

SIMPLIFICATION AND AGGREGATION STRATEGIES APPLIED FOR FACTORY ANALYSIS IN CONCEPTUAL PHASE USING SIMULATION

Matías Urenda Moris
Amos H.C. Ng
Jacob Svensson

Virtual Systems Research Centre
University of Skövde
Skövde, SE-541 28, SWEDEN

ABSTRACT

Despite that simulation possesses an establish background and offers tremendous promise for designing and analyzing complex production systems, manufacturing industry has been less successful in using it as a decision support tool, especially in the conceptual phase of factory design. This paper presents how simplification and aggregation strategies are incorporated in a modeling, simulation and analysis tool, with the aim of supporting decision making in conceptual phase. Conceptual modeling is guided by a framework using an object library with generic drag and drop system components and system control objects. Data inputs are simplified by the use of Effective Process Time distributions and a novel aggregation method for product mix cycle time differences. The out coming specification is through a Web Service interface handle by modeling system architecture, automatically generating a simulation model and analysis. Case studies confirm a breakthrough in project time reduction without appreciable effects on the model's fidelity.

1 INTRODUCTION

Today, industry has very limited support from working procedures, methods, and tools for analysis of complete plants in early program stages. This leads to difficulties in predicting the consequences from the early decisions that are basic for robust, flexible and cost effective production. Despite that simulation possesses an establish background and offers tremendous promise for designing and analyzing complex production systems, manufacturing industry has been less successful in using it as a decision support tool, especially in the early conceptual phase of factory design (McNally and Heavey 2004). Unfortunately, simulation analysis for system design is used in later phases when cost and system performance are more or less locked.

There are several reasons to the poor use of simulation at this phase. Industrial experience shows that decisions are many times based on heuristic and historical approaches. If

tools are used they are mainly limited to analytical tools. The arguments suggesting an analytical solution, instead of a discrete event simulation (DES) approach, are among others a more effective use of time and the lack of detail process data (Jägstam and Klingstam 2002; Patchong et al. 2003). Remember that many conceptual models have a very short life span, due to the nature of the task which is evaluating different production concepts and control strategies towards each other. The production manager is therefore dependent on the help of a simulation expert and needs to order models for the different concepts. If the company has in-house experts there is still a time-consuming dialog and process between the manager and the expert. That particular dialog is somewhat intricate. The simulation specialist, used to work with detail models for planning and scheduling of existing production lines, gets into an "ethical dilemma". Not used to build simple models he has his own lack of credibility to the models an the use of estimated data. The troubles with models at conceptual phase are therefore summarized in lack of data, time and knowledge. Can these problems be overcome or should production managers continue to design production lines based on guesses, "this is the way we have always done" attitude or be limited to basic queuing network models which will most certain result in bigger assumptions than a DES model?

The objective of this research is to overcome these problems and "frontload" the use of virtual methods to analyze complete factories. The project targets were to increase the use of simulation analysis, make them faster and more accurate in the early conceptual phase. This is through developing a new work method and a corresponding toolset that tightly integrates model abstraction, input data management and simulation-based optimization under an innovative framework that is specifically designed for production system's designer/managers in the conceptual design phase. As a part of the research work within the Factory Analyses in ConcepTual phase using Simulation (FACTS) project, which is funded by VINNOVA in Sweden (see Acknowledgement) and took place between Janu-

ary 2006 and March 2008. The toolset is named FACTS Analyzer. The project contained three work packages and an integration package.

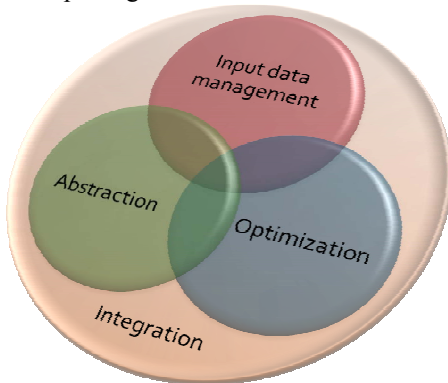


Figure 1: The project's work packages

The aim of this paper is to present the results from the abstraction work package, in particular on how simplification and aggregation strategies are used to simplify conceptual modeling in FACTS Analyzer. For the sake of clarity, the general concept and system design of FACTS Analyzer will be briefly presented in Section 2. The rest of the paper is organized as follows: Section 3 will describe the work done in the abstraction work package, the objectives, theoretical approach and modeling framework. Section 4 will present a novel simplification and aggregation approach. Section 5 gives details and validation results from applying FACTS analyzer to industrial-based test cases. Conclusions and further development are found in section 6.

2 FACTS ANALYZER SYSTEM ARCHITECTURE

FACTS Analyzer provides a novel modeling framework and a Web Services based client/server system architecture, in which when combined together, can synthetically provide the features required for making DES easier to use as well as speed up the time-consuming model building and experimentation process.

FACTS Analyzer is designed with the principle of *illusion of simplicity* and *system neutrality*. In terms of the *illusion of simplicity*, unlike any other simulation software, FACTS Analyzer is designed to be a “thick” client application that accesses the server components through the Web Services interface (see Figure 2). The server components, including model generator, DES, optimization algorithms, data analysis functions and the underlying integrated data-base management system, enable the FACTS users to carry out the following functions in a transparent manner:

- Generate simulation model automatically. FACTS Analyzer provides a unique graphical user interface (GUI) that is customized for manufacturing executives to quickly model a conceptual system design and make evaluations. The concept of system neutrality and

automatic model generation allows the conceptual design to be stored into a “neutral” standardized format, e.g. the format specified in NISTIR 7198 (McLean et al. 2005) so that simulation can be generated and executed, in principle, in any available simulation software.

- Optimize decision variables, through the build-in support of simulation-based optimization (SBO) functionality. For a decision maker, the process of finding a sufficiently good design setting could be too time-consuming and in many cases impossible if the search space is huge. SBO is a relatively new technique that can be applied to seek the “optimal” setting for a complex system based on one or multiple performance measures generated from simulation by using various searching methodologies (Fu et al. 2000, April et al. 2004). In the FACTS system architecture, these services are virtually provided by connecting the FACTS server components to the OPTIMIZE server components. Further details of OPTIMIZE and the advantages of running SBO in a client/server architecture based on Web Services technology can be found in (Ng et al. 2008).

3 ABSTRACTION WORKPACKAGE

Models are, almost always, simplifications or abstractions of reality. Sánchez (2006) defines a model as “a system which we use as a surrogate for another system”. Discrete event models are no exception. Their aim is to correctly describe the system of interest and mimic its behavior. “The purpose of modeling is to simplify and abstract to gain insight” (Sánchez 2006). Modeling is consequently a commission of making suitable abstractions and simplification of the system of interest. It demands experience and judgment. There are no tools that can substitute experience, but great things can be achieved with the right guidance and support. The abstraction work package has that endeavor, to give the modeler the tools and necessary guidance. The aim was to provide a toolset for conceptual modeling within the manufacturing industry domain. The outcome is to simplify conceptual modeling and increase the accuracy of the out coming model specification.

Model abstraction, according to Frantz (1995), is “...a method for reducing the complexity of a simulation model while maintaining the validity of the simulation results...” There is always the possibility of modeling a system at different abstraction levels. But independent of which level, the model needs to be valid for its purpose or objective. Frantz (1995) identifies different model abstraction techniques. He distinguishes three main groups within a taxonomy: model boundary, model behavior and model form modifications. All these modifications demand a certain degree of simplification of the model, whether it is related to how data is aggregated or the limitation of the model's scope.

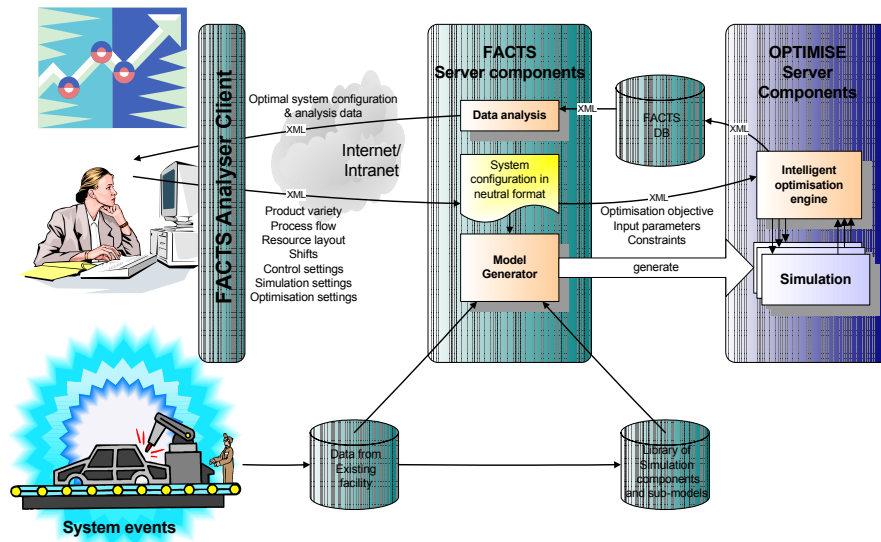


Figure 2: The system architecture of FACTS Analyzer

The choice of abstraction level is therefore closely related to the set of question the simulation model is expected to address. It can be said that the abstraction level is affecting the utility and performance of the model. But the relation between utility/performance and abstraction does not conform to a linear relation (Brooks and Tobias 1996). It is true that a more detail model with a lower level of abstraction might be able to address additional questions and being more accurate, but at the same time if the higher complexity leads to extended project time and increase cost it might counteract its utility. Experience presents that a complex model is for many reasons not wanted and not always more valid or credible than a simple one (Chwif et al. 2000, Sánchez 2006).

3.1 Objectives

This discussion highlights an important limitation of the work currently presented. The objective is to be able to address questions that are essential and feasible to answer at the concept definition phase of an industrial project (see Figure 3). At this phase there is a clear lack of information and data. The proposed system has not been defined. The production manager has got information on expected volumes, maybe information on product mix. But there are still construction changes to expect and/or configuration changes on the final product. The significance at this phase is to be able to choose the right production concept for the expected product and production volume. Having stated this, it becomes obvious that the level of abstraction of these models is higher, compare to models of existing production lines, and the assumptions likewise. It is therefore, reasonable to expect that the model performance in terms of output accuracy would be lower (Brooks and To-

bias 1996, Madan et al. 2005), however this would not necessarily be the case when it comes to the model's utility or other performance measurements. From a management point of view the results from the simulation analysis could be invaluable, both in terms of concept evaluation as when it comes to system understanding and project communication.



Figure 3: The work procedure used at Volvo AB-Global Development Process

3.2 Conceptual Modeling

Conceptual modeling is considered to be, if not the most important, at least one of the most critical steps in a successful simulation study (Robinson 2006). The reason is that the model design has an impact in many aspects of a simulation project. Robinson (2006) outlines in particular the following aspects; data requirement, model development time and validity, as well as speed of experimentation and the confidence that is placed on the model results.

There is a close relation between conceptual modeling and model abstraction. According to Robinson (2006), "Conceptual modeling is the abstraction of a model from a

real or proposed system”. The conceptual model is, therefore, the result of the process of using abstraction techniques to derive a model (Frantz 1995).

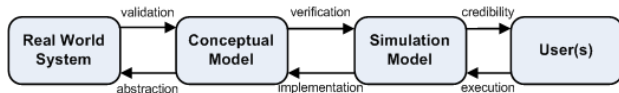


Figure 4: Model abstraction in the simulation process (Frantz 1995)

It is evident that conceptual modeling is of huge significance and closely related to simplification and aggregation strategies, at the same time it is an area that is not well understood and that has received little attention in practice and research. It is still an art, even though many opinions about formalizing the procedure and the need for using guidelines have been raised (Brooks and Tobias 1996, Pace 2000, Robinson 2006).

In addressing several related issues, Robinson (2006) points out the need for guidelines into “how to develop conceptual models”. He identifies three basic approaches as sub topics: principles of modeling, methods of simplification and modeling frameworks. Of which modeling frameworks is the most complete approach including both principles and methods, but also providing a “specific set of steps that guide a modeler through the development of a conceptual model”. Pace (2000) identifies four steps in conceptual model development: 1) defining the simulation context or problem domain through collection of authoritative information, 2) identifying entities and processes that need to be represented for the simulation to accomplish its objectives (this step includes basic decisions about the model’s level of details and aggregation), 3) development of simulation elements (implies decisions about the level of accuracy and precision), 4) addressing relationships among simulation elements to ensure that constraints and boundary conditions are met. This four-step modeling procedure meant to be iterative by nature. Despite the need of suggested formalism to the process, the author’s opinion is that the complexity of the task still demands “one ... [to] be very pragmatic in developing a simulation conceptual model” (Pace 2000).

3.3 Objects and Modeling Steps

The FACTS modeling framework support the aim of formalizing and set up guidelines for conceptual modeling, but it is far from covering a complete modeling context. It is domain specific and mainly focusing in models with high abstraction level. It supports modeling of production and manufacturing lines, manual or automatic, of single or mixed product variety with a selection of production control strategies. It does not, hitherto, support batch production.

The general approach to document conceptual models is to use a neutral notation like Unified Modeling Language (UML) diagrams or Simulation Activity Diagrams

(SAD) (Pace 1999, 2000; Heavey and Ryan 2006). The aim is to find a descriptive and unambiguous way of documenting the system of interest. But application domain experts which are the one developing and approving the simulation requirements are seldom experts in formal methods, specially if the notations used are from the software engineering field (Pace 2000). The consequence is that the simulation requirements are stated in natural language, passed orally or in written documents to an external modeler who needs to ensure that he has collected the necessary “authoritative information” and defined the right simulation context (see step one in Pace (2000)). Since the FACTS concept is aiming at these very domain experts, production managers/engineers, a more intuitive way of documenting the system was required. The modeling objects and their settings, needed to be easily identified from the “real world”. The process of determining the objects that are indispensable for modeling purposes, is hence an iterative bottom-up process based on several industrial case studies and interviews with domain experts. This has resulted in the following kinds of objects, divided into two main categories:

- Flow objects (resources) – *Source, Sink, Operation, Buffer, Junction, Pallet system, Portal* (gantry crane).
- Logic objects – *Shift* (working or production hours settings), *Kanban, ConWIP* and *Takt* control¹.

Their drag and drop icons in FACTS Analyzer can be viewed in the left column of figure 5. The set of objects, including a flow connection function and a “Note” object for documentation, are continuously evaluated, both in terms of the current objects’ functionality as in terms of including additional objects. The modeling steps, observable as arrows at the top of figure 5, are: 1) product definition, 2) flow definition, 3) resource settings, 4) simulation settings and 5) experiments settings /results view.

After defining the product(s), the user builds the system and its flow by drag and dropping modeling objects from the object column into the modeling window. The user can copy, duplicate, delete and move objects. Flow objects need to be connected according to the expected production flow of the product. This includes material handling systems, *Pallet, Portal* and *Buffers* with a transportation time (or *Buffers* used as inline store), sequential or parallel *Operations* (manual, semi- or automatic stations), etc. Logic objects are, on the other hand, also introduced into the modeling window, but they do not use connections.

The next step “resource settings” is where the object’s settings are specified e.g. size of buffers, transportation time, number of pallets, times for the *Portal* object, cycle time, availability, mean time between failure or an operations type (manual, semi- or automatic) and product mix variation (described in section 4).

¹ Takt control means using a pre-determined Takt time to control the pacing rate of parts from stations to stations in a synchronized manner.

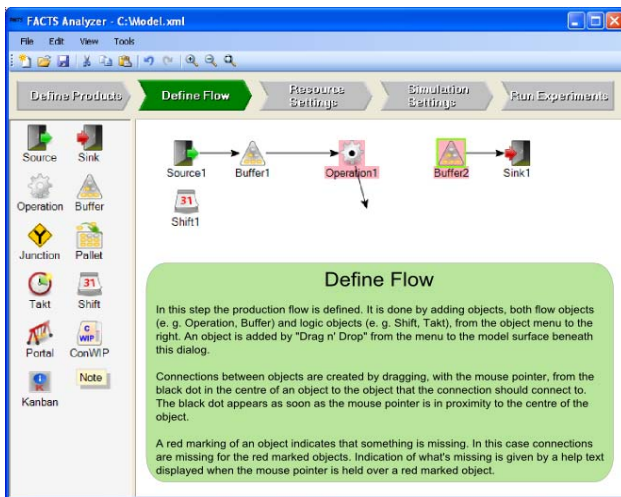


Figure 5: View of the flow definition step including a description text

“Simulation settings” includes settings like simulation horizon and warm-up period and number of replications. An additional user choice is the selection of shifting bottleneck detection analysis (Roser et al. 2002). The shifting bottleneck detection algorithm demands a higher number of calculations which prolog the simulation execution time. This feature should not be used when the experiments involve optimization analysis.

In the final step, “Run experiments”, the user has a choice to either make a simple simulation analysis, including a shifting bottleneck analysis or to define a single or multi objective optimization experiment. If optimization is the user’s option, FACTS analyzer provides a wizard to guide the user in defining the right settings. The simulation results, including the shifting bottleneck analysis is presented in two views, either in a graphical view (Visual), or a numeric view (Plain). In the graphical view, the data is presented by charts and in some outputs also by highlighting the three objects with the highest values e.g. utilization for operations and occupation for buffers.

Considering that the outcome of the conceptual model is an automatically generated simulation model and that the user is not expected to have a simulation background, there is a strong need to limit the user’s modeling mistakes. This is achieved by applying a strict guidance and notification system. The modeling framework highlights in every step if the modeling and analysis specification is lacking necessary information to fulfill the minimum requirements for the generation of a successful simulation model and its experimentation (see highlight of operation 1 in figure 5). This function, together with the results analysis, helps the user in the verification and validation of the conceptual model.

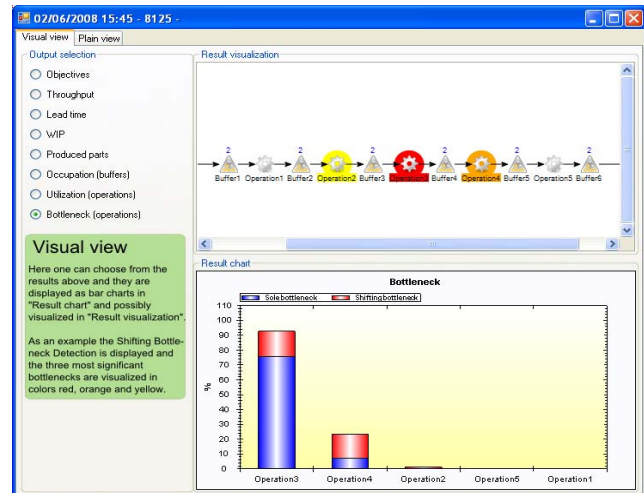


Figure 6: Visual view of bottleneck operations from the experiment/results step

4 SIMPLIFICATION AND AGGREGATION STRATEGIES

There are two main objectives behind the simplification of simulation models. One of them is to decrease the work effort, time and cost of the project (Chwif et al. 2000) (Madam et al. 2005). The second one is to decrease the execution time of the simulation run (Johnson et al. 2005). Simulation runs based on models of the same scope but with different detail level could differ up to 10 times in execution time. If the analyst is aiming at using simulation optimization or using the output of the model to verify scheduling or planning alternatives in real time, a fast execution time is extremely important. For instance, SBO may use several hundreds of simulation runs before an optimal solution is found, every second of reduced execution time is therefore crucial. The reason behind the abstraction work package has obviously been to achieve both these objectives. But many of the simplification and aggregation methods suggested within the simulation society have focus on how to simplify or aggregate an existing simulation model or model an existing system (Brooks and Tobias 2000, Johnson 2005). These procedures are infeasible in the early conceptual phase because the system does not exist and detail data is not available. Therefore it was necessary to look at this task with a different mindset and evaluate new approaches. Two of the contributions are described in the following subsections.

4.1 Processing Time Aggregation

Input data collection has been proven to be a time-consuming task when conducting DES projects (Klingstam and Johansson 2000). Recent research even shows that input data analysis may take 31% of the total time in a DES project (Skoogh and Johansson 2007). The problem of low input data availability is even more compelling when simula-

tion studies are done in early conceptual phases. In terms of input data for simulation studies, uncertainties in the early phases are usually associated with the variations in, e.g., processing time, failure rate, repair time, setup time and/or even product variants that cannot be known but need to be taken into account to estimate the steady-state performance of the new design and compare with several alternatives. Although there can be numerous kinds of uncertainties when building an abstract model for a non-existing production line, it is not wise to compromise the accuracy of the abstract model, e.g. to use only constant values, when variability of the system can be estimated by the designer, using their knowledge or experience from some existing systems. In other words, an aggregation methodology that can guide the designer to concisely and effectively choose the most reasonable, albeit uncertain, input data distributions for the abstract model is needed. In FACTS Analyzer, we use an EPT-based data input as such an “effective” methodology.

Introduced by Hopp and Spearman in their award winning text book (2001), EPT is described as “the time seen by a lot from a logistical point of view”. The concept is based on aggregating all time-taking events that a lot/part experiences in a workstation, cell or resource. Instead of using different distributions to describe the variability of the workstation, the EPT distribution embeds all disturbances and includes cycle time variations. As it captures all the disturbances, this has made EPT-based aggregation as a more accurate approach when compared with the traditional MTBF and MTRR distributions in simulation modeling. Another advantage is that for each resource, there are only two parameters to keep track of, calculate and update, namely, the aggregated mean of processing time, t_e , and its corresponding coefficient of variation, c_e . They differ from the traditional cycle time and cycle time variation because they contain all the disturbances and are thus more “effective”.

The EPT concept has been adopted and extended by the research group at Eindhoven University of Technology (Jacobs et al. 2003). Their work focus on building simple, yet accurate models, of manufacturing networks using operational factory data. While their techniques can be applied to both DES and analytical queuing models, most of their research is focusing on the latter one for the development of abstract models and parameter identification for queuing networks using decomposition techniques. At the same time, their EPT-based framework is also based entirely on operational data collected from existing systems. The challenge of the FACTS approach is to use EPT in early phases of a design project when operational data is absent or very limited. In order to do this there is an imperative need to identify some generic EPT distributions, with the suitable shapes, that can be used for modeling the statistical behavior of various typical resources. For instance, we expect that automated workstations in a manufacturing line would behave similarly, in

terms of pattern in cycle time variations, breakdowns and repair times. But we would expect that a manual assembly workstation would generally behave in a very different pattern when compared to an automated workstation. The same kinds of observations have already been made by simulation experts, but techniques to address these are limited to adjust arrival rate or MTBF/MTTR (Law and Kelton 2000).

The EPT distributions that need to be identified are expected to be scalable and easily adapted to the specific cycle time, but as mentioned earlier, have the shape that corresponds with the type of station e.g. automatic, manual. This means that the FACTS user would only need to define the type of station (automatic, semi-automatic or manual) and the cycle time. Currently, the distribution family that has been chosen and implemented in FACTS Analyzer to describe manual stations variability is *Johnsson S_B* , because of its high scalability and bounded nature. Validation results in the industrial-based test cases have shown that this probability distribution is suitable but more empirical evaluations are required before a general conclusion can be made.

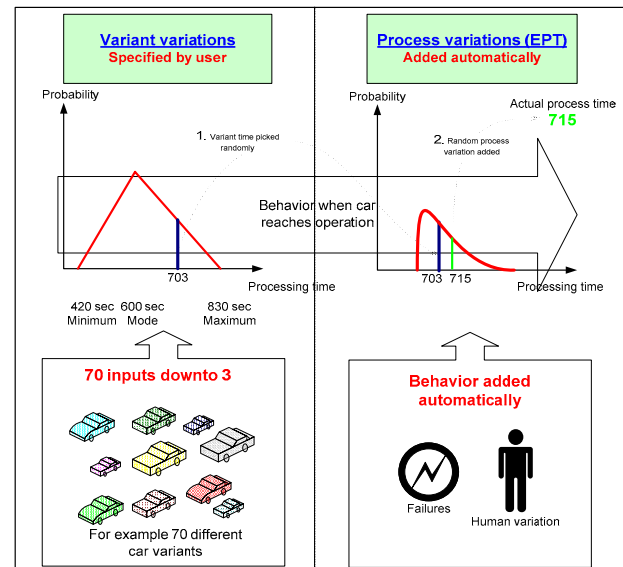


Figure 7: Processing time generation from probability distributions that cope with product variants and EPT aggregation

4.2 Abstraction for Product Variants

Another issue that is related to product mix and cycle time estimation has to do with how we could effectively model large number of different products in mix product lines. A manual assembly line could for instance assemble 1000 variants of a product. Each variant might not differ physically much from the other, but when it comes to assembly time the difference could be significant. At the same time, it is easy to realize the time consuming work of feeding the simulation model with all possible cycle times. EPT has the possibility to include product mix, but it is impossible to de-

fine a generic EPT distribution shape that is able to take this into consideration. Another way of doing it without alter the shape of the EPT distribution is by using a separate distribution to stochastically generate the cycle time mix. FACTS Analyzer adopts an approach similar to the one described in (Law and Kelton 2000) using a Triangular probability distribution for representing the products variation so that the users need only to feed the system with three values; the minimum, the mode and the maximum value of the estimated cycle time range of different product types. This product mix distribution then generates the cycle time expected for the product type at the workstation. This randomly chosen cycle time is then fed to the EPT distribution that will add the disturbances/variability that is expected that the system contains. This procedure is illustrated in Figure 7. Note that the Triangular distribution is used to represent the cycle time variations resulting from product mix while the EPT distribution models the variations resulting from the disturbances at the stations.

5 INDUSTRIAL TEST CASES

Industrial-based test cases are essential in the FACTS project for verifying the FACTS concepts and demonstrate the applicability of the toolset. The test cases took place, at Volvo Car Corporation (VCC) at Torslanda Gothenburg, at Volvo Powertrain plant in Skövde and at Volvo Trucks in Tuve, Sweden. The following discussion will focus on the work done at Volvo Trucks, but general comments from the other test cases will be given.

Final assembly of Volvo trucks is done in an assembly plant in Tuve, located in the outskirts of the city Gothenburg. In the final assembly the engine and the transmission (gearbox) is joined before the complete engine (engine and gearbox) is assembled on the base frame on which the cab then is assembled. The focus of this case has been on the first part of this production system, namely, the two engine lines where the final assembly of the engine is done including the marriage with the gearbox. These two lines are identical with respect to the number of stations, the number of operators, and the production control logic. Each line consists of nine manual stations with one operator each. In addition to the station dedicated operator there is one extra operator per line as well as five extra operators that serve as extra resources on heavy variants on Station 7 and Station 8 on both lines. Each line is controlled by a predetermined Takt rate of its own. The MHS is consisted of an Automated Guided Vehicle System (AGVS). Both lines are fed with engines from an outside goods terminal by the same pallet system which is built of roller conveyor tracks and a Loading Pallet System (LPS).

5.1 Detailed Model vs. Abstract Model

A detailed model of these two engine lines was built manually using Siemens-UGS Plant Simulation (see Figure 8). An abstraction of this detailed model was then done in two steps. First the MHSs, including the LPS and AGVS, and all of its control logic were replaced with a simple pallet system using the standard objects in FACTS Analyzer. Despite the lack of a complex control logic such a simple pallet system manages to capture the main characteristics of both the LPS and the AGVS, namely the limited number of AGVs and their re-transportation time from the last station back to the first. The re-transportation time is modeled with a FIFO (First-In-First-Out) buffer that has a processing time equal to the re-transportation time.

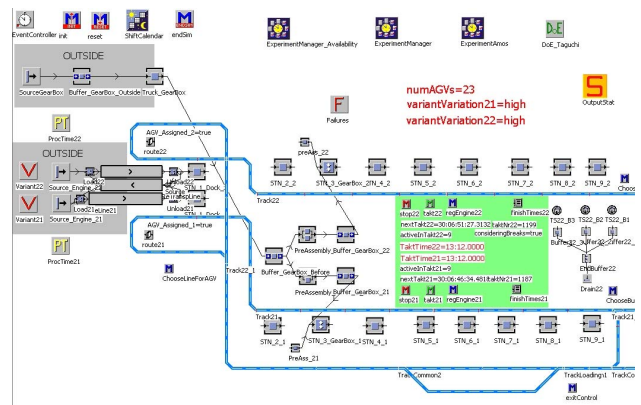


Figure 8: Manually built detailed model for the two truck engines assembly lines

The second part of the abstraction concerns the product (engines) variants, their processing times as well as the failure rates of the stations using the aggregation method described in Section 4.2. In contrast to the detailed model where each variant is modeled with a single entity type in the simulation, the single-variant abstraction method is used in FACTS Analyzer. In this sense, only one product type would be generated and variations on processing times in each station are represented using the triangular distribution as mentioned before. The parameters of the triangular distribution are chosen so that the minimum and maximum value corresponds to the recorded processing time of the lightest and heaviest variant on each station whereas the mode is calculated by using the weighted mean of the processing time of all variants and their frequencies on each station. Additionally, production assembly time variations and disturbances due to station failures are put into FACTS Analyzer using the EPT-based aggregation method.

5.2 Results and Lessons Learned

Two of the three test cases (Engine line at Volvo Trucks and Paint shop line at VCC) involved a validation procedure towards the detail models of the corresponding production lines. This gave the advantage of being able to make more

thorough test on system behavior and not only focus on outputs like throughput, lead time and WIP levels. In the third case study, the model validation was done towards the engine block line at Powertrain, Skövde.

The results from the test cases were very promising, both in terms of output equivalence and system behavior. In the above presented Tuve case the error in production rate between the detail and abstract model was only 3.1%, more details are found in (Ng et al. 2007b). Even better values were obtained from the Paint shop case with output values only 1-4% difference on throughput, WIP average level and lead time. In the Powertrain case the comparisons were not as straightforward. There were no comparative data on average lead time and WIP, but the model showed very good corresponding to the system outputs.

The really astonishing values are the ones obtained when compared the detail vs. abstract model's development time. In the VCC case the detail model took 4 weeks to build using a commercial simulation package. The abstract model took 40 minutes for a new modeler without prior knowledge of FACTS Analyzer. In the Tuve case the detailed model took almost two months to complete. The FACTS model took 45 minutes to build. It is true that the development of the detailed models gave valuable insight in the system behavior and a good foundation for finding input data for the abstracted models during the studies, but the time difference is still huge. What is even more convincing is that when it came to experimentation and analysis the most valuable insight was gained from the FACTS Analyzer's models. The ability to develop new concepts and use of the optimization feature to find the right mix of buffer levels and control strategies were by far the superior work procedure and much more time efficient.

Two important lessons were learned under the test cases:

1. It can be difficult to estimate lead time and WIP levels. The modeler needs to be able to correctly approximate, e.g. the number of spaces on conveyor tracks, transportation time and/or actual buffer places.
2. MHS can have a decisive effect on the outputs and cannot be too simplified generally. The Powertrain case forced the modelers to develop a new object, *Portal* (a gantry crane that works sequentially which is a typical MHS for Powertrain).

The *Portal* object is an example of a specialized object that may not be part of a generic library of modeling features. But its customized development represents a feasible and eligible way into how FACTS Analyzer can be adapted to a specific company's needs. The development of company specific objects and/or stochastic distributions increases the accuracy and user friendliness of the tool.

6 FUTURE DEVELOPMENT AND CONCLUSIONS

The future development work demands a balance between the human obsession of trying to address all possible modeling scenarios and the user friendliness of a tool for fast and accurate concept analysis. During the coming year will the FACTS development group gain more insight and data from industrial users. The work will focus on improving the modeling features and estimated distributions for industrial systems. Nevertheless, considering that the FACTS architecture and platform is generic, developing objects for other domain areas is highly feasible.

By using the FACTS Analyzer's framework it is believed that the modeling work of production systems during the conceptual design phase will be more accessible for non-simulation experts. The tool's guiding features, simplification and aggregation methods, will not only increase the fidelity of the abstracted model, but thanks to its analysis options will give insights and answers. It will, together with knowledge and experience, be of vital importance for manufacturing executives so that they can support their decisions with the "facts" that are proved through simulation analysis.

ACKNOWLEDGMENTS

The authors gratefully acknowledge VINNOVA, Sweden, for the provision of research funding (within the MERA framework) and the industrial partner companies, including Volvo Cars Corporation, Volvo Powertrain, Volvo Technology and Volvo Trucks, for their supports in the FACTS project.

REFERENCES

- April, J., M. Better, F. Glover, and J. Kelly. 2004. New advances for marrying simulation and optimization. In *Proceedings of the 2004 Winter Simulation Conference*, 80-86. Washington, D.C.: IEEE.
- Brooks, R.J., and A.M. Tobias. 1996. Choosing the Best Model: Level of Detail, Complexity and Model Performance. *International Journal of Production Research* 24:1-14.
- Brooks, R.J., and A.M. Tobias. 2000. Simplification in the simulation of manufacturing systems. *International Journal of Production Research* 30:1009-27.
- Chwif, L., M.R. Pereira Barretto, and J.P. Ray. 2000. On simulation model complexity. In *Proceedings of the 2000 Winter Simulation Conference*, 449-455.
- Frantz, F. 1995. A Taxonomy of Model Abstraction Techniques. In *Proceedings of the 1995 Winter Simulation Conference*, 1413-1420.
- Fu, M.C., S. Andradóttir, J.S. Carson, F. Glover, C.R. Harell, Y.-C. Ho, J.P. Kelly, and S.M. Robinson. 2000. Integrating Optimisation and Simulation: Research and Practice. In *Proceedings of the 2000 Winter Simulation Conference*. Arlington, VA: IEEE.

- Hopp, W.J., and M. L. Spearman. 2001. *Factory physics*, 2nd edition. New York: Irwin McGraw-Hill.
- Jacobs, J. H., L.F.P. Etman, E.J.J. van Campen, and J.E. Rooda. 2003. Characterization of Operational Time Variability Using Effective Process Times. *IEEE Transactions on Semiconductor Manufacturing*. 16:511-520.
- Johnson R.T., J.W. Fowler, and G.T. Mackulak. 2005. A discrete event simulation model simplification Technique. In *Proceedings of the 2005 Winter Simulation Conference*, 2172-2176.
- Jägstam, M., and P. Klingstam. 2002. A handbook for integrating discrete event simulation as an aid in conceptual design of manufacturing systems. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1940-1944.
- Klingstam, P., and B. Johansson. 2000. Towards a Strategic Framework for Logistic and Production Flow Simulation. In *The New Simulation in Production and Logistics: Prospects, Views and Attitudes*, 45-54. Berlin, Germany: Eigenverlag.
- Law, A., and W. D. Kelton. 2000. *Simulation modeling and analysis*, 3rd edition. Singapore: McGraw-Hill International Series.
- Madan, M., Y. Son, H. Cho, and B. Kulvatunyoo. 2005: Determination of Efficient Simulation Model Fidelity for Flexible Manufacturing Systems. *International Journal Computer Integrated Manufacturing* 18: 236-250.
- McLean, C., Y. T. Lee, G. Shao, and F. Riddick. 2005. Shop Data Model and Interface Specification, *NISTIR, 7198*. National Institute of Standards and Technology, Gaithersburg, MD.
- McNally, P., and C. Heavey. 2004. Developing simulation as a desktop resource. *International Journal of Computer Integrated Manufacturing* 17:435-450.
- Ng, A., M. Urenda Moris, J. Svensson, A. Skoogh, and B. Johansson. 2007. FACTS Analyser: An innovative tool for factory conceptual design using simulation. In *Proceedings of the Swedish Production Symposium*. Gothenburg, Sweden, 2007.
- Ng, A., H. Grimm, T. Lezama, A. Persson, M. Andersson, and M. Jägstam. 2008. OPTIMISE: An Internet-Based Platform for Metamodel-Assisted Simulation Optimization. In *Recent Advances in Communication Systems and Electrical Engineering*, ed. X. Huang, Y-S. Chen and S-L. Ao, 281-296. Springer.
- Pace, D.K. 1999. Development and Documentation of a Simulation Conceptual Model. In *Proceedings of the 1999 Fall Simulation Interoperability Workshop*. <www.sisostds.org> [Accessed May 2008].
- Pace, D.K. 2000. Simulation Conceptual Model Development. In *Proceedings of the 2000 Spring Simulation Interoperability Workshop*. <www.sisostds.org> [Accessed May 2008].
- Patchong, A., T. Lemoine, and G. Kern. 2003. Improving Car Body Production at PSA Peugeot Citroën. *INTERFACES* 33:36-49.
- Robinson, S. 2006. Conceptual modeling for simulation: issues and research requirements. In *Proceedings of the 2006 Winter Simulation Conference*, 792-800.
- Roser, C., M. Nakano, and M. Tanaka. 2002. Shifting bottleneck detection. In *Proceedings of the 2002 Winter Simulation Conference*, 1079-1086.
- Sánchez, P.J. 2006. As simple as possible, but no simpler: a gentle introduction to simulation modeling. In *Proceedings of the 2006 Winter Simulation Conference*, 2-10.
- Skoogh, A., and B. Johansson. 2007. Time Consumption Analysis of Input Data Activities in Discrete Event Simulation Projects. In *Swedish Production Symposium*. Gothenburg, Sweden.

AUTHOR BIOGRAPHIES

MATÍAS URENDA MORIS. is a final-year Ph.D. student at the Virtual Systems Research Centre, University of Skövde, Sweden. He holds a B.Sc. degree in Automation Engineering and M.Sc. in Manufacturing Management at the University of Skövde in collaboration with Loughborough University, UK. His main research area is DES for manufacturing and healthcare systems with emphasis on system modeling and analysis <matias.urenda.moris@his.se>.

AMOS H.C. NG is a Senior Lecturer at the University of Skövde, Sweden. He holds a B.Eng. degree and a M.Phil. degree, both in Manufacturing Engineering from the City University of Hong Kong and a Ph.D. degree in Computing Sciences and Engineering from De Montfort University, Leicester, U.K. His research interests include virtual engineering for manufacturing machinery and machine systems as well as simulation-based optimization. His e-mail address is <amos.ng@his.se>.

JACOB SVENSSON is currently working as a Research Assistant in the field of Conceptual modeling, and Simulation Based Optimization at the Virtual Systems Research Centre, University of Skövde, Sweden. He received his M.Sc. in Industrial Engineering at Linköping University, Sweden, 2007. <jacob.svensson@his.se>.