

A FRAMEWORK FOR CONFIGURABLE HIERARCHICAL SIMULATION IN A MULTIPLE-USER DECISION SUPPORT ENVIRONMENT

Wenlong Zhao
Alexander Verbraeck

Systems Engineering Department
Faculty of Technology, Policy and Management
Delft University of Technology
P.O.Box 5015, 2600 GA Delft, THE NETHERLANDS

ABSTRACT

In this paper we discuss the new challenge for using simulation modeling tools to support decision making in a multiple-user environment when organizations make business process changes in complex logistics systems. Current simulation based decision support tools are mainly single-level oriented and lack the capability and efficiency to support the business process analysis used by multiple users. Operational freedom for configuring executable simulation models is required by users to conduct customized services. This challenge motivated to design an architecture for configurable hierarchical simulation to support multiple-user decision making in a flexible way. The preliminary idea of a new design principle is presented in this paper.

1 INTRODUCTION

Nowadays, with the impact of the fast developments in information technology, the global political and economic situation and ever-changing customer demands, organizations have to operate in a dynamic and turbulent environment that requires fast and flexible responses to their changing business needs. As a result, they have to innovate continuously to align their structure, processes and technologies for survival (Den Hengst and de Vreede 2004).

Many organizations, like harbours and airports, are faced with the problem that managing their businesses is becoming increasingly difficult. Business threats from external competitors force them to have to find ways to increase their competitive advantages. However, due to limited development spaces, i.e. a lack of land space to build new terminals, they have to pay considerable attention either to improving current operational processes in their logistics systems, or to thinking of new development opportunities, like building a new harbour or building an extra runway. Designing new infrastructures and associated business activities is not a straightforward task because or-

ganizations have to deal with many dynamic and complex situations both from the internal and external business environment. Frequently, organizations face difficulties with the “reachability” of their business strategies. Organizations cannot implement a strategy effectively if they are unable to map it seamlessly into processes and applications (Ebstrategy 2004). How to bring in operational insight to support discovering the feasibility of high level strategies when organizations make changes is quite challenging.

Considering the current business environment, organizations can be treated as dynamic and complicated agents that have constantly to adapt to “best fit” their environment. When making changes, for instance, in logistics systems which can be treated as complex systems, such as changing or (re)-designing logistic and transportation issues in operational processes at airports and harbours, many stakeholders are involved in the decision making process. Communications among stakeholders regarding design issues usually become very difficult. Stakeholders have their specific roles in a project, and their responsibilities change when their roles change. Based on their different roles, various stakeholders may work at different hierarchical job levels and look into operational processes at different levels of detail, i.e. at a strategic level, at a tactical level or at an operational level, both for setting up a process and during its execution (Henderson 2002). For example, senior managers, middle managers, analysts, engineers and technicians, all work on different parts at various levels of the process design and analysis. They will also be interested in different parts or aspects of a system, in which the aspect issues and the coordination problem have to be taken into account when dealing with the system as a whole. For instance, system structure changes or logistics process changes may cause many aspect concerns, like environmental issues, safety issues, economic issues, etc. It is really hard to evaluate these aspects and consider all the related detailed operations. Predictions and measurements are difficult to pinpoint prior to implementation (Henderson, 2002).

In order to predict and evaluate the outcome of the changes made in systems, organizations need supportive tools. Simulation modeling tools provide a structural environment in which one can understand, analyze and improve business processes (Pegden et al.1995). The big advantage of using simulation in business process management is that the proposed strategies and alternatives can be qualitatively and quantitatively evaluated prior to make actual changes. However, current simulation based decision support tools are mainly single-level oriented, which means that simulation models are constructed for a certain purpose and it is almost impossible for users to choose and change the levels of detail in the models in an easy way. This kind of design lacks the capability and efficiency to support business process analysis in a multiple-user decision making environment, which requires multiple views on the operational processes. The contents and information provided by supportive tools should cover multiple users' work scopes and interests, i.e. multiple detailed levels and multiple-aspect evaluation, in which they can share their knowledge and coordinate better. These defined requirements increase the difficulties met when using current simulation modeling tools.

The multiple users' view implies a hierarchical view on systems or system processes. Hierarchy can be used to deal with the complexity in reality. Hierarchy implies a framework that permits complex systems to be built from simpler ones, which can be used to help people to organize, to understand, to communicate and to learn about systems complexity (van Gigch 1991). Using hierarchical modeling design and development as the way of modeling means that models can be quickly constructed in a "rough form", and later implemented or replaced in more detail.

Hierarchical modeling enables model aggregation (model abstraction) and model decomposition. Users can take highly aggregated simulation models to address high level questions quickly, and take detailed simulation models to obtain detailed insight into critical parts of the systems. Our research is aimed at designing an architecture for hierarchical simulation which can be easily configured by multiple users to support their decision making in business process management in logistics systems.

In the next section, we will discuss the major theoretical foundation for configurable hierarchical simulation. We will discuss the main services required in configurable hierarchical simulation in section 3. Then a design framework will be presented and discussed in section 4. We will summarize the main contributions of this work in section 5, and make the final conclusion in section 6.

2 THE THEORETICAL FOUNDATION OF CONFIGURABLE HIERARCHICAL SIMULATION

Hierarchical modeling and simulation has been aware for many years, see Fishwick's hierarchical reasoning (1986), Zeigler's Hierarchical DEVS (2000). We use the term "configurable" in relation to the flexibility of the models. The models should be designed in such a way that multiple users have the possibility to construct the models at different levels of detail without too much effort. The intention is not to pre-construct the models or model components at different levels of detail. Instead, these operations will be done by simulation applications automatically. Configurable hierarchical simulation (CHS) is designed to achieve this goal. The theoretical foundation of CHS partly covers Zeigler's hierarchical DEVS, Fishwick's multimodeling and Kleijnen's regression metamodeling.

2.1 Zeigler's Hierarchical DEVS

Hierarchical DEVS emphasizes on constructing models at several levels of abstraction. Abstraction in DEVS refers to a method or algorithm applied to a model to reduce its complexity while preserving its validity in an experimental frame, see Zeigler (2000).

Aggregation in DEVS is defined as an abstraction method that maps a coupled model into another, becoming a less complex coupled model, with the intent of preserving behavior in some applicable experimental frame. There are two steps to do aggregation:

- first step-partition: the component set is first partitioned into blocks in which each block can be described as a coupled model. This step does not involve any abstraction process, but makes changes in structure. Therefore there is no any model reduction as yet.
- second step-lump: the partitioned blocks are abstracted to a simpler version and are formed into a final set of coupled models, called lumped models. These models should show a significant reduction compared with original models.

Various coupling conditions and principles were introduced by Zeigler, et al. (2000) to complete these two steps to achieve valid reduced models. One of the principles to do partition and lump is to require block homogeneity which means all components in a block must have the same model structure, which also implies that different blocks may have different characteristic component model structures. These aggregation mappings are called identity-erasing. The algorithms and theorems defined by Zeigler for model reduction form the core theories for our architecture design and development in terms of model aggregation/abstraction.

2.2 Fishwick’s Hierarchical Reasoning and MOOSE Project

Another important theoretical foundation for our research is Fishwick and his co-work’s research. Fishwick’s hierarchical reasoning (1986, 1995) places the emphasis on process abstraction which permits multiple levels of abstraction during the simulation and gives the user methods for focusing on different aspects of the simulation at different levels of abstraction. This is more based on qualitative reasoning. The underlying principle is that each process abstraction level should be defined so that the maximal amount of knowledge is updated with each event. Fishwick’s hierarchical reasoning provides a good framework and workable methodology for producing models at multiple levels of representation even if Fishwick uses a hand-driven approach to predefine models at each level. Fishwick does not intend to realize these four stages, especially for higher levels of abstraction in an automatic way. However we still consider his work to be a very valuable reference for our architecture development.

Another research field we are going to discuss concerns Fishwick’s MOOSE (Multimodeling Object-Oriented Simulation Environment) research project. A multimodel is a heterogeneous hierarchy of models where a model component at one level of "abstraction" is sub-refined into a model, possibly of a different type, at the next lower level. Multimodeling endows the simulationist with the capability to blend different model types together to form hybrid models (Fishwick 1993, 1995), such as Petri networks, automata, Markov models and block models. The underlying motivation for this research is that most systems will contain model components whose types are different and the “all in one” simulation language does not match most real world problems. With regard to our research focus: trying to deal with the difficulty of “levels of model representation” and “coupling aspect models and eventually making model reduction for multiple-user decision support, the multimodeling framework is helpful for dealing with aspect models. We therefore consider Fishwick and his co-worker’s work to be one of the cornerstones in our research context.

2.3 Kleijnen’s Regression Metamodeling

Another theoretical foundation is Kleijnen’s regression metamodeling, which is taken from the operations research field. A regression model is a metamodel of the simulation model, which approximates the input/output behavior of the simulation model that generates the input/output data to which the regression analysis is applied (Kleijnen 1995). The basic idea behind regression metamodeling is to reduce ad hoc situations, and give more general results in less time, see Kleijnen (1995). Kleijnen focuses on the design of experiments (DOE), making large variance reduc-

tions for models, sometimes up to 99% reduction, see Kleijnen (1971), to improve simulation efficiency. It is proposed to apply DOE combined with regression analysis to sensitivity analysis and optimization in simulation. DOE treats the simulation model as a black box. It can apply to any type of simulation. Only the input and output are observed and analyzed. This approach eventually leads to valid and reduced models by reducing the factors required to improve simulation efficiency. Kleijnen’s regression metamodeling is valuable for our hierarchical simulation design with regard to reduced models

2.4 The Relation of the Discussed Theories with Our Research Context

As discussed, Zeigler’s hierarchical DEVS, Fishwick’s multimodeling and Kleijnen’s regression metamodeling form the cornerstones of our research theory. The relationships between their work and our research design are illustrated in figure 1.

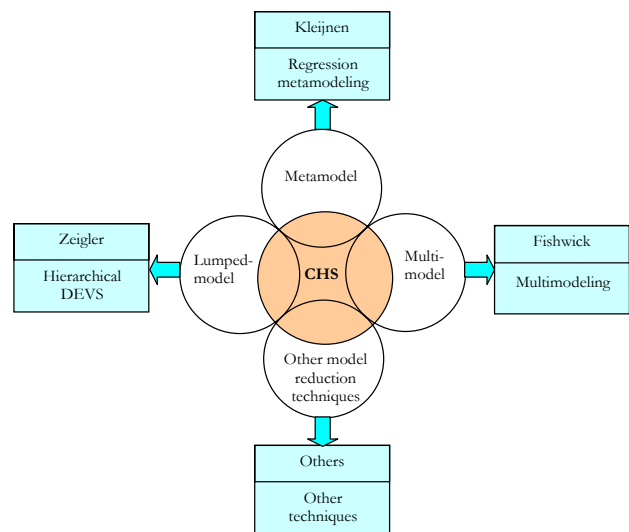


Figure 1: The Theoretical Foundation of Configurable Hierarchical Simulation (CHS)

The underlying theories of configurable hierarchical simulation partly cover Zeigler, Fishwick and Kleijnen’s work, and also other existing model reduction techniques, like state aggregation and notional behavior aggregation as defined by Frantz (1995), etc. Designing configurable hierarchical simulation (CHS) may also help us to elaborate a new theory or an approach to combine these existing theories into a consistent framework.

3 SERVICE-ORIENTED DESIGN FOR CONFIGURABLE HIERARCHICAL SIMULATION

3.1 Main Services in the Architecture

In this section, we are going to discuss the key issues in our research and the preliminary idea of a design principle for configurable hierarchical simulation.

Supporting multiple-user decision making enabled by models and simulation enhances a new user-oriented service design that should better cover “multiple-level” and “multiple-aspect” concerns of users. Models and simulations can play the role of services for service design. There are a lot of definitions of service. According to Grönroos (1990), service is defined as “A (series of) activities of more or less intangible nature that normally, but not necessarily, take place in interactions between the customer and service employees and/or physical resources or goods and/or systems of the service provider, which are provided as solutions to customer problems.” Service has the intangible feature that may add value to users, in our research context, to support decision making and problem solving efficiently. As already discussed, the architecture design should create a multiple-user workable decision support environment in which users have the “operational freedom” to do the focused analysis with subsystems or aspect concerns without being isolated from the view of a whole system. However, each operation with models, i.e. model selection (sub models, aspect models), model abstraction (model aggregation), leads to model reduction, and targets reduced models. We call *the activities of producing reduced models as model reduction services*, which contain two major services: *model selection services* and *model aggregation services*. These services, aiming at enhancing the flexibility of simulation model construction, should eventually provide a better support for multiple users. Therefore model reduction services form the main functionality in the architectural design for configurable hierarchical simulation.

We will now illustrate the idea discussed above in detail. Our research focuses on the domain of logistics systems. Each logistics system may contain many subsystems, and these subsystems may interconnect. Simulation models or model components can be designed to represent subsystems with simplification. The changes made in operational processes within one or several subsystems can lead to aspect concerns, where the selected aspect models can be coupled with simulation models of subsystems to provide aspect evaluation. Figure 2 is designed to present this idea graphically.

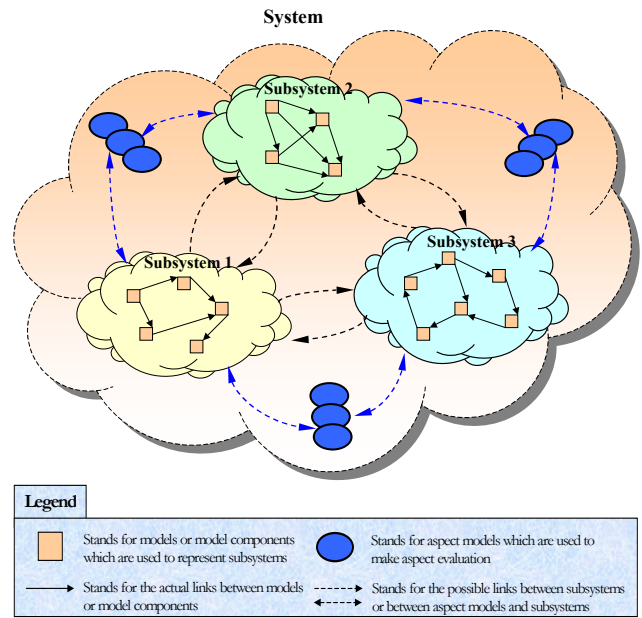


Figure 2: The Representation of a System and its Subsystems by Models

Figure 2 gives an overview of using models to represent a complex system and its internal relationships among the subsystems. As already discussed, the situation becomes complicated for modeling a system and its associated operational processes when considering multiple-user decision making support. Figure 3 is designed to show an overview of a multiple-user decision support environment in our research context.

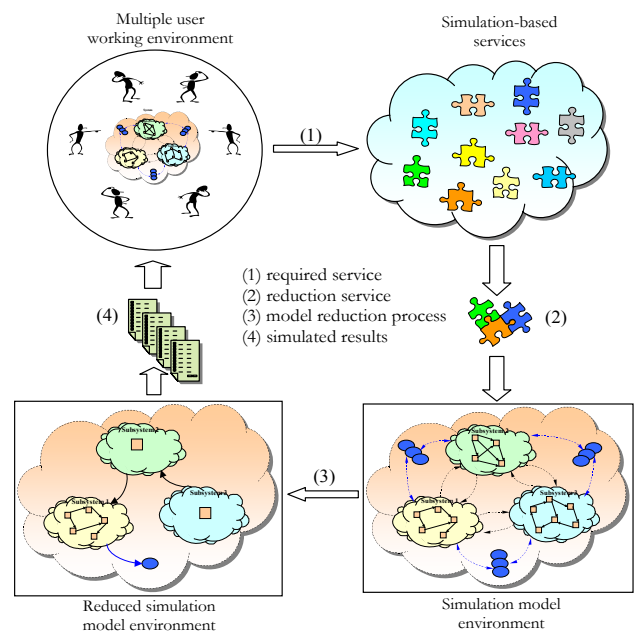


Figure 3: A Multiple-User Decision Support Environment

The simulation-based services are proposed and will be designed and developed for any service request with regard to operational insight into subsystems or systems with the “level” and “aspect” concerns, see (1). The services mainly consists of model reduction service components. With regard to each specific service request from users, suitable service components will be formed, see (2), to provide certain reduction services on the top of an executable simulation model environment, i.e. aspect model selection, model aggregation for subsystems. The foundation theories mentioned for CHS play major roles here. Finally a reduced simulation model environment will be generated and this will be executed to provide the required decision support services for users, see (4). The aggregation service is one of the core reduction services. An example is given in figure 4.

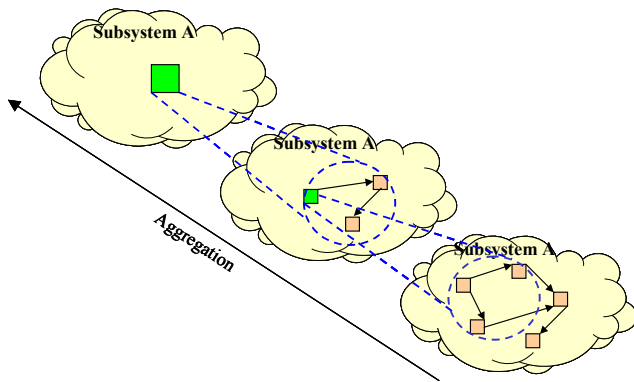


Figure 4: An Example of the Process of Model Aggregation

Models that are designed to represent a system or subsystems can be aggregated to many levels of abstraction based on what is requested. This operation will lead to multiple levels of model representation, and the model representation at each level will be executable. This is designed to satisfy user’s requirements, from the point of view of the quality of information produced, and from the perspective of a better understanding for operational processes in logistics systems. This is our main research direction and focus. It is quite obvious and logical to consider an opposite situation, only designing a single level of model representation to produce multiple levels of aggregated information for multiple-user decision support. Actually this is a common design approach in current simulation applications used in organizations. This idea is illustrated in figure 5.

Once the simulation models are designed and constructed by model developers, the detailed levels of the simulation models are fixed or predefined. Users do not have the possibility to zoom in and zoom out in a flexible way with constructed models. What users can usually do is to select the output they need which is generated by a fixed simulation model environment, see (2). Then the simulation application can make some proper operations, i.e. selection, aggregation, for the simulated results through designed reduction algorithms or services, see (3). Finally the selected

or reduced results will be presented to users in various forms, see (4). In this way, simulation applications enable users to make output reduction, instead of having to make model reduction to produce the required results.

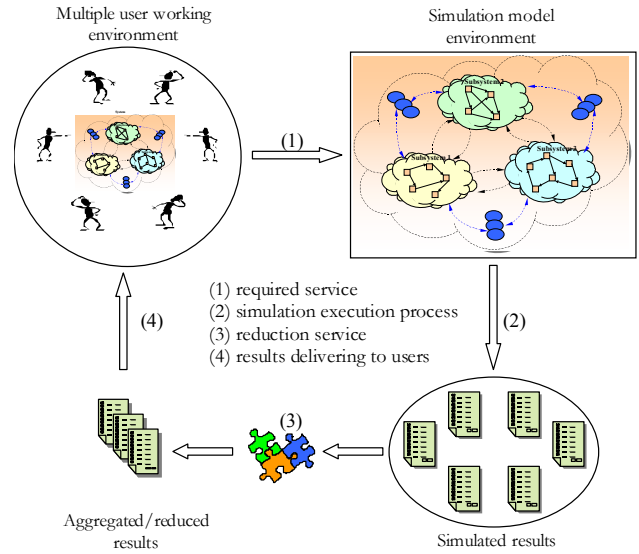


Figure 5: A Single Level of Model Representation for Multiple-User Decision Support

Compared with the multiple levels of model representation, the approach where a single level model representation is constructed needs relatively less effort, and it is not necessary to consider validation problems for multiple levels of model representation. However, a single level of models cannot meet the needs for multiple users because users require the produced information at different levels of detail, and require the different levels of insight into processes. Furthermore, for modeling a very complex system, like modeling passenger flows in airport terminals, only using a single level of models to perform all the model operations is very costly.

Rioux and Nance (2002) imply the need to reproduce systems at different degrees of accuracy and precision in a cost effective manner without significant degradation in achieving certain goals in the course of model construction. Many other researchers have similar perspectives and arguments, such as Kachitvichyanukul, et al. (2001), Kasputis and Ng (2000), Benjamin, et al. (2000), Davis, et al. (2000). The effort required to construct models at the different levels of detail is rather sensitive to the cost-benefit effect. Davis, et al. (2000) argue that an overly detailed model is difficult to verify, expensive to modify/maintain, and most importantly, takes significantly longer to run. However, an overly abstract model may not show the problems in such a way that they are “actionable” for the personnel who must use the results. The ideal model is one that will achieve the stated goals within the

desired level of accuracy with minimal effort. This argument is quite interesting and also valid for our research.

As already discussed in this paper, we depart from a multiple-user point of view, and come up with the idea that users need “operational freedom” to configure simulation models to produce their required information. Designing an architecture to support models with multiple levels of representation is proposed to meet the new challenge.

4 A DESIGN FRAMEWORK FOR CHS

Before we discuss our preliminary design framework for configurable hierarchical simulation (CHS), we would like to address the design requirements first.

4.1 Design Requirements

Using configurable hierarchical simulation should create an environment in which multiple users may feel comfortable to deal with their needs. This environment should :

- *support web-enabled loosely-coupled service oriented design*
Multiple users means that the users work at different locations. This requirement is expected to support distributed users.
- *support different aspect models*
Aspect models should be tuned to plug-play with simulation models.
- *support aggregation services for multiple levels of model representation*
A automatic model aggregation service is desirable to achieve flexibility.

- *support model selection services*
The focused analysis by different users should be supported by this service.
- *support validation services*
These services should function as consistency checkers to avoid errors made in the operations.

4.2 A Design Framework

Based on the previous discussion, especially in section 3.1, and also taking into account our research context, we come up with the first idea of a design framework for configurable hierarchical simulation, which is illustrated in figure 6. In the following section, we will discuss each element/component that appears in this framework in detail.

4.2.1 Service Requirements from Multiple Users

The service requirements used here are drawn from specific domains. This leads to several concerns:

- In the initial phase, the different user groups and their basic service requirements will be investigated to help us construct our library of base models.
- During the operational phase to test and evaluate different scenarios or policies, users may specify their preference: i.e. which aspects, which subsystems, which processes, of what duration and at which level of detail they want to measure.

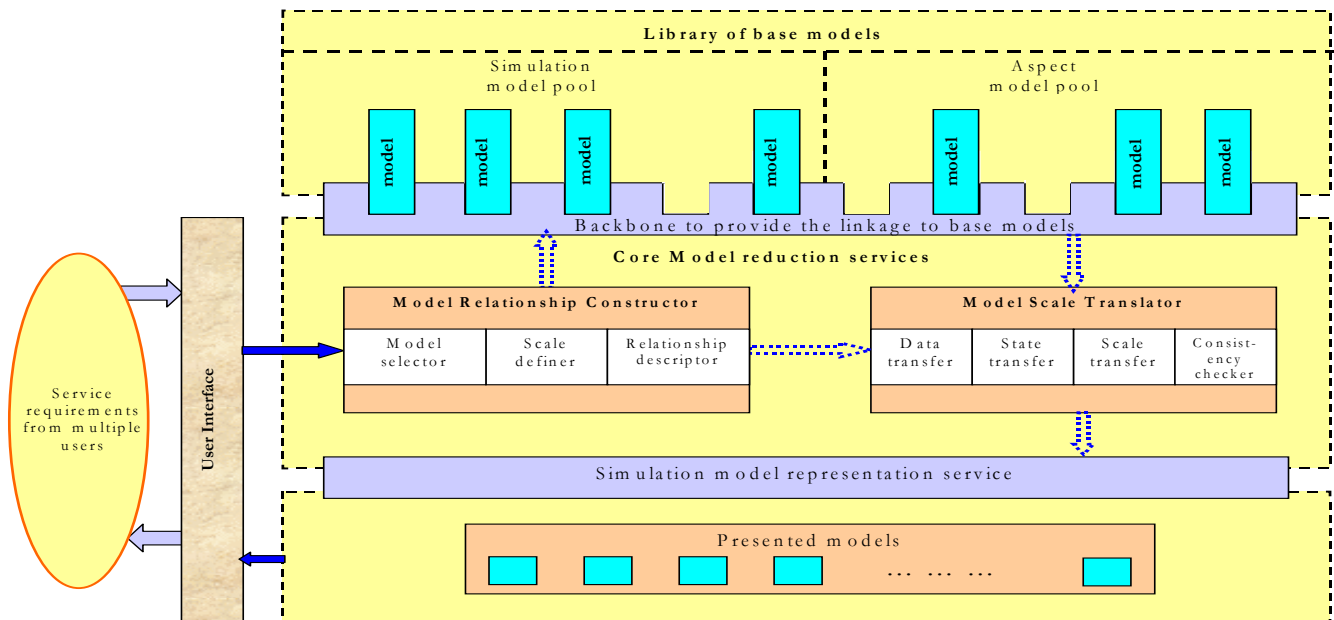


Figure 6: A Framework of Model Reduction Services for Multiple Users

4.2.2 User Interface

This acts as intermediary between users and models. Users do not see the actual executable simulation models, but interact with a user interface. There are several concerns here:

- Users may have the opportunity to customize their required services through a certain interface with regard to options on aspect, process, model scale and time duration, all of which have already been discussed above.
- The provided information/data will be presented to users through this interface, such as tables, graphs, text files and spread sheets.

4.2.3 Library of Base Models

We divide this library into two categories.

- One, simulation model pool: this contains all the predefined simulation models in certain scales/levels based on the domain specific user groups and their basic service requirements.
- Two, aspect model pool: this contains all the aspect models which already exist in organizations, i.e. noise calculation models, transport models, economic models, business models, clustering design models, etc. These models will be loosely-coupled with simulation models for the evaluation of system aspect concerns.

These base models are then used to provide the basic services for users, and the presented multiple representations of simulation models are built based on these models. How to construct a library of base models is an important issue. There are two concerns of the way of modeling.

- One: the models should be built on certain scales: either as generic as possible, can be specified into more situation, or as detailed as possible, can be aggregated into required scale, but they should be executable.
- Two: the models should be scalable and relatively reusable: therefore component-based modeling will take place here.

4.2.4 Backbone Providing the Linkage to Base Models

This is the core service used to link the base models. The base models might not be necessary in client computers. Instead, some very heavy models may be deployed in different servers. The backbone should provide a web-enabled loosely coupled service to link and also “adapt” these required base models. Some of the challenges here are to “wrap” or code aspect models so they become workable.

4.2.5 Core Model Reduction Services

This is the core architecture development part for constructing simulation models at multiple aggregation levels. There are two main functional services named “Model relationship constructor” and “Model scale translator”.

4.2.5.1 Model Relationship Constructor

The “model relationship constructor” (MRC) can be treated as the bridge or enabler to transfer user specification to model construction eventually. The main function of an MRC is to help to define all the relations and logic required to construct the multiple levels of aggregated models needed to satisfy user requirements transferred from user interface. A MRC consists of three components.

- Model selector: this is used to choose the models from the library of base models according to the user requirements.
- Scale definer: this is used to define the scale of the chosen models and the logic of scale realization.
- Relationship descriptor: this is used to describe the interconnected relationship among chosen base models, and can help model integration.

4.2.5.2 Model Scale Translator

The “model scale translator” (MST) enables automatic model aggregation and model integration controlled by a MRC. Four components are needed to support the MST.

- Data transfer: this is used to aggregate, refine and transfer I/O data between two executable models at different scale levels during execution.
- State transfer: this is used to transfer the states of objects between two executable models at different scale levels during execution;
- Model scale translator: this is used to realize the multiple levels of aggregated model generation.
- Consistency checker: this is used to check the consistency of all above operations, including data transfer consistency check, state transfer consistency check and multi-scale model transfer consistency check.

4.2.6 Simulation Model Representation Services

Simulation models at multiple scales or multiple aggregated levels can be built up using “core model reduction services” (CMRS). However these models need not be deployed in client computers. They might be stored and executed server side and display output to the user through “user interface”. These services should provide the func-

tions of web-enabled execution and storage for presented simulation models during execution.

5 CONTRIBUTIONS

Three major contributions are concerned by doing this research work.

- A new theory in configurable hierarchical simulation, dealing with the main concerns of multiple-level and multiple-aspect simulation, will be formulated.
- A new design for experiments to validate models with multiple levels of representations will be elaborated and the underlying principles and algorithms used for this method will be defined.
- Multiple users will be better supported if they use configurable hierarchical simulation.

6 CONCLUSIONS

In this paper, we have described the new challenges for supporting multiple-user decision making by using simulation modeling tools. We proposed a new design principle for hierarchical simulation which can be configured by users. Finally we addressed a design framework for configurable hierarchical simulation.

In the next step, we will explore a new theory or approach to integrate Zeigler's hierarchical DEVS, Fishwick's multimodeling, Kleijnen's regression metamodeling, and other model reduction techniques into a consistent framework. This can be achieved through the research instrument of laboratory experiment. This defined theory will thus be used to implement our architectural design for configurable hierarchical simulation. The validation issues when producing models at multiple levels of representation play a key role in a successful architectural design. The final step will be to exam user satisfaction with regard to the new design principle.

REFERENCES

- Benjamin P., D. Delen, R. Mayer, and T. O'Brien. 2000. A Model-Based Approach for Component Simulation Development, In: J.A. Joines, R.R. Barton, K. Kang, and P. A. Fishwick (Eds.) *Proceedings of the 2000 Winter Simulation Conference*, pp.1831-1839.
- Davis, P.C., P.A. Fishwick, C.M. Overstreet, and C.D. Pegden. 2000. Model Composability as a Research Investment: responses to the featured paper. In: J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick (Eds.) *Proceedings of the 2000 Winter Simulation Conference*, pp.1585-1591.
- Den Hengst, M and G. de Vreede. 2004. Collaborative business engineering: a decade of lessons from the field". *Journal of Management Information Systems/Spring 2004, Vol. 20, No. 4. pp.85-113.*
- Ebstrategy, web page www.ebstrategy.com/services/creating/creating.htm [accessed December 15, 2004].
- Fishwick, P.A. 1986. Hierarchical Reasoning: simulating complex processes over multiple levels of abstraction. PhD dissertation, published by University of Pennsylvania.
- Fishwick, P.A. 1993. Multimodeling as a Unified Modeling Framework", In: G.W. Evans, M. Mollaghasemi, E.C.Russell, and W.E.Biles (Eds.) *Proceeding of the 1993 Winter Simulation Conference*.
- Fishwick, P.A. 1995. Simulation Model Design and Execution: building digital worlds. Prentice Hall, Englewood Cliffs, New Jersey.
- Frantz, F.K. 1995. A Taxonomy of Model Abstraction Techniques. In: C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Go]dsman (Eds.) *Proceedings of the 1995 Winter Simulation Conference*.
- Grönroos, C. 1990. Service Management and Marketing: managing the moments of truth in service competition. Lexington, MA: Lexington Books
- Henderson, Peter (Ed.).2002. Systems Engineering for Business Process Change: new directions, Springer, London, UK.
- Kachitvichyanukul, V., J.O. Henriksen, R.G. Ingalls, C.D. Pegden, and B.W. Schmeiser. 2001. Simulation Environment for the New Millennium (Panel). In: B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer (Eds.) *Proceedings of the 2001 Winter Simulation Conference*, pp.541-547.
- Kasputis, S., and H.C. Ng. 2000. Composable Simulation. In: J.A. Joines; R.R. Barton; K. Kang; P. A. Fishwick (Eds.) *Proceedings of the 2000 Winter Simulation Conference*, pp.1577-1584.
- Kleijnen, J.P.C. 1971. Variance Reduction Techniques in Simulation. PhD dissertation, published in University of Tilburg.
- Kleijnen, J.P.C. 1995. Sensitivity Analysis and Optimization in Simulation: design of experiments and case studies. In: C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman (Eds.) *Proceedings of the 1995 Winter Simulation Conference*, pp. 133-140.
- Pegden, C.D., R.E. Shannon; and R.P. Sadowski. 1995. Introduction to Simulation Using Siman, 2e. The McGraw-Hill Companies, Inc., London.
- Rioux,G.P., and R.E. Nance 2002. Generalizing: is it possible to create all-purpose simulations? In: E.Yucesan, C.-H.Chen, J.L.Snowdon, and J.M.Charnes (Eds.) *Proceedings of the 2002 Winter Simulation Conference*, pp.783-790.
- Van Gigch, J.P. 1991. System Design Modeling and Meta-modeling. Plenum Press, New York.

Zeigler, B.P., H.Praehofer, and T.G.Kim. 2000. Theory of Modeling and Simulation: integrating discrete event and continuous complex dynamic systems. Academic Press, San Diego.

AUTHOR BIOGRAPHIES

ALEXANDER VERBRAECK is chair of the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology, and a part-time full professor in supply chain management at the R.H. Smith School of Business of the University of Maryland. He is a specialist in discrete event simulation for real-time control of complex transportation systems and for modeling business systems. His current research focus is on development of generic libraries of object oriented simulation building blocks in C++ and Java. His email and web addresses are a.verbraeck@tbm.tudelft.nl and www.tbm.tudelft.nl/webstaf/alexandv.

WENLONG ZHAO is a researcher in the Systems Engineering Department of the Faculty of Technology, Policy and Management of Delft University of Technology. His main research interest focuses on developing a fully customized simulation service to support multiple-user decision making. His email and web addresses are zwloug@yahoo.com and www.tbm.tudelft.nl/webstaf/wenlongz.