricity company, *Electricidade de Portugal* – EDP. It is located in Sines, in the Southwest of Portugal.

The power plant uses coal for fuel and is composed of four identical groups of 314 MW each, making it the most powerful thermal electricity-producing center in Portugal. The normal thermal electricity generation process is employed, as represented in figure 1 in the following page.

The plant possesses a coal depot, from which the coal is sent to the coal processing unit. Each coal processing unit comprises a set of grinders that pulverise the coal into powder. The coal is then transported by a warm air current to the combustion chamber of the steam generation unit.

The combustion air is heated prior to combustion in two rotating air heaters (RAHs) which use the heat from the gas leaving the steam generator. This heat is also used for the air current transporting the coal. Insufflation of the steam generator is performed by primary air (PA) and secondary air (SA) ventilators, which feed the transportation and the combustion air respectively. Two induced-extraction ventilators (IEVs) extract the combustion gas. The PA ventilators, SA ventilators and IEVs perform what is referred to as the air-smoke circuit (CPPE, 1999).

In the boiler, the heat from burning coal is used to produce steam from water. After the steam has been expanded in the high-pressure body of the turbine, it returns to the steam generation unit where it is reheated, and goes back to the turbine to be expanded in the medium and low-pressure bodies. The calorific energy in the steam is transformed into mechanical energy by the turbine shaft and finally into electricity by the alternator (CPPE, 1999). Before the electricity is sent to the electrical grid, a transformer reduces power from 314 MW to 298 MW (the remaining 16 MW are for internal use). After the expansion, the steam flows out into a condenser. The condensed water passes through a feedwater system before entering the boiler, completing the water/steam cycle.

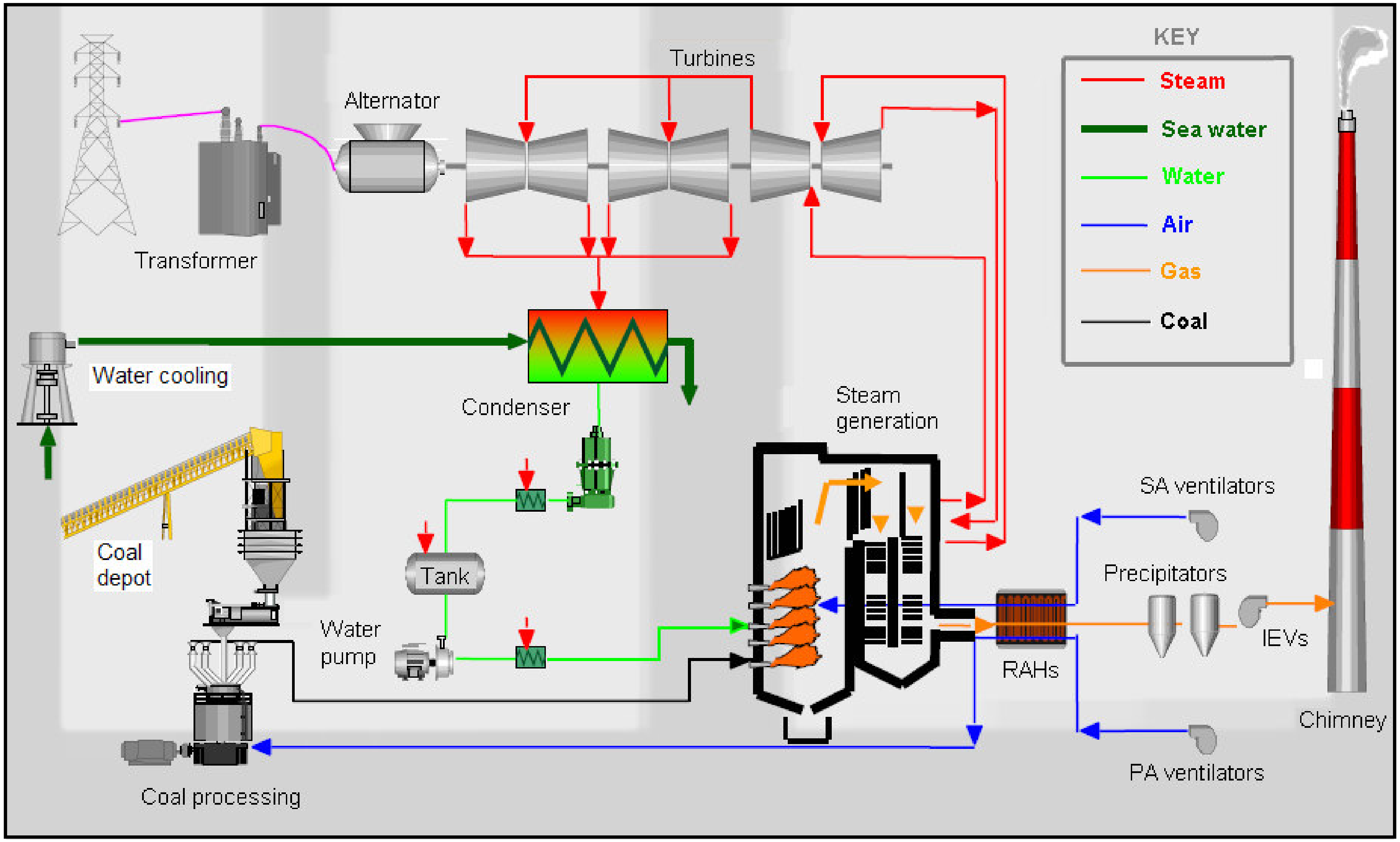

Figure 1: Electricity production process at the Sines power plant

1.4 PROBLEM DESCRIPTION

One of the performance measures of fuel-based power plants is the amount of electricity that can be generated per unit of fuel (coal in the case of the Sines power plant). This is an efficiency ratio, which translates the general need of industrial plants to produce more while consuming less. An improvement in efficiency involves changes in the manufacturing process, such as the use of new boilers for steam production, for instance. Changing the basic manufacturing process means redesigning the plant in a major way, which is by all means impossible to achieve with this kind of thesis. The increase in efficiency of the Sines power plant is therefore outside the scope of this document.

Another performance measure of power plants is unplanned unavailability. Unlike some industrial systems, which strive for maximising production, power plants produce only the amount of electricity that is demanded by the operators of the electrical grid. In the case of the Sines power plant, REN – *Rede Eléctrica Nacional*, the national Portuguese electrical grid operator, decides when and how much electricity should be produced by the power plant. The demand is almost always 298 MW, which is the maximum power output that can be emitted by the power plant.

Unavailability means that the power plant is not able to meet the grid's demand, i.e. the total power output of the plant is below what is required by REN. Unavailability could be planned or unplanned. Planned unavailability corresponds to periods of preventive maintenance. Every four years, one of the groups in the power plant stops production during six weeks to be serviced (i.e. subject to maintenance). During those six weeks the total power output of the plant is smaller because one of the groups is out of work, leading to an unavailability situation. This is planned unavailability because it is predicted and chronological. Unplanned unavailability, on the other hand, is caused by random events like failures and "peaks". Failures diminish the plant's power output and can even set it to zero temporarily. As for peaks, they occur when a given safety interval is breached. For instance, there is a safety interval for the temperature of the steam exiting the steam generation unit. If the steam's exit temperature falls out of that interval, electricity production is interrupted so the steam's temperature can be put back to within acceptable limits. That is called a "peak".

Some sections of the electricity production process may be responsible for more unplanned unavailability in the power plant than others. In other words, some sections may be more important "bottlenecks" than others. Nevertheless, the identification of the main bottlenecks is not a straightforward matter. It is possible that one section fails more often than another and still be responsible for less unplanned unavailability because it has shorter repair times. The presence of redundancy is also an influent aspect. For instance, the coal processing unit is made up of five grinders, but only four are used. If one grinder fails, there is an extra grinder to replace the one that failed, and no production will be lost. If two grinders fail, production will be lost as the total coal output throughput of the coal processing plant is only 3/4 of its normal value. A failure in the turbine, on the other hand, usually means zero power output. All these interrelationships render the identification of the bottlenecks a rather complex task.

It is convenient that unplanned unavailability stays as low as possible. Unplanned unavailability means there is a deficit of produced electricity and consequently shorter profits. According to Fontes (2007), "(Unplanned unavailability) *is the **no. 1 enemy** of thermal power plants because* (…) *it has high costs both from a financial perspective and from an availability perspective*." "Reducing unplanned unavailability thus means increasing electricity production and revenues.

A computer model will be developed to show whether a decrease in unplanned unavailability at the Sines power plant, with the consequent gain in electricity production, is feasible. The model should identify the sections of the production process that are responsible for the most unplanned unavailability (bottlenecks) and make it possible to compare alternative scenarios (for instance, added capacity at bottleneck vs. decrease of repair times at bottleneck), while accommodating the characteristics of the system previously described. This is the main goal of the research presented in this document.

1.5 CONCLUSION

In this chapter an introduction to the subject of this thesis is made. The structure of the document is presented and the problem is described.

EDP's Sines thermal power plant uses coal as primary fuel and employs the normal thermal electricity production process. Coal is burned to produce steam, which passes through a turbine. The mechanical energy generated by the rotation of the turbine shaft is then converted into electrical energy.

One of the performance measures of power plants is unplanned unavailability. A period of unavailability is a period of time when the power plant is not able to meet the demand of the electrical grid, in this case operated by REN. Unavailability can be planned or unplanned. Planned unavailability is deterministic as it is caused by preventive maintenance (services). Unplanned unavailability, on the other hand, is due to stochastic events such as failures and peaks. Some sections in the production process can act as important bottlenecks, i.e. be responsible for more unplanned unavailability than others.

The aim of this thesis is to develop a computer model of the power plant that accommodates its characteristics. The model should identify the main bottlenecks (in terms of caused unplanned unavailability) and allow for alternative scenarios to be compared.

In the next chapter, an investigation on decision support tools, on a former stage, and on different simulation modelling approaches, on a latter stage, is performed.

CHAPTER 2

APPROACH TO THE PROBLEM

2.1 INTRODUCTION

In this chapter, different approaches for solving problems of the nature described in chapter 1 are considered. In section 2.1, several decision support tools are compared in their ability to handle the problem and simulation is chosen. In section 2.2, existing approaches in the field of simulation are discussed. One of them is selected and the reasons for its choice are presented.

2.2 SIMULATION MODELLING AS A DECISION SUPPORT TOOL

Management is the art of making decisions without having all the information available (Albertyn, 2004). As leaders, managers must be able to make decisions. According to Garvin and Roberto (2001), "*Decision making is a job that lies at the very heart of leadership*".

One way to increase decision quality is by resorting to a decision support tool. The point of using decision support tools is to increase the probability that the decision made is correct, with a consequent increase in the decision cost. Figure 2 depicts this relationship for several decision support tools (Kleinschmidt, 1990):

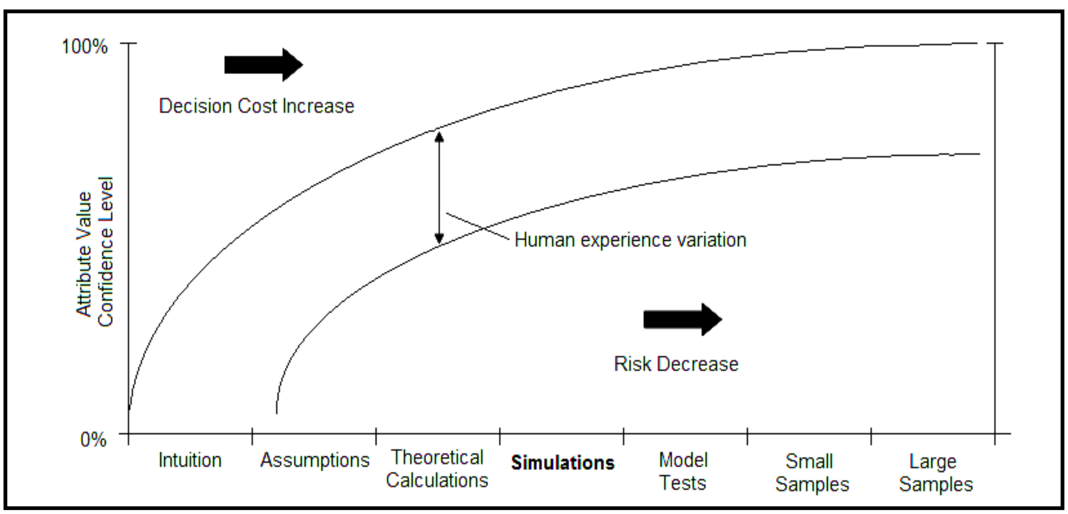

Figure 2: Decision support tool confidence level

Suppose that the value of an attribute or characteristic is relevant for a given decision. The vertical axis in figure 2 represents the confidence level of the decision tool in the determination of that value, whereas the horizontal axis represents different decision tools. The two curves represent the variation due to human experience; for example, the assumptions made by an experienced person are better estimates than the assumptions of a beginner.

Intuition, also called "gut feel", is the least confident way of making decisions. As French (1986) writes, "*It appears that unguided, intuitive decision-making is susceptible to many forms of inconsistency*". Assumptions are a slightly more accurate procedure but still very inaccurate. As for theoretical calculations, they are usually deterministic, and therefore unable to cope with the problem exposed previously as they cannot accommodate randomness. These three decision support tools are therefore excluded from the set of possible candidates to the resolution of the problem described in chapter 1.

Mathematical models lie between theoretical calculations and simulations. Mathematical models include operations research methods such as linear programming, inventory theory, etc. The main difference between mathematical models and simulation models is that, in the first case, decision variables and the objective function possess direct mathematical relations, which does not happen in simulation, where an "experimentalist" view is privileged (Tavares et al., 1996).

Perhaps the most significant advantage of simulation models over mathematical models is that the former can handle much more complex problems than the latter. Real-world problems are often impossible to represent through direct analytical relations, at least without excessive simplification. Taha (1987) supports this argument: "*Simulation models, when compared to mathematical models, do offer greater flexibility in modelling complex systems*". Hillier and Lieberman (2005) agree: "*Many problems are too complex to permit* (mathematical modelling). *Thus, simulation often provides the only practical approach to a problem*". In other cases, mathematical modelling may be viable but the analytical methods for solving the model either do not exist or are inefficient, which renders numerical experimentation (simulation) an interesting option. Another advantage is that managers usually find it harder to understand an analytical approach than a simulation approach (Tavares et al., 1996).

Model tests and samples are already in the domain of real-world experiments and samples of the actual hardware. Considered against real experimentation, simulation has advantages as well. Real experiments are more expensive than simulation, especially if something goes wrong. Secondly, even though it may take a considerable amount of time to develop a simulation model, once it is finished it allows the modeller to simulate months or years of system behaviour in seconds, something that is not feasible with experiments. Thirdly, it is rarely possible to replicate experiments in management science. Simulation circumvents this impossibility (Pidd, 2004). It could be argued that model tests and samples provide the highest confidence level possible in attribute value determination, but simulation seems to offer a better risk/cost trade-off.

All things considered, simulation emerges as the best suited approach to the problem. The most important benefits of using simulation are (Banks et al., 2005 and Porta Nova, 2000):

- It allows the study of very complex real-world systems that are difficult or impossible to model mathematically;
- It allows bottleneck analysis to be performed;

- New features, alternative configurations, etc. can be explored and compared without disrupting operations or committing resources, for existing and new systems alike;
- Insight can be obtained about the interaction of variables and their importance to the performance of the system;
- Simulation enables the modeller to manipulate the time-scale according to the necessities, by virtually speeding up or slowing down the phenomena under investigation.

Simulation also has its shortcomings (Banks et al., 2005 and Porta Nova, 2000):

- The development of simulation models is generally quite complex and time-consuming;
- Each run of the model is lengthy and produces only one observed value for each variable, which forces the modeller to run the model several times to obtain confidence intervals for those values;
- Simulation results can be difficult to interpret, because they are usually generated in a big amount, and because it can be hard to distinguish whether an observation is the result of system interrelationships or of randomness.

Still, for everything that has been discussed, it is reasonable to say that simulation is the best approach to the problem and that its benefits largely surpass its pitfalls.

2.3 SIMULATION APPROACH

2.3.1 Classification of simulation models

According to Law and Kelton (2000), simulation models may be classified in three dimensions:

- Static vs. dynamic simulation models. This dimension concerns the role of time in the model. A static simulation model represents the system at a particular point in time, thereby assuming that the influence of time in the behaviour of the system can be neglected. A dynamic simulation model, on the other hand, represents the system as it evolves over time;
- Deterministic vs. stochastic simulation models. This dimension concerns the role of randomness in the model. If a system displays no random behaviour, then it can be modelled by a deterministic model, i.e. one without probabilistic components. However, if the behaviour of the system is at least partly random (which is the case of many real-world systems), a stochastic model is required;
- Discrete vs. continuous simulation models. This dimension concerns the way in which changes in the state of the system are addressed by the model. In a discrete-event simulation model, the state variables that describe the state of the system change instantaneously at precise points in time. In a continuous model, on the other hand, state variables change continuously with respect to time. These models usually contain differential equations that define the rate of change of the state variables with time.

The classification discrete vs. continuous models follows the classification of the actual systems they represent. It should be mentioned, however, that discrete models are not always employed to represent discrete systems, and vice-versa (Law and Kelton, 2000). Also important is not to confuse the discretisation necessary to the resolution of differential equations with discrete-event simulation. According to Nutaro (2007), in a continuous system both state and time can be made discrete. When state is discrete, the differential equation is approximated by a discrete-event system. When time is discrete, the differential equation is approximated by a difference equation and its solution is calculated at fixed points in time. Time discretisation, therefore, is not *per se* indicative of a discrete model; it may be established to permit the application of numerical integration methods necessary to the resolution of the differential equations. Hence we see that even true continuous simulation is in fact somewhat discrete, but that must not be misregarded as discrete simulation. Still, because of this characteristic, continuous simulation can be performed using a discrete-event approach (Kalisz, 1993 and Nutaro, 2007).

It is easy to see that a simulation model of the Sines power plant must be dynamic and stochastic. The third dimension is not as clear. Although events such as failures occur discretely, the plant possesses a continuous process (characterised by material flow rates instead of material units). Systems with both discrete and continuous characteristics may be modelled by a *combined* simulation model. First, however, it is convenient to make an analysis of existing simulation tools that could be used to build a model of the plant.

2.3.2 Process simulation

A flowsheet is a graphical representation of a given process, typically with blocks representing operations and arrows interconnecting those blocks. Flowsheet simulators are omnipresent in the chemical industry. Actually, as Eich-Soellner et al. (1997) point out, "*The central role of flowsheets has made the term 'flowsheeting' a synonym for the simulation of chemical engineering plants.*" Flowsheet simulators fall in two basic categories: steady-state flowsheet simulators and dynamic flowsheet simulators.

Steady-state flowsheet simulators have been widely used in chemical process engineering since the 1960s. Steady-state simulators describe the process as a set of modules connected by flows of material and energy between them. The modules correspond to mass and energy balances together with physical and thermodynamic data necessary for calculations. The calculations may be performed using one of two basic techniques. The sequential approach computes modules one by one, in a direction which generally follows that of the physical flows in the system (Leiviskä, 1996). Algorithms such as those proposed by Sargent and Westerberg (1964) and Gundersen and Hertzberg (1983) help perform these calculations. The other technique is equation-oriented, where information is collected from the flowsheet and converted into a set of equations that are solved simultaneously (Shacham et al., 1982 and Stadtherr and Vegeais, 1985). There is also a hybrid approach, the simultaneous modular, which partitions a process into several modules or module clusters (Fagley and Carnahan, 1990 and Lee and Yoon, 1994). Well-known commercial steady-state flowsheet simulators are ASPEN PLUS, PRO/II, ProSimPlus and CHEMCAD. The main shortcoming of steady-state simulators is that they are static, i.e. they capture the state of the system at a precise point in time. It has been demonstrated that the state of the Sines power plant system changes over time, with random events such as failures taking place in a scale of weeks. It will thus be necessary to model the behaviour of the system for a period of one year or more, something that is difficult to achieve with a steady-state simulator.

Dynamic flowsheet simulators, on the other hand, possess the capability of system simulation over time. Like steady-state simulators, they may assume a sequential, equation-oriented or simultaneous modular approach. They normally comprise a set of ordinary and/or partial differential equations (ODEs/PDEs) which describe the state of the system, thereby making use of continuous simulation principles. Associated to this is the fact that dynamic simulators tend to be quite complex, representing the system in great detail. The focus of dynamic simulators is on the study of the system's transient behaviour, for instance during start-up and shut-down periods, system disturbances or operation under extreme conditions. Transient behaviour is the kind of behaviour verified in specific situations which last for a short time (as opposed to steady-state behaviour, which may be considered as the "general" behaviour of the system over a long period of time). Dynamic simulators have been developed namely for predictive control strategies (Ordys and Clarke, 1993 and Prasad et al., 2000), operator training (Vasandani et al., 1989) and student teaching (Diáz and Garrido, 2004 and Diáz et al., 2006).

2.3.3 The most adequate approach

It has been mentioned that the Sines power plant is a system with both continuous and discrete characteristics. Thus, apparently, combined simulation could be used to model the system. There are two main approaches to combined simulation. The first approach, proposed by Fahrland (1970), advocates the decomposition of the system into a continuous part and a discrete part, so that, during the execution of the model, either purely discrete simulation or purely continuous simulation is being performed (Cellier, 1977). An interface element, or co-simulation bus, is necessary to support the communication and synchronisation between the two parts (Gheorghe et al., 2007). The second approach, proposed by Barton and Pantelides (1994), argues that combined systems are more naturally viewed as a single physical subsystem, on which external actions are imposed in order to achieve certain objectives.

It seems as though combined simulation could be used to build a model of the Sines power plant. Under the view of Fahrland (1970), for example, the model would have a discrete part and a continuous part. The discrete part would handle discrete events such as failures and peaks (system disturbances), while the continuous part would study the detailed effect of those events on the transient behaviour of the system, for instance using a dynamic flowsheet simulator. However, there are two reasons for not including transient behaviour in the model:

a) One of the goals of this work is to study unavailability in the power plant. Unavailability depends on preventive and corrective maintenance, which take place in a scale of weeks. Therefore it will probably be necessary to run the model for a period of one year or more to gain an understanding of unavailability in the system and identify the bottlenecks. Hence the focus of the model should be on the output throughput of the power plant and the effect of random events, for instance, on that throughput rather than on the detailed characterisation of the system's behaviour during disturbances (transient behaviour). According to Law and Kelton (2000), it must not be included more detail in the model than is necessary to address the issues of interest;
b) In power plants, transient behaviour is essentially associated to the change in the power output in the time periods immediately after unavailability situations (e.g. after a failure is repaired, the power output of the plant progressively increases until it reaches the projected value). Power plants strive for maintaining the projected power output and thus the duration of such time periods is kept to a minimum. The cumulative time spent in transient behaviour is therefore a very small percentage of total production time. That tends to negate the effect of transient behaviour.

These reasons suffice to exclude transient behaviour, and consequently continuous simulation, from the model. A discrete-event simulation approach is better suited to the expected level of model detail. Nevertheless, it must not be forgotten that the system to be modelled (the Sines power plant) is characterised by a continuous process and that should be accommodated by the model.

However, the majority of important discrete-event simulation books and software packages focus on purely discrete systems. Pidd (2004) notes that "(…) *management scientists seem to be more often concerned with systems that can satisfactorily be simulated discretely*". As a result, discrete-event simulation books spend very little time with continuous systems, because they are interpreted as necessarily requiring continuous simulation. Law and Kelton (2000), for instance, dedicate about 0,4% of their book (3 pages in 705 pages) to continuous systems. Pidd performs approximately the same (1 page in 295 pages). This phenomenon occurs with other discrete-event simulation books.

Albertyn (2004) shares such view on his work. This propelled him to develop a generic discrete-event methodology for the simulation of stochastic continuous systems. The methodology is applied to the construction of a simulation model of a petrochemical plant where an identification of the main bottlenecks is made and an analysis of alternative scenarios is performed, which corresponds to the aim of a simulation model of the Sines power plant. Although the model was specific of a petrochemical plant, the methodology is sufficiently generic to permit its application to other systems with similar characteristics to the petrochemical plant. Those characteristics are the presence of a continuous process, the presence of failures (stochastic events) and services (preventive maintenance or deterministic events) and the presence of complex interrelationships. The Sines power plant has been shown to exhibit all these characteristics, which renders it eligible for the use of the generic methodology. Furthermore, simulation models that are developed with the generic methodology possess beneficial design characteristics, namely (Albertyn, 2004):

a) Short model development and model maintenance times;
b) Short simulation runtimes;
c) Robust modelling ability;
d) Accurate modelling ability.

The generic methodology is thus adequate to the resolution of the problem focused in this research. The methodology seems to be able to handle the problem in a more satisfactory manner than any other approach that was presented. In the light of the results of the literature research presented in this chapter, it was decided to use the generic methodology developed by Albertyn (2004) to build a simulation model of the Sines power plant. The explanation of the methodology will not be done here, but together with the work description, as the two are deeply connected. Work description is made in chapters 3 and 4.

2.4 CONCLUSION

In this chapter a critical research on possible approaches to the problem was performed. In section 2.2 simulation is compared to other decision support tools. Theoretical calculations do not make provision for random behaviour, while mathematical models often cannot adequately accommodate complex systems. Real-world experimentation also has evident disadvantages when compared to simulation.

In section 2.3 different simulation approaches were presented. Simulation models can be static or dynamic, deterministic or stochastic and discrete or continuous. The Sines power plant is a system with dynamic, stochastic and discrete as well as continuous characteristics. In continuous process simulation, which is essentially related to chemical processes, there are steady-state simulators and dynamic simulators. Steady-state simulators are static and therefore are not a good option. Dynamic simulators could be an option to build a combined simulation model of the Sines power plant, with a discrete part and a continuous part for transient behaviour modelling. However, when the objectives of the present research are taken into account, the effect of transient behaviour in a system like the Sines power plant does not appear significant. Thus a purely discrete-event model seems more appropriate. Still, the Sines power plant has a continuous process and this should be accurately represented by the model. Albertyn (2004) developed a generic methodology for the simulation of systems with characteristics exhibited by the Sines power plant – the presence of a continuous process, stochastic and deterministic events and complex interrelationships. The generic methodology developed by Albertyn (2004) is selected as the basis for this work. Simulation models developed with this methodology show beneficial design characteristics.

CHAPTER 3

MODEL CONCEPTUALISATION

3.1 INTRODUCTION

In this chapter, the conceptual development of the proposed simulation model of the Sines power plant is thoroughly explained.

Since this research is a simulation study, it is should follow the main steps usually identified in the realisation of such studies. Porta Nova (2000) presents these steps in a flowchart:

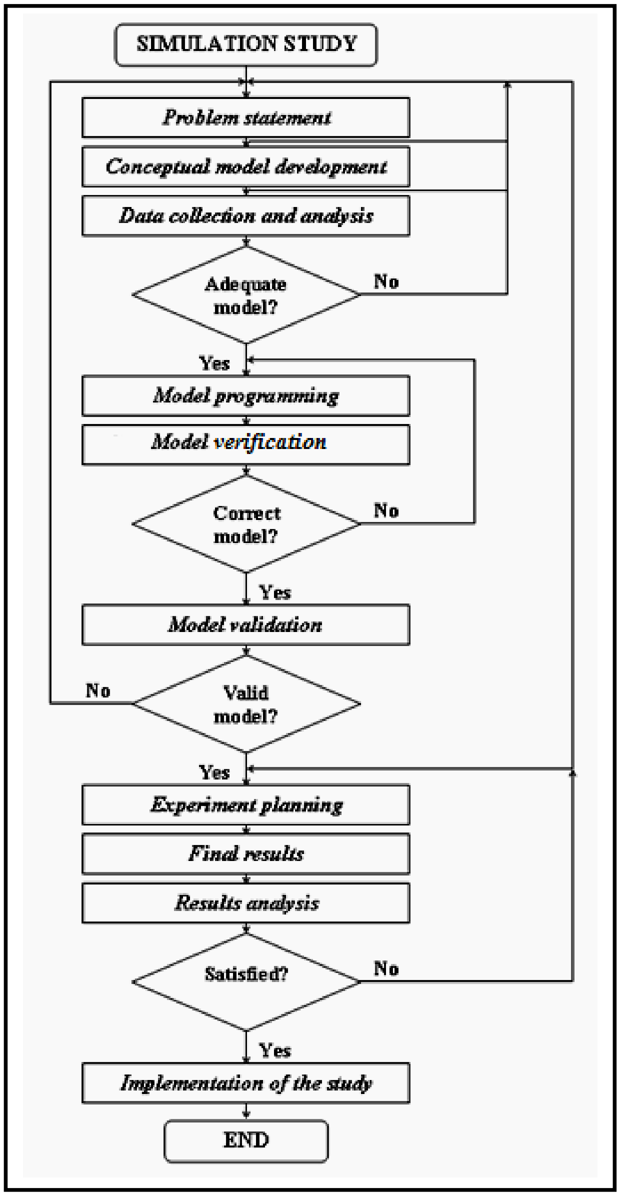

Figure 3: Steps of a simulation study

In figure 3, each step is contained in a rectangle. The start and end of the process are marked by round-cornered rectangles. Rhombuses represent conditions that must be met in order to advance in the algorithm. This implies that the development of a simulation study is not unidirectional, but is rather an iterative process, where it may be necessary to return to previous steps, depending on the verification of the conditions in the rhombuses (Porta Nova, 2000).

To put it shortly, the most important steps in a simulation study are:

1) Problem statement;
2) Conceptual model development;
3) Data collection and analysis;
4) Model "programming" using an adequate language or software;
5) Simulation model verification;
6) Simulation model validation;
7) Experiment planning;
8) Analysis of the results;
9) Implementation of the study.

Steps 7 and 9 were not performed in this research, while step 1 has already been performed. All the other steps are covered in the following sections and in chapter 4.

In this chapter the simulation model is conceptualised, which corresponds to the second step in a simulation study. It is the step where theoretical knowledge is applied to the definition of a set of methods and techniques that allow the treatment and modelling of the system under study. The term "method" is considered to describe a more accomplished procedure with a broader range of application, while the term "technique" is considered to describe a less accomplished procedure with a more restricted range of application (Albertyn, 2004).

Before the methods and techniques of the model can be developed, it is necessary to analyse the characteristics of the system and to describe the system in a way that permits its easier treatment by the model. Section 3.2 deals with these aspects. The methods and techniques used in the simulation model are then properly presented in section 3.3.

3.2 THE SYSTEM

3.2.1 System characteristics

Before endeavouring to develop a model of any sort it is essential to fully understand the characteristics of the system under scrutiny. The research of Albertyn (2004) focused on a specific kind of systems:

a) The systems are continuous process systems;

b) The systems are subject to two types of discrete events – chronological events (services) and stochastic events (failures);
c) The systems have complex interrelationships.

The Sines power plant is undoubtedly a continuous process system. This means that the different raw materials, such as coal or water, as well as the steam and electricity that are produced, constitute continuous flows, measured in terms of mass or volume per time. The continuous flows in the Sines power plant will henceforth be referred to as "commodities" and the Sines power plant itself will be referred to as "power plant" for greater simplicity.

The complex interrelationships are manifested in the process flow of the plant. The complexities are present in the power plant due to the fact that, in virtue of the sequential nature of the process, each part of it is deeply connected to the other parts as far as events like failures are concerned. A failure in one given piece of equipment has a significant effect on both upstream and downstream operations.

As far as characteristic b) is concerned, the power plant is slightly different from the systems focused by Albertyn (2004). Preventive maintenance, i.e. the chronological services mentioned previously, is, at least in theory, performed every four years for one production group, during a period of six weeks. The services are staggered in time to minimize the impact on production (i.e. if group 1 is serviced in year *n*, then group 2 is serviced in year *n*+1, group 3 in year *n*+2, etc.) It is said "in theory" because in recent years preventive maintenance has been somewhat irregular, namely due to the installation of dessulphuration systems (one per group), which has not been done for all groups simultaneously. This is one difference with respect to the situation in Albertyn's work, where services follow clearly defined cycles. A second difference resides in the fact that, in the systems focused by Albertyn's research, each group of equipments has its own service cycle (services vary from equipment to equipment), with a particular length and duration, whilst in the power plant, one whole group is serviced in the six week interruption; that is, all equipments inside a group undergo the same service cycle. The manner in which preventive maintenance was handled by the simulation model will be explained in section 4.4. As explained in chapter 1, preventive maintenance is responsible for planned unavailability, while events such as failures are responsible for unplanned unavailability. From now on the terms "lost production", "bottleneck time", "power unavailability" or "energy unavailability" (the meaning of these terms will be uncovered in the following sections) are referent to **unplanned** unavailability only.

Failures are very important in the generic methodology because, like services, they affect the power output of the power plant. Given that the focus of this research, in the case of the power plant, is on unavailability and in particular on unplanned unavailability, failures must play a key role in the model. It seems important, therefore, to study the failure characteristics of the equipments in the power plant.

Only failures that may result in unavailability (referred to as “relevant” failures), i.e. power output lower than the demand, are relevant for the simulation model. Failures that do not cause unavailability have no effect in the power output and therefore are not considered. A preliminary analysis to the data that was collected (see section 4.2) indicated that there are quite a number of different systems and equipments in the power plant whose failure can be relevant. The maintenance division of the plant does not record failures of individual components, but groups those components into smaller systems. Still, the systems prone to relevant failure are very diverse. Figure A1 in Appendix A shows this diversity.

Figure A1 displays the electricity production process in the power plant through a block-diagram. All the systems whose failure is relevant are represented in the figure. The different systems are designated by their *KKS* code (see figure A2 in Appendix A for more information). An arrow indicates the flow of a commodity, say coal or steam, while a line indicates a functional connection. The larger blocks represent the main systems and are interconnected by arrows because commodities flow between them, meaning they intervene directly in the production process. The smaller blocks represent auxiliary systems and are connected between themselves and the main systems by lines, meaning they do not intervene directly in the production process but perform functions necessary to its completion.

A simulation model characterising the failures of all the systems and subsystems depicted in figure A1 would be a rather detailed one. It is arguable that such a level of model detail is required for an accurate representation of the system – that is, for the time being, there is no evidence showing that a less detailed model cannot handle the system satisfactorily. Law and Kelton (2000) have some words of advice concerning the initial level of model detail: “*It is rarely necessary to have a one-on-one correspondence between each element of the system and each element of the model. (…) We recommend starting with a ‘moderately detailed’ model, which can later be embellished if needed.*”

Therefore, it was decided to group the smaller equipments and systems into entities called “smaller plants” (so as to distinguish them from the power plant). Smaller plants correspond to functional sections of the electricity production process and were established and named according to the specific task they perform. Figure A3 in Appendix A shows the systems grouped into smaller plants. Each smaller plant is represented by a bold typeface “box” that comprises the main and auxiliary systems in that smaller plant. A total of eight smaller plants were established: coal processing, steam generation, turbine, electricity generation, water cooling, condenser, feedwater and air-smoke.

The validity of this grouping procedure can only be considered reasonable after the final model is built and its results are compared to real-world data (see section 4.4). Nevertheless, as Law and Kelton point out, it is more reasonable to begin with a simpler model at first, which can then be made more complex if necessary, than starting off with a very detailed model that might not have been needed in the first place.

3.2.2 System description

After the plants have been established, it is now possible to make a first description of the power plant as a system. Harrell and Tumay (1999) propose a hierarchical way of making a description of a system. A schematic representation is provided by figure 4.

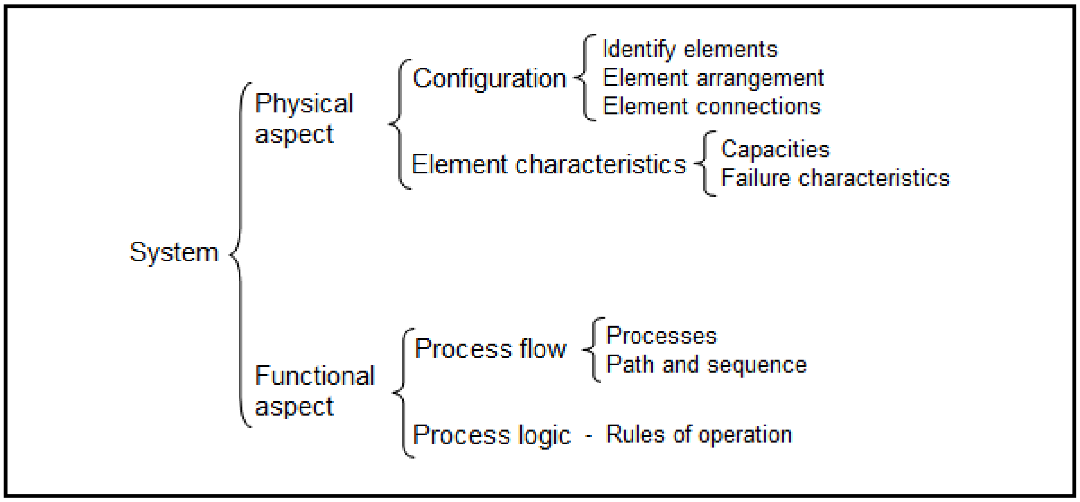

Figure 4: System description breakdown

The physical aspect in the description of a system regards the configuration of the elements and their characteristics. The configuration of the elements is the way they are arranged and connected. In the case of the power plant, the elements are the smaller plants and their relevant characteristics for this matter are their capacities and failure characteristics.

As for the functional aspect in the description of a system, it regards both the process flow and the process logic of the system. The process flow describes the processes undergone by the commodities and the path or sequence of those processes. The process logic details the rules of operation of a system. For instance, if element I supplies commodity *a* to elements II and III, then a rule of operation could be that element III is to be supplied only when the capacity of element II is surpassed.

The hierarchical structure proposed by Harrell and Tumay (1999) shall be followed when describing the power plant system. A visual representation of the configuration and the process flow of the power plant is given by figure 5 in the following page. In figure 5 the grouping of systems into smaller plants is already considered.

Through inspection to figure 5 one can already take conclusions about the configuration and process flow of the power plant. It is possible to visualise the arrangement of the elements (the smaller plants) and to follow the nature and sequence of the operations that each commodity in the system is subject to. However, figure 5 does not give information on element characteristics.

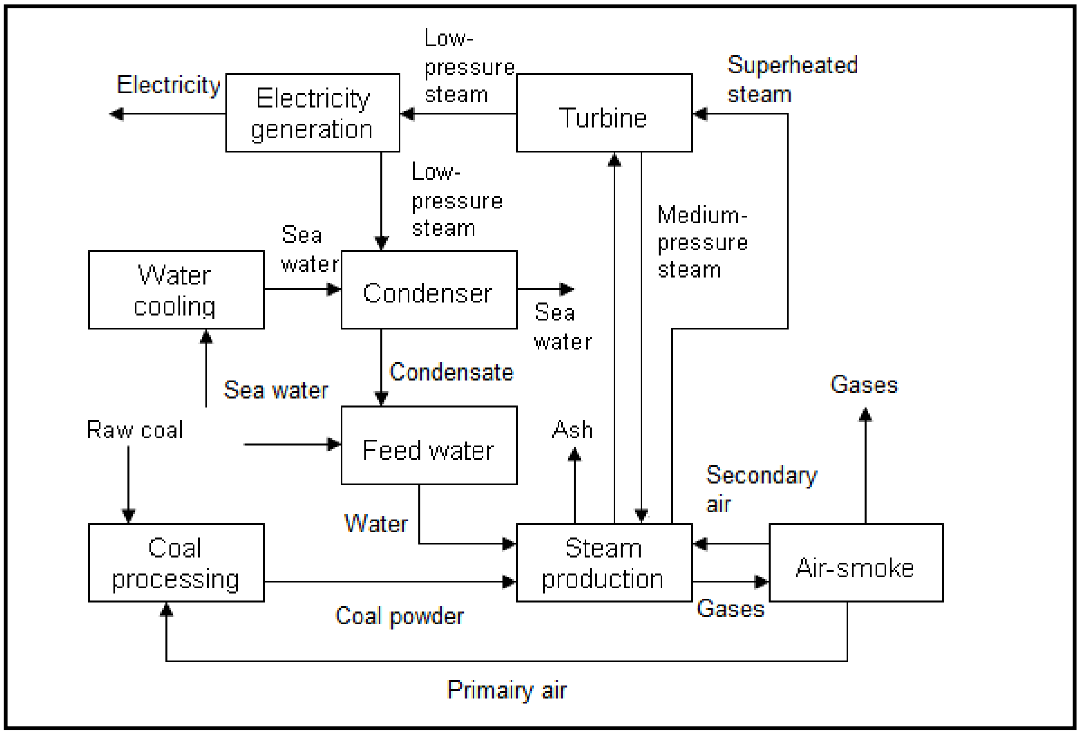

Figure 5: Schematic representation of the process

A more formal way to combine and present this information, suggested by Albertyn (2004) is to do it in a table such as table A1 in Appendix B. Table A1 is a very succinct manner to describe the configuration and process flow of the system. Each line of the table corresponds to one smaller plant, identified by a number and the corresponding name.

The third column in the table indicates the number of modules in the smaller plant. One module corresponds to one "replication" of the smaller plant. For instance, there are five coal grinders in each production group of the power plant; therefore, it is said that the coal processing plant has five modules. The modules obey to the same component-grouping principle of the smaller plants; that is, in the coal processing plant, for instance, each module has a bunker system, a feeder system and a pulverizing system, as indicated by figure A3 in Appendix A. Figure A3 may be used to verify the physical constitution of each module, as far as the systems prone to relevant failure are concerned (besides the systems represented in figure A3, each smaller plant includes all the other systems necessary to its functioning. Thus, the coal processing plant includes all the systems necessary to process the coal, the steam generation plant includes all the systems necessary to generate steam, etc.).

The final four columns of the table are very useful in that they provide a picture of the process flow of the power plant in a simple and effective manner. The fourth column lists the commodities that flow into each smaller plant module and the respective input capacity, while the fifth column indicates the smaller plant from which those commodities flow. The final two columns do exactly the same for the outward flow. As an example, the first line in table A1 would read thus: smaller plant no. 1 is the coal processing plant and it possesses five modules. Each module has an input capacity of 29,07 ton/h of raw coal, which come from no smaller plant in particular (it is fresh), and of 92,61 ton/h of primary air, which come from plant no. 8 (air-smoke plant). Finally, each module has an output capacity of 29,07 ton/h of powder (grinded coal), which flow into smaller plant no. 2 (steam generation plant). This logic is identical for all other smaller plants and therefore it will not be necessary to make further description.

In the case of electricity generation plant, which includes both the alternator and the transformer, the input would be mechanical energy, but it appeared somewhat strange to refer to mechanical energy in the same way as a commodity, and therefore it was considered that low-pressure steam flows from the turbine to the electricity generation plant. Obviously this has no effect in the final results of the model. The final output of the electricity generation plant is 298 MW and not 314 MW, because that is the amount of power that is actually emitted to the electrical grid. Therefore, whenever the term “electricity production” is used throughout this document, in fact it designates the amount of electricity that is emitted and not produced.

The feedwater plant presents a particularity. The feedwater plant is composed of a water turbo-pump and auxiliary equipments, namely an auxiliary turbine. The turbo-pump and auxiliary equipments make up one module of the feedwater plant. When the module fails (i.e. when the pump or an auxiliary system fails), it is replaced by two electrical water pumps, which consume 10 MW of electrical power. The 10 MW are supplied by the electricity generation plant. This situation is a characteristic of the process logic and may thus be considered as a rule of operation of the system, according to figure 4. The indication in table A1 that the feedwater plant possesses two backup modules is part of a procedure of the simulation model to accommodate this rule of operation; this procedure will be detailed in section 4.3.

It must not be forgotten that the power plant possesses four identical production groups and that the system description that has been made is valid for each of those groups individually. It is assumed that the four groups are identical (i.e. that they are all described by the same system description) and that the conclusions reached with the model for one group are extendable to the whole power plant. The validity of this assumption is supported in section 4.2 with the validation of the model.

There is one other element characteristic that has not yet been described, which is failures. These are addressed in section 4.2. However, besides plant failures, there are other events in the power plant that affect its final output throughput. Analysis of failure data revealed that there are situations when the plant’s control system may have to interrupt production due to a variety of reasons. In other situations, a failure external to the power plant occurs; a very common example is a technical problem in the electrical grid. A third kind of situations that is not a smaller plant failure but disturbs production as well is the previously mentioned peaks (see section 1.4). These “special” events – control actions, external failures and peaks – do not correspond to a failure of one the smaller plants but have an impact on the final output throughput of the power plant and therefore must be accounted for by the model. The manner in which they are handled by the model and combined with the system description that was performed is thoroughly explained in the following section.

3.3 METHODS AND TECHNIQUES OF THE SIMULATION MODEL

In this section the methods and techniques used by the simulation model are presented. All of them are featured in the generic methodology developed by Albertyn (2004). Therefore his work will not be referenced repeatedly throughout the section for the sake of the readability of the document. An analysis to the impact of the system characteristics is first made.

3.3.1 Implications of the characteristics

The system characteristics mentioned in subsection 3.2.1, namely characteristics a) and c), have implications in a simulation model of the power plant.

According to Albertyn (2004), continuous flows may be problematic if the simulation model is built in a in a discrete-event simulation software package (which was the case in Albertyn's research), as most software packages (i.e. those oriented to discrete simulation, not chemical flowsheet simulators) are oriented to the modelling of discrete quantities. Normally each individual unit in the system is represented by an entity in the software package; the problem is that continuous flows cannot be decomposed into individual units.

Albertyn (2004) mentions two techniques, proposed by Harrell and Tumay (1999), to overcome this difficulty. One technique suggests the conversion of the commodity flows into sets of entities or "packages" in the software package. For instance, if the output throughput of the water cooling plant is 36.000.000 m^3/h of sea water, then, according to this technique, the flow of water could be converted into, say, 36.000 entities, if each entity is considered to represent a "package" of 1000 m^3/h of water. The other technique, referred to as variables technique, suggests simply the representation of the flows with variables. Albertyn (2004) uses the variables technique because the first one poses accuracy problems – accuracy in determining the flow rate cannot be greater than the size of each "package" – and runtime problems – simulating the flow requires the generation of one event per "package" or, in the case of the water flow alone, 36.000 events per hour of simulated time, which hugely increases runtime. None of these difficulties arise with the variables technique.

Additionally, the first technique is only worthy of consideration if a simulation software package is used. In this research, the simulation model will be implemented in a general programming language (see section 4.3); therefore, the variables technique, besides being the most accomplished of the two presented techniques, is also the only which makes sense.

In this document the term “output throughput” designates the flow rate of some commodity coming out of a given smaller plant (for instance, in the previous example the output throughput of the water cooling plant is 36.000.000 m^3/h). When the discussion involves the final electricity output throughput of the power plant, the term “power output” (which has been used previously) may be used as a synonym, since an electricity flow rate corresponds to a power output.

The system is also characterised by complex interrelationships. The fact that the electricity production process is made up of different steps, each performed in a different smaller plant, means that all plants are intrinsically connected as far as events such as failures are concerned. A failure in one smaller plant has an immediate effect both on its output throughput as well as in the output throughput of all the other smaller plants. The dimension of this effect is not simple to assess. A failure in the steam production plant causes the plant to be unable to produce any steam whatsoever because the plant possesses only one module (see table A1 in Appendix B). That means that all the remaining plants are temporarily switched off. If, however, the failure happened in the air-smoke plant, this plant can still handle half of its normal output throughput because it has two modules. Then all the other plants are also operating at half-load. This could mean that some modules are switched off while others are switched on. In the steam production plant, for instance, the single existing module must be switched on (albeit with a output throughput that is smaller than its normal value), but on the coal processing plant two modules may be switched off, as in normal conditions four modules are used. On the other hand, if two modules of the coal processing plant happen to be under repair after failure, then only one of the available modules may switched off (the plant has five modules).

Furthermore there are the “special” situations to be considered. The “special” situations correspond either to an action of the control system, a peak or an external failure. When one of these situations takes place, production is interrupted and all the modules in the plants are switched off, except for those that are being repaired.

The main goal of this work, as stated in section 1.4, is the development of a simulation model to study unplanned unavailability at the power plant. Unavailability, in terms of power, is defined as the difference between the power demand of the operator of the electrical grid (REN) and the power output of the power plant. Even though the power demand is not constant, it is almost always 298 MW, which is the designed capacity of the power plant. This also means that the power demand never exceeds the production capacity of the power plant, which implies that unavailability is never due to lack of installed capacity, but rather due to a decrease in the power output of the power plant (for instance because of a failure).

One can therefore assume that the power demand has a constant value of 298 MW and write

$$
\begin{aligned}
Unavailability_{Power}(t) &= Demand_{Power}(t) - Throughput_{PowPlt}(t) = \\
&= 298(MW) - Throughput_{PowPlt}(t)
\end{aligned}
$$

(Eq. 1)

Where:

$Unavailability_{Power}(t)$ – The power unavailability at the power plant, as a function of time, in MW.

$Demand_{Power}(t)$ – The power demand of the electrical grid, as a function of time, in MW.

$Throughput_{PowPlt}(t)$ – The power output of the power plant, as a function of time, in MW.

If energy unavailability over a period of time Δ*t* is desired, one has simply to multiply power unavailability by the period of time during which power unavailability remained constant, because energy equals power times period of time. If power unavailability varied during the time period, then equations 2 and 3 are applicable:

$$Unavailability_{Energy}(\Delta t) = \sum_{i=1}^{n} Unavailability_{Power}(\Delta t_i)\Delta t_i \qquad \text{(Eq. 2)}$$

$$\Delta t = \sum_{i=1}^{n} \Delta t_i \qquad \text{(Eq. 3)}$$

Where:

$Unavailability_{Energy}(\Delta t)$ – The energy unavailability at the power plant during time period Δ*t*, in MWh.

$Unavailability_{Power}(\Delta t_i)$ – The power unavailability at the power plant during time period Δt_i, in MW.

Δt and Δt_i – The total time period and the i^{th} time period during which power unavailability remained constant, respectively, in hours.

n – The number of time periods.

Equation 1 shows that the power output throughput of the power plant, as a function of time, needs to be known in order to determine the value of unavailability. The power output throughput function does not have an analytical form because of the randomness associated to failures, peaks, etc. Therefore, a method is required that can determine the power output throughput of the power plant as a function of such events. That method is the subject of the next subsection.

3.3.2 The ERM method

The actual power output of the power plant is a function of time because the modules in the smaller plants are subject to failures, which affect the availability of the modules. This has a direct impact on the actual output throughput of the smaller plants, which in turn determines the actual output throughput of the power plant. To know the actual output throughput of the power plant it is thus necessary to know the actual output throughput of each of the smaller plants, or, in other words, the actual output throughput of the power plant is a function of the actual output throughput of each of the smaller plants.

The actual output throughput of each of the smaller plants depends on the maximum possible output throughput of each of the smaller plants (this concept will be clarified in the next subsection). The maximum possible output throughput of a plant is the number of available modules in the plant multiplied by the capacity of each module:

$$Throughput_{PltMaxPos}(t) = (n_{PltModAv}(t))(Capacity_{PltMod}) \qquad \text{(Eq. 4)}$$

Where:

$Throughput_{PltMaxPos}(t)$ – The maximum possible output throughput of the smaller plant, as a function of time, in the appropriate flow measure unit.

$n_{PltModAv}(t)$ – The number of available modules in the smaller plant, as a function of time.

$Capacity_{PltMod}$ – The capacity of each module in the smaller plant, as a constant, in the appropriate flow measure unit.

Equation 4 is only true because all the modules in each smaller plant have the same input and output capacities and because they operate in parallel rather than in sequence. Equation 4 also indicates that the number of available modules in each smaller plant is a function of time because it depends on failures, which are themselves time dependent. More specifically, the number of available modules in a given smaller plant is the number of modules of the plant minus the number of modules being repaired after failure:

$$n_{PltModAv}(t) = n_{PltMod} - n_{PltModFail}(t) \qquad \text{(Eq. 5)}$$

Where:

n_{PltMod} – The total number of modules of the smaller plant, as a constant.

$n_{PltModFail}(t)$ – The number of modules of the smaller plant that are being repaired after failure, as a function of time.

The previous equations are also applicable when considering the input throughput of the smaller plants, but it is more common to express the throughput as the output throughput. Each smaller plant has several throughput values, each value corresponding to an output commodity. The set of throughput values of a smaller plant is referred to as the throughput vector of that smaller plant. The throughput vectors of all the smaller plants constitute the throughput vector of the power plant. If the values of the input throughputs of raw coal in the coal processing plant, sea water in the water cooling plant and fresh water in the feedwater plant are known, then the throughput vector of the power plant can be determined at a given time *t*.

The simulation model must also determine the number of modules that are switched on and switched off in each smaller plant. The number of modules that are switched on in a smaller plant is the actual output throughput of the plant divided by the capacity of each module of the plant. The ceiling of if this ratio must be computed because the ratio might result in a non-integer number:

$$n_{PltModOn}(t) = Ceiling(\frac{Throughput_{PltAct}(t)}{Capacity_{PltMod}}) \qquad \text{(Eq. 6)}$$

Where:

$n_{PltModOn}(t)$ – The number of modules that are switched on in the smaller plant, as a function of time.

$Throughput_{PltAct}(t)$ – The actual output throughput of the smaller plant, as a function of time, in the appropriate flow measure unit.

$Capacity_{PltMod}$ – The output capacity of each module of the smaller plant, as a constant, in the appropriate flow measure unit.

The information about the number of modules that are switched on permits the determination of the number of modules that are switched off. The number of modules that are switched off in a smaller plant equals the number of available modules minus the number of modules that are switched on:

$$n_{PltModOff}(t) = n_{PltModAv}(t) - n_{PltModOn}(t) \qquad \text{(Eq. 7)}$$

Where:

$n_{PltModOff}(t)$ – The number of modules that are switched off in the smaller plant, as a function of time.

So far this discussion this discussion has been focused in the establishment of equations characterising the behaviour of the modules that are necessary to the operation of the ERM method.

ERM stands for Entity Represent Module. The method gets its name from the fact that, when applied in the context of a simulation software package, it uses entities to represent the modules. Most simulation software packages feature basic building blocks as well as entities. The building blocks, which have names like *Servers* or *Work Centers*, are usually employed to model a physical location in a plant where tasks are performed on the units being processed. The units being processed are the entities. The ERM method uses entities to represent the modules in the smaller plants, which is a counter-intuitive procedure because, in the real world, modules are where "work" takes place (in the case of the power plant, "work" corresponds to the physico-chemical processes that the commodities are subject to). Thus, in a conventional simulation method, modules would be modelled by building blocks. By modelling modules with entities instead of building blocks, the size of the simulation model is considerably reduced and manipulation of the modules becomes easier.

In this research, the simulation model will be implemented in Mathematica 5 (see section 4.3), which is a mathematics-oriented programming language but may be regarded, for this matter, as a general purpose language. Therefore the benefits of the ERM method mentioned above will not be directly experienced, as Mathematica is not a specific simulation software package. Still, the ERM method allows for a simple and effective manipulation of the modules and the information pertaining the modules for the purposes of the simulation model, and was thus used in this research. The name of the method, *Entity Represent*, will lose its real significance but it was decided to keep it anyway.

From the previous discussion it is possible to conclude that the behaviour of the modules in the smaller plants of the power plant is characterised by three possible states:

a) On (available and switched on);
b) Off (available and switched off);
c) Failure (failed and being repaired).

The aim of the ERM method is the determination of the state of the modules in each smaller plant, at any given moment in time. To reach that aim, it advocates the construction of two parts in each smaller plant – the part responsible for the available modules, which will be designated as the availability part, and the part responsible for the modules being repaired after failure, which will be designated as the repair part.

Before the start of a simulation run, each smaller plant is populated with the corresponding number of modules. In every smaller plant, all modules are placed on the availability part. The repair part is therefore empty. Each module is attributed a next-failure time, which indicates the instant of time when the module is going to fail, and a next-repair time, which indicates the duration of the repair of the next failure (i.e. the next repair). Both the next-failure time and the next-repair duration are sampled from theoretical probability distributions (see section 4.2).

As the simulation progresses in time, the model checks for the occurrence of scheduled failures or finished repairs of the modules in the smaller plants. The manner in which the model handles this procedure depends on the simulation perspective that is selected. The simulation perspective is detailed in section 4.3. First the model verifies if any modules have finished being repaired. If so, the corresponding modules are attributed new next-failure and next-repair times, because they have been restored to an approximately "as good as new" configuration. The modules are then moved from the repair part to the availability part and the number of repairs in the smaller plant, which will be later used for verification of the simulation model, is incremented by one. Afterwards the model verifies if any modules have failed. If so, the corresponding modules are moved from the availability part to the repair part, where they must be delayed by a time period that is at least equal to the next-repair time of the modules. The actual delay of a module in the repair part may be longer than its next-repair time because other modules may already be under repair. The repair part therefore works like a queue with FCFS (first-come-first-served) logic.

The ERM method can thus calculate the number of available modules in each smaller plant at any given moment in time. This is vital information for the determination of the maximum possible throughput of each smaller plant, which influences the final power output throughput of the power plant, as will be explained in the next subsection.

Figure 6 is a schematic representation of the ERM method:

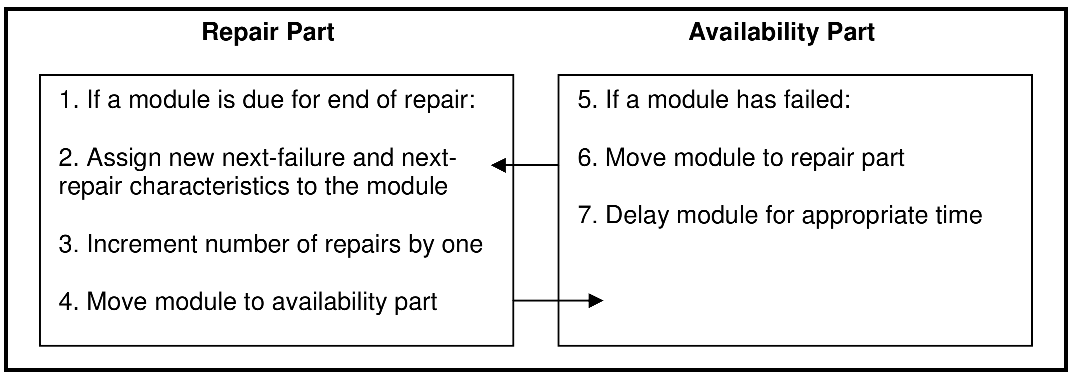

Figure 6: The ERM method

These concepts will be addressed in more detail in section 4.3.

It is true that several smaller plants in the power plant have but one module. These plants act as “switches” in the system, where one failure causes the interruption of production (zero output throughput of the power plant or total unavailability). One could argue that perhaps a simpler representation could be devised for these smaller plants as no module queues are ever formed in them during a simulation run. Such a technique would reduce the amount of computer memory used by the simulation model. This is not done for three reasons:

a) The amount of memory saved by this procedure would be insignificant as it would only spare a few variables in some plants, and even “big” Mathematica variables like lists or matrices occupy very little memory;
b) The method would lose generality. Among other things, it could become difficult or even impossible to evaluate the effect of an increase in the number of modules on the behaviour of the smaller plant;
c) A standardisation principle is preserved.

The ERM method can thus handle all smaller plants but, apparently, it cannot handle the three kinds of “special” situations – actions of the control system, external failures and peaks – because they are not smaller plants. Nonetheless, through analysis of the failure data, it was possible to observe that every time a “special” situation occurred, production was interrupted. Therefore the “special” situations, similarly to single-module plants, act as switches in the system and can be modelled by the ERM method in exactly the same way. “Special” situations can be modelled as fictional smaller plants with one module and dummy availability and repair parts. Again, one could argue that a specific procedure to handle the “special” situations could be devised. Reasons a) and c) once more justify the decision not do so.

3.3.3 The FC method

The designation of this method stands for Fraction Comparison (FC). The FC method is considered to be the “jewel in the crown” of the generic methodology (Albertyn, 2004).

The aim of the method is the determination of the actual power output throughput of the power plant at any given moment in time. The FC method determines the actual power output of the power plant by identifying, at any given moment in time, the momentary bottleneck. The momentary bottleneck may be defined as the point in the power plant that is momentarily limiting its power output. Thus, for the FC method to function, it is first necessary to specify which points can act as bottlenecks in the system. It has been demonstrated that every smaller plant is subject to failures, which may be responsible for more or less unavailability. This implies that all eight smaller plants can act as bottlenecks. It has been demonstrated that the “special” situations can also be responsible for unavailability. They qualify as possible bottleneck points as well. There are thus eight smaller plants and three kinds of “special” situations which can act as possible bottleneck points, or a total of eleven possible bottleneck points.

The FC method is based on the fact that the actual output throughput values of the smaller plants are in fixed relations of one another for all possible output vectors of the power plant. The fixed relations correspond to the input/output ratios of the smaller plants and are derived from the actual output throughput values of the smaller plants when the power plant is operating under steady-state conditions (no unavailability). This is not necessarily true for all chemical processes but it can be assumed so for the purposes of the simulation model. The validity of this assumption is supported by the verification and validation of the simulation model in section 4.4.

The steady-state actual output throughput values of the smaller plants constitute what is referred to as the FC method parameter set. The steady-state output throughput of each smaller plant can be determined by multiplying the number of modules of the plant by the capacity of each module. The FC method parameter set is presented in table A2 in Appendix B. The real capacity of each module is in fact slightly larger than the one that is presented in table A1, but this extra capacity is very seldom used, which means the difference between the two can be neglected. The "steady-state output throughput" values of the "special" situations are not considered to be part of the FC parameter set because they are only fictional and are determined arbitrarily.

Equation 8 expresses the steady-state output throughput of each smaller plant, as defined above:

$$Throughput_{PltSSAct} = n_{PltMod} Capacity_{PltMod} \qquad \text{(Eq. 8)}$$

Where:

$Throughput_{PltSSAct}$ – The steady-state actual output throughput of the smaller plant, as a constant, in the appropriate flow unit of measure.

If the values of the steady-state output throughput of the smaller plants are known, the FC method is able to determine the momentary bottleneck, and consequently the actual output throughput of all the smaller plants, by using the values of the maximum possible output throughput of the smaller plants. The maximum possible output throughput of a smaller plant, at any given moment in time, is the number of available modules of the smaller plant multiplied by the capacity of each module (see equation 5). The number of available modules of each smaller plant is determined by the ERM method.

According to the FC method, at any given moment in time a fraction is calculated for each smaller plant. It is the fraction of the maximum possible output throughput of the smaller plant to the steady state actual output throughput of the smaller plant:

$$Fraction_{Plt}(t) = \frac{Throughput_{PltMaxPos}(t)}{Throughput_{PltSSAct}} \qquad \text{(Eq. 9)}$$

Where:

$Fraction_{Plt}(t)$ – The fraction value of the smaller plant, as a function of time.

The fraction may assume positive values larger, equal or smaller than one. A smaller plant with a fraction value larger than one is operating below its maximum possible output throughput. A smaller plant with a fraction value equal or smaller than one is operating at its maximum possible output throughput, which in this case is smaller than the steady-state output throughput of the smaller plant.

The fraction values of the smaller plants can be compared thanks to the division process, which renders the values dimensionless. The smaller plant with the smallest fraction value is the momentary bottleneck point in the power plant. This value is referred to as the "Benben" value, as a reference to the sacred stone found in the temple of Heliopolis, Egypt, because it is the "sacred" value that determines the actual output throughput of the power plant (Albertyn, 2004):

$$Benben(t) = Min(Fraction_1(t), Fraction_2(t), ...Fraction_n(t)) \qquad \text{(Eq. 10)}$$

Where:

$Benben(t)$ – The Benben value of the power plant, as a function of time.

The actual output throughput or throughput vector of each smaller plant can be determined multiplying the Benben value with its steady-state actual output throughput vector:

$$Throughput_{PltAct}(t) = (Benben(t))(Throughput_{PltSSAct}) \qquad \text{(Eq. 11)}$$

The actual output throughput of the smaller plant is then used to determine the number of modules that are switched on and off in the smaller plant, through equations 6 and 7, respectively.

The FC method is applied, at any given moment in time, to the set of eleven possible bottleneck points, including the "special" situations. The "special" situations are modelled as if they were smaller plants (see subsection 3.3.2), each with a fictional module, and therefore their "steady-state actual output throughput" and "maximum possible output throughput" (they do not actually exist) can be calculated for the purposes of the FC method. An occurrence of one of these situations is equivalent, in the simulation model, to a failure of the single "module" in the smaller plant that corresponds to that "special" situation. The "maximum possible output throughput" of the corresponding fictional smaller plant will thus be zero, and the fraction value is also zero, which will certainly be the minimum fraction value of all the possible bottleneck points, i.e. the Benben value. That smaller plant, and hence the corresponding "special" situation, will be set as the momentary bottleneck, which is correct – in the real system, when one of the "special" situations occurs, it acts as a bottleneck in the system because production is interrupted.

For a better understanding of the FC method, a simple example featuring only the coal processing plant and steam production plants (considering only the steam as output of the steam generation plant) will be presented.

a) The steady-state actual output throughput values of the two plant system are (equation 8):
   i) Coal processing: 116,28 ton/h of powder
   ii) Steam production: 950,40 ton/h of SH (superheated) steam
b) Now suppose that the ERM method determined that two modules of the coal processing plant are being repaired. The number of available modules in each smaller plant is (equation 7):
   i) Coal processing: 3
   ii) Steam production: 1
c) The maximum possible output throughput values of the two smaller plants are (equation 4):
   i) Coal processing: 87,21 ton/h of powder
   ii) Steam production: 950,40 ton/h of SH steam

d) The fraction values of the two smaller plants are (equation 9):
    i) Coal processing: 0,75
    ii) Steam production: 1
e) The Benben value is therefore 0,75 (equation 10), which corresponds to the coal processing plant. The momentary bottleneck of the two plant system is thus the coal processing plant.
f) The actual output throughput values (the throughput vector of the two plant system) of the two plants are (equation 11):
    i) Coal processing: 87,21 ton/h of powder
    ii) Steam production: 712,80 ton/h of SH steam

3.3.4 Identification of the bottlenecks

Section 2.3 indicates that of the objectives of the simulation model of the power plant is to identify the sections of the electricity production that are responsible for most unplanned unavailability or, in other terms, which sections are most significant as bottlenecks.

The previous subsection indicates how the FC method can determine the momentary bottleneck in the power plant. The FC method does not directly reach the above mentioned goal of identifying the smaller plants that are the most significant bottlenecks, which requires an analysis to the system over a long period of time. However, it is quite clear that for the most significant bottlenecks to be identified, information on the momentary bottleneck must be collected over time. At the end of the simulation run that information must then be analysed using some sort of criteria to identify the most significant bottlenecks.

The generic methodology features two techniques for bottleneck identification and prioritisation - the lost production technique and the bottleneck time technique.

The lost production technique keeps a record for every smaller plant of the electricity production that is lost due to that smaller plant. The lost production due to each smaller plant, over a given period of time, is equal to the energy unavailability due to the smaller plant during that time period (the terms “lost production” and “energy unavailability” are synonyms). Lost production can therefore be calculated using equation 3, if the equation is applied to the smaller plant in particular (instead of to the whole power plant). At the end of the chosen time period, the lost production values of all the smaller plants are summed up and each lost production value can then be divided by the total, to generate comparable percentages.

We can therefore write

$$Bottleneck_{PltPrdLst}(\Delta t) = \left(\frac{\text{Production}_{PltLst}(\Delta t)}{\text{Production}_{TotLst}(\Delta t)}\right)100\% \qquad \text{(Eq. 12)}$$

Where:

$Bottleneck_{PltPrdLst}(\Delta t)$ – The percentage of the total lost electricity production that is due to the smaller plant.

$Production_{PltLst}(\Delta t)$ – The lost electricity production due to the smaller plant, over the period of time that is chosen, in MWh.

$Production_{TotLst}(\Delta t)$ – The total lost production of the power plant, over the period of time that is chosen, in MWh.

All variables are written as (Δt) to show that they depend on a period of time, because the two bottleneck identification techniques identify bottlenecks over a period of time, typically one year or more, depending on the length of the simulation run.

The bottleneck time technique works in a similar manner to the lost production technique. It measures the relative time that each smaller plant was the momentary bottleneck, over a given period of time. The bottleneck time technique keeps a record for every smaller plant of the time that smaller plant has been the momentary bottleneck. At the end of the chosen time period, the bottleneck time values of all smaller plants are summed up and each bottleneck time value can then be divided by the total, to generate comparable percentages.

Equation 13 translates the bottleneck time technique:

$$Bottleneck_{PltTm}(\Delta t) = \left(\frac{Time_{PltBtt}(\Delta t)}{Time_{TotBtt}(\Delta t)}\right)100\% \qquad \text{(Eq. 13)}$$

Where:

$Bottleneck_{PltTm}(\Delta t)$ – The percentage of total bottleneck time that is due to the smaller plant.

$Time_{PltBtt}(\Delta t)$ – The total time that the smaller plant is the momentary bottleneck over the chosen period of time, in hours.

$Time_{TotBtt}(\Delta t)$ – The total bottleneck time over the chosen period of time, in hours.

When a given smaller plant becomes the momentary bottleneck, one of the attributes of the smaller plant registers the current simulation time, in hours, while another attribute registers the power unavailability. When the smaller plant ceases to act as a bottleneck, the current simulation time is registered and the time difference between this time and the time the smaller plant became the bottleneck is calculated, for the purposes of the bottleneck time technique. The time difference is then multiplied by the power unavailability to determine the lost production due to the smaller plant, for the purposes of the lost production technique.

A simple example featuring the coal processing, steam production and electricity generation plants (whose only output will be considered as electricity) will be devised.

a) Suppose that at simulation time = 10 hours the FC method determined the following fraction values for the smaller plants:
   i) Coal processing: 0,75
   ii) Steam production: 1,0
   iii) Electricity generation: 1,0
b) The momentary bottleneck is the coal processing plant (equation 10). Equation 11 gives the actual output throughput values of the smaller plants:
   i) Coal processing: 87,21 ton/h of powder
   ii) Steam production: 712,8 ton/h of SH steam
   iii) Electricity generation: 223,5 MW
c) The current power unavailability is 298 – 223,5 = 74,5 MW (equation 1).
d) The unavailability is due to the momentary bottleneck, which is the coal processing plant. The corresponding attributes of the coal processing plant register the time and power unavailability:
   i) Time: 10h
   ii) Power unavailability: 74,5 MW
e) Now suppose that at time = 15 hours the FC method determined the following FC values:
   i) Coal processing: 0,75
   ii) Steam production: 0,0
   iii) Electricity generation: 1,0
f) The momentary bottleneck is the steam production plant (equation 10). Equation 11 gives the actual output throughput values of the smaller plants:
   i) Coal processing: 0,0 ton/h of powder
   ii) Steam production: 0,0 ton/h of SH steam
   iii) Electricity generation: 0 MW
g) The current power unavailability is 298-0 = 298 MW (equation 1).
h) The coal processing plant ceased to act as a bottleneck. The bottleneck time and lost production techniques compute the bottleneck values for the coal processing plant:
   i) Time as bottleneck = 15 – 10 = 5h
   ii) Lost production = (5h)*74,5 MW = 372,5 MWh (equation 3)
i) The unavailability is due to the momentary bottleneck, which is now the steam production plant. The corresponding attributes of the steam production plant register the time and power unavailability:
   i) Time: 15h
   ii) Power unavailability: 298 MW
j) Suppose that at time = 17 hours the steam production plant ceased to act as a bottleneck. The bottleneck time and lost production techniques compute the following bottleneck values for the steam production plant:
   i) Time as bottleneck = 17 – 15 = 2h
   ii) Lost production = (2h)*298 = 596 MWh (equation 3)

k) According to the bottleneck and lost production techniques, at the end of the simulation run unplanned unavailability is characterised thus:
    i) Total lost production = 372,5 MWh + 596 MWh = 968,5 MWh
    ii) Total bottleneck time = 5h + 2h = 7h
    iii) Coal processing lost production percentage (equation 12): 38,5%
    iv) Steam production lost production percentage (equation 12): 61,5%
    v) Coal processing bottleneck time percentage (equation 13): 71,4%
    vi) Steam production bottleneck time percentage (equation 13): 28,6%

If two or more smaller plants act as bottlenecks simultaneously, the lost production and bottleneck time are divided equally among the smaller plants.

In section 3.2 a rule of operation of the power plant is presented. The rule states that when the module (main water pump system) of the feedwater plant fails, it is replaced by two backup modules (two electrical pumps). These two pumps consume 10 MW of electrical power. The 10 MW are withdrawn from the electricity generation plant, decreasing its final power output to 288 MW (298 MW – 10 MW) or, in other words, causing a temporary power unavailability of 10 MW. The unavailability is in this case due to the feedwater plant. The output throughput values of all the smaller plants, however, remain unchanged because the output throughput of the two backup modules is the same as the output throughput of the main module.

This particularity could pose a problem in the application of the FC method. When the main module in the feedwater plant fails, no bottleneck is detected by the FC method because the fraction value of the feedwater does not change. Hence the FC method would not “know” that there is a 10 MW unavailability due to the feedwater plant.

A simple way to eliminate this problem is to consider that the two backup modules of the feedwater plant have indeed a joint output capacity that is smaller than the capacity of the main module Under the assumption that the output throughput values of the smaller plants are in fixed input/output ratios of one another, if the joint capacities of the two backup modules are considered to be a 288/298 fraction of the capacities of the main module, then that will translate into a final power output of 288 MW. The capacities of each backup module would thus be a (288/298)*(1/2) = 144/298 fraction of the capacities of the main module. This is why it is indicated under table A1 that the feedwater plant possesses two backup modules of capacities approximately 48% of the capacities of the main module. The real fraction is in fact 144/298.

This procedure ensures that, if the main module of the feedwater plant fails, the resulting 10 MW of unavailability are identified by the FC method and the appropriate bottleneck values can be assigned to the feedwater plant by the bottleneck identification techniques. However, the application of this procedure alone would result in an error in the total output throughput values of the smaller plants which are calculated at the end of the simulation run, because during the time that the feedwater plant was the bottleneck, the model assumed that the actual output throughput values of the smaller plants are a 288/298 fraction of their steady-state values, which is not the case of the real-world situation, where the actual output throughput values remain constant. For each smaller plant, the difference between the real-world actual output throughput value and the model-calculated value is thus (1-288/298)*(steady-state output throughput value of the smaller plant), but only during the time that the feedwater plant was the bottleneck. The total difference at the end of a simulation run equals (1-288/298)*(steady-state output throughput value)*(total time that the feedwater plant was the bottleneck). Therefore, the model simply has to add this difference to the total output throughput values of each smaller plant at the end of the simulation run to compensate for the error.

3.3.5 Summary

It is convenient to make a summary of all the methods, techniques and concepts that were presented in this section:

a) The variables technique that uses variables to represent the commodity flows in the system. This technique addresses the continuous process characteristic of the system;
b) The ERM method that determines the status of the modules in the smaller plants at any given moment in time, and, using that information, the maximum possible output throughput of each smaller plant at any given moment in time. This method addresses the discrete event characteristic of the system;
c) The FC method that determines the momentary bottleneck at any given moment in time and, using that information, the actual output throughput of each smaller plant (the throughput vector). This method addresses the complex interrelationships characteristic of the system;
d) The determination of the number of modules that are switched on and the number of modules that are switched off in each smaller plant, at any given moment in time;
e) The updating of variables that keep a record of the functioning of the simulation model, during the simulation run (number of repairs, total output throughput so far, total lost production and time as bottleneck so far) and at the end of the simulation run (mean number of available modules and of modules under repair, mean number of modules that are switched on and off);
f) The prioritisation of the bottlenecks at the end of the simulation run, using the lost production and bottleneck time techniques. These techniques address the complex interrelationships characteristic of the system.

It is possible to observe a separation of these methods, techniques and concepts into two parts. The ERM method addresses the discrete-event characteristic of the system by defining an availability part and a repair part in each smaller plant, which are used to determine the status of the modules. The modules are “real”, in the sense that they exist physically; therefore the ERM method is referred to as the “real” part of the simulation model. The other techniques, methods and concepts, on the other hand, address the continuous process and complex interrelationships characteristics through the use of variables and logical equations, whose existence is “virtual”. Therefore these methods, techniques and concepts are referred to as the “virtual” part of the simulation model. Figure 7 illustrates the division of the simulation model into the “real” and “virtual” parts:

“Real” part of simulation model	**“Virtual” part of simulation model**
- ERM method	- Variables technique - FC method - Determination of the number of modules switched on and off - Updating of variables - Determination of mean values of relevant variables - Bottleneck prioritisation

Figure 7: “Real” and “virtual” parts of the simulation model

3.4 CONCLUSION

In this chapter the conceptual development of the model is presented.

In section 3.2 the characteristics of the system under scrutiny, the power plant, are described. It was concluded that it would be convenient to group the equipments in each section of the production process into smaller plants organised in modules. Such organisation is more suitable to the required level of model detail. A formal system description is then performed, breaking the system down to its physical and functional aspects. The physical aspect comprises the elements configuration and module capacities and failure characteristics, while the functional aspect comprises the process flow and the process logic.

In section 3.3 the most important concepts of the generic simulation methodology developed by Albertyn (2004) are presented. These concepts correspond to different methods and techniques which are divided in a “real” part and a “virtual” part. The “real” part features the ERM method, which determines the state of the modules in every smaller plant at any given moment in time. The “virtual” part features: the variables technique, which represents the commodity flows with variables; the FC method, which, at any given moment in time, identifies the momentary bottleneck and determines the actual output throughput vector of every smaller plant; the determination of the number of modules that are switched on and off in each smaller plant, at any given moment in time; the updating of variables that keep a record of the functioning of the simulation model; the determination of the mean values of the relevant variables at the end of the simulation run; and bottleneck prioritisation using the lost production and bottleneck time bottleneck identification techniques.

CHAPTER 4

MODEL CONSTRUCTION

4.1 INTRODUCTION

In this chapter the work description initiated in the previous chapter is continued. After the conceptualisation step has been performed, the model may now be constructed. Model construction is the subject of this chapter.

In section 4.2 the aspects related to data collection and analysis are covered. In section 4.3, the reasons behind the decision to use one particular computer tool to implement the model are presented. One of different possible simulation perspectives is selected and its application to the implementation of the model is explained. In section 4.4 the verification and validation of the model are described. Finally, in section 4.5 simulation model enhancement is discussed.

4.2 DATA COLLECTION AND ANALYSIS

Data collection and analysis corresponds to the third step in a simulation study (see section 3.1). Data collection and analysis is made with two goals in mind (Porta Nova, 2000):

a) Fitting theoretical probability distributions to the data and estimation of parameters, in order to adjust the model to the reality of the problem under scrutiny;
b) Gathering information on the performance of the system, so as to validate the model on a latter stage.

This section is organised in two subsections. The first subsection focuses on data collection while the second subsection focuses on data analysis.

4.2.1 Data collection

The first thing to do when collecting data is to specify what kind of data to collect.

If figure 5 is recalled, it is possible to observe that there is one element of the system which has not yet been described. That element is the failure characteristics of the modules in the smaller plants.

The study of the failure characteristics of the modules implies the collection of failure data, which are fit to theoretical probability distributions whose parameters must be determined. This exactly the first goal of data collection and analysis stated above. Of course that the rest of the system description (element configuration and capacities, process flow) also meant the collection of the corresponding data, but it did not involve any statistical procedures and therefore were not considered as data collection *de facto*.

The failure characteristics of the smaller plants are vital information for the simulation model to work properly, because they affect the availability of the modules (see subsection 3.3.2). The two major failure characteristics affecting module availability are the time between consecutive failures of the modules (in some circumstances referred to as MTBF or Mean Time Between Failures) and the time to repair the modules (in some circumstances referred to as MTTR or Mean Time To Repair). Time between failures and time to repair are random variables, which will almost certainly differ from smaller plant to smaller plant. Therefore, data must be collected on the time between failures and time to repair of the modules of each smaller plant, to find out if theoretical probability distributions can be fitted to them. The term “smaller plant” includes the real smaller plants and the “special” situations. In the case of the “special” situations, the “time between failures” corresponds the time between successive occurrences of the “special” situation, while the “time to repair” corresponds to the duration of the “special” situation.

The power plant has four groups. Each group uses the same electricity production process (see figure 1) and features the same basic smaller plants (see figure 6), with identical numbers of modules and input/output capacities (see table A1). Nonetheless, each group possesses, of course, its own equipments and systems. An expert opinion from the management of the plant indicated that the equipments and systems of the four groups are approximately identical, as far as their failure characteristics are concerned.

An assumption that all four groups can be considered to have the same failure characteristics is beneficial for the simulation model, for two reasons:

a) The failure data of only one group, instead of four, have to be collected and analysed.
b) If the four groups are simulated in each run of the model, each run takes four times more time than if only one group is simulated. If, on the other hand, it is decided to run each group separately, it is necessary to run the model four times, instead of only one time for one group.

For the reasons that were presented, it seems reasonable to develop a simulation model for one group and to assume that the results of the model are extendable to all four groups. The historical data that were collected and analysed were provided by plant operators and were referent to group 2. The reason for the choice of group 2 is presented in section 4.4.

The beginning of this section indicates that additional data on the performance of the system must also be collected for validation purposes. The data that was collected for validation is the total electricity production of the power plant during the period 2006-2007. These data are presented in table A5 in Appendix C. The reasons for the choice of these particular data and time period will not be discussed here but on section 4.4, which concerns the verification and validation of the simulation model.

4.2.2 Data analysis

After the data have been collected they have to be analysed. The data collected for validation will be compared with the results of the model later on and do not require further analysis. The failure data, on the other hand, must be carefully analysed so that the failure characteristics of the modules may be defined.

The data corresponding to the time between failures and the time to repair of the modules of each smaller plant was collected and organised. From this it was possible to fit a probability distribution to each data sample and to estimate the respective distribution parameters, through the following procedures:

1) Calculation of sample descriptive statistics (average, standard deviation, etc.);
2) Construction of an histogram of the sample data;
3) Fitting of a theoretical probability distribution to the sample data using the method of the Chi-Square hypothesis test;
4) Estimation of the distribution parameters using the maximum-likelihood estimators.

The final results are presented in tables A3 and A4 in Appendix C. In table A3, each is line represents one smaller plant. Column 3 indicates the theoretical probability distribution that was fitted to the time between failures data corresponding to the modules of that smaller plant. Columns 4 and 5 indicate the parameters of the distribution. Columns 6, 7 and 8 work in the same way as columns 3, 4 and 5 but are referent to the time to repair of the modules. Table A4 concerns the "special" situations and it has the same logic.

It can be observed that the Weibull distribution is used to model the time between failures (or, for the "special" situations, the time between occurrences), which is not surprising. The smaller plants are composed of many different systems and equipments (see figure A3 in Appendix A). Many failures occur in the systems and equipments of the smaller plants, but only a few of those failures are "relevant", i.e. serious enough to render a module inoperable. As Banks et al. (2005) note, the Weibull distribution is adequate under such circumstances: "*When there are a number of components in a system and failure is due to the most serious of a large number of defects, or possible defects, the Weibull distribution seems to do particularly well as a model*". Therefore, even in situations where more than one distribution fitted well to the sample data, the Weibull distribution was chosen.

As for the time to repair (or, for the "special" situations, duration), it was verified that the Weibull and the exponential distributions fitted well to the data. This also makes sense because both distributions are widely used to model the time to complete some task or activity. In this case, the task is the repair of the smaller plant (or solving the "special" situation).

4.3 MODEL IMPLEMENTATION

In this section the implementation of the model is presented. Two parts can be identified in the section. The first part corresponds to subsection 4.3.1, which discusses the choice of a programming tool for implementation of the model. The second part is directly concerned with the implementation of the model and comprises two subsections. In subsection 4.3.2, a simulation perspective is selected. In subsection 4.3.3, the implementation of the model using the chosen programming tool is explained in more detail.

4.3.1 Mathematica – a programming tool

A vast number of computer aids is available nowadays for the development of simulation models. These computer aids fall in two major categories: general purpose programming languages and specific simulation software packages. Well known examples in the first category are FORTRAN, C, Java, Mathematica, etc. In the second category, important references are GPSS\H, Arena, SIMUL8, SLAM II, and others.

It was decided to build the simulation model in Mathematica 5.0. The main reason for this choice was the great familiarity the author possessed with the language.

It must be recognised that the use of a specific simulation software package has significant advantages over using a general programming language (such as Mathematica) for the development of a simulation model. The main advantage is that, in a simulation software package, many basic functions necessary to the construction of the model (time-advance mechanisms, event management, random number generation, etc.) are already implemented in the language, which speeds up the model implementation process (Porta Nova, 2000).

However, general purpose languages are still often used to the construction of simulation models and do present some advantages with respect to simulation software packages (Law and Kelton, 2000):

a) Most modellers already know a programming language, but not a simulation package (it was the case of the author);
b) Models written in general programming languages have the potential to be more flexible and run faster because they can be tailored to the specific problem in hand, whereas simulation packages are designed to address a wide variety of systems.

The use of Mathematica in particular, also has benefits for the purposes of a simulation model:

a) Mathematica can handle lists, matrices and other data structures with great ease and efficiency (unlike other programming languages, e.g. FORTRAN). This is of special importance when it comes to the development of a simulation model;
b) Mathematica has good random number generation capabilities;

c) Mathematica has extensive statistics functionalities, generally better than those of other programming languages. This allows Mathematica to be used both for data analysis and implementing the final simulation, meaning that the simulation model is entirely developed using the same tool, which might not be feasible with other languages.

Additional evidence that Mathematica can efficiently accommodate simulations is the fact that one of the major applications demonstrated in Mathematica teaching courses is precisely discrete-event simulation, as Carmo et al. (1999) confirm.

In the context of this research, two more aspects regarding the use of a general purpose language are worth noticing. The first is that the nature of the methods and techniques presented in section 3.3, namely the ERM method (which suggests the representation of modules with entities instead of building blocks), imply that an eventual simulation software package used for the construction of the model would not be employed in the "usual" way. Also, the implementation of the methods and techniques in general seems to require a good deal of "pure" programming. Albertyn (2004) implemented one version of a simulation model developed under the generic methodology in SIMUL8, and suggested that a very important part of this model was the Visual Logic block. Visual Logic is a feature of SIMUL8 and is basically a programming tool, which allows the manipulation of elements in the model through assignment of values to variables, conditions, loops, and other instructions common in general programming languages. Therefore, the simulation package would be used much in the same way as a programming language, so in fact it appears logical to use an actual programming language for the development of the model. Another relevant aspect is the fact that the two model versions developed by Albertyn (2004) were built in different simulation software packages, so as to prove that the generic methodology was truly generic and independent of the package being used. Thus, the use in this research of a general purpose programming language goes one step further to prove that the generic methodology is also independent of the kind (general programming language vs. simulation package) of computer aid being used.

4.3.2 Elements of the simulation model

In any simulation model there are a number of elements. The model elements under discussion in this subsection may be present in any kind of simulation model, although they are usually more intrinsically connected with discrete-event models. The following discussion is based primarily on the work of Porta Nova (2000) and Pidd (2004).

There are essentially three kinds of model elements: objects, attributes and operations. The following paragraphs present each of these elements.

The model objects are primarily the entities. The concept of entities has already been mentioned previously in subsection 3.3.2 but will be presented again here for the sake of the continuity of the argument. Entities are the elements in the system whose individual behaviour has a relevant effect on the state of the system, and whom the model keeps track of. From the discussion in subsection 3.3.2, it is possible to conclude that the only relevant entities of this simulation model are the modules in the smaller plants. The individual behaviour of each module affects the state of the system, which is in accordance with the definition of entities presented above. As for the organisation of the entities, there may be permanent groups of similar or identical entities in the model, which are called classes. It is also possible that there are temporary groups of entities, which are called sets; these often correspond to entity queues of some sort. In the case of this simulation model, modules are grouped in smaller plants and therefore each smaller plant can be considered as a class. The availability part and the repair part of the smaller plants (see subsection 3.3.2) act as sets, because during a simulation run modules spend some of the time in the availability part and some time in the repair part.

A second type of model elements is entity attributes. Attributes are the characteristics of the entities that define their behaviour. Usually there are several attributes for each entity. The attributes of the modules will be detailed later on in this section.

The third type of model elements is entity operations, which are referred to as activities. Activities are the operations or procedures that change the state of the entities. The start (or restart) and end (or interruption) of activities correspond to events in the simulation model. The sequence of activities and events of each individual entity in the system is called the process of that entity. A simulation model may follow different simulation perspectives, depending on the manner in which it handles entity operations.

In a process perspective, the sequence of activities and events (i.e. the process) of each individual entity in the system is followed throughout the course of a simulation run. This perspective is extremely difficult to implement, since it requires that the process of each entity is followed and updated simultaneously with the processes of all the other entities. This calls for sophisticated mechanisms of synchronisation present in some complex computational systems. The process perspective was therefore not chosen for the simulation model under development in this research.

In an activity perspective, the focus is, as indicated by the name, in the activities involving the entities. In this perspective it is necessary to: first) identify the relevant entities and entity classes, second) identify the activities in which the entities engage, third) link those activities together and fourth) define the conditions which preside to the realisation of each activity. In this perspective, the simulation time usually advances in fixed increments. At each instant in time, the conditions of every activity specified are checked, usually by inspection of one or more attributes of the entities. All the activities whose conditions are met are performed by the model.

Finally, there is the event perspective. In an event perspective it is necessary to: first) identify the relevant events that can alter the state of the system, and second) specify the actions the model must undertake when each of those events occurs. This perspective requires the construction and updating of a list of future events that indicates at least the time of occurrence and the kind of each event. The simulation time is usually advanced to the time of the next event (variable increment).

Some simulation packages are oriented to a specific perspective. However, since the model will be developed in Mathematica and not in a simulation package, this has no influence in the decision for which simulation perspective to use. The activity and the event perspectives are of equally feasible implementation in Mathematica. It was decided to use the activity perspective to build a first version of the model, because it is the most simple to implement.

The first step in this perspective, according to what was said above, is the identification of the relevant entities and classes. It has already been demonstrated that the relevant entities in this instance are the modules and that they are grouped in smaller plants.

The second and third steps, activity identification and connection, may be performed using an activity cycle diagram. This technique was popularised by Hills (1971). An activity cycle diagram is basically a map of the life cycle of an entity, which is considered as being characterised by a series of states, schematically represented by the diagram. Throughout the simulation, entities alternate between active and dead states. Active states usually involve the cooperation of different entities, for instance one entity which performs actions, or active entity, and one entity on which actions are performed, or passive entity. On dead states, on the other hand, “nothing happens”, and entities wait there until they move back to an active state. Active states are associated to rectangles while dead states are associated to circles. Active states and particularly dead states are frequently modelled by queues.

It is possible to observe such dynamics in the behaviour of the modules of the smaller plants. Figure 8 depicts the activity cycle diagram of the simulation model:

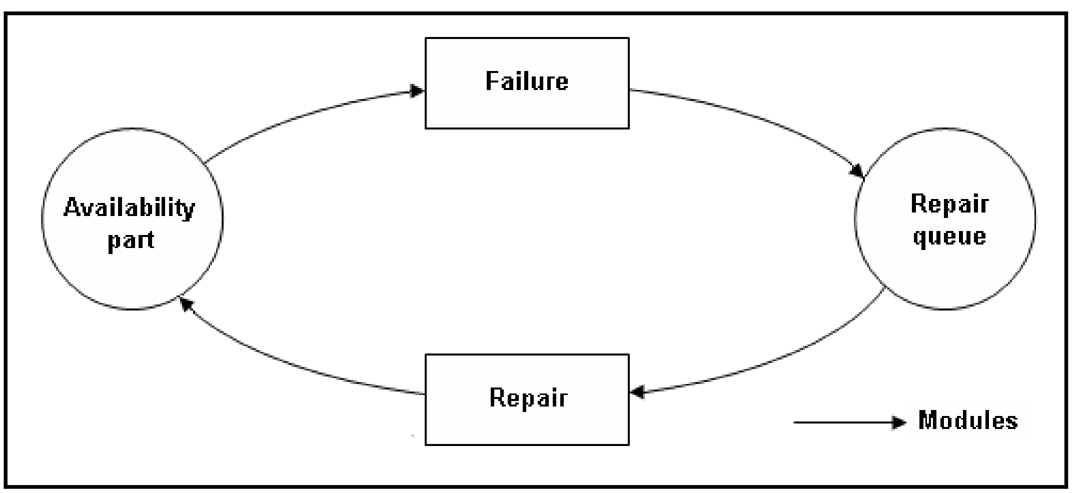

Figure 8: Simulation model activity cycle diagram

A module will probably spend most of the time in the availability part of the corresponding smaller plant, which corresponds to a dead state, because although the module may be switched on, nothing "happens" from the simulation point of view. The availability part is thus represented in a circle. Then the module could fail; this is an active state because the simulation model must perform actions – in this case, move the module to the repair part. Failure is thus represented in a rectangle. The repair part works exactly like a queue, where modules must wait for their turn to be repaired in case there is already another module under repair. The repair queue has a FCFS discipline (see subsection 3.3.2). The modules' waiting for repair constitutes a dead state. In the real-world situation, the first module in the repair queue can begin repair, which is once more an active state, as soon as the repair team is free. However, the repair team is not actually modelled as an entity in the simulation model; the modules are simply delayed in the repair part. After the module has been repaired, it is again available for work and can return to the availability part.

It is now possible to define the actions that must be performed by the model in each activity. According to the activity cycle diagram, one such activity is failure. The logic associated to the failure activity is represented in figure 9:

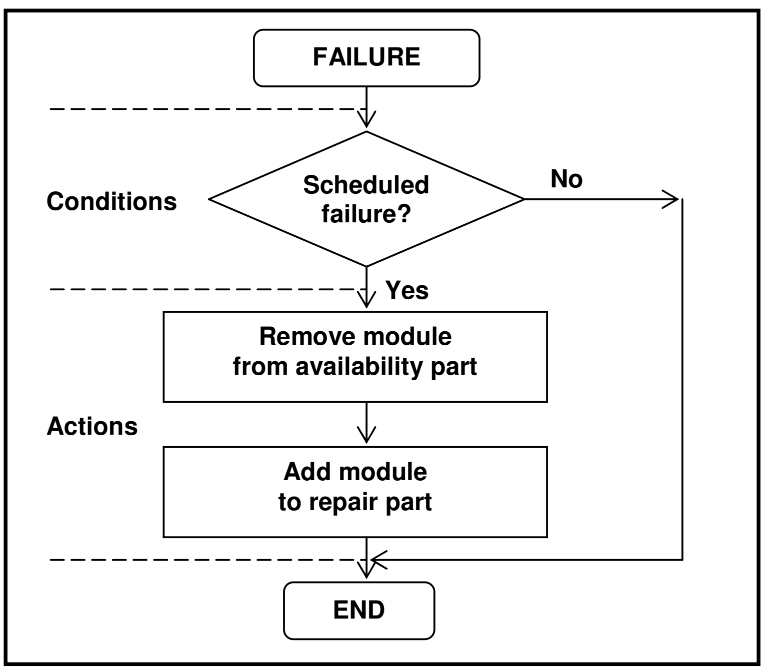

Figure 9: Failure activity – flowchart

The failure activity is rather simple. In case there is a scheduled failure (condition), the module is removed from the availability part and added to the repair part (actions).

The activity cycle diagram shows that there is one other activity in the model, which is that of repair. There is not, however, an actual *repair activity* because the modules are simply delayed in the repair part until they have finished being repaired and can be moved back to the availability part. The actions to be performed by the model are thus associated to the end of repair of a module. The end of repair activity is specified in figure 10 in the same fashion as the failure activity (using a flowchart):

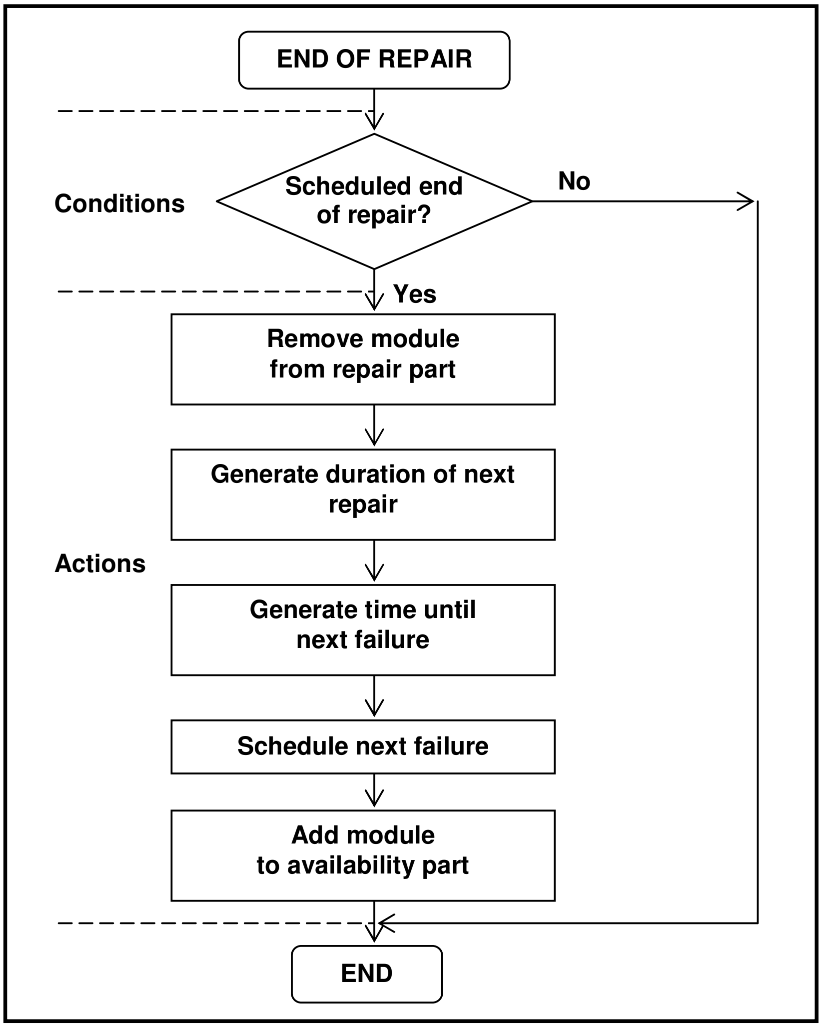

Figure 10: End of repair activity – flowchart

In case there is a scheduled end of repair, the corresponding module is removed from the repair part and its next repair and next failure attributes are attributed new values, because the module has been restored to an approximately "as good as new" configuration. The next failure is scheduled and the module is returned to the availability part.

It is possible to see that every activity follows the same basic structure, beginning with a set of conditions that must be met for the actions to be performed. This is the basic principle of the activity simulation perspective: the model increments the simulation time and, at each instant in time (each position of the simulation clock), the model verifies the conditions of every activity and if they are met, then the corresponding activities are performed. A basic condition for the realisation of an activity is that the activity must be scheduled to happen at that instant. The model verifies this condition by inspecting the attributes of the module. In the simulation model each module has the following attributes: 1) a number which identifies the smaller plant the module belongs to; 2) a number which identifies the module individually inside that smaller plant; 3) the duration of the next repair of the module; 4) the time of the next failure of the module; 5) the time of the beginning of the next repair of the module; 6) the time of the end of that repair; and 7) the current status of the module (on/off/being repaired). Therefore, to verify if a failure of a given module has occurred, the module compares the time of the next failure attribute of the module (4^{th} attribute) with the current simulation time. If they are equal, the failure activity is performed for that module. The process is identical for the end of repair activity, in which case the model compares the end of repair attribute (6^{th} attribute) with the current simulation time. After the necessary activities have been performed, the simulation time is incremented to the next time instant.

The previous discussion focused on the manipulation of the entities of the model (modules). It corresponds to the manner in which the ERM method manipulates the modules in each smaller plant. It therefore addresses the “real” part of the simulation model. The “virtual” part of the simulation model is, nevertheless, also necessary for the simulation model to work properly. In the next subsection, the interaction of all these concepts in the context of the computer implementation of the model will be clarified.

4.3.3 Mathematica implementation

When writing a large computer program, such as a simulation model, it is convenient that the program is divided in several files. This kind of organisation facilitates the program development and is also user-friendly. Mathematica files are called *notebooks* (extension *.nb*). The simulation model program was thus organised in four notebooks:

a) One notebook containing the definition of abstract data types in packages;
b) One notebook containing the definition of functions used in the course of the simulation;
c) One notebook handling the input variables;
d) One notebook containing the main function (the “simulator”).

Figure 11 schematically represents these concepts in a flowchart-like manner. It is a comprehensive, global view of the simulation model logic. The figure contains four boxes, each box corresponding to one Mathematica file or notebook. Inside each box the content of the respective notebook is represented. The following paragraphs provide insight about figure 11.

Mathematica offers a variety of data types – integer numbers, real numbers, lists, etc. Such data types can be referred to as *concrete*, meaning they are already implemented in the language.

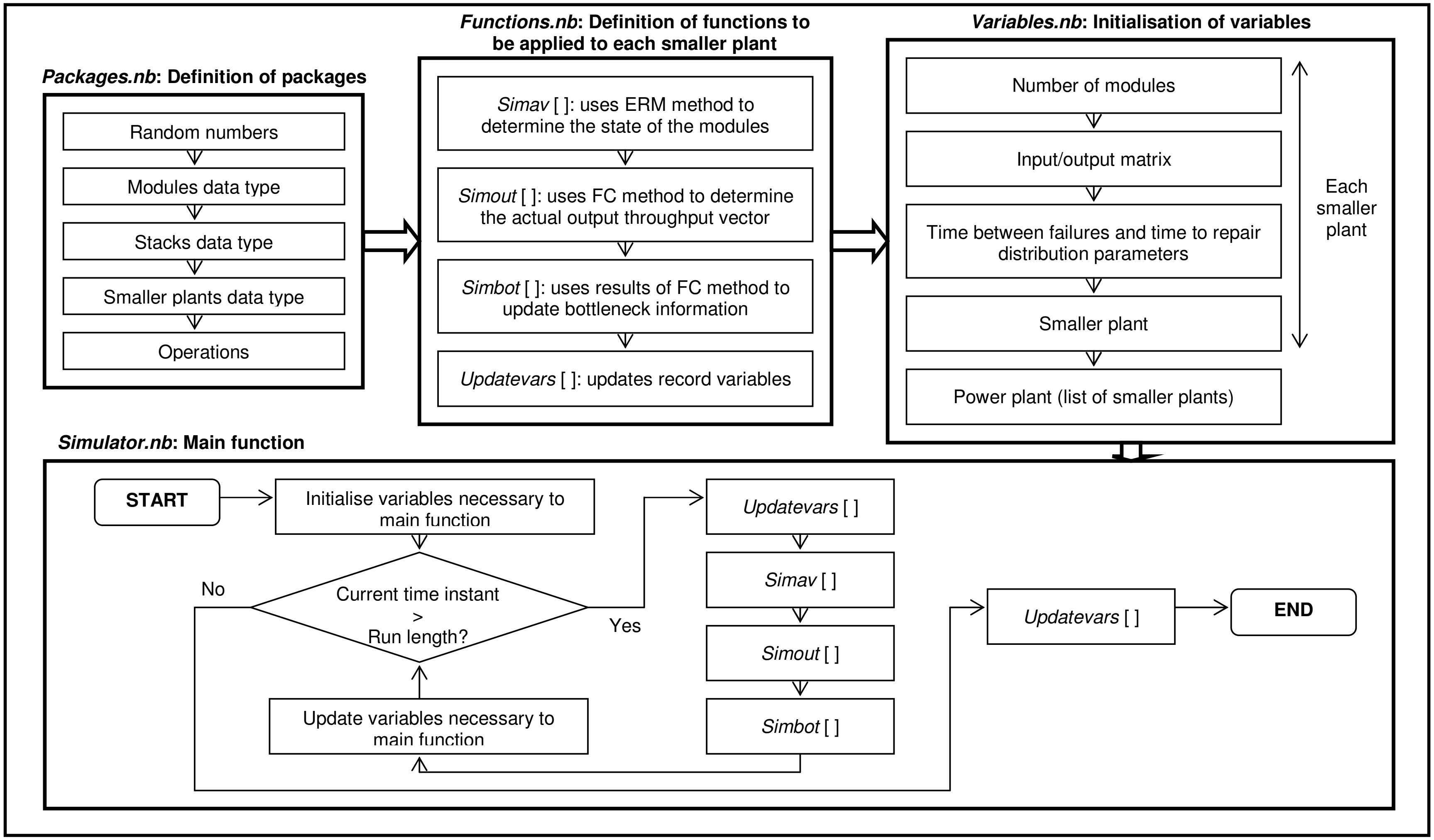

Figure 11: Simulation model diagram

However, the developer may wish to implement a specific type of data, with certain characteristics and operations that enable it to handle the problem more easily than concrete data types. These data types can be referred to as *abstract*, because they are not provided by the language.

Maeder (1994) stresses the importance of the definition of abstract data types in program development and the ability of Mathematica to do so: "*The theoretical concept of abstract data types provides a useful tool for program development. (This) can be realised in Mathematica very easily. Mathematica's interactive nature makes it well suited for rapid prototyping and testing of designs.*"

The implementation of abstract data types is, nevertheless, made upon existing (concrete) data types, usually lists, which Mathematica handles very well. The definition in Mathematica of an abstract data type is made using a functionality called "package". Besides the direct definition of data types, packages may also be used specifically for the definition of operations and functions in general, whether upon on those data types or not.

A total of five packages were implemented for the development of the simulation model: random number generation, definition of the modules data type, definition of the stacks data type, definition of the smaller plants data type, and operations on smaller plants.

The random numbers package was implemented to allow a more intuitive generation of random numbers, particularly when sampled from a Weibull or exponential distribution. The modules, stacks and smaller plants data types may be regarded as the **building blocks** of the simulation model and were used to represent modules, availability and repair parts, and smaller plants, respectively. Each building block is implemented in a separate package.

The modules building block corresponds to a list containing all the relevant information about the module. As for stacks, they are basically lists of modules used to represent the availability and repair parts of the smaller plants. Inside the stacks, the modules are sequenced from top to bottom according to their next failure attribute. This characteristic allows stacks to behave like module queues with a FCFS discipline, and it decreases simulation runtime by ensuring that only the top module of the stack needs to be verified by the model to determine if an activity needs to be performed.

The smaller plants package makes use of the functions and building blocks defined in the three previous packages (random numbers, modules and stacks) and it defines the smaller plants building block. In essence, the smaller plant building block is a list containing all the relevant information about that smaller plant. The last package defines several operations used by the simulation model. These operations include, for instance, the ERM and FC method procedures.

The Mathematica representation of the modules, stacks and smaller plants building blocks is displayed in Appendix D.

The previous discussion is devoted to explaining the meaning of the first "step" in figure 11. The five packages that were presented are implemented in a notebook called *Packages.nb*. Each of the four notebooks represented in figure 11 must be evaluated in Mathematica for the program to run; figure 11 presents the notebooks in the sequence (starting top left) in which they should be evaluated by the user.

In another notebook, *Functions.nb*, the simulation functions are defined. The simulation functions use the operations defined in the operations package. The functions are *Simav* [ ], whose purpose is to apply the ERM method to every smaller plant, *Simout* [ ], whose purpose is to apply the FC method, and *Simbot* [ ], whose purpose is to apply the lost production and bottleneck time techniques. The *Simav* function is the function responsible for inspecting the module attributes and performing the necessary activities. The *Simout* function manipulates the maximum, steady-state and actual output throughput vectors of every smaller plant for the purposes of the FC method. The *Simbot* function operates by updating the "bottleneck vector", which is the last vector in every smaller plant (see Appendix D). There is also the *Updatevars* [ ] function, whose purpose is to update the variables that keep a record of the functioning of the simulation model and to determine the values of the relevant variables at the end of the simulation run (see subsection 3.3.5).

In the *Variables.nb* notebook the main variables are then initialised. In this instance the main variables are the smaller plants. The elements that characterise each smaller plant, and which are defined by the user in this notebook, are the number of modules, the input/output matrix and the time between failures and time to repair distribution parameters. The list that represents each smaller plant (see Appendix D) is then automatically initialised. This logic is valid for every smaller plant; after all the smaller plants have been initialised, a power plant variable, which is the list of all the smaller plants, is constructed. It is upon the elements of this list (the smaller plants) that the simulation model operates.

The main function is defined in the last notebook, *Simulator.nb*. The main function makes use of all the packages, functions and variables defined in the previous three packages and hence it is the final step in figure 11. The function begins by initialising the variables necessary to its operation (these variables are only for internal use). The simulation time is initialised as zero. It then runs a loop, the simulation loop, which goes on until the simulation time exceeds the run length that was specified. In each step of the loop, the four functions *Updatevars* [ ], *Simav* [ ], *Simout* [ ] and *Simbot* [ ] are applied in this sequence to each smaller plant in the power plant variable. The internal variables are updated, namely the simulation time which is incremented by a fixed amount, and the cycle restarts. When finally the simulation time exceeds the run length, the *Updatevars* function is applied one last time and the simulation results are printed. One run of the model is thus completed.

In the next section, aspects such as the run length and the value of the fixed time increment will be addressed in more detail.

4.4 VERIFICATION AND VALIDATION

In this section, the verification and validation of the simulation model are performed. Verification and validation are two important steps in the development of a simulation study. Only after a simulation model has been acceptably shown to accurately represent the system under scrutiny may it be used to evaluate the effect of changes in the configuration of that system.

A simulation model may be validated using a variety of methods. The most widespread of these methods is the method of independent runs, because it is the most simple to interpret and the most robust (Porta Nova, 2000). This method was therefore selected to validate the simulation model. In the method of independent runs, the model is run several times, each run having a given length. Therefore, before attempting to validate or verify the model, it is convenient to determine the adequate length of each simulation run and number of model runs.

4.4.1 Length of the simulation run

According to Porta Nova (2000), the determination of the length of a simulation run is conditioned by the time span of the simulation. There are two kinds of simulations as far as the time span is concerned (Porta Nova, 2000):

a) Terminating simulations, where the modelled systems have well defined start and finish times, and the systems' probabilistic behaviour is not uniform. An example could be a bank agency (e.g. open from 8am to 3pm);
b) Steady-state simulations, where the modelled systems work continuously, and the systems' probabilistic behaviour eventually stabilises.

Is it fairly straightforward that the power plant falls within the second class of systems. The power plant works in continuous operation 24 hours per day. A steady-state simulation is therefore adequate.

A common problem in steady-state simulations is the initialisation bias. The initialisation bias is a distortion in the statistical distributions of the system variables, caused by the fact that the system is initialised far from steady-state conditions. Usually the system is initialised as empty (no entities) or inactive (Porta Nova, 2000), leading to the existence of a warm-up period in the simulation during which the system "fills up" with entities. Statistics collected during this warm-up period would therefore be biased.

An advantage of using the variables technique (see section 3.3) to represent commodity flows is that it is not necessary to wait for the system to "fill up" with entities before steady-state conditions are reached. According to the variables technique, the commodity flows are represented by variables, which assume the steady-state values from the beginning of the simulation. The only entities in the model are the modules, which are generated by the simulation model before the simulation starts and do not pose an initialisation bias problem either. Therefore it is not necessary to account for a warm-up period when deciding the length of the simulation run.

A run length that is frequently found in steady-state simulations is one year (i.e. when the simulation time reaches one year, the simulation is terminated). For the simulation model of the power plant, however, it was decided that a two-year period would be a more adequate basis for the length of each run. The main reason for this was the fact that the failure rates of the different smaller plants are in general relatively low, which leads to a possibly larger error in the number of failures generated by the simulation model if a shorter run length is used.

Therefore, the model would simulate the operation of one production group (section 4.4 explains why only one group should be considered in the model) of the power plant for a period of two years or 2*365*24 = 17520 hours. Validation of the simulation model could thus be performed by comparing the historical performance of one group during a particular period of 17520 hours (say 2006-2007), with the performance in the simulation model, if 17520 hours of simulated time are used as run length.

However, section 3.2 indicates that each production group in the power plant is subject to interruptions in production. These interruptions correspond to preventive maintenance service cycles of four years, each service taking six weeks. The cycles are different for each production group because they are staggered in time to minimise the impact on total production. Recently some groups have also interrupted production for the installation of the dessulphuration process. These interruptions are planned and are thereby responsible for planned unavailability. Such planned interruptions have an impact on the production of each group and hence must be accounted for by the simulation model. During the interruption the group is not producing any electricity and therefore its total electricity output throughput at the end of the year will be smaller than in normal years. One solution is to simply define an event in the model which corresponds to the beginning of the interruption. When the simulation time reaches the time of occurrence of that event, the simulation time is incremented directly to the time of the end of the interruption and the next-failure and next-repair attributes of the modules are given new values, because they have meanwhile been restored to an approximately “as good as new” configuration. The simulation then proceeds with its normal course. This technique ensures that the interruption is reflected on the final results of the model, because “nothing will happen” in the model between the beginning and the end of the interruption.

Data on the historical performance of the power plant must then be collected for verification and validation, for a particular two year period and a particular group. If during those two years the group has had a planned interrupted production (service cycle or other issues), then the interruption must be handled by the model in the manner described above. It was considered that the more recent this two-year period, the better (it is more appealing to prove that the model accurately represents the power plant in the year 2007 than proving that it accurately represents the power plant in the year 1993, for instance), and the two year period selected was thus 2006-2007, given that data were available for that time period.

Any of the four production groups would be eligible for the simulation model, since they are assumed to be identical as far as the failure characteristics of the equipment are concerned (see section 4.2). It was decided, however, that groups with longer production interruptions would present an opportunity for a more sound validation of the model. In the presence of longer stops, transient behaviour may assume a more relevant role in the behaviour of the system; if the simulation model, which excluded transient behaviour (see section 2.3), is proven to be valid even in such circumstances, this constitutes further evidence that the assumption that transient behaviour can be excluded from the model is correct. In the period 2006-2007, the groups subject to the longest stops were groups 1 and 2. Group 2 was chosen arbitrarily. During 2006-2007, group 2 interrupted production for a total of 2400 hours (505 hours due to a maintenance service and 1895 hours due to the installation of the dessulphuration process). In other words, it was subject to 2400 hours of unplanned unavailability. These interruptions were consecutive and took place from the 1st of April 2007 to the 9th July 2007. In the context of the simulation model, this means that the interruption will last from simulation time = 10920 hours (period 1st January 2006 – 1st April 2007), until 2400 hours from there or at simulation time = 13320 hours. The structure of the simulation model run, for validation purposes, is represented in figure 12:

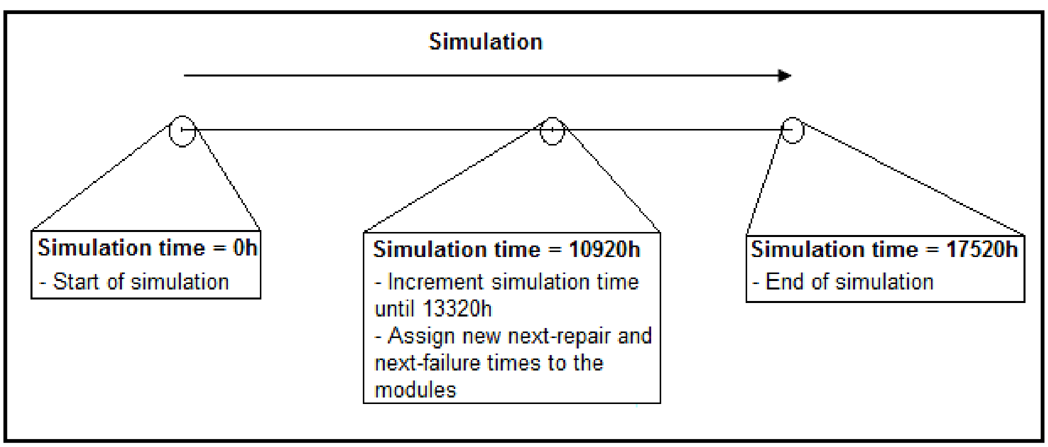

Figure 12: Structure of the simulation run for validation

The data collected for validation of the model is presented in table A5 in Appendix C. Table A5 shows the total electricity production of group 2 in the period 2006-2007, which will be used for validation of the model. The lost production in MWh in group 2 over that period will be used for complementary validation and it is also presented. **This value refers to the lost production due to unplanned unavailability only.** The lost production corresponding to planned unavailability is already accounted for by the model thanks to the interruption procedure.

After validating the model, it will have to be run again for obtaining the final results. The simulation run structure will be slightly different here than for validation purposes, because each run does not need to mimic the historical interruptions of a group. It is convenient, however, that the model runs somewhat reflect the “normal” or “steady-state” behaviour of the system as far the planned interruptions are concerned. The management of the power plant indicated that after the dessulphuration process has finished being installed in all groups, which is due to happen in 2008, the planned interruptions will be due only to preventive maintenance, which will return to its normal configuration of four year cycles, six weeks each service. This may be regarded as the “steady-state” behaviour of the power plant as far as planned unavailability is concerned. One may also consider that an interruption of six weeks every four years is equivalent to an interruption of three weeks every two years, and that these interruptions happen in the middle of the two year period, i.e. from the beginning of the first year’s last $1^{1/2}$ week or simulation time = 8436 hours, until the end of the second year’s first $1^{1/2}$ week or simulation time = 8940 hours, as represented in figure 13:

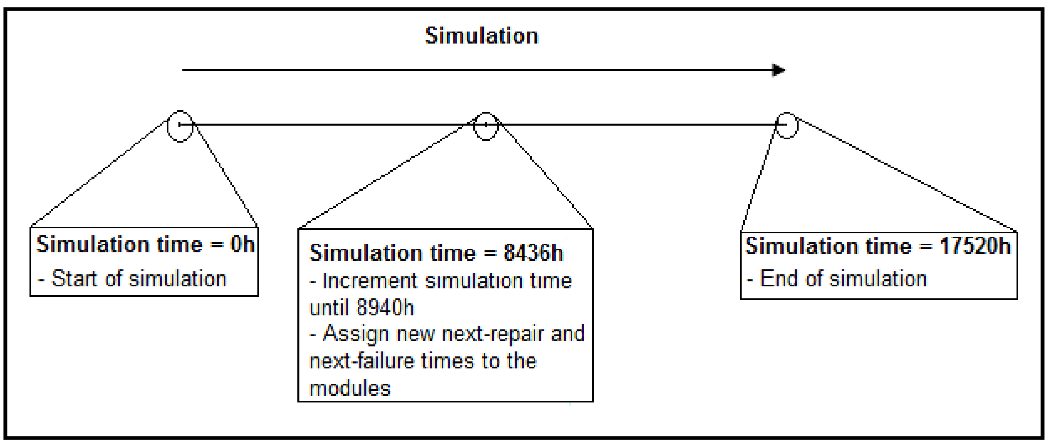

Figure 13: Structure of the simulation run for obtaining results

It is said in the previous section that the simulation time is incremented by a fixed amount (in the activity-based simulation perspective). The value of that increment may be determined using statistical techniques. Such techniques, however, do not fall within the scope of this document. Albertyn (2004) uses a statistical procedure to reach the conclusion that a time increment of one hour is appropriate. The value of one hour seems an adequate time increment in the case of the power plant simulation model because all the commodity flows are expressed as hourly rates. The time between failures and time of repair of the smaller plants are also in a scale of hours, as is the run length. Therefore it seems quite reasonable to establish a one-hour time increment for the purposes of the simulation model.

4.4.2 Number of runs

In the beginning of this section it is indicated that the method of independent runs is used to validate the simulation model. The method of independent runs consists in running the model *n* times, each run producing one value for each of the relevant variables. The values produced by the different runs of a stochastic simulation model are usually not identical due to the random behaviour of the system. Thus, after *n* runs, a sample of size *n* is obtained for each relevant variable. The samples can then be analysed by common statistical techniques (Porta Nova, 2000).

A larger number of model runs means a smaller variance and hence a greater confidence in the estimation of the values of the relevant variables. However, one cannot simply increase the number of runs *ad æternum* as that would translate in unacceptably large runtimes. The aim is to know the minimum number of model runs necessary to reach an acceptable level of confidence for the values of the relevant variables (Porta Nova, 2000).

Crow et al. (1960) suggest the following rule for the minimum sufficient number of runs: to reach a confidence interval of probability 100(1 – α)% whose length does not exceed $2H$, increase the number of model runs n and calculate the standard deviation *S* until

$$t_{\alpha/2;n-1} \cdot \frac{S}{\sqrt{n}} \leq H \qquad \text{(Eq. 14)}$$

Where:

n – The number of model runs

100(1-α)% – The confidence interval, as a percentage.

S – The standard deviation of the relevant variable(s), calculated after n runs of the model, in the appropriate unit of measurement.

t – The upper percentage point of the t distribution value.

H – Half of the expected length of the confidence interval for the relevant variable, in the appropriate unit of measurement.

The simulation model will be validated by comparing the total electricity production of group 2 in the period 2006-2007 with the total electricity production value returned by the model. Therefore, the relevant variable used in equation 14 is the total electricity production of group 2 in the period 2006-2007, whose value is indicated in Table A5 in Appendix C.

At a 99% confidence interval, the half-length of the interval H equals 0,5% of the electricity output throughput or approximately 21711,5 MWh. Hence the model will be run several times, successively increasing the number of runs and substituting the appropriate values in the left side of equation 14, until it returns a value smaller than the maximum 21711,5 MWh. The first n

(number of runs) to satisfy such a condition is the minimum number of model runs sufficient to determine the value of the electricity output throughput of group 2 during 2006-2007, with an expected confidence of 1%.

Table 1 presents the results for different numbers of model runs:

Table 1: Half-length of confidence interval for different numbers of runs

n	Runtime (mins.)	$t_{\alpha/2;n-1}$	S (MWh)	Left side eq. 14 (MWh)	H_{max} (MWh)
17	24,3	2,921	32541,1	23053,6	21711,5
18	25,8	2,898	31520,2	21530,4	21711,5
19	27,2	2,878	31308,4	20671,6	21711,5
20	28,4	2,861	30837,6	19728,0	21711,5

From table 1, it is possible to observe that any number of runs larger than 17 satisfies equation 14. The minimum number of runs of the simulation model is thus 18. It is a common practice to establish a number of runs slightly larger than the minimum, e.g. the closest "round" number. Therefore, 20 model runs appear to be an adequate number.

4.4.3 Verification and validation of the simulation model

Many authors have stressed the importance of performing simulation model verification and validation before the model can be used to take conclusions about the system under scrutiny. Such authors include Law and Kelton (2000), Pidd (2004) and Banks et al. (2005).

Law and Kelton (2000) provide the following definition of verification: "*Verification is concerned with determining whether the conceptual simulation model (model assumptions) has been correctly translated into a computer 'program', i.e. debugging the simulation computer program.*"

The computer program implemented until this stage had approximately 600 lines of Mathematica code. It is obviously not a trivial task to verify computer programs of this size. A detailed discussion of the verification of the computer program of the Sines power plant simulation model is outside the scope of this document; still, the most important procedures are presented in the rest of this section.

Some verification techniques mentioned by Law and Kelton (2000), and that were used in the Mathematica implementation of the power plant simulation model, are:

a) Writing the program in modules or subprograms. Such practice hugely facilitates program debugging. It is well known in the computer programming world that even the simplest patches of code rarely work at the first attempt. Therefore, it would be simply suicidal to write the whole program and only then try to debug it, because the possible sources of error would be extremely diverse;

b) Running the model under a variety of settings of input parameters and checking that the model output is reasonable. As an example, the application of the FC is an arduous task because, at each increment of the simulation clock, there are many possible interactions in the state of the modules in the different smaller plants. The model was thereby tested with "extreme" values, i.e. very short times between failures. This ensured that all possible interactions were uncovered and that the model was apt to cope with them;

c) Tracing the model during its execution. Tracing the model means following the state of the simulated system, e.g. values of variables, statistical counters, etc., throughout the simulation. The Mathematica print function was extremely useful in tracing the model, namely in the "extreme" conditions mentioned above.

As a complement to this discussion, a test to the output of the model will be performed.

A basic yet powerful way of verifying the power plant simulation model is to compare the number of failures generated by the model in each smaller plant with the real number of failures that occurred, during the simulated time period. The number of failures generated by the model after 20 runs of 17520 hours (each following the run structure for validation) was thus compared to the real number of failures in each smaller plant, during 2006-2007. Table 2 makes such comparison.

Table 2: Verification of the simulation model

Name	**No. modules**	**Dist. / param.**	**Real no. failures**	**Model no. failures**	**Dev. (%)**
Coal processing	5	We(1,19;2510)	31	31,25	0,81
Steam production	1	We(1,35;5500)	3	2,25	-25,00
Turbine	1	We(1,24;3690)	4	3,85	-3,75
Electricity generation	1	We(1,20;3100)	5	5,10	2,00
Water cooling	1	We(1,17;13030)	2	1,00	-50,00
Condenser	2	We(1;33;24480)	1	0,95	-5,00
Feedwater	1	We(1,20;805)	18	17,15	-4,72
Air-smoke	2	We(1,22;13600)	2	1,85	-7,50
Control system	-	We(1,10;6205)	2	2,55	27,50
Peaks	-	We(1,15;2290)	6	7,00	16,67
External failures	-	We(1,18;1950)	8	8,65	8,13

In table 2, the real-world number of failures in the smaller plants is compared to the number of failures generated by the simulation model. The first column of the table lists the smaller plants and "special" situations, with the second column indicating the respective number of modules. The third column is entitled "Dist. / param." and it indicates the type of theoretical probability distribution used to model the time between failures of the corresponding smaller plant / "special" situation ("We" stands for Weibull), as well as the respective parameters. Then, in the fourth column, the number of failures that actually occurred in group 2, in the period 2006-2007, is indicated for each smaller plant. This is equally applicable to the "special" situations, where the number of "failures" corresponds to the number of occurrences of the situation. Finally, the fifth column indicates the mean number of failures or occurrences calculated from 20 runs of the simulation model of 17520 hours each, while the sixth column indicates the deviation ("Dev.") of that number to the real-world number, as a percentage.

An inspection of table 2 reveals that the deviation of the number of failures generated by the model to the real number of failures assumes values ranging from 0,81% (coal processing) to -50,00% (water cooling). This and several other deviation values are considerably large. However, one must keep in mind that these smaller plants have relatively low failure rates, as suggested by the large values of the β parameter of the Weibull distributions (this was one of the reasons that led to the decision to use a two year basis for the length of the simulation run in the first place). The mean time between failures of the water cooling plant, for instance, is 12340 hours, which translates in about one failure every 17 months. This low failure rate could easily result in a large relative deviation; the same goes for other smaller plants with large deviations. Still, the overall impression is that the simulation model works correctly, as far as the number of generated failures is concerned.

It may finally be attempted now to validate the simulation model. Law and Kelton (2000) define validation as "… *the process of determining whether a simulation model* (…) *is an accurate representation of the system*, *for the particular objectives of the study*." It must nevertheless be mentioned that one can never be completely sure that a model is "valid"; Law and Kelton (2000) say that "*There is no such thing as absolute model validity.*" Pidd (2004) goes even further to claim that "(…) *in practice it can be very difficult to validate a simulation model properly.* (…) *Validation is impossible, but desirable*".

Validation is "impossible" in the sense that it is extremely difficult to prove that the model mimics the actual system in every aspect, under every possible circumstance. A simulation model may appear to be a very accurate representation of a system but *it is not* the system. However, as Law and Kelton (2000) point out, one must always concentrate on the particular objectives of the study, namely on the values of the relevant variables. Indeed, validation usually involves a comparison of the real values of the relevant variables, for a known scenario, with the values of the same variables returned by the simulation model, in the same scenario.

It has been previously demonstrated that the total electricity production of group 2 during 2006-2007 would be used for validation. The total electricity production of group 2 during 2006-2007 was 4342301 MWh (see Table A5 in Appendix C). Therefore, in order for the simulation model to be validated, this value must be compared to the mean total electricity production generated by the model, when it is run with the specified run length and number of runs (17520 hours and 20 runs, respectively). Table 3 makes such comparison:

Table 3: Validation of the simulation model

Run length (hours)	No. runs	Runtime (mins.)	Real elec. prod. (MWh)	Model elec. prod. (MWh)	Dev. (%)
17520	20	28,4	4342301	4340073	-0,051

The value of the electricity output throughput (electricity production) obtained by the simulation model is indicated in the fifth column of table 3 ("Model elec. prod."). This value was reached after 20 runs of the model of 17520 hours each. Each run followed the structure presented earlier in this section. The runtime was approximately 28,4 minutes. The first three columns of table 3 provide this information. The simulation model used an activity-based simulation perspective and a time increment of one hour, according to what was previously discussed. The real-world value of electricity production is indicated in the fourth column of the table ("Real elec. prod."), which is then compared to the value obtained by the model (in the fifth column) to determine the deviation of the latter to the former, as a percentage. The deviation is indicated in the final column of the table. The value obtained by the model for the electricity production is the mean of the electricity production values returned by each of the 20 runs.

It may be observed from table 3 that the percentage deviation of the model value to the real value is very small (less than 0,1%). Another variable that must be computed by the model, and which may be used for complementary validation, is the ratio of lost production to total electricity production. The value returned by the model in this instance was of 3,81%, which correlates closely with the real-world value (for group 2 during 2006-2007) of 3,76%. The real-world value of the ratio is computed with the information presented in table A5 in Appendix C.

Even though absolute model validation is not feasible, these results – very small deviation for the electricity production and close correlation between the real-world and model-predicted ratio of lost production – indicate that it can be accepted that the simulation model (which was built under the activity simulation perspective and used an iteration time interval of one hour) is a valid representation of the power plant.

The next section is concerned with simulation model enhancement.

4.5 MODEL ENHANCEMENT

After the simulation model was built, it was investigated whether an improvement of any sort to the model was possible. It was decided to attempt an improvement in two model performance measures: the variance and the runtime.

4.5.1 Variance reduction

In the previous section it is demonstrated that the minimum number of runs of the model to reach an acceptable level of confidence is 18. This appears a somewhat large number, in comparison with many other simulation models which use for instance 15 runs. Additionally, the majority of simulation models do not use more than 20 runs, which was the selected number for this simulation model, and 20 runs represents a relatively low “safety margin” with respect to the minimum number of 18 runs.

Therefore it seems reasonable to claim that the model may be improved if the minimum number of runs is decreased. According to equation 14, the minimum number of runs of a simulation model increases with the standard deviation of the model results. A decrease in the minimum number of runs can thus be achieved if the standard deviation, or alternatively the variance (given that variance is the square of the standard deviation) is somehow reduced. Variance may be reduced using a variance-reduction technique (VRT).

Law and Kelton (2000) point out five different VRTs:

a) Common random numbers (CRN);
b) Antithetic variates (AV);
c) Control variates (CV);
d) Indirect estimation;
e) Conditioning.

The last three VRTs, control variates, indirect estimation and conditioning, are not as straightforward as the first two and are more seldom employed. As such they were rejected as candidates to be used in the enhancement of the simulation model.

The main difference between the first two VRTs, common random numbers and antithetic variates, is that the common random numbers technique applies when two or more alternative system configurations are being compared, while the antithetic variates technique applies to the analysis of a single configuration. The stage where alternative power plant configurations are compared has not yet been reached; the aim is to reduce the variance of the simulation model that was developed, which represents the power plant as it is now. Therefore the AV technique looks more appropriate.

Law and Kelton (2000) describe the principle of AV: "*The central idea* (of AV) *is to make pairs of runs of the model such that a 'small' observation on one of the runs in a pair tends to be offset by a 'large' observation on the other one; i.e., the two observations are negatively correlated.*" AV thus endeavours to induce correlation between the values of pairs of model runs. If the average of the two observations in the pair is used for analysis, it will tend to be closer to the average μ of the variable whose value is to be determined by the model than if the two observations in the pair were independent, which is the normal case (Law and Kelton, 2000).

AV makes use of random variates generation methods, a matter that will be briefly presented here. A common way to sample an observation from a theoretical probability distribution is to first generate a random observation from a uniform distribution in the interval [0,1] and then to perform some transformation on it, depending on the desired final distribution. This is known as the *inverse-transform* method for generating random variates. The inverse-transform method is used in the Mathematica program to generate module failure and repair times from either a Weibull or an exponential distribution. To generate an observation from a Weibull distribution of parameters (α, β), a random observation from a uniform distribution in the interval [0,1] or $U(0,1)$ is generated (using the Mathematica *Random* function) and $\beta(-\ln U)^{1/\alpha}$ is computed. The process for generating an observation from an exponential distribution of parameter β is similar, the final transformation being $-\beta\ln U$. It is possible to demonstrate that $\beta(-\ln U)^{1/\alpha}$ and $-\beta\ln U$ follow Weibull and exponential distributions respectively.

The principle of AV is to use complementary (and hence the name *antithetic*) random numbers in pairs of runs. If U_k is used in the first run, then $(1 - U_k)$ is used in the second run. It is valid to do this since $(1 - U_k)$ also follows a $U(0,1)$ distribution. The idea is that a possibly "large" observation in the first run (U_k) will be compensated by the corresponding "small" observation in the second run ($(1 - U_k)$). For AV to work properly, however, it is essential that the two random numbers be used for the same purpose; otherwise the benefit of using AV could be lost or even backfire (Law and Kelton, 2000). Consider the following example: a random number is used in the first run of a run pair to generate the time until the next failure of a module, and it results in a small time. That will tend to increase the bottleneck effect of the corresponding smaller plant in the first run. If the complementary number is used in the second run of the pair, it will result in a large time, which will tend to decrease the bottleneck effect of the corresponding smaller plant in the second run and "compensate" for what happened in the first run. The purpose of using AV is therefore achieved. If, however, the complementary random number is used to generate a repair time in the second run, it will again result in a large time, but now that will tend to once more increase the bottleneck effect of the smaller plant, which is the opposite of the intended effect. Hence the importance of synchronisation between the two runs.

One way to ensure that the random numbers used in one run are based on the random numbers of the previous run is to use random number streams. A stream is a list of random numbers that are successively used throughout the simulation run to generate the desired observations. In the case of AV, model runs are organised in pairs and the second run of each pair uses the complementary stream of the first run. Synchronisation can be achieved by using stream dedication, i.e. different streams for different purposes – one stream for generating failure times and one stream for generating repair times, for instance.

The AV technique was thus applied to the simulation model by implementing these concepts in the Mathematica program. The smaller plants building block was enhanced with the inclusion of two random number streams, one for failure times and one for repair times. This means that each individual smaller plant has its own dedicated streams, ensuring maximum synchronisation so as to increase the potential benefit of using AV. Some functions were also added to the program to enable stream-sharing in pairs of runs.

After the AV technique was implemented the model was run several times, varying the number of runs. Table 4 presents the results in a similar fashion to table 1:

Table 4: Half-length of confidence intervals for different numbers of runs using the AV technique

n	Runtime (mins.)	$t_{\alpha/2;n-1}$	S (MWh)	Left side eq. 14 (MWh)	H_{max} (MWh)
11	17,4	3,169	25107,5	23990,0	21711,5
12	18,2	3,106	24018,4	21535,5	21711,5
13	19,6	3,055	23941,3	20285,6	21711,5
14	21,1	3,012	23568,7	18972,6	21711,5
15	22,3	2,977	23114,8	17767,4	21711,5
16	23,5	2,947	22727,0	16744,1	21711,5
17	24,3	2,921	22035,6	15611,1	21711,5
18	25,8	2,898	21472,8	14672,4	21711,5
19	27,2	2,878	20334,7	13426,2	21711,5
20	28,4	2,861	19951,3	12763,6	21711,5

If the results presented in table 4 are compared with the results presented in table 1, it is possible to see that the AV technique has a significant impact on the overall performance of the simulation model. The minimum number of runs to reach a 99,5% confidence interval (of dimension 21711,5 MWh) is now 12 instead of the original 18. The same is to say that the variance in the model results has been greatly reduced. If the square of the standard deviation (i.e. the variance) from n = 17 to n = 20 in table 4 is compared to the square of the standard deviation from n = 17 to n = 20 in table 1, it is verified that the variance experienced a reduction of between 54% (n = 18) and 58% (n = 20). In addition, if 20 runs is kept as the adequate number of model runs, the half-length of the confidence interval is now 12763,6 MWh, which represents a decrease of approximately 35% with respect to the situation without the AV technique. Furthermore, the inclusion of the AV technique did not increase model runtime.

The model results for n = 20 were the following:

Table 5: Validation of the simulation model using the AV technique

Run length (hours)	No. runs	Runtime (mins.)	Real elec. prod. (MWh)	Model elec. prod. (MWh)	Dev. (%)
17520	20	28,4	4342301	4339999	-0,053

There is no discernible difference between the accuracy of the model with the AV technique and the accuracy of the model without it (see table 1).

These results indicate that the inclusion of the antithetic variates variance reduction technique is highly beneficial for the overall performance of the simulation model.

4.5.2 Event simulation perspective

Section 4.5 presents different simulation perspectives, with the conclusion that the activity and event perspectives are of feasible implementation in Mathematica. It was then decided to implement the model under the activity perspective, because it is the simplest approach. In the activity perspective, the simulation time is incremented in fixed amounts (1 hour in this case), and at each time instant the activity conditions are checked. If the conditions are met, the activities are performed; if the conditions are not met, the activities are not performed. In either case time is afterwards incremented to the next instant.

The use of the activity perspective implies that there are many instants where "nothing happens", i.e. no activity is performed. This is particularly evident when there are few events during a simulation run, which is the case of the simulation model of the power plant. It thus seems that it would be a more effective solution for the simulation model to advance the simulation time automatically to the time of the next event, instead of using a fixed-time increment and checking whether an event is scheduled to happen at the current instant.

The event simulation perspective uses such approach. In this perspective, a list of future events is kept. The events are characterised namely by the respective time of occurrence and are sequenced according to it. Therefore the first event in the list is always the next event to be simulated. The basic principle is that the simulation time is advanced to the time of the next event, and the model performs the necessary actions corresponding to the event. The event is then deleted from the list and the process continues with the next event in the event list.

Before the model can be implemented it is necessary to specify the different events that may occur in the simulation model. This is analogous to what was done in the activity perspective, where the different activities of the model were defined. Indeed, the activities that were then defined, failure and end of repair, are "pseudo" activities because they actually correspond to events. The logic associated to the failure and end of repair events is exactly identical to the logic associated to the failure and end of repair activities, with the exception that there are no conditions to be met (the simulation time is advanced to the time of the next event).

A simulation model using the event perspective was thus implemented in Mathematica. This version of the model included the antithetic variates technique discussed in the previous subsection. The event-based simulation model possesses an identical logic to the logic of the activity-based simulation model presented in figure 11. An extra package, the event package, had to be created in the *Packages.nb* notebook. In this package the basic functions on events were defined – creation of events, construction of a list of future events and insertion of an event in the list. Events are modelled as lists of four elements, where the first element is the ID of the smaller plant involved in the event, the second element is the ID of the particular module involved in the event, the third element is the kind of the event (failure / end of repair) and the fourth element is the time at which the event takes place. Module attributes also changed. The duration of the next repair, time of beginning of next repair and end of next repair attributes were eliminated, while the next failure attribute was kept in order to facilitate the sequencing of the modules according to their next failure in the availability and repair parts.

After the model was implemented it was validated in the same fashion as before. 20 runs of the model were completed and the results are presented in table 4:

Table 6: Validation of the event-based simulation model

Run length (hours)	No. runs	Runtime (mins.)	Real elec. prod. (MWh)	Model elec. prod. (MWh)	Dev. (%)
17520	20	0,166	4342301	4341050	-0,029

The deviation between the real-world value of the electricity production and the value returned by the model is very small (smaller than 0,1%). The lost production ratio computed by the model was 3,79%, which correlates very closely with the real-world value of 3,76%. These results indicate that it can be accepted that the event-based simulation model is a valid representation of the power plant.

Furthermore, the event-based simulation model represents a phenomenal decrease in runtime with respect to the activity-based model. The activity-based model took almost 30 minutes to complete 20 runs (see table 5), whereas the event-based model takes approximately 10 seconds. That is a 99,5% improvement in runtime with no discernible change in accuracy. The event-based model, when compared to the activity-based model, has the disadvantage of being slightly more complex to implement. It is necessary to manipulate the list of future events, whilst on the activity-based model everything is done solely with the attributes of the modules. This also means that the activity based model uses a lesser amount of computer memory. However, these are small advantages when compared to the decrease in runtime permitted by the event based model. Therefore, the event-based model, and not the activity-based model, shall be used to take conclusions and compare alternatives regarding the system configuration of the power plant. In chapter 5 these aspects are covered.

4.6 CONCLUSION

In this chapter the construction of the simulation model is presented.

In section 4.2 the aspects related to data collection and analysis are covered. It was assumed that the four production groups of the power plant could be considered identical as far their failure characteristics are concerned. Failure data of the modules in the smaller plants were collected and analysed for fitting of probability distributions. The Weibull and exponential distributions were used to model the time between failures and repair times of the modules.

In section 4.3 the reasons and advantages to use Mathematica to implement the simulation model are presented. The activity simulation perspective is, due to its greater simplicity, selected among other perspectives as the basis for the implementation of the model. The elements of the simulation model, namely the entities which in this case are the smaller plant modules, are characterised and the logic of the activities in which the modules engage – failure and end of repair – is defined. Finally, the Mathematica implementation of the model is described in some detail. The model is organised in four Mathematica notebooks. In one notebook the abstract data types or alternatively the building blocks are defined using packages, in a second notebook the simulation functions are defined, in another notebook the main variables are initialised and in a final notebook the main simulation function is operated.

In section 4.4 specific parameters of the simulation model are defined, namely the structure of each model run and the number of runs. The length of each model run was defined as 17520 hours, with the appropriate interruptions corresponding to planned stops. The number of module runs was established as 20. The simulation model is then verified and validated by comparing the results of the model with real-world data on the performance of the power plant. Such comparison allowed the activity-based simulation model to be considered as a valid representation of the power plant.

In section 4.5 the simulation model is enhanced by including the antithetic variates variance reduction technique and implementing the event simulation perspective. The enhanced simulation model performs better than the original simulation model because it reaches equally accurate results while reducing variance by around 55%, thanks to the antithetic variates technique, and reducing runtime by approximately 99,5%, thanks to the implementation of the event simulation perspective.

CHAPTER 5

MODEL APPLICATION

5.1 INTRODUCTION

In this chapter the simulation model is finally used to its main purpose, which is to obtain results and reach conclusions about the system under scrutiny, the power plant.

The enhanced version of the simulation model is used to obtain results. In all circumstances, results are reached after 20 runs of the model, each run following the structure for obtaining results specified in section 4.4. The model results are obtained for one group, and it is assumed the results are extendable to the whole power plant (see section 4.2). The results generated by the model in each run are the following:

a) The total electricity production of the power plant;
b) The total lost production (due to unplanned unavailability);
c) The lost electricity production due to each smaller plant;
d) The total time each smaller plant has been the momentary bottleneck;
e) The total and mean output throughput values of each smaller plant;
f) The number of repairs in each smaller plant;
g) The mean number of available, switched on and switched off modules of each smaller plant.

In section 1.4 it is stated that the main goal of this research is to show whether a decrease in unplanned unavailability at the power plant is feasible namely by identifying the main bottlenecks. In the list of results above, the first four results are those more primarily concerned with these aspects. Therefore these will be the results under focus throughout this chapter.

It was decided to evaluate three different scenarios. Scenario I represents the “business as usual” situation, i.e. it studies the current configuration of the power plant. Scenarios II and III represent two possible alternative configurations of the power plant.

In section 5.2 the main bottlenecks are identified (in scenario I) according to the lost production and bottleneck time identification techniques. In section 5.3 scenario II is evaluated and its results are compared to the scenario I results. In section 5.4 scenario III is evaluated and its results are compared to the scenario I results.

5.2 SCENARIO I

In this section the scenario under study is the current configuration of the power plant, or scenario I. Figures 14 and 15 provide the scenario I results in terms of bottleneck performance. Figure 14 characterises each bottleneck according to the lost production technique:

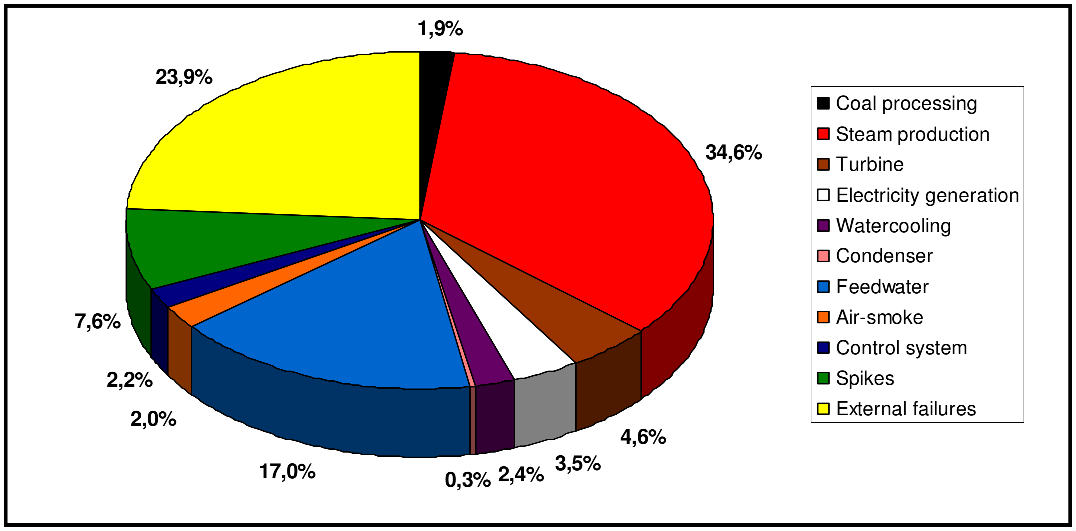

Figure 14: Scenario I bottlenecks according to the lost production technique

It is possible to see that currently the smaller plants responsible for the most lost production are, in this order, the steam production plant, external failures and the feedwater plant. External failures are not really a smaller plant but are referred to in this manner because the "special" situations were modelled as smaller plants. The steam production plant is responsible for 34,6% of the total lost production, while external failures are responsible for 23,9% and the feedwater plant is responsible for 17,0%.

It may be concluded from figure 14 that the steam production plant, external failures and the feedwater plant are responsible for most of the lost production. Together, the three smaller plants account for 75,5% of the total lost production of the power plant.

Figure 15 characterises each bottleneck according to the bottleneck time technique:

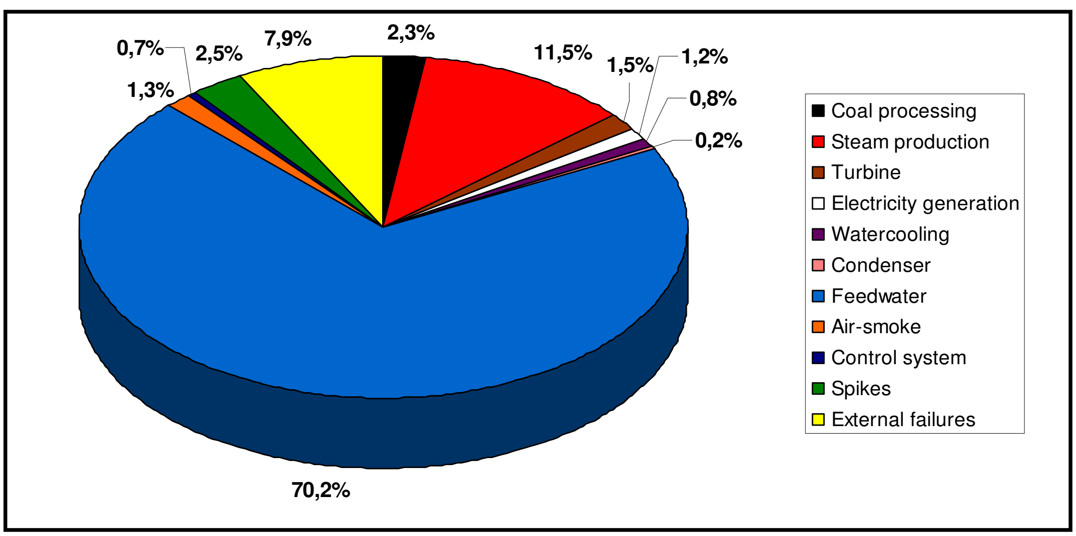

Figure 15: Scenario I bottlenecks according to the bottleneck time technique

It is possible to see that currently the smaller plants responsible for the most bottleneck time are, in this order, the feedwater plant, the steam production plant and external failures. The feedwater plant is responsible for 70,2% of the total bottleneck time, while the steam production plant is responsible for 11,5% and external failures are responsible for 7,9%.

It may be concluded from figure 15 that the feedwater plant, the steam production plant and external failures are responsible for most of the bottleneck time. Together, the three smaller plants account for 89,6% of the total bottleneck time of the power plant.

An important conclusion about the results presented in figures 14 and 15 is therefore that, in both the lost production and the bottleneck time criteria, the top three smaller plants account for most of the bottleneck effect in the power plant. This proves that there are indeed *main* bottlenecks in the power plant. If the bottleneck effect was equally spread through several or all the smaller plants, one could not speak of the existence of one or more main bottlenecks.

Additionally, the main bottlenecks are the same in the two criteria. It is thus possible to say that those smaller plants are the main bottlenecks regardless of the bottleneck criterion that is used, since both criteria point to the same three smaller plants. This is an expectable result, because if one smaller plant is the bottleneck for long periods of time, it is likely that it will also be responsible for much lost production.

These results lead to the conclusion that the main bottlenecks in the power plant are the steam production and feedwater plants and external failures. The first part of the goal of this research, which was the identification of the main bottlenecks in the power plant, has thus been accomplished.

The results on the identification of the main bottlenecks are in accordance to what was expected by the management of the power plant, at least concerning the steam production and feedwater plants. The steam production plant and the feedwater plant were regarded by the management as the main lost production and time bottlenecks, respectively.

Furthermore, the fact that external failures are one of the main bottlenecks is a positive symptom for EDP. It means that a large portion of the bottleneck effect in the power plant is due to external factors for which the company cannot be held responsible. This also allows one to concentrate exclusively on the two other main bottlenecks which are internal to the power plant. The two main bottlenecks of interest are thus the steam production plant and the feedwater plant.

The steam production plant is the main bottleneck as far as lost production is concerned while the feedwater plant is the main bottleneck as far as bottleneck time is concerned. Scenarios II and III deal with these situations separately. In scenario II a proposed change in the configuration of the steam production plant is evaluated and in scenario III a proposed change in the configuration of the feedwater plant is evaluated.

5.3 SCENARIO II

The fact that the steam production plant is one of the main bottlenecks is hardly surprising. The steam production plant, which corresponds to the boiler, may be considered as the “heart” of the power plant, where coal is burned to produce steam. It is thus natural that failures in this section of the process have a pronounced effect in the output throughput of the power plant.

The steam production plant is the main responsible for lost production. Therefore, when attempting to decrease lost production at the power plant, the steam production plant should be the first candidate to configuration changes.

In a preliminary analysis, the following options regarding the steam production plant could be evaluated:

a) Adding one or more modules;
b) Increasing the time between failures;
c) Decreasing the repair time.

The options are qualitatively different. The first option seeks to reduce lost production by adding capacity, while the second and third options seek to reduce it by decreasing the bottleneck time.

The first option, adding modules to the steam production plant, signifies in practice to install more boilers in each group. Boilers are extremely large, expensive and complex systems. Installing an extra boiler would probably mean the construction of a fifth group in the power plant, which obviously does not make sense. The first option is therefore not viable.

The second option, which advocates trying to increase the mean time between failures (for instance through reliability programs or substitution of components), is viable but on the other hand does not appear very promising. The mean time between failures of the steam production plant is around 5043 hours, which is already quite large. Increasing the time between failures would lead to fewer failures, but from table 2 it is possible to see that the steam production plant only failed three times during 2006-2007, for instance. The prosecution of the second option would therefore have a reduced impact in the amount of lost production.

The third option is decreasing the repair time. The mean time to repair a failure in the steam production plant is 67 hours (see table A3 in Appendix C), which is the second longest in the power plant (only the feedwater plant has longer repair times). This characteristic, allied to the fact that a failure in the steam production plant reduces the power output of the power plant to zero (total unavailability), is the main factor behind the large amount of production that is lost at the steam production plant. It seems therefore reasonable to evaluate the effect of a decrease in its mean repair time.

The decrease in the mean repair time of the steam production plant is actually going to be attempted in the real-world system. The main reason why the mean repair time of the steam production plant is so large is that, whenever there is a failure, it is necessary to wait for a long time before the inside of the boiler is cool enough for the repair team to enter the boiler. For the cooling of the boiler to be as fast as possible, the induced-extraction ventilators of the air-smoke plant (see figure 1) are kept extracting air from the interior of the boiler. The management of the power plant proposes to have the other equipments in the air-smoke plant, the primary air and secondary air ventilators, force fresh air into the boiler. This inward air current, together with the outward current maintained by the induced-extraction ventilators, will strengthen the air circulation inside the boiler and accelerate the cooling process.

An opinion from the management indicated that the total time elapsed from the occurrence of a failure in the steam production plant (which includes the time for the boiler to cool down) until the repair team can enter the boiler is approximately 80% of the total repair time. It was also indicated that the previously explained procedure of performing extra ventilation in the boiler will reduce the pre-enter time in 25 to 30%. If a reduction of 30% in the mean repair time of the steam production plant is considered, then the total mean repair time will experience a reduction of (30%)(80%) = 24%. Given that the current (scenario I) mean time to repair of the steam production plant is of 67 hours (see table A3 in Appendix C), then the new repair time would be of (100% – 24%)*67 hours = 50,9 hours or approximately 51 hours.

Scenario II will thus represent the power plant with a mean time to repair in the steam production plant of 51 hours. The first thing to do before comparing the results of scenario II with the results of scenario I is to prove that the two scenarios are in fact different. Remember that a simulation model does not calculate the values of the relevant variables but rather establishes confidence intervals for those values. If the confidence intervals for the values of the relevant variables in two scenarios overlap, then it cannot be considered that the two scenarios represent different outcomes (Pegden et al., 1995).

In this case the relevant variable is the total electricity production in one group during two years of simulated time (see section 4.4). Scenarios I and II were therefore run separately and 99% confidence intervals for the total electricity production were compared. The half-length H of the 99% confidence intervals is determined using equation 14 in section 4.6. The half-length is then added and subtracted to the mean electricity production value to determine respectively the upper and lower endpoints of the interval. Table 7 makes such comparison:

Table 7: Confidence intervals for the electricity production in scenarios I and II

Scenario	Elec. production (MWh)	S (MWh)	H (MWh)	Lower endpoint (MWh)	Upper endpoint (MWh)
I	4884650	23533,9	15055,6	4869594	4899706
II	4912045	24358,4	15583,0	4896462	4927628

Table 7 shows that the confidence intervals do overlap. Therefore the two scenarios cannot be considered as representing two different situations. Of course that a mean time to repair of 67 hours is different from a mean time to repair of 51 hours; however, the overlapping of the confidence intervals indicates that it cannot be statistically proved that such a decrease in the mean time to repair of the steam production plant will have an impact in the performance of the system, at least with a confidence level of 99%.

5.4 SCENARIO III

The feedwater plant is primarily a time bottleneck. From figure 15 it is possible to see that the feedwater plant alone is responsible for more than 70% of the total bottleneck time. This result is a strong motivation to evaluate a change in the configuration of the feedwater plant.

Similarly to the case of the steam production plant, the following changes in the configuration of the feedwater plant might be considered:

a) Adding one or more modules;
b) Increasing the time between failures;
c) Decreasing the repair time.

The mean repair time of the feedwater plant is approximately 68 hours. This is only slightly larger than the 67 hours of the steam production plant (in scenario I). The reason why the feedwater plant has a much more pronounced effect in bottleneck time than the steam production plant is the fact that, in the feedwater plant, the long repair times are allied to the relatively short time between failures (805 hours is a short time in the context of the power plant). Therefore, the evaluation of options b) and c) appears attractive.

The repair times of the feedwater plant are long because, after every failure in the main module, it is necessary to wait for the material to cool down before the repair can begin. This is analogous to the situation in the steam production plant. However, in this case it is not possible to hasten the cooling of the material. Decreasing the repair time is thus not a viable scenario.

Increasing the time between failures could be achieved for instance by resorting to reliability programs or replacing components. Nevertheless, it is not clear what kind of reliability program to use, and it would be difficult to specify which components should be replaced. Additionally, it would not be a straightforward matter to assess the final impact of such measures in the time between failures. Therefore it was decided not to evaluate this option.

The major source of unavailability in the feedwater plant, according to what was exposed in section 3.3, is the electricity consumed by the two backup modules (electrical pumps) when these have to replace the main module (a turbo-pump and auxiliary systems). The unavailability caused is low (10 MW) but because the repair times are long, it eventually translates into relatively large values of lost production, besides the significant effect on bottleneck time.

Adding an extra module to the feedwater plant hence becomes an interesting scenario to analyse. This supposes that the current main module and the extra module would not share the output throughput of the smaller plant; in that case, a failure in one of the modules would mean a half-load situation or 149 MW (298 MW / 2) of unavailability. The objective is that when of the modules is working, the other module is idle (switched off). When the working module fails, it is replaced by the second module. This replacement never generates any loss of production because the proposed extra module is exactly equal to the main module and hence does not consume electricity. The point of having an extra module would be to not having to resort to the two backup modules with the consequent bottleneck effect.

Scenario III therefore represents the power plant with two modules in the feedwater plant instead of one (main) module. Like in scenario II, it is necessary first of all to statistically prove that the two scenarios are, in fact, different, by establishing 99% confidence intervals:

Table 8: Confidence intervals for the electricity production in scenarios I and III

Scenario	Elec. production (MWh)	S (MWh)	H (MWh)	Lower endpoint (MWh)	Upper endpoint (MWh)
I	4884650	23533,9	15055,6	4869594	4899706
III	4917960	25241,6	16148,0	4901812	4934108

It is possible to see that the 99% confidence intervals for the electricity production in scenarios I and III do not overlap. Hence scenarios I and III can be considered as representing two distinctive configurations of the power plant and the electricity production difference may be used to determine the effect of introducing an extra module in the feedwater plant.

Figures 16 and 17 show the bottleneck performance results in scenario III:

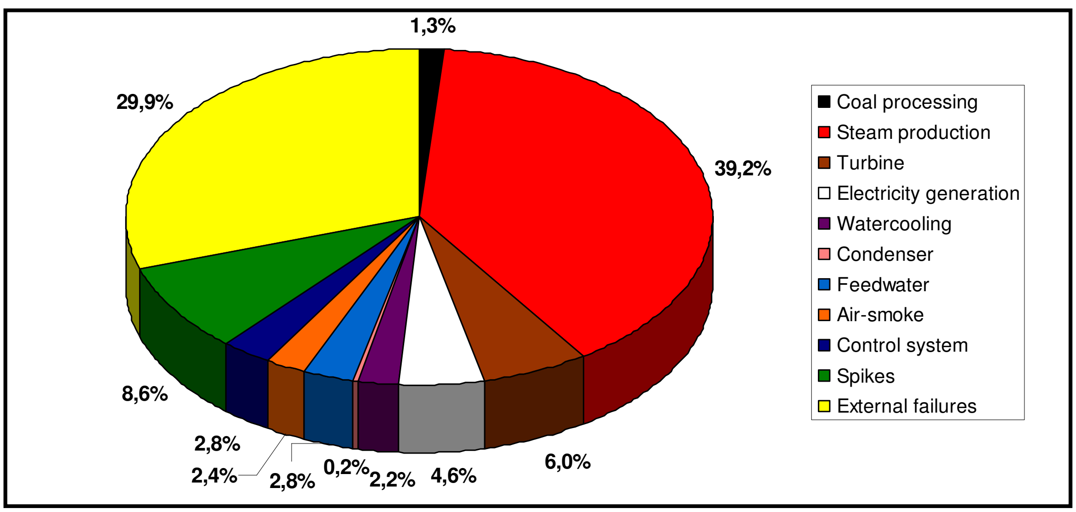

Figure 16: Scenario III bottlenecks according to the lost production technique

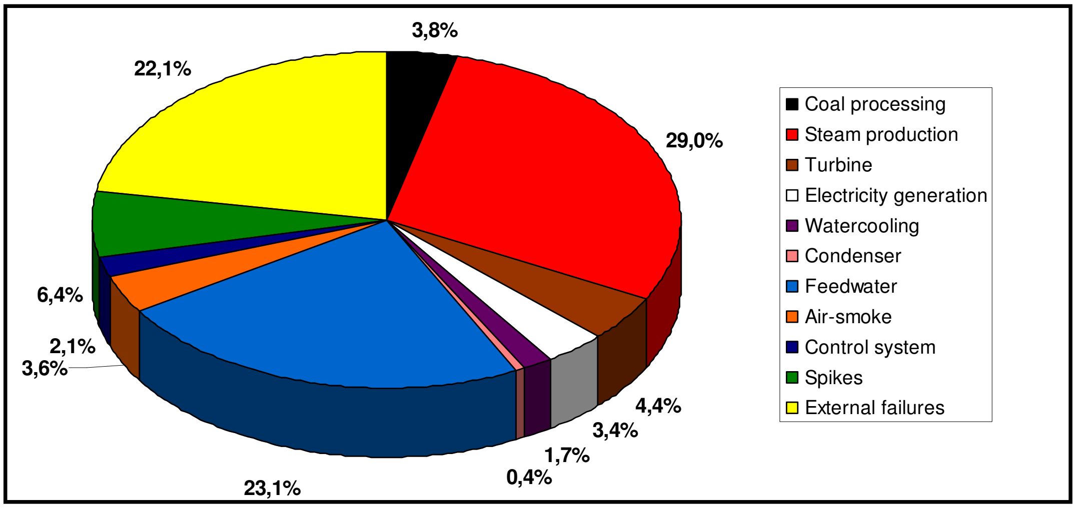

Figure 17: Scenario III bottlenecks according to the bottleneck time technique

When these results are compared to those of scenario I, it is possible to see that the inclusion of an extra module in the feedwater plant has a significant impact in bottleneck performance.

The feedwater plant in scenario III is responsible for only 2,8% of the lost production (see figure 16), a major decrease with respect to the 17,0% verified in scenario I. **In scenario III, the feedwater plant no longer qualifies as an important lost production bottleneck.**

In terms of bottleneck time the impact of the introduction of an extra module has also been visible. The contribution of the feedwater plant, in scenario III, is of 23,1%, a major decrease from the 70,2% verified in scenario I. On the other hand, the bottleneck time percentage of all the other smaller plants increased, but this is because the bottleneck time of each other smaller plant remained approximately constant while the total bottleneck time experienced a decrease. In fact, this is perhaps the most significant impact of the introduction of an extra module in the feedwater plant. The total bottleneck time in scenario I was of 1889 hours, while in scenario III it is of 692 hours. This represents a reduction in total bottleneck time (total time the power plant is producing below the desired value of 298 MW) of approximately 63%.

In table 9 a final comparison between the scenario I results those of scenario III is made:

Table 9: Scenarios I and III – final results

Scenario	Elec. production (MWh)	Lost production (MWh)	Ratio lost production (%)	Bottleneck time (h)
I	4884650	186122	3,81	1889
III	4917960	152811	3,10	692

In scenario III, the total electricity production registers an increase of 33310 MWh (for one group), which corresponds to approximately 112 hours of production or 4,6 production days.

The 4,6 days are gained every two years (remember the model run length is of two years), which means that approximately 2,3 production days are gained each year with the inclusion of an extra module in the feedwater plant.

Table 9 also shows that lost production decreases from 186122 MWh to 152811 MWh or alternatively from 3,80% to 3,10% of the total electricity production.

It is thus possible to conclude that, when an extra module is added to the feedwater plant, the bottleneck effect at the power plant and hence unplanned unavailability is decreased, while the electricity production of the power plant is increased.

5.5 CONCLUSION

In this chapter the main results of the model are presented and discussed.

In section 5.2 the main bottlenecks in the power plant are identified. The identification of the main bottlenecks is made in scenario I, which is representative of the current configuration of the power plant. According to the lost production identification technique, the three main bottlenecks are the steam production plant, external failures and the feedwater plant. According to the bottleneck time identification technique, the three main bottlenecks are the feedwater plant, the steam production plant and external failures. It can be said that the main bottlenecks in the power plant are the steam production and feedwater plants and external failures, because both bottleneck identification techniques point to them and in both criteria these smaller plants together account for most of the bottleneck effect. Only the steam production and feedwater plants need to be analysed given that external failures are not the responsibility of EDP.

Several options regarding the configuration of the steam production plant, with the goal of decreasing its bottleneck effect, are discussed. It is concluded that the most reasonable option would be a decrease in the mean total repair time of the steam production plant. The management of the power plant proposes to enhance the cooling process of the boiler after every failure, resulting in an expected decrease of 24% of the mean repair time of the steam production plant. Scenario II is therefore representative of a 24% decrease in the mean time to repair in the steam production plant. This scenario, however, cannot be proved to represent a different situation than that of scenario I because the confidence intervals of the relevant variable in the two scenarios overlap.

Several options regarding the configuration of the feedwater plant, with the goal of decreasing its bottleneck effect, are discussed. It is concluded that the most reasonable option would be adding an extra module identical to the main module. Scenario III is then defined as representing the inclusion of an extra module in the feedwater plant. The results of this scenario, which was concluded could be considered as different from scenario I, indicate that when such a change is made, the bottleneck effect in the power plant, and in particular that of the feedwater plant, is significantly alleviated and the electricity production increases.

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

In this chapter the most important conclusions of this thesis are presented.

In section 6.2 a summary of the main aspects dealt with in each chapter is made. Each chapter is focused in a specific subsection. In section 6.3 the strengths and weaknesses of the generic methodology are presented. Finally, in section 6.4 the contributions and limitations of the work developed in this thesis are discussed and guidelines for further research are provided.

6.2 SUMMARY

6.2.1 Chapter 1

In chapter 1 the subject of this thesis is presented. Background information about the system under scrutiny in this research, the Sines power plant, is provided. The power plant is run by EDP – *Electricidade de Portugal*, the Portuguese national electrical company, and it employs the normal thermal process for the generation of electricity.

One of the performance measures of power plants is unplanned unavailability. Unavailability means that the power output of the power plant is below the required value by the electrical grid. Unavailability may be planned, if it is caused by a preventive maintenance situation, or unplanned, if it caused by stochastic events such as failures. Unplanned unavailability reduces electricity production and consequently profits. Some sections in the production process can act as important bottlenecks, i.e. be responsible for more unplanned unavailability than others.

The aim of the thesis is defined as developing a computer model of the power plant that allows for the main bottlenecks to be identified and alternative scenarios to be compared. The main goal of the model is to show whether a decrease in unplanned unavailability and hence an increase in electricity production is possible.

6.2.2 Chapter 2

In chapter 2 different decision support tools are compared in their ability to handle the problem presented in chapter 1. It is concluded that, given the complex, stochastic nature of the problem, simulation can handle it more readily than other decision support tools, namely theoretical calculations, mathematical models and samples of the actual hardware.

Different simulation approaches are discussed. It is concluded that a simulation model of the Sines power plant must be dynamic and stochastic and it is shown that the plant possesses both discrete characteristics (present in events such as failures) and continuous characteristics (the continuous production flows). Two kinds of flowsheet simulators are then presented. Steady-state flowsheet simulators cannot be used to build a model of the power plant for the purposes of the research because they are static. Dynamic flowsheet simulators cannot be used either because they represent the system in a greater level of detail than that required.

It is indicated that the most adequate simulation approach would be discrete-event simulation. In this field, however, there are some deficiencies in what regards the simulation of systems with continuous characteristics such as the power plant. The generic methodology developed by Albertyn (2004) is then selected as the basis for this work as it allows for discrete-event simulation of systems exhibiting the characteristics of the power plant. Simulation models developed under this methodology present beneficial design characteristics, namely (Albertyn, 2004): short model development and model maintenance times, short simulation runtimes, robust modelling ability and accurate modelling ability.

6.2.3 Chapter 3

In chapters 3 and 4 the most significant work undertaken in this research is described. It was decided to structure the two chapters following the main steps of a sound simulation study, namely: conceptual development, data collection and analysis, computer implementation, verification and validation and analysis of the results. In chapter 3 the conceptual development of the model is addressed.

Firstly the system under scrutiny, the power plant, is characterised and described. The system has a continuous process, is subject to chronological as well as stochastic events and it manifests complex interrelationships. It was thence concluded that it would be convenient to group the equipments in each section of the production process into smaller plants organised in modules. A formal system description is then performed, breaking the system down to its physical and functional aspects. The physical aspect comprises the elements configuration and module capacities and failure characteristics, while the functional aspect comprises the process flow and the process logic.

The most important concepts of the generic simulation methodology developed by Albertyn (2004) are then presented. These concepts correspond to different methods and techniques which are divided in a “real” part and a “virtual” part. The “real” part features the ERM method, which determines the state of the modules in every smaller plant at any given moment in time. The “virtual” part features: the variables technique, which represents the commodity flows with variables; the FC method, which, at any given moment in time, identifies the momentary bottleneck and determines the actual output throughput vector of every smaller plant; the determination of the number of modules that are switched on and off in each smaller plant, at any given moment in time; the updating of variables that keep a record of the functioning of the simulation model; the determination of the mean values of the relevant variables at the end of the simulation run; and bottleneck prioritisation using the lost production and bottleneck time bottleneck identification techniques.

6.2.4 Chapter 4

In chapter 4 the construction of the simulation model using the concepts developed in chapter 3 is presented. The aspects related to data collection and analysis covered. It was assumed that the four production groups of the power plant could be considered identical as far their failure characteristics are concerned and therefore the model would represent but one group. Failure data of the modules in the smaller plants were collected and then analysed for fitting of theoretical probability distributions. The Weibull and exponential distributions were used to model the time between failures and repair times of the modules.

The reasons and advantages to use Mathematica to implement the simulation model are presented. The activity simulation perspective is, due to its greater simplicity, selected among other perspectives as the basis for the implementation of the model. The elements of the simulation model, namely the entities which in this case are the smaller plant modules, are characterised and the logic of the activities in which the modules engage – failure and end of repair – is defined. Finally, the Mathematica implementation of the model is described in some detail. The model is organised in four Mathematica notebooks. In one notebook the abstract data types or alternatively the building blocks are defined using packages, in a second notebook the simulation functions are defined, in another notebook the main variables are initialised and in a final notebook the main simulation function is operated.

Specific parameters of the simulation model are defined, namely the structure of each model run and the number of runs. The length of each model run was defined as 17520 hours, with the appropriate interruptions corresponding to planned stops. The number of module runs was established as 20. The simulation model is then verified and validated by comparing the results of the model with real-world data on the performance of the power plant. Such comparison allowed the activity-based simulation model to be considered as a valid representation of the power plant.

Finally the simulation model is enhanced by including the antithetic variates variance reduction technique and implementing the event simulation perspective. The enhanced simulation model performs better than the original simulation model because it reaches equally accurate results while reducing variance by around 55%, thanks to the antithetic variates technique, and reducing runtime by approximately 99,5%, thanks to the implementation of the event simulation perspective.

6.2.4 Chapter 5

In chapter 5 the model is used to obtain results, primarily those related to the original goals of the thesis which were to identify the main bottlenecks and show whether an increase in electricity production was possible.

According to the lost production bottleneck identification technique, the three main bottlenecks are, in this order, the steam production plant, external failures and the feedwater plant. According to the bottleneck time identification technique, the three main bottlenecks are, in this order, the feedwater plant, the steam production plant and external failures. It can be said that the main bottlenecks in the power plant are the steam production and feedwater plants and external failures, because both bottleneck identification techniques point to them and in both criteria these smaller plants together account for most of the bottleneck effect. Only the steam production and feedwater plants need be concentrated upon given that external failures are not the responsibility of EDP. The identification of the main bottlenecks is made in scenario I, which is representative of the current configuration of the power plant.

Several possible scenarios regarding the configuration of the steam production plant, with the goal of decreasing its bottleneck effect, are discussed. It is concluded that the most reasonable scenario would be a decrease in the mean total repair time of the steam production plant. The management of the power plant proposes to enhance the cooling process of the boiler after every failure, resulting in an expected decrease of 24% of the total mean repair time of the steam production plant. Scenario II thus represents the situation with a mean time to repair in the steam production plant that is 24% shorter than its current value. This scenario, however, cannot be proved to represent a different situation than that of scenario I because the confidence intervals of the relevant variable in the two scenarios overlap.

Several possible scenarios regarding the configuration of the feedwater plant, with the goal of decreasing its bottleneck effect, are discussed. It is concluded that the most reasonable scenario would be adding an extra module identical to the main module. Scenario III is then defined as representing the inclusion of an extra module in the feedwater plant. The results of this scenario, which was concluded could be considered as different from scenario I, indicate that when such a change is made, the bottleneck effect in the power plant, and in particular that of the feedwater plant, is significantly alleviated and the electricity production increases.

6.3 STRENGTHS AND WEAKNESSES OF THE GENERIC METHODOLOGY

In section 2.3 it is indicated that simulation models developed under the generic methodology possess certain beneficial design characteristics, namely: short model development and model maintenance times, short simulation runtimes, robust modelling ability and accurate modelling ability. These characteristics can be considered as the strengths of the generic methodology. The following list explains why:

a) The development time of a simulation model depends on several factors, such as the purpose for which the model will be used, time and money constraints, the kind of software tool that is used, the skills of the analyst/programmer, etc. Still, it can be said that the generic methodology contributes to reduce development time. The methods and techniques featured in the methodology, namely the variables technique, the ERM method, the FC method and the bottleneck identification techniques are all of simple comprehension to persons with a minimal background in Industrial Engineering (and not only). This does not mean that those methods and techniques are in any way too simplistic or unsophisticated, but it does allow for a reduction of the development time. As an example, the simulation model of the power plant was built in approximately 150 hours, which seems a very acceptable time when compared to the development time of other simulation models;
b) It is reasonable to believe that the above mentioned characteristic also leads to a shorter model maintenance time;
c) The use of the variables technique contributes to short simulation runtimes because it ensures that it is not necessary to make provision for a warm-up period in the beginning of each simulation run. The FC method is a very efficient way to determine the output throughput vector of the power plant, which also impacts positively in the simulation runtime. Furthermore, the generic methodology is independent of the simulation perspective that is used, which means that if the event perspective is selected, short simulation runtimes can be achieved;
d) Robustness in this context means that the generic methodology can be readily employed to accommodate a variety of systems, as long as they are stochastic continuous systems displaying complex interrelationships (Albertyn, 2004). Up until this work the generic methodology had only been applied to the construction of a simulation model of a petrochemical plant, which made Albertyn (2004) claim that it was difficult to substantiate the robustness characteristic of the methodology. The successful application of the generic methodology to the power plant, however, constitutes evidence that it is, indeed, robust;
e) The methods and techniques featured in the generic methodology support its accurate modelling ability:
    i) The variables technique uses real numbers to represent commodity flow rates, posing virtually no limit to the accuracy with which their values can be determined;
    ii) The ERM method assists the calculation of the actual output throughput vector of the power plant by correctly determining the state of the modules in each smaller plant at any given moment in time;
    iii) The FC method accurately determines the actual output throughput vector of the power plant at any given moment in time;
    iv) The updating of variables during and after the simulation run obviously ensure the accuracy of the results;

v) The lost production and bottleneck time bottleneck identification techniques can effectively determine the values of lost production and bottleneck time due to each smaller plant. They do so by covering all possible situations as far the bottleneck status of the power plant at any given moment in time is concerned.

The generic methodology has its weaknesses as well.

One weakness is that the generic methodology does not make provision for transient behaviour (see section 2.3). Such shortcoming means that it is difficult for the methodology to be applied in the context of a dynamic flowsheeter, for instance. Nevertheless, the exclusion of transient behaviour may also be perceived as an advantage, given that it greatly reduces the complexity of the models by paving the way for discrete-event simulation.

Another weakness is the relatively low level of detail in which the various systems and equipments of the power plant, or of any industrial system that is modelled with the methodology, are represented. In order for the generic methodology to be successfully applied to the power plant, the equipments had to be grouped into broader sections or smaller plants. The grouping procedure leads to the loss of information on individual equipments or components. Again, this weakness may be perceived as an advantage because the complexity of the models is reduced. Law and Kelton (2000) warn against the dangers of designing a simulation model with a deeper level of detail than the one necessary.

A final weakness of the generic methodology is the fact that, while it allows for a sound study of the behaviour of the system under scrutiny in terms of material flows, it does not contemplate any energetic aspects. Mass balances to each section in the process are made (through the listing of the module input and output capacities) but energy balances are not considered. That implies that simulation models built under the generic methodology do not make provision for energetic integration as a means to achieve greater energy efficiency in the process, for instance.

6.4 CONTRIBUTIONS AND LIMITATIONS OF THIS WORK

The most important contribution of this work is the fact that the Sines power plant has been provided with a simulation model that accurately represents its behavior, or at least some part of its behaviour. The main benefit of the model, however, resides not on its capability to identify the major bottlenecks in the system given the initial conditions (i.e. the diagnosis of the situation), but on its applicability to study the effect of proposed changes in the system and to compare alternative scenarios. The simulation model thus constitutes an effective decision tool that may be used by the management of the power plant.

It must be said that the simulation model, because it offers a relatively global picture of the behaviour of the system, can be employed for purposes other than bottleneck identification and production increase. Besides bottleneck performance, the model provides information on the output throughput, the number of repairs and the mean number of available, switched on and switched off modules of each smaller plant, at the end of each simulation run. This information may be used as well when taking decisions regarding the configuration of the system.

A strength of the simulation model that was built is that it is to a large extent generic, which is essentially due to the fact that the model building blocks (especially the smaller plants building block) are themselves generic. The building blocks can be used freely to represent what is desired. The model is therefore quite independent of the initial input, namely of the number of smaller plants, the input/output commodities of each smaller plant, the number of modules in each smaller plant, and the characteristics of the modules (input/output capacities and failure characteristics). This indicates that any system exhibiting the characteristics of the systems focused by the generic methodology (having a continuous process, subject to chronological and stochastic events and displaying complex interrelationships), in particular other power plants, can be readily accommodated by the simulation model. An exception to the generic character of the model is the rule of operation regarding the feedwater plant.

Another strength of the simulation model is the separation of the Mathematica program into four different files. That kind of organisation, and namely the fact that the manipulation of variables is made in a specific file, renders the model more user-friendly. Additionally, the choice for Mathematica to implement the model resulted in considerably smaller model size and runtimes than if a specific simulation software package had been used. For example, the event-based simulation model implemented in Simul8 by Albertyn (2004) was 937 KB in size and took approximately 6,8 minutes to perform 20 runs, while the event-based Mathematica model was 168 KB in size and took approximately 10 seconds to perform 20 runs. However, the use of Mathematica is not without disadvantages, as will be demonstrated.

This work also constitutes an innovative contribution to the work of Albertyn (2004), in that:

a) It has contributed to demonstrate the robustness of the generic methodology, by showing that it could be successfully applied to a system other than the petrochemical plant that was the basis for the work of Albertyn (2004). The demonstration of the robustness of the generic methodology is enhanced by the fact that the system under scrutiny, the power plant, is not even in the petrochemical industry;
b) The model was implemented in a general programming language (Mathematica), which is further evidence of the generality and robustness of the generic methodology. The original model developed by Albertyn (2004) was implemented in two simulation software packages. The successful implementation of this simulation model in Mathematica shows that the generic methodology is truly independent of the computer tool that is used;
c) It has demonstrated that the antithetic variates variance reduction technique, which is a significant enhancement to the simulation model, is compatible with the generic methodology.

As for the limitations of this work, one weakness of the simulation model that was built is the loss of generality with respect to the feedwater rule of operation. If this rule were to be modified or eliminated it would be necessary to make changes in the Mathematica code. This seems to be a weakness not only of this simulation model but of the generic methodology itself. It is difficult to develop a generic representation of the particular rules of operation (process logic) of a system, as the rules of operation are always specific of the particular system under scrutiny.

Also, the use of a general programming language to implement the simulation model impacted positively on its size and runtimes but had a negative effect on its user-friendliness. In this aspect the use of a specific simulation software package proves to be much more advantageous. For instance, the Simul8 model built by Albertyn (2004) is entirely presented in one Simul8 graphical screen (with exception for input and output files), whilst in this Mathematica model there are four different files. Moreover, the Simul8 model makes use of graphics and animation – as examples one could mention the buttons for performing actions, the representation of smaller plants with graphical icons or a bottleneck animation that works during the simulation run. Obviously none of this exists in the Mathematica simulation model, which basically is a list of programming code lines that must be evaluated for the model to run.

6.5 POSSIBILITIES FOR FURTHER WORK

In this section some possibilities for further work are mentioned.

One possibility for further work is the application of the simulation model that was developed to power plants other than the Sines power plant. The model is sufficiently generic to permit its application, notwithstanding the specificity of the rule of operation regarding the feedwater plant.

The loss of generality in the rules of operation is certainly a possibility for future developments in the generic methodology. Even though it is impossible to make provision for all possible rules of operation in industrial systems, many kinds of rules are common to a great number of systems because they represent more or less universal concepts. The generality of the generic methodology may thus be improved by the inclusion of a “library” from where such concepts could be easily selected and implemented in the model under construction (Albertyn, 2004).

The generic methodology has the potential to propel the development of a specific simulation software package solely for its application. Alternatively, the methodology could be implemented in licensed templates of existing software packages, for instance. This would facilitate its widespread use by analysts, engineers, students, etc. The user-friendliness of the generic methodology could also be improved by enhancing the manipulation of the building blocks, namely through the use of icons, the “drag and drop” functionality and building block menus. The development of a concise, simplistic and user-friendly generic methodology manual is intrinsically related to these concepts and therefore constitutes a possibility for further work as well (Albertyn, 2004).

REFERENCES

Albertyn, M. (2004). *Generic Simulation Modelling of Stochastic Continuous Systems*. *PhD* Thesis. Faculty of Engineering, Built Environment and Information Technology, University of Pretoria, South Africa.

Banks, J., Carson II, J., Nelson, B. and Nicol, D. (2005). *Discrete-Event Simulation*. 4th Edition, Pearson Prentice Hall, Upper Saddle River, New Jersey.

Barton, P. and Pantelides, C. (1994). Modeling of Combined Discrete/Continuous Processes. *AIChE Journal*, *40*(6).

Carmo, J., Sernadas, A., Sernadas, C., Dionísio, F.M. and Caleiro, C. (1999). *Introdução à Programação em Mathematica*. IST Press, Lisbon.

Cellier, F. (1977). Combined Discrete/Continuous System Simulation Languages – Usefulness, Experiences and Future Development. *ACM SIGSIM Simulation Digest*, *9*(1), 18-21.

Companhia Portuguesa de Produção de Electricidade (CPPE) (1999). *Central Termoeléctrica de Sines*. Edition of GCD – Gabinete de Comunicação e Documentação CPPE, Lisbon.

Crow, E.L., Davis, F.A. and Maxfield, M.W. (1960). *Statistics Manual*. Dover Pubs., New York.

Diáz, M. and Garrido, D. (2004). A Simulation Environment for Nuclear Power Plants. *Proceedings of the Eighth International Symposium on Distributed Simulation and Real-Time Applications*, pp. 98-105.

Diáz, M., Garrido, D., Romero, S., Rubio, B., Soler, E. and Troya, J. (2006). Experiences with Component-Oriented Technologies in Nuclear Power Plant Simulators. *Software - Practice and Experience*, *36*, 1489-1512.

Eich-Soellner, E., Lory, P., Burr, P. and Kröner, A. (1997). Stationary and Dynamic Flowsheeting in the Chemical Engineering Industry. *Surveys on Mathematics for Industry*, *7*, 1-28.

Fagley, J. and Carnahan, B. (1990). The Sequential-Clustered Method for Dynamic Chemical Plant Simulation. *Computers and Chemical Engineering*, *14*, 161-177.

Fahrland, D. (1970). Combined Discrete-Event Continuous Systems Simulation. *Simulation*, *14*(2), 61-72.

French, S. (1986). *Decision Theory: An Introduction to the Mathematics of Rationality*. Ellis Horwood, Chichester, West Sussex.

Fontes, J. (2007). *Degradação de Materiais e Vida Restante em Caldeiras de Centrais Termoeléctricas Convencionais*. Master's Thesis. Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisbon.

Garvin, D., Roberto, M. (2001). What you don't know about making decisions. September 2001 Issue, Harvard Business Review, Cambridge, Massachussets.

Gheorghe, L., Bouchhima, F., Nicolescu, G. and Boucheneb, H. (2007). A Formalization of Global Simulation Models for Continuous/Discrete Systems. *Proceedings of the Summer Computer Simulation Conference,* pp. 559-566.

Gundersen, T. and Hertzberg, T. (1983). Partitioning and Tearing of Networks – Applied to Process Flowsheeting. *Modelling, Identification and Control*, *4*(3), 139-165.

Harrell, C. and Tumay, K. (1999). *Simulation Made Easy: A Manager's Guide*. 5th Edition, Engineering and Management Press, Norcross, Georgia.

Hillier, F. and Lieberman, G. (2005). *Introduction to Operations Research*. 8th Edition, McGraw-Hill, New York.

Hills, P.R. (1971). *HOCUS*. P-E Group, Egham, United Kingdom.

Kalisz, E. (1993). Continuous Systems Simulation using GPSS/H. *ACM SIGSIM Simulation Digest, 22*(3), 3-10.

Kleinschmidt, H. (1990). Study Notes: Systems Engineering Post Graduate Course in the Department of Industrial and Systems Engineering. University of Pretoria, Pretoria.

Law, A. and Kelton, W. (2000). *Simulation Modeling and Analysis*. 3rd Edition, McGraw-Hill, New York.

Lee, K. and Yoon, E. (1994). The Flexible Modular Approach in Dynamic Process Simulation. *Computers and Chemical Engineering*, *18* (Suppl.), s761-s765.

Leiviskä, K. (1996). Simulation in Pulp and Paper Industry. Process Simulation Course, Control Engineering Laboratory, University of Oulu, Oulu.

Maeder, R. E. (1994). *The Mathematica Programmer*. Academic Press, 1994.

Nutaro, J. (2007). Discrete event simulation of continuous systems. In *Handbook of Dynamic System Modeling*, P. Fishwick (ed.), Arizona Center for Integrative Modeling and Simulation, University of Arizona & Arizona State University, Arizona.

Ordys, A. and Clarke, P. (1999). Constrained Predictive Control for Multivariable Systems with Application to Power Systems. *International Journal of Robust Non-Linear Control*, *9*, 781-797.

Pegden, C., Shannon, R., Sadowski, R. (1995). *Introduction to Simulation Using SIMAN*. 2nd Edition, McGraw-Hill, New York.

Pidd, M. (2004). *Computer Simulation in Management Science*. 5th Edition, John Wiley & Sons, Chichester, West Sussex.

Porta Nova, A. (2000). *Apontamentos de Simulação*. Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisbon.

Prasad, G., Irwin, G., Swidenbank, E. and Hogg, B. (2000). Plant-wide Predictive Control for a Thermal Power Plant based on a Physical Plant Model. *Control Theory and Applications*, *147*(5), 523-537.

Sargent, R. and Westerberg, A. (1964). Speed-up in Chemical Engineering Design. *Trans. Inst. Chem. Eng.*, *42*, 190-197.

Shacham, M., Machietto, S., Stutzman, L. and Babcock, P. (1982). Equation Oriented Approach to Process Flowsheeting. *Computers and Chemical Engineering*, *6*(2), 79-95.

Stadtherr, M. and Vegeais, J. (1985). Recent Progress in Equation-Based Process Flowsheeting. *Proceedings of the Summer Computer Simulation Conference*, pp 325-330.

Taha, H. (1987). *Operations Research: An Introduction*. 4th Edition, MacMillan, New York.

Tavares, L., Oliveira, R., Themido, I. and Correia, F. (1996). *Investigação Operacional*. McGraw-Hill Portugal, Lisbon.

Vasandani, V., Govindaraj, T. and Mitchell, C. (1989). An Intelligent Tutor for Diagnostic Problem Solving in Complex Dynamic Systems. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2,* pp. 772-777.

APPENDIX A

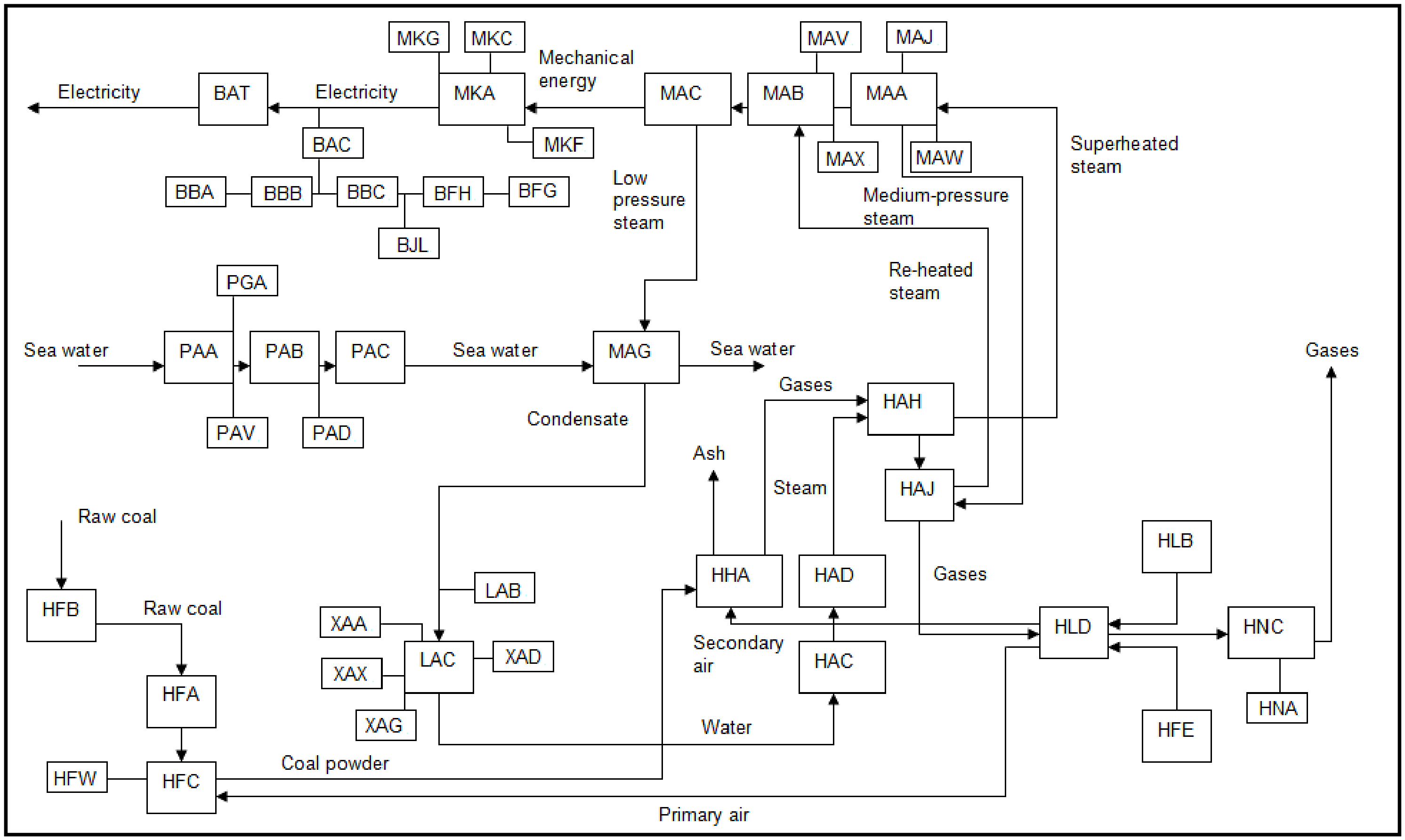

Figure A1: Block-diagram of the systems prone to relevant failure in the power plant

Remarks about figure A1

Both the main systems and the auxiliary systems are referred to by their *KKS* designation. The *KKS* (*Kraftwerkskennzeichnungssystem*, in German) is an international designation system for power plants. Here is the key of figure A1 with the *KKS* system designations:

BAC – generator circuit breaker	MAA – high-pressure turbine
BAT – generator transformer	MAB – medium-pressure turbine
BBA-C – medium voltage distribution boards	MAC – low-pressure turbine
BFG/H – low voltage distribution boards	MAG – main condenser
BJL – low voltage subdistribution boards	MAJ – air removal system
HAC – economiser system	MAV – lubricant supply system
HAD – evaporator system	MAW – sealing, heating and cooling system
HAH – high-pressure superheater system	MKA – generator
HAJ – re-heat system	MKC – generator exciter set
HFA – bunker for pulverizing system	MKF – stator/rotor liquid cooling system
HFB – feeder system	MKG - stator/rotor gas cooling system
HFC – pulverizing system	PAA – extraction system
HFE – mill air system	PAB – circulating water piping/culvert system
HFW – sealing fluid supply system	PAC – water pump system
HHA – main burners	PAD – recirculating/outfall cooling system
HLB – forced-draught fan system	PGA – closed cooling water system
HLD – air heating system	XAA – auxiliary high-pressure turbine
HNA – ducting system	XAD – bearings
HNC – induced-draught fan system	XAG – condensating system
LAB – feedwater piping system	XAX – control and protection equipment
LAC – feedwater pump system	

Figure A2: *KKS* codes of the systems prone to relevant failure in the power plant

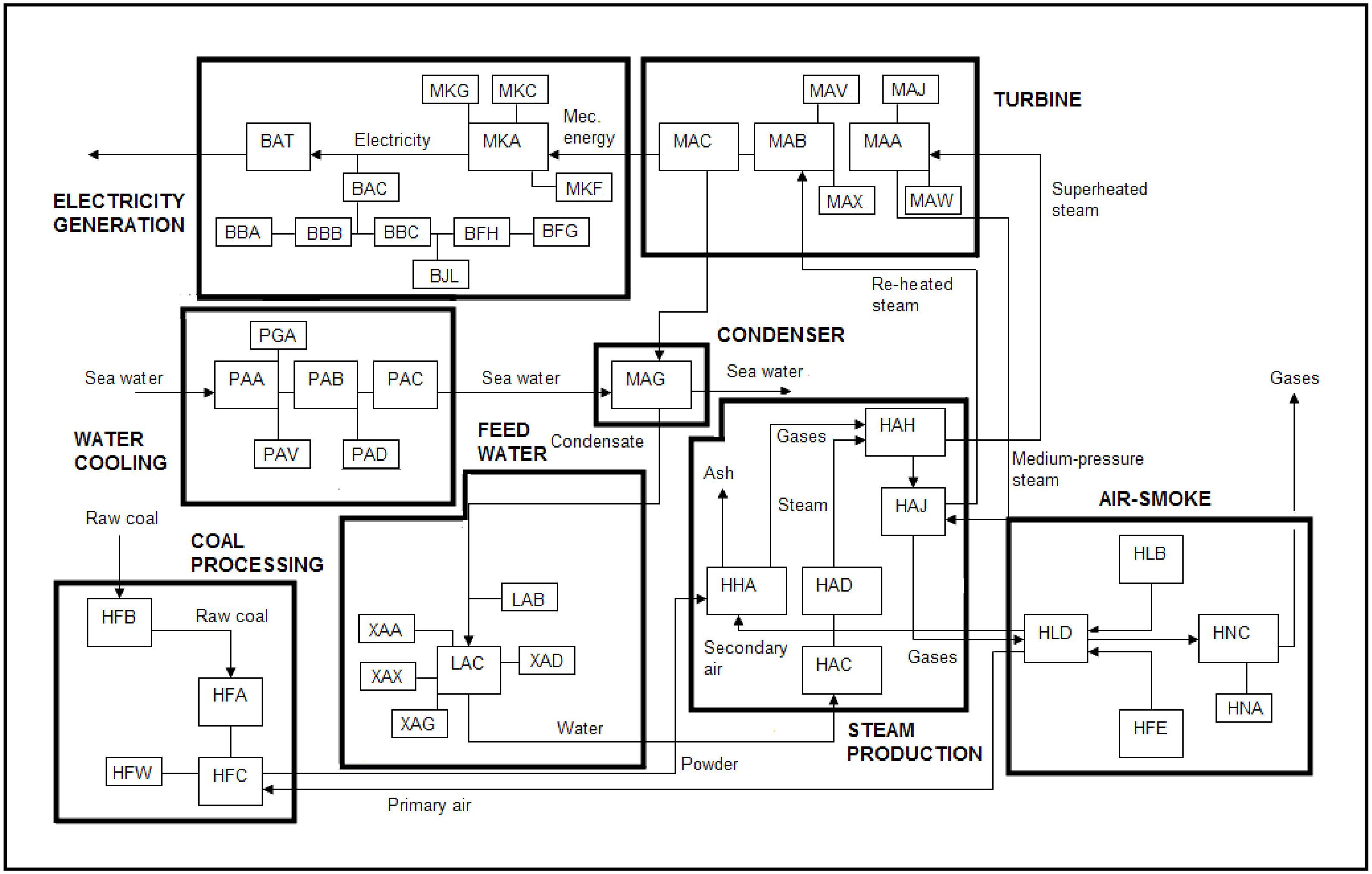

Figure A3: Systems grouped into smaller plants

APPENDIX B

Table A1: Number of modules and input/output capacities

No.	Name	Mod.	Capacity in (ton/h)	From	Capacity out (ton/h)	To
1	Coal processing	5	Raw coal - 29,07 Primary air – 92,61	(-) No. 8	Coal powder – 29,07	No. 2
2	Steam production	1	Water – 952,56 Coal powder – 116,28 MP steam – 846,72 Secondary air – 941,76	No. 7 No. 1 No. 3 No. 8	SH steam – 950,4 RH steam – 846,72 Gases – 1312,20 Ash – 7,44	No. 3 No. 3 No. 8 (-)
3	Turbine	1	SH steam – 950,40 RH steam – 846,72	No. 2 No. 2	LP steam – 846,72 MP steam – 846,72	No. 4 No. 2
4	Electricity generation	1	LP steam – 846,72	No. 3	Electricity – 298 (MW) LP steam – 846,72	(-) No. 6
5	Water cooling	1	Sea water – 36000000*	(-)	Sea water – 36000000*	No. 6
6	Condenser	2	LP steam – 423,36 Sea water – 18000000*	No. 4 No. 5	Condensate – 423,36 Sea water – 18000000*	No. 7 (-)
7	Feedwater[1]	1	Condensate – 846,72 Fresh water – 105,84	No. 6 (-)	Water – 952,56	No. 2
8	Air-smoke	2	Gases – 656,10	No. 2	Gases – 656,10 Primary air – 185,22 Secondary air – 470,88	(-) No. 1 No. 2

* - m^3/h
1 – Possesses two backup modules, each of capacities approx. 48% of the ones indicated.

Table A2: FC method parameter set

No.	Name	Steady-state actual output throughput
1	Coal processing	Coal powder – 116,28 ton/h
2	Steam production	SH steam – 950,40 ton/h RH steam – 846,72 ton/h Gases – 1312,20 ton/h Ash – 7,44 ton/h
3	Turbine	LP steam – 846,72 ton/h MP steam – 846,72 ton/h
4	Electricity generation	Electricity – 298 MW LP steam – 846,72 ton/h
5	Water cooling	Sea water – 36.000.000 m^3/h
6	Condenser	Condensate – 846,72 ton/h Sea water – 105,84 ton/h
7	Feedwater	Water – 952,56 ton/h
8	Air-smoke	Gases – 1312,20 ton/h Primary air – 370,44 ton/h Secondary air – 941,76 ton/h

APPENDIX C

Table A3: Module failure characteristics

No.	Name	Time between failures			Time to repair		
		Distribution	α	β	Distribution	α	β
1	Coal processing	Weibull	1,19	2510	Weibull	1,23	30,0
2	Steam production	Weibull	1,35	5500	Exponential	-	67,0
3	Turbine	Weibull	1,24	3690	Exponential	-	6,5
4	Electricity generation	Weibull	1,20	3100	Weibull	1,30	4,5
5	Water cooling	Weibull	1,17	13030	Weibull	1,18	15,0
6	Condenser	Weibull	1,33	24480	Exponential	-	3,0
7	Feedwater	Weibull	1,20	805	Weibull	1,23	73,0
8	Air-smoke	Weibull	1,22	13600	Weibull	1,13	14,0

Table A4: “Special” situation occurrence characteristics

Name	Time between occurrences			Duration		
	Distribution	α	β	Distribution	α	β
Control system	Weibull	1,10	6205	Weibull	1,40	6,0
Peaks	Weibull	1,15	2290	Exponential	-	6,0
External failures	Weibull	1,18	1950	Weibull	1,09	18,0

Table A5: Data for validation of the simulation model

Period	Electricity production (MWh)	Lost production (MWh)
2006-2007	4.342.301	163.460

APPENDIX D

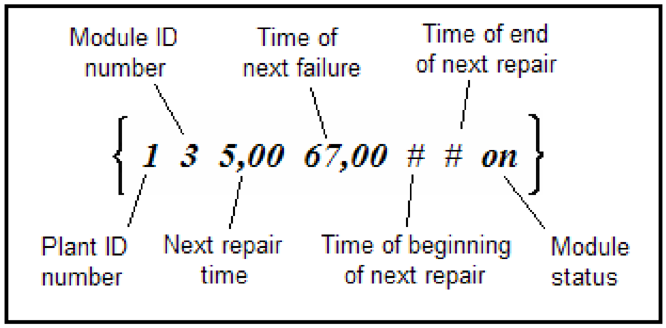

Figure A4: Module Mathematica building block (example)

Note: the times of the beginning and end of the next repair can only be specified after the module fails and the status of the repair queue is known, so they are represented by the symbol "#" in the rest of the time.

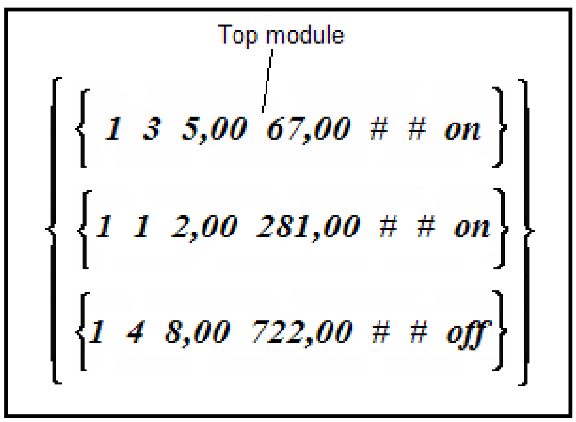

Figure A5: Stack Mathematica building block (availability stack example)

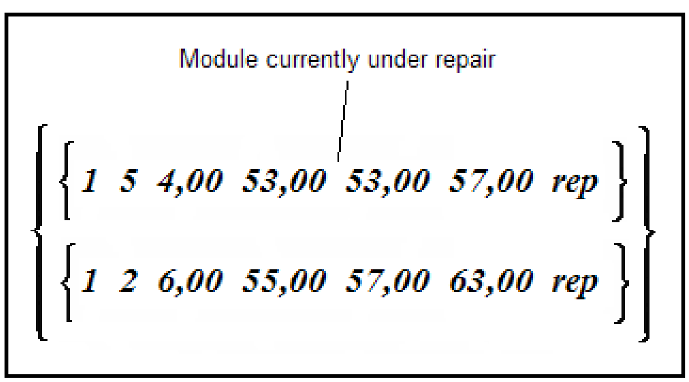

Figure A6: Stack Mathematica building block (repair stack example)

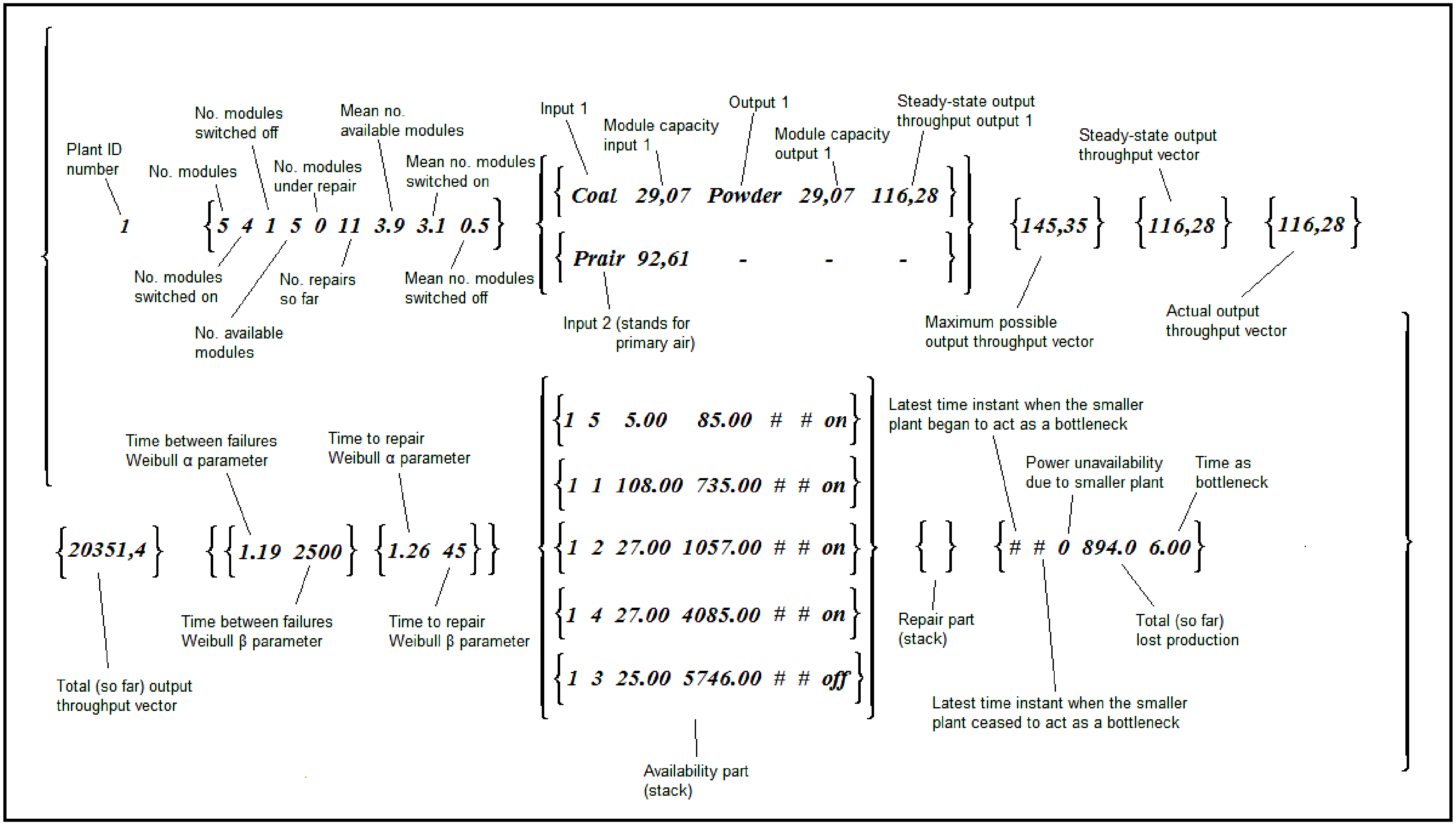

Figure A7: Smaller plant Mathematica building block (example)