Advances in Simulation for
Production and Logistics Applications
Markus Rabe (ed.)
Stuttgart, Fraunhofer IRB Verlag 2008

Deploying Decision Support Systems Based on Simulation Models

Einsatz von Entscheidungsunterstützungssystemen auf der Basis von Simulationsmodellen

Rienk Bijlsma, Vincent De Gast, Edwin Valentin,
Systems Navigator, Delft (Netherlands)

Abstract: Simulation environments hardly offer support to perform a set of experiments and analyze these experiments with a group of people. Automod and Arena provide some functionality and optimization and experimental design are available as techniques, but in practice everyone makes their own spreadsheets with overviews of parameters and performance indicators. We define a range of requirements to enhance the support of performing simulation experiments together with a group of stakeholders, whereby the stakeholders should be able to trigger and analyze scenarios without support from a simulation expert. Based on these requirements we have designed a database application called Scenario Navigator. That results in a user friendly interface to define scenarios within a simulation study for several simulation models. The performance indicators of the scenarios are represented in customized reports so stakeholders in an organization benefit from the ability to perform scenarios faster and evaluate results better.

1 Introduction

Shannon (1975, p. 2) defines simulation as "the process of designing a model of a concrete system and conducting experiments with this model in order to understand the behaviour of a concrete system and/or to evaluate various strategies for the operation of the system". Focus in a simulation study is to understand a system by performing experiments. The process of performing these experiments require that all stakeholders that are involved in a simulation study can define possible alternatives that they would like to be evaluated, so they can understand the effects of their suggestions. The understanding of a system and possible alternatives also requires that the team that is working in the simulation study, i.e. model developers, model analysts, data providers and problem owners together, can introduce new possible solutions and analyze the outcome of the simulation runs of the possible solutions (van Meel 1994).

Taylor et al (2001) describe the use of a tool called GroupSim to generate ideas and suggestions as part of a group session to identify new experiments. GroupSim stops at the moment that the experiments are defined and GroupSim does not offer features to the stakeholders for analyzing the outcome. Further, the preparation of a simulation model to run the simulation experiment and to combine the performance indicators is a time consuming process. During this process the stakeholders can easily define a dozen new simulation experiments. As a result the stakeholders are all waiting during the meeting for the first simulation experiment to finish.

The time consumption of experimentation with a simulation model is mainly the activity of entering the parameters in a model and evaluation of the outcome. There are several possibilities which are statistically solid approaches to minimize the number of experiments to perform (Kleijnen 1977) or result in automatic optimization and providing the best fitted result (Krug 2001). The strength of these processes is in situations where a mathematical equation can be defined to judge what is the best performance of a system. Unfortunately, in situations with several stakeholders involved there is not one best solution, but there are a range of possibilities all of them having advantages and disadvantages. The approaches as described by Kleijnen (1977) and Krug (2001) are thus not suited as a generic solution for support to stakeholders in the simulation experimentation process.

Some commercial simulation environments offer tools and systems to support stakeholders in a simulation study. The support is mainly provided to a model developer who can adjust faster a simulation model and thus start faster running the simulation experiments. For example, the simulation environment Arena offers a tool called Process Analyzer (Kelton et al. 2007) and the simulation environment Automod provides an instrument for performing experiments called AutoStat (Banks 2000). In section 2 is described what are the shortcomings of these additional instruments, together with some custom-made interfaces to a simulation model.

We describe a generic database tool that combines the power of the instruments Process Analyzer and AutoStat with automatic optimization in this paper. We used our lessons learned from years of simulation experimentation and designed this database tool in such a way that multiple stakeholders can cooperate to initiate and evaluate simulation experiments. The need for such an application is described in section 2 where we evaluate current ways of interfacing to simulation models. In section 3 we translate the identified shortcomings to requirements for a new tool. Section 4 describes a data model and interaction between a database tool and simulation environments. Section 5 introduces the implemented solution which we call Scenario Navigator and in chapter 6 we draw some conclusions about applicability and future research in the field of practical simulation studies and support of experimentation.

2 Background

Hillen (1993) describes that performing experiments in a simulation study consists of three steps:

1. Define experiment conditions

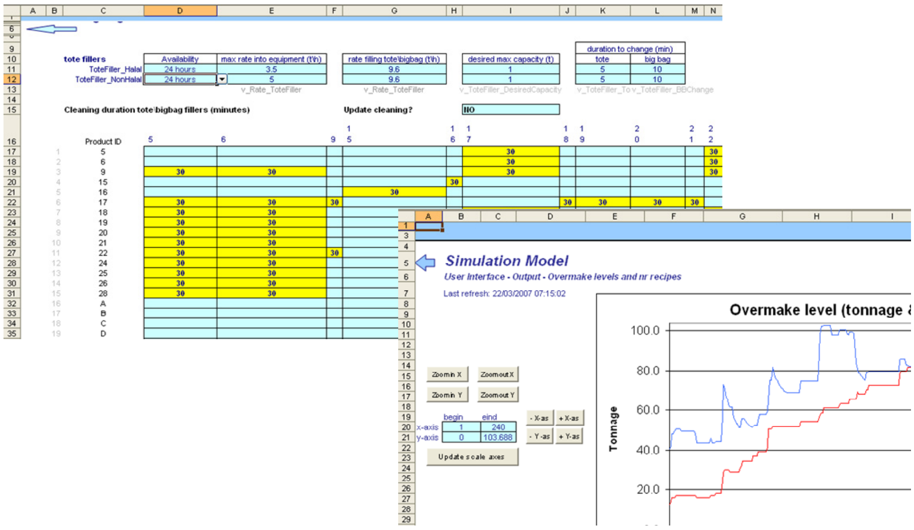

Figure 1: *Dedicated Excel Interface for Data Entry*

2. Execute simulation run

3. Register results

These three steps are performed just as long as until the problem owners are satisfied and have gathered sufficient insight in the possible solutions. According to Robinson (1999) the simulation study duration or the budget are limited and thus the insight should be provided before the end. Developers of simulation environments have been working over the past 10 years in enabling model developers to be able to handle all three activities in performing simulation experiments faster and better. They help model developers by entering data more easily via user interfaces to modules, links with Excel sheets or model developers can make customized Visual Basic for Applications (VBA) interfaces. The run time of a simulation run has been mainly improved by faster more powerful computers and registration of results is improved by standardized reports, writing data to databases and export to Excel sheets.

The development of VBA interfaces to a simulation model or an Excel sheet (example in fig. 1) with data, are time consuming efforts and require a lot of development time of model builders. These VBA interfaces or Excel sheets are often customized for a simulation model and are developed over and over again. Only the simulation experts who are regularly developing simulation models in the same domain can benefit from previously developed interfaces and will be able to work with minor adjustments. The advantage of these custom interfaces is that advanced reports can be made with fancy graphs of different variables in the simulation model. Nevertheless, our experience is that these interfaces are fixed with one simulation model and they do not offer the ability to evaluate performance indicators across different scenarios.

The simulation environment Arena offers an instrument that is specifically developed for defining a range of simulation experiments and comparing performance indicators of these experiments. This tool is called Process Analyzer (Kelton et al.

2007) and also enables scheduling of simulation experiments using different simulation models. The ability to use different simulation models is required to enable simulation experiments that are more than just a parameter setting, for example a complete new production line or a new control mechanism to allocate jobs to a set of resources. Figure 2 shows Process Analyzer of Arena including 15 different scenarios that are ready to be executed. The grey column at the right of figure 2 will contain the values that are returned from the simulation environment once the simulation runs for these scenarios are performed. In Process Analyzer only quantitative performance indicators can be observed, in this case average waiting time. Process Analyzer does not show graphs of change values over time, no confidence intervals and no nice representation for a lot of performance indicators.

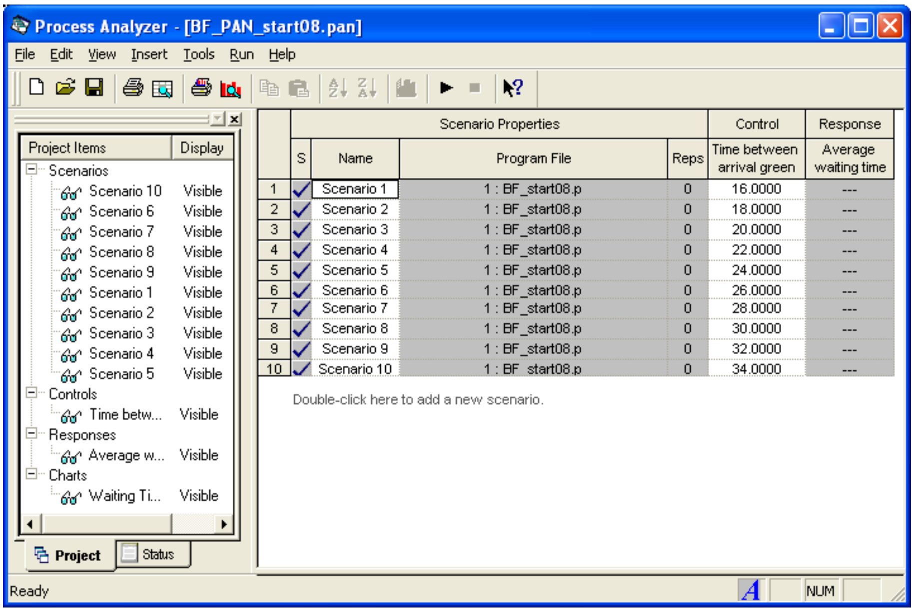

Figure 2: *Process Analyzer of Arena*

Another custom tool is provided by Automod and named AutoStat (Banks 2000). AutoStat is a tool that provides data analysis like confidence intervals and required warm-up period for a simulation run. Unfortunately, this custom tool can not provide analysis of several experiments. All mathematical power is organized for one experiment and thus does not support multiple experiments as commonly performed in a simulation study.

Optimization technology in combination with simulation is available for support to identify the optimum in a system by changing single parameter settings. Optimization tools like OptQuest (April et al. 2004) or Issop (Krug 2001) provide different optimization algorithms to find the optimum configuration of a set of parameters. The optimization instrument feeds the simulation with a parameter setting, the simulation runs and returns one number (the goal function). Figure 3 shows the value of an object value, the best identified set of parameters and the current evaluation for one simulation model after over 35 experiments. The disadvantage of optimization is that a goal function needs to be defined. A single goal function can not always be defined. For example, a higher utilization versus a lower cost is hard to express in a goal function.

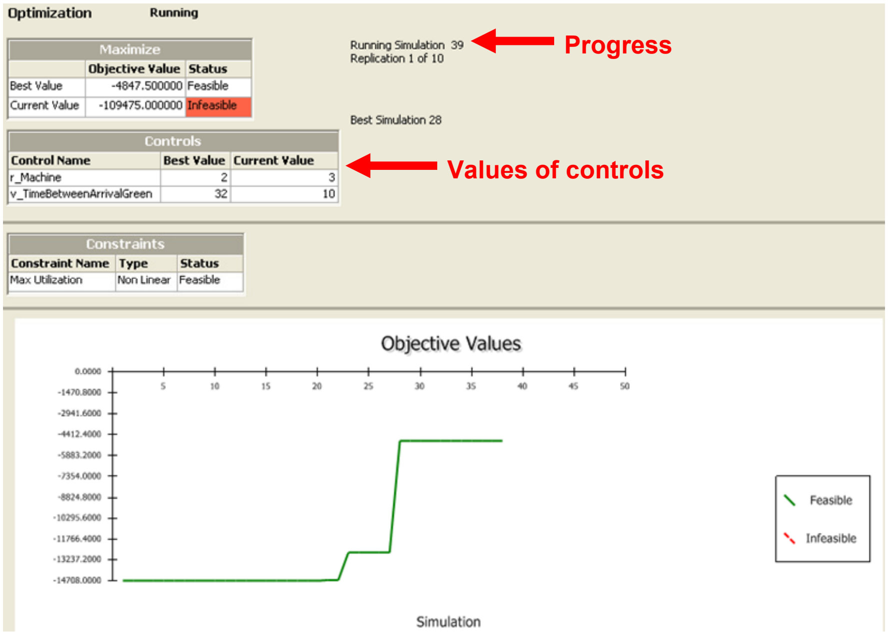

Figure 3: *Results of Optimization: Best and Current Run*

Kleijnen (1977) offers mathematical approaches to reduce the number experiments. He describes a methodology of metamodels to evaluate the influence of values and their sensitivity. The metamodels result in matrixes that define the solution space for a set of experiments. Via the metamodels is determined what a success factor is of a simulation experiment. Some of the unlikely experiments can be excluded by determining the success of an experiment be forehand and thus reducing the number of experiments to be performed. This approach has theoretical advantages, but has one major disadvantage: stakeholders do not like that their suggestions are ignored, even with the best mathematical proof. Further, setting up a valid metamodel of a system is not a straightforward job and requires stakeholders to be strong in concepts.

3 Requirements Support Experimentation

In order for multiple stakeholders to work effectively together while performing experiments in a simulation study we have identified the need for a generic support tool that can interact with simulation modeling environments. We identify requirements for the users (3.1), the different phases of model experimentation as described by Hillen (1993), i.e. model adjustment (3.2), perform simulation run (3.3) and analysis of output (3.4). Further we identify requirements to support the advanced mechanisms such as optimization (Krug 2001) and experimental analysis (Kleijnen 1977). Finally we define requirements for computer systems and network availability (3.6).

3.1 Requirements for Stakeholders Simulation Study

The functionalities and features offered by simulation environments for performing experiments are mainly targeted at the model builders. The application for performing simulation experiments and comparing results should focus on all stakeholders in a simulation study, thus also the people who are involved and do not have simulation expertise. Enabling simulation novices to use such an application provides the following requirements:

- *Application should be easy understandable;* problem owners can not be expected to spend weeks of training to define a new experiment or to compare two experiments
- *All stakeholders should be able to define experiments*
- *All stakeholders should be able to view experiment results*
- *Data entry should be user friendly;* data providers should be willing to use the application and not be scared by unfriendly use
- *Reports should be advanced and provide insight;* less is often more. The reports that come out of a normal simulation run can be dozens of pages long, while only a couple of key performance indicators have to be evaluated.
- *Stakeholders should be able to use the application at their own desktop;* if the computer application is a standalone system, then users will not use the application and are not using the possibilities.
- *User rights should be defined to enable confidentiality of simulation study results;* In a simulation study with different stakeholders some insight might be confidential and cannot be set, viewed or adjusted by all individuals in the simulation study. One must be able to define multiple roles for different type of users. Possible roles and user rights can be; model builder/solution designer, data administration, administrator, decision maker and viewer.

3.2 Model Data Management

All experiments that are performed in a simulation study require a set of data, run definitions like length of the run and warm up period and a simulation model. The application should provide support for all stakeholders in the simulation study to introduce new experiments and view already defined experiments. This requirement is divided in the following requirements for the application:

- *Manual data entry should be possible*
- *Automatic data entry from external sources should be possible;* most simulation models require data from external systems for example historic order entries, sales forecasts, or process durations and failure characteristics. In order

to make this available to the end user, solid data connections must be made to external systems, which can be uploaded on user demand. For example sourcing form MIS, MES and ERP systems

- *Reusing data already entered in a previous experiment should be possible;* experiments are not individual events, but triggered by outcomes of a previous defined experiment.
- *Application should keep history of defining experiments;* with different stakeholders involved, it is needed to know who has developed or adjusted an experiment and when this has happened.
- *Stakeholders should be able to compare input parameters between experiments;* besides results of a simulation study, stakeholders should be able to check what where the differences between experiments and what effects these experiments have.
- *Stakeholders should be able to easily define a set of experiments;* If a stakeholder wants to perform 10 experiments in which one input parameter changes values from 1 to 10, then this stakeholder does not want to define these experiments all one by one. He or she wants to use a one button approach that automatically defines these 10 experiments.
- *Experiments should be defined with different simulation models;* two experiments might use the same input data, but use an alternative simulation model that contains a new process or logic.
- *The experiment should contain information for run setup;* every simulation experiment requires to know a run length, warm up period and number of replications to be able to collect statistics and represent a system valid.
- *Should be able to schedule experiments later in time;* Taylor et al. (2001) describe that generating ideas for simulation runs is easily done, but that collecting all data might take some time and running an experiment might require some computer power. Therefore, performing the actual experiments might be postponed until sufficient data is collected and CPU power is available.

3.3 Simulation Engine Connectivity

Performing the simulation experiments requires automatic control of a simulation environment. The automatic activities are mentioned in the following requirements:

- *Application should be able to trigger a simulation environment to open a simulation model*
- *Application should be able to adjust parameters of the simulation model according to parameters defined in experiment*
- *Application should be able to adjust run setup of a simulation model according to run setup defined in experiment*
- *Application should be able to perform a simulation run with the simulation model in the simulation environment*

- *Application should be able to collect performance indicators of simulation run;* the collection of the performance indicators should be done during the run of the simulation to be able to make graphs and at the end of each replication to be able to define some overall performance indicators.
- *Application should be able to connect to different simulation environments;* it is uncommon that within one simulation study several different simulation environments are used, but it does occur, thus the application should be able to work together with different simulation environments, possibly even simulation environments that apply different formalisms and not just discrete event.

3.4 Scenario Analysis

The key issue of the application is to provide insight to the stakeholders in the simulation study. This insight is achieved by providing reports that contain performance indicators of the simulation experiments. The reports should match the following requirements

- *Reports should contain focused performance indicator;* not all performance indicators should be provided in one report
- *Stakeholders should be able to export a report;* reports of performance indicators should be added to Word documents or Powerpoint presentations to higher management.
- *Simulation model should represent performance indicators in several reports;* depending on the analysis, the stakeholders in the simulation study might want to view a different set of performance indicators of the simulation experiment
- *Reports should contain data of one or more scenarios;* reports should be defined with information of single experiments, but also reports that show performance indicators of different experiments to enable comparison between experiments.
- *Stakeholders should be able to define their own customized reports;* stakeholders know what they would like to see in a report of a simulation experiment. A couple of predefined generic reports can be made, but the stakeholders should be able to define their own report in a customized way.

3.5 Advanced Decision Support

The simulation framework should be able to support the user with advanced decision support technology that can be applied in combination with simulation models. The aim for this is to take manual work out of the hands of the end user who otherwise has to design each scenario by himself, which results in the quality of the decision support being highly dependant on the quality and creativity of the user. By automatically suggesting new scenarios or even running them, the quality of decision support can be improved.

- *Application should be extendable with optimization algorithms;* embedding optimization for simulation models can result in automatic scenario evaluations, and finding a “best” solution based on the goal function.

- *Application should be able to define new experiments with sets of parameters with optimization algorithms;* Experiment suggestion should be available in the framework so that models can suggest what value to change for a new scenario. For example a resource that has a high utilization with WIP building up and the highest recorded value in the resource queue is the last value, would result in the suggestion to increase the resource capacity.

3.6 System Requirements

The system requirements for such enterprise wide decision support systems can be split up in requirements important for the end user, as well as corporate requirements.

End user system requirements are a combination of speed of analysis and accessibility. Scenarios must be executed as fast as possible, although it never is quick enough. The best way of implementing fast model execution is to implement several dedicated simulation servers that run the engine(s) using grid computing. While waiting on scenario execution, users should be able to prepare other scenarios, or view reports of scenarios run in the past. Accessibility requirements determine that corporate users can access their projects at any given time, from almost any location in the world. Accessibility over a secured web page is highly preferred.

Corporate requirements will emphasize the data security of such a framework, as most of these solutions are using key and sensitive data sets, and or involve strategic directions that the enterprise is considering. Therefore, all servers and services should be run on networked corporate servers, preferably platform independent.

4 Concept Scenario Navigator

We defined a database application that matches the requirements and called this Scenario Navigator. The database of Scenario Navigator contains data of all scenarios that can be accessed by any user and can trigger simulations. Figure 4 shows the interaction of data input for experiments and simulation models into Scenario Navigator. Within the database a range of scenarios are then triggered to be performed using one or more simulation models in one or more simulation environments. The outcome of the simulation models is stored again in the Scenario Navigator database and allows the user to view different reports, either a report of an individual scenario, or a report of selected scenarios.

In figure 4 at the left are four reports shown of a scenario and at the right hand side three reports with different topics, i.e. utilization, wait time and queue length. These reports are simple examples, but can be customized and contain different sets of scenarios and performance indicators.

Figure 5 is the process of working with Scenario Navigator. A model developer creates a simulation model according to normal processes (Banks 1999). This simulation model is added to the Scenario Navigator database and the problem owner (or any stakeholder in the simulation study) can add data for experiments. The process of identifying the experiments to be run in some kind of group process is left out of the scope of this process flow, we assume that the stakeholders in the simulation study together define the types of experiments to be performed. One by one the experiments are added into the Scenario Navigator database, whereby the user can apply copying of already defined experiments or adjusting already defined experiments. The activity of data entry is all tracked in the database, to enable users to return to a previous state. If for some simulation scenarios an alternative simulation model is required, then the model developer can develop this alternative and add the model to the Scenario Navigator database.

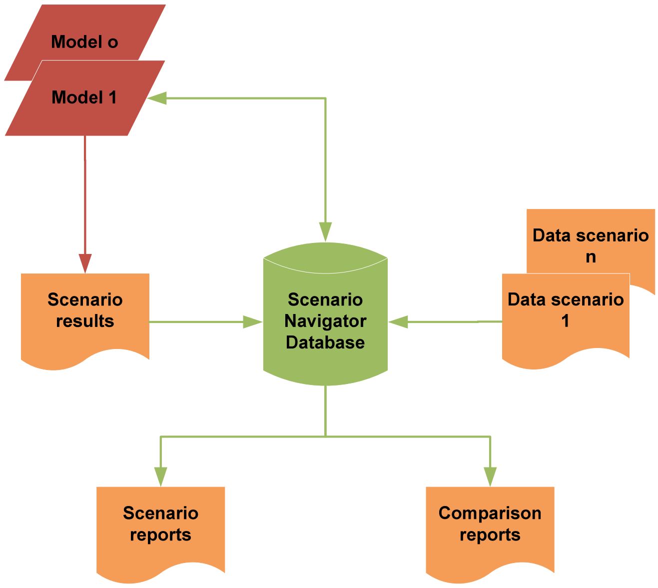

Figure 4: *Interaction Scenario Navigator*

The next step is to perform the simulation experiments. The users of Scenario Navigator can identify a list of simulation runs to be performed. Scenario Navigator will trigger the simulation application to perform the experiments. Thanks to grid computing and server capabilities it is possible to run several simulation experiments that are defined in Scenario Navigator at the same time.

The performance indicators of the simulation experiment are automatically imported and stored in the Scenario Navigator database. The collection of performance indicators enables stakeholders to view results in fancy reports at any time they want.

The reports enable the stakeholders to evaluate whether they now have gathered sufficient insight in the system thanks to the simulation study. If they do not have sufficient insight yet, then they will define some additional experiments and perform them. Once they have collected sufficient insight, they will stop the simulation study and act according to their findings.

The parameters and performance indicators of scenarios are kept in the database. If in a later stage some more scenarios are required or a report needs to be adjusted slightly, then this can be performed easily, without performing the simulation experiments again.

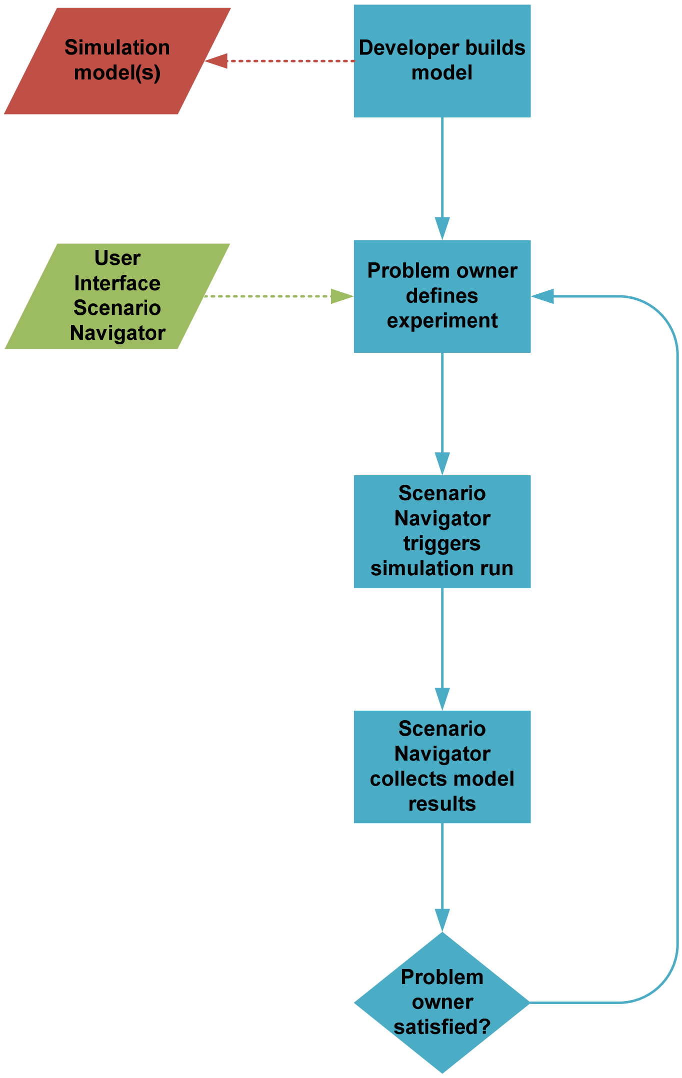

Figure 5: *Process of Performing Simulation Experiments with Database Application*

The Scenario Navigator application is designed for multiple users. A database server will be connected to the network and all computers connected to this computer can define new scenarios or view the results, given the correct user rights. Figure 6 provides an example of computers connects to enter input and to view output, all to the same Scenario Navigator database within a simulation study.

5 Scenario Navigator

In this chapter we describe the functionalities, capabilities and use of Scenario Navigator and how Scenario Navigator matches the mentioned requirements.

5.1 Core

The user can define the project structure by adding input & output tables and adding available models. Models can be added from different simulation environments; the system currently supports Arena®, Enterprise Dynamics®, Extendsim® & SIMUL8® models. Thanks to the addition of the simulation model a range of new scenarios can be defined and used to gather insight in the performance of the factory. Once a project is completely defined, scenarios can be defined, executed and compared in dashboards, charts & reports. Depending on their status the input parameters can be adjusted or reports of performance indicators can be viewed.

5.2 Scenario Analysis with Charts & Reports

Scenario Navigator has a build in Chart Wizard & Report Design engine for allowing great flexibility in displaying scenario results. Charts can be displayed on dashboard screens, and can be equipped with filters and allow selection of data series. The report designer allows users of Scenario Navigator to define their own customized reports, which can display results of a single scenario, or multiple scenarios, as a comparison report. Figure 7 shows an overview of scenarios with their status. Depending on their status the input parameters can be adjusted or reports of performance indicators can be viewed.

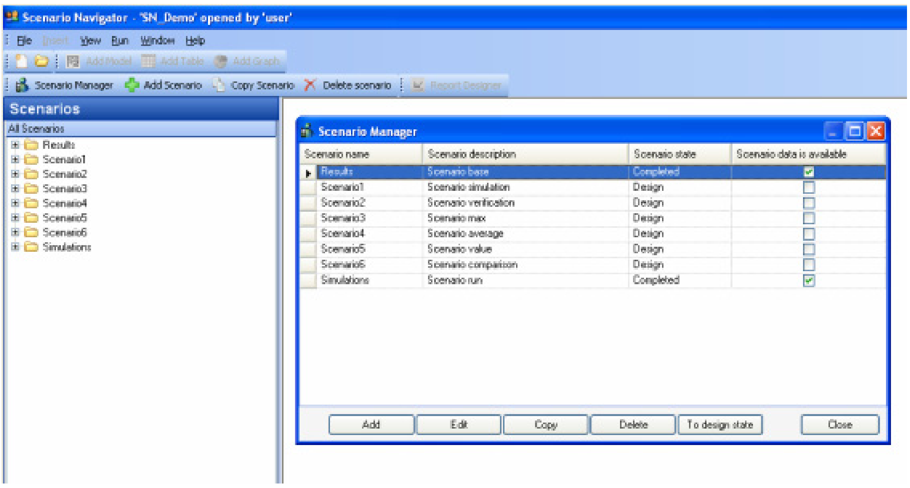

Figure 7: *Overview Scenarios in Simulation Study*

5.3 Domain and Customer Extensions

Scenario Navigator can be customized for particular domains or extended with specific functionality. The generic Scenario Navigator application is currently implemented in one of the worlds' largest food manufacturers. This is done by using the software in combination with the generic extensions concept. A complete dedicated data structure in combination with engine library and customized reports is added to the software. This allows the end users to generate a complete model of an existing or new plant, and analyze its performance in detail.

The Scenario Navigator application is also used in decision support in hospitals together with the dedicated simulation library called Hospital Navigator (Hay et al. 2006). This is currently in use by several NHS hospitals in the United Kingdom. This enables hospitals to use the application for regular evaluation of allocation of beds and achieve a higher utilization of their beds without denying patients service.

Optquest, by Opttek, is available as extension, allowing complete and customized simulation optimization within Scenario Navigator projects.

5.4 Usage in Real Projects

Scenario Navigator software has been used in a dozen simulation projects and is currently in use at several end users of simulation projects. Based on our project experiences and user feedback, it has become clear that the concept is a success. Using the software has resulted in a reduction of time spent for validation and experimenting by at least 20 %. Delivery of experiments to end users and customers has been received well, almost always leading to the request for more experiments, either done by the consultant, or by the end users themselves.

User interface development as well as report design tends to become very important in delivering the project. Since Scenario Navigator has a wide range of functionalities here, focus is needed in order to design and implement user interfaces in such a way, that KPIs are shown in a format that is easy to understand. With more project experience, users will become better at designing such interfaces.

Existing customers are actively using Scenario Navigator in their operational decision support process. Amongst several feature and functionality requests, making Scenario Navigator a web enabled application ranks highest.

6 Conclusions

Developing the ideas behind this decision support framework for simulation, Scenario Navigator, has been a process of many years. In those years we have been involved in practicing simulation in a wide variety of industries. The lessons learned in these projects, the encountered limitations in software and the customer needs have directed us in our development work thus far. Our motivation for developing this framework comes primarily from our goal to bring the power of simulation models to the desktop of problem owners. The development of the Scenario Navigator software has been a process of 2 years, resulting in use with several corporate organizations that base their weekly planning on the outcome of the simulation.

Although so far results of using the application have been very positive, the scope of use is limited to discrete event simulation and simulation studies performed at one location. We observe a need for supporting the stakeholders in using simulation within a group and as a group decision instrument. Organizations could provide worldwide access to their database, but this is not feasible with security mechanisms. More important is that simulation technology is a still a tool of engineers sitting around a computer. We focus our future research in further providing managers to understand and perform the experiments as part of their group meetings. This requires further research in cooperation in a simulation study and enhancements in changing simulation models for scenarios, beyond adjusting parameters. Additional support can then be provided by sensitivity analysis, confidence intervals and supporting a reduction of the model variability.

References

April, J.; Better, M.; Glover, F.; Kelly, J. (2004) New advances and applications for marrying simulation and optimization. In: Ingalls, R.G.; Rossetti, M.D.; Smith, J.S.; Peters, B.A. (eds.) Proceedings of the 2004 Winter Simulation Conference. IEEE, Piscataway, pp. 80-86

Banks, J. (1999) Introduction to simulation. In: Farrington, P.A.; Nembhard, H.B.; Sturrock, D.T.; Evans, G.W. (Eds.) Proceedings of the 1999 Winter Simulation Conference. IEEE, Piscataway, pp. 7-13

Banks, J. (2000) Getting started with Automod. Brooks Automation, Salt Lake City

Hay, A.M.; Valentin, E.C.; Bijlsma, R.A. (2006) Modeling emergency care in hospitals: A paradox – The patient should not drive the process. In: Perrone, L.F.; Wieland, F.P.; Liu, J.; Lawson, B.G.; Nicol, D.M.; Fujimoto, R.M. (eds.) Proceedings of the 2006 Winter Simulation Conference. IEEE, Piscataway, pp.439-445

Hillen, D.W. (1993) Simulation in Produktie en Logistiek (Dutch). Academic Service, Schoonhoven

Kelton, W.D.; Sadowski, R.P.; Sturrock, D.T. (2007) Simulation with Arena, 4th ed. McGraw-Hill, New York

Kleijnen, J.P.C. (1977) Design and analysis of simulation: Practical statistical techniques. Simulation 28 (1977) 3, pp. 81-90

Krug, W. (2001) Modellierung, Simulation und Optimierung. SCS-European Publishing House, Erlangen

van Meel, J.W. (1994) The dynamics of business engineering – Reflections on two case studies within the Amsterdam Municipal Police Force. Doctoral Dissertation, Delft University of Technology, Netherlands

Robinson, S. (1999) Three sources of simulation inaccuracy (and how to overcome them). In: Farrington, P.A. ; Nembhard, H.B. ; Sturrock, D.T. ; Evans, G.W. (eds.): Proceedings of the 1999 Winter Simulation Conference. IEEE, Piscataway, pp. 1701-1708

Shannon, R.E. (1975) Systems simulation: The art and science. Prentice Hall PTR, Upper Saddle River

Taylor, S.J.E; Hlupic, V.; Robinson, S.; Ladbrook, J. (2001) GROUPSIM: Investigating issues in collaborative simulation modelling. In: Proceedings of the UK ORS Simulation Study Group Workshop, UK, pp. 11-18