Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

HIERARCHICAL MODELING OF A SHIPYARD INTEGRATED WITH AN EXTERNAL SCHEDULING APPLICATION

Ali S. Kiran
Tekin Cetinkaya
Juan Cabrera

Kiran Consulting Group
4411 Morena Blvd. Suite 105
San Diego, CA 92117, U.S.A.

ABSTRACT

This paper presents a hierarchical approach on the simulation of large-scale discrete event systems used recently by Kiran Consulting Group (KCG) to model shipyard operations. Because of the dynamic, stochastic and complex nature of the shipbuilding processes, bottleneck identification and estimation of the impact of new technology implementation is extremely difficult to derive via analytical methods. The simulation model of a large-scale discrete event system can be considered as a collection of sub-systems, which are represented by the simulation models that are independently created, modified, and saved. This approach also includes methods that integrate these sub-models into an overall model in order to run different scenarios and identify global performance measures.

1 INTRODUCTION

The modeling and simulation of large-scale discrete event systems needs to be approached in a distinct manner. A simulation model of a large-scale system should be flexible, easy to change and still have fast model execution for experimentation. Currently, no major discrete event simulation software includes the tools necessary for hierarchical modeling. The objectives of a simulation project determine the level of detail for the modeling. Large scale-models involve the measurement and study of individual functional areas as well as system as a whole. This factor often makes large-scale models complex and slow to run.

Zeigler, Oren, and Sargent, et al., have documented hierarchical modeling concepts since the early 1980's. Hierarchical modeling provides a way to manage large-scale complex systems by considering them as a collection of sub-systems that are represented by simulation models that are independently created, modified and saved. The sub-models generate statistics specific to the sub-system in question. These models provide sufficient detail of the sub-system in order to derive finer performance metrics, such as resource utilization and sub-assembly/component cycle times. This type of modeling generally allows for quick scenario analysis that can be focused on the impact of a single or a specified set of sub-systems.

This hierarchical approach was recently used in a project for a leading shipbuilding company. The shipyard was jmplementing a new scheduling system and desired a way to verify that the schedule was feasible. The company chose simulation in order to evaluate the generated schedule as well as various automation strategies and their effect on system performance. Specifically, this hierarchical modeling method was used to allow shipyard managers and planners to 1) identify bottlenecks within the shipyard as a whole and within specific work centers, 2) determine the impact of technological improvements, resource maintenance plans and resource assignment on the shipyard schedule and 3) establish the appropriate levels of manning, resource maintenance and resource assignments in order to comply with that schedule.

A ProModel simulation package, comprised of an overall shipyard model and nine sub-models that represent key work centers in the shipyard, along with an integrated Visual Basic Interface was developed. The interface controls the execution of all the simulation models, as well as allows the user to enter parameters and control options for each model. In addition, the interface automatically extracts scheduled shipyard activity for each model and parameters such as entity attributes (e.g., plate dimensions) and resource constraints (e.g., crane lifting capacities) that are defined in the shipyard's scheduling software (also a KCG product).

The main challenge was twofold: to integrate the individual sub-models into an overall model in order to evaluate the overall system performance under different scenarios; and to seamlessly tie the Visual Basic interface with the scheduling software. Applying the hierarchical modeling approach, the overall model was designed to mix and match the different variations of the sub-models in a user-friendly manner via the Visual Basic interface.

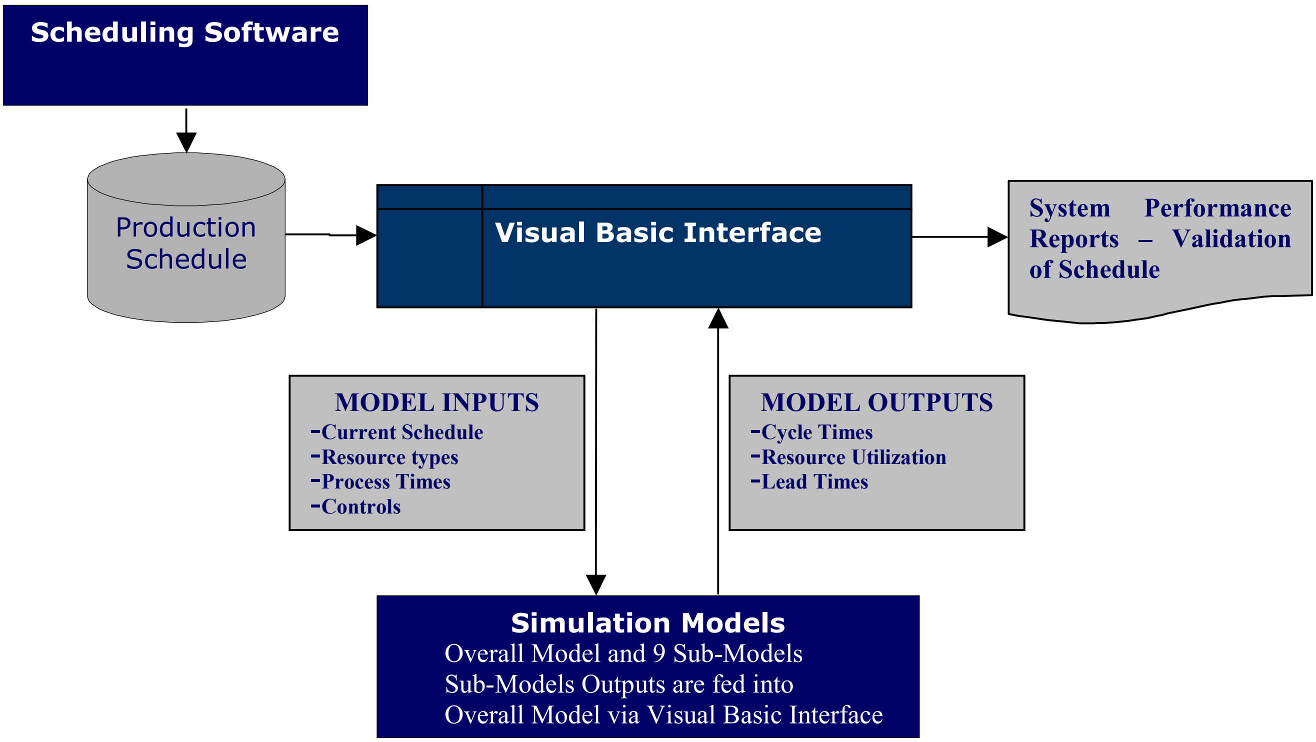

Figure 1: VB Shell Scenario-Runner Configuration

2 SIMULATION METHODOLOGY

The project involved the integration of the scheduling database with a series of 9 sub-models and an overall model. ProModel simulation software was used for the development of these sets of models. The data from scheduling had to be filtered to derive the information pertinent to each work center that was to be modeled as well as manipulated into a format that could be read into ProModel while keeping the integrity of the schedule. Based on the model that was selected to be run in the Visual Basic interface, the appropriate tables were queried from the schedule database for extraction of the correct data to feed into the selected model.

A major component of the project was the ability for shipyard management to use the modeling tool for all types of ships that are built by their company both currently and in the future. This led to various complexities that had to be accounted for in the models. For example, different ship types may require different resource types to work on the sub-assemblies and assemblies that comprise the ship. In addition, the processing times for similar sub-assemblies may vary between ship types. To accommodate for these differences, the Visual Basic interface was comprised of various input areas that were used to specify the required resources and processing times specific to each ship type present in the shipyard's schedule.

In a shipyard, material-handling equipment, such as overhead cranes, carriers, and forklift trucks, is a very important part of the process, in addition to the human resources, such as the welders and layout personnel. These resources were modeled as part of the sub-systems to identify the bottlenecks in the process sequences for both manual and automated activities. Each sub-model was developed with enough detail to be consistent with the overall project objectives and the input and output requirements. For example, downtime parameters as well as factors for process complexities based on component attributes were definable by the user via the front-end to feed into the sub-models. Other user-controlled factors included fatigue, delays due to weather, and lifting capacities of various material-handling equipment, just to mention a few.

Once the sub-models were developed, the next step was to develop an overall model with a minimum level of detail that was also controllable by the user via the Visual Basic interface. The objective of the overall model was to tie the independent sub-models to derive the cycle time for the entire ship from entry of raw material into the fabrication shops to erection and completion of the hull in the launching ways. Figure 1 shows the VB Shell Scenario-Runner Configuration that integrates the overall model, the sub-models and the scheduling database that contains the feeder data for the models.

As can be seen in Figure 1, the basic modeling hierarchy is described as follows: The user can alter parameters, including start and end dates that are used to query the appropriate data in the schedule in order to run any or all of the simulation sub-models. The overall model uses the output from these sub-models to specify the global cycle time averages and standard deviations for the work centers. These cycle time statistics were differentiated by the hull types extracted from the schedule and the sequence by which the hulls were built. The output from the overall model includes the overall cycle time for each hull scheduled within the time window specified by the user via the VB interface as well as the utilization rates for heavy lift transporters that run between work-centers.

2.1 Sub-Models

Each sub-model was run individually and experiments were conducted to evaluate the system's performance. The sub-models included the different ship components/part types (such as plates, structures and sub-assemblies) as well as the material handling equipment and the resources for each model specification. The sub-models were designed so that the current system could be run and the results analyzed via the Visual Basic interface. This analysis included the automated identification of the bottlenecks in the process sequences and the possible areas that could benefit most from new technologies and automation. The sub-models could then be manipulated to account for the process times, the downtime frequencies and downtime durations for the new technologies. For each sub-model, experimentation also included the development of an output file, which included the following information:

- Part type/component
  -For some sub-models, this component was as detailed as a plate; in others as large as a unit or hull.
- Flow time parameters
  -The flow time of a part is the time it spends in the system from its arrival to its completion.

2.2 Overall Model

An overall model was also developed in order to integrate the individual sub-models and also to evaluate the overall system performance.

The sub-models were represented as "black boxes" in the overall model. Since this did not allow for an accurate representation of detail, it was necessary to estimate the overall flow time (process time) for each sub-model. This was done via an output file, which was created by experimenting with each sub-model. To be able to represent the overall system accurately, the following assumptions had to be made:

- The process flows between the sub-systems can be defined for each part type.
- There is sufficient staging or storage areas for the semi-finished parts
- First-come-first-serve is used as a dispatching rule
- Each sub-model may have its own orders (arrivals) if necessary
- Each sub-model has a probabilistic process time, which is based on the estimated flow time.

This last assumption about the probabilistic process time is the most important assumption. This assumption is based on previous research conducted by Kaplan and Unal (1993). They concluded that the number of entities in a system and the average load of the sub-system are sufficient enough in order to estimate the flow times. They successfully used multiple regression analysis to estimate the flow times as a function of these two factors. Before reaching the above conclusion they tested several other parameters through a rigorous testing procedure. Since the basic assumptions of this shipbuilding case were similar to the ones given by Kaplan and Unal, a similar approach was devised here. Because the work centers were not modeled in as much detail in the overall model as in the sub-model, the probabilistic process time of the work center was determined using multiple regression analysis. In the overall model:

- Each sub-model was represented as a single infinite capacity location in the overall model
- The process time for the location was defined as a random variable with the parameters given in the appropriate sub-model output file
- The heavy lift material handling equipment for movement between the sub-models was included
- Key measurements such as system throughput and were written to an output file.

2.3 VB Shell Scenario-Runner

Because ProModel simulation software version 4.2 has OLE capability, MS Excel's Visual Basic was used to create a user friendly shell which is called a Scenario-Runner. Prior to the running of the sub-models, the user specifies the start and end hulls to be included in the modeling session. This is then interpreted into start and end dates by which the Visual Basic interface can extract the appropriate data from the schedule database. Another important feature of the Visual Basic interface is that it allows a user that is not experienced with ProModel to run the models without having to enter or understand the models' internal code. In addition, outputs from the simulation sessions are generated and presented via the same interface.

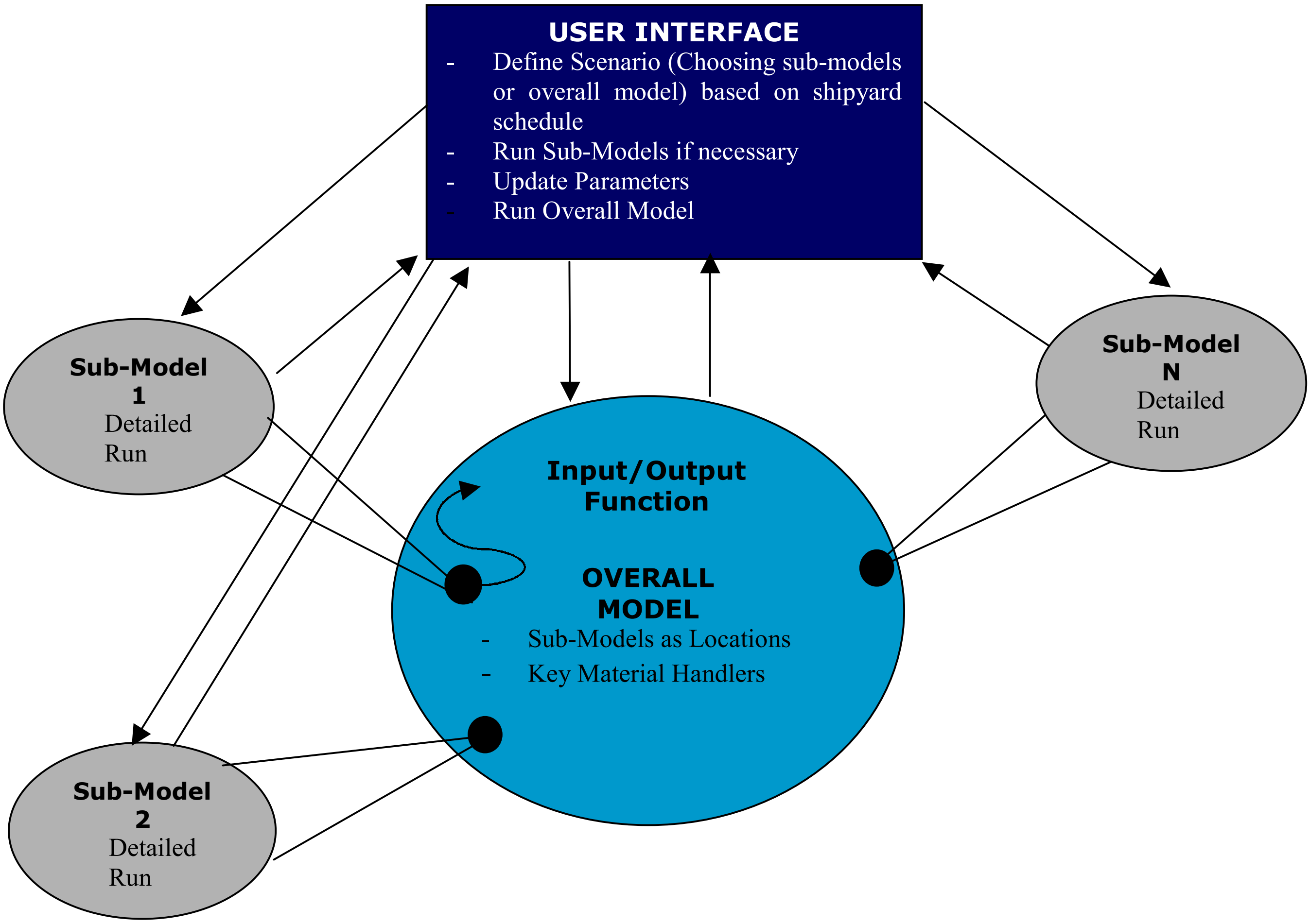

Figure 2: Visual Basic Interface Details

In this manner, key measurements are presented to the user without having to enter ProModel's internal output report. Figure 2 illustrates the link that the Visual Basic interface provides between the overall model and sub-models. In summary, this VB shell was designed to perform the following functions:

- Extract data from the schedule database based on the start and end hulls selected by the user.
- Help to define the scenarios for each of the sub-models that are to be run.
  -This includes definition of parameters such as down times, weather delays, etc.
- Maintains the scenario definitions
- Produces customized reports and graphs of key measurements.

3 CONCLUSION

The hierarchical modeling approach described here is very useful and flexible for the simulation analysis of large-scale systems. Using the VB Shell–Scenario Runner provided, the user can easily mix and match variations of the sub-models and achieve the overall project's objective. Hierarchical modeling also simplifies the development of the overall model into a simple selection of sub-models from a library of models by using the estimated flow time for the selected sub-model.

Currently, experiments of the overall model are not yet complete; therefore, we can not summarize the benefit of this approach other than simply comparing the model speed. Nevertheless, KCG has experienced in a project with another major U.S. shipyard with a similar model hierarchy, the use of an overall model parametrically tied to sub-models vs. the merging all the sub-models into an overall model decreased the expected run time by to less than a fourth of the run-time of a merged overall model.

REFERENCES

Kaplan, A.C and A.T. Unal, "A probabilistic cost-based due date assignment model for job shops" Int. Journal of Production Research 1993.

Kiran, A.S. and T. Cetinkaya, "Unpublished Project Report", Kiran & Associates, 1998.

Luna, J.J., "Hierarchical Relations in Simulation Models" Proceedings of 1993 Winter Simulation Conference.

AUTHOR BIOGRAPHIES

ALI S. KIRAN is President of Kiran Consulting Group and is an internationally recognized authority in Industrial and Systems Engineering, Simulation and Scheduling. He and his group actively consult to Fortune 100 and 500 companies here and abroad in these areas. He is also the co-author of the Mixed Model Scheduler (MMS®) Advanced Planning and Scheduling System. He previously taught Industrial Engineering at leading universities, including the University of Southern California and Texas Tech University. He holds an M.S. degree in Mechanical Engineering and a Ph.D. in Industrial Engineering. His e-mail address is `<ali.kiran@kiran.com>`.

TEKIN CETINKAYA is Director of Industrial Engineering Services and has been with Kiran Consulting Group since 1997. Before joining KCG, Mr. Cetinkaya worked as a Senior IE at Packard Bell NEC for over two years and worked as a Mechanical Engineer in an aerospace company in Los Angeles for five years. Since joining KCG he worked with numerous companies including Avery Dennison, Compaq, Columbia Sportswear, Motorola and several others. Areas of expertise include Business Process Re-Engineering, World Class Manufacturing, and Simulation Modeling. He holds B.S. and M.S. degrees in Industrial and Systems Engineering. His email address is <tekin.cetinkaya@kiran.com>.

JUAN CABRERA is an Industrial Engineering Services Consultant with Kiran Consulting Group since 2000. Prior to joining KCG, he has served as a U.S. Navy officer and as an engineer for seven years. He holds a BSE in Naval Architecture from the University of Michigan and a ME in the same area from the University of California - Berkeley as well as an MSE in Industrial Engineering from the University of Michigan. Since joining KCG, he has worked on several projects, including documentation writing for a Consilium software component, time standards and workload analysis study for Qwest Dex and the simulation project for the subject shipyard of this paper. His email address is <juan.cabrera@kiran.com>.