A CONTEXT-BASED MULTI-PERSPECTIVE MODELING AND SIMULATION FRAMEWORK

Çağrı Tekinay
Mamadou Seck
Michele Fumarola
Alexander Verbraeck

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

ABSTRACT

Constantly increasing complexity of organizational environments and changing demands of the stakeholders severely affects the strategic capabilities of organizations and reduces their decision making ability dramatically. Large-scale complex systems like energy grids, air-ground traffic control systems and logistics systems are designed in multi-actor environments and hence require various perspectives (e.g. financial, operational, and environmental) to serve different actors. Although this is the case, current simulation based decision support environments lack the capability of covering multiple system perspectives at once and in different levels of details to provide better understanding of the systems. In this paper, the key challenges and preliminary design ideas are discussed to provide a multiresolution modeling capability. We introduce a context-based 'view' concept as an enabler to support multi-perspective modeling in multi-actor environments.

1 INTRODUCTION

Current large scale organizations are multi-actor systems embedded in complex multi-actor environments. Each actor has its own interests and as a result will perceive the system in its own way. This makes it difficult to perform a system analysis reflecting all parties' interests simultaneously. Support tools are used by organizations to evaluate and foresee the consequences of their actions. Modeling and simulation provides organizations with the ability to analyze existing systems and to design improvements and evaluate them just before their real world implementation (Pegden, Shannon, and Sadowski 1995). Besides, to cover the interests of all actors involved, models tend to become larger and more complex, which reduces their usability. This leads to extensibility and maintainability issues.

Hierarchical structures have been considered an efficient way of coping with complexity. Simon's notion of near decomposability indicates that systems can be studied in a sufficiently accurate way by adopting a hierarchical decomposition (Simon 1962). As a consequence, hierarchical modeling became a cornerstone in M&S research. However, it must be realized that a hierarchical decomposition can only be done once a perspective has been chosen, which conflicts with the multi-actor nature of organizational environments.

Large-scale complex systems like energy grids, air and ground control systems or logistic systems offer services to different actors interested in many different facets of the system like financial, environmental, operational and safety related issues. Researchers have been striving to introduce new features to im-

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

prove the capabilities of the existing multi-actor decision support environments. In the field of modeling and simulation, the most focus was put on aggregation and abstraction for providing multi-resolution models. Less emphasis has been put on multi-perspective modeling and simulation environments.

In this paper we discuss the key issues and preliminary ideas to provide a context-based multi-resolution and multi-perspective modeling approach. Section 2 discusses the literature about multiresolution modeling. Section 3 presents a conceptualization of an approach for context-based multiresolution modeling. In section 4, this approach is applied to a demonstrative example. Finally, in section 5 we conclude the paper and present future work.

2 RELATED WORKS

Before talking about the literature on multiresolution modeling (MRM) some system theory and system hierarchy background is discussed.

2.1 System Theory, Hierarchy and Complex Systems

The major focus of system theory is to give a formal description of a system by using both its structural and behavioral attributes. Schmidt and Taylor (1970) and Roberts et al. (1983) define a ‘‘system’’ as collective interactions of entities working together for some logical reason. Depending on the investigator’s goal, a system which is constituted by interacting entities might be seen as a subsystem of another. This is fully compatible with Simon’s explanation of hierarchical systems (1962). Instead of just referring to the classical and narrower description of hierarchical systems, which basically identify the connection between subsystems with an authority relation, Simon extended the hierarchy concept by introducing the notion of inclusion hierarchy (Lane 2006). A general agreement exists that hierarchy is one of the enabler concepts for the understanding of complex systems.

Important contributions have been made in system theory by researchers such as, Fishwick (1986) and Zeigler, Praehofer, and Kim (2000), who have studied hierarchical modeling and simulation. The primary focus of Fishwick’s (1986) hierarchical reasoning is process abstraction, whose core ideas are derived from qualitative reasoning. Process abstraction permits multiple levels of abstraction during the simulation and gives the user methods for focusing on different levels of abstraction. Ziegler discussed different abstraction strategies in the context of hierarchical DEVS for reducing the model complexity, within a given experimental frame, without losing its consistency and validity. Among other simplification strategies like omission or linearization, aggregation is presented as a technique allowing for the combination of groups of components into a simpler single component which represents their collective behavior (Zeigler, Praehofer, and Kim 2000).

Abstraction becomes even more important in the context of ‘‘complex large-scale systems,’’ which Takahara et al. characterized as a kind of system where the number of nested components, which are both decision making and processing units, are extremely large and the system is consistently open to the information exchange with the surrounding environment (Takahara, Iijima, and Zhao 1988). This complexity causes communication problems between the organizational roles (i.e., at strategic, tactical and operational levels).

2.2 Theoretical Background of MRM

Davis and Bigelow (1998) define MRM as the process of constructing an individual model or model families providing variety of resolutions to different types of users for the same phenomena. The term resolution is used to describe the detail level of system components and their behaviors. For example, ‘‘high resolution’’ models have a more detailed description of the system components. Also, they provide the relations between the components in a more structural way. On the other hand, only having a high resolution perspective has its limitation on the scope of the system. Such pros and cons exist for the ‘‘low resolution’’ models. Low resolution models provide analytical agility but they need the assistance of high resolution models in order to provide an operational overview of a system (Bigelow and Davis 2003).

MRM is a highly applicable approach when dealing with composability (Davis and Tolk 2007). Composability is the capability of collecting the right components and forming them together in different combinations to complete a set of predefined user requirements (NRC 2006). The fundamental concepts have been studied and published by several researchers (Davis and Huber 1992; Davis and Hillestad 1993; Reynolds, Natrajan, and Srinivasan 1997) as a result of the multifaceted, multiparadigm studies (Zeigler and Ören 1986; Fishwick and Zeigler 1992). Further, from a top-down perspective, MRM enables the understanding of decomposition of a system into subsystems without losing the conceptual meaning provided by them. It also has the ability to provide the bottom-up meaning of the phenomena to the user (Davis and Tolk 2007). MRM helps the designer to form information at different levels of details with different levels of insights based on the user requirements. As a result, multiresolution representation of a complex system provides better control and understanding to the system users.

On the other hand, MRM has a few feasibility concerns and limitations which are well described by Bigelow and Davis (2003). One of the primary concern about the MRM is the issue of ''imperfect decomposition''. Such limitation can only be eliminated by only applying context-dependent approximations. However, the usage of such approximations inside the MRM scope is still limited (Davis and Tolk 2007).

Another feasibility issue about the MRM mechanisms is that it lacks a certain trigger mechanism which understands the situational changes and alerts the model to reconfigure itself. Providing such modeling capability not only helps the users to deal with emergent conditions (Yılmaz and Ören 2004) but also helps them to see the outcome of their strategies without any efforts of pre-construction.

The next section contains some preliminary design ideas and introductions of some new concepts for our multi-resolution multi-perspective modeling and simulation framework.

3 EXPLORING THE DESIGN SPACE

A ''view'' is a multiresolutional model description of an existing phenomenon inside a complex system which enables the run-time model updating by providing the necessary information about the certain context. As a complementary description, a ''view'' is a type of model on a certain perspective of a system which can be interpreted into multiple levels of representations for different organizational levels, in such a way that the model representation at each level is executable. The main concern of our ''view'' concept is to provide extensibility and adaptability that is not addressed in the Multi-resolution Entity design scheme.

The reason why we use the name ''view'' is to emphasize that the concept not only addresses the different looking perspective of stakeholders but it also defines their understandings about an existing phenomena inside a specific context or a jargon. It is very likely that a user might be interested in analyzing the operational, environmental, financial or safety perspectives of a complex systems. (or at least a certain part of the system). All these system perspectives constitute the user's ''view'' space for the desired system. However, a view is required to provide some contextual knowledge to the model so that the model can infer some meaning out of the user's query. For example, a user might be curious to observe the energy consumption of a vehicle, group of vehicles or machines inside a complex system during the runtime of the simulation. The energy consumption perspective in here forms one of the views inside the complete view space for this complex system.

The following section provides a detailed example of our discussed ideas to provide a better understanding about the introduced view concept. For the example implementation, we use a DEVS simulation model of automated container terminals which was previously represented in Fumarola, Seck, and Verbraeck (2010a). We select the quay crane equipment out of the whole container terminal model to show the necessary steps to provide a multiresolution description for the quay crane model and have it ready for the view concept implementation discussions.

4 PROVIDING A DEMONSTRATIVE EXAMPLE: QUAY CRANE

Automated container terminals are peculiar for their lack of manual labor, replace by robotized equipment that is centrally controlled by a terminal operating system. Automated container terminals can be studied as technical hierarchical systems by different perspectives.

During the design of automated container terminals key performance indicators are used as design requirements. Given performance goals, designers have to optimize their design till the requirements are being fulfilled. To cope with this difficult task however, designers need to be supported throughout the design process. In Fumarola, Seck, and Verbraeck (2010b), we presented how these processes can be supported by going from high abstracted models to more concrete and detailed models. Ideally, these models would therefore be at the designer's disposal during the process. Container terminals are however designed in multi-actor environments, hence requiring different perspectives to serve different actors. One could therefore be interested in emissions (which have an ever-increasing importance in the design of modern technical systems), energy consumption, safety, or finance.

The, ideally automatic, construction of these different perspectives begins with the implementation of a highly detailed model of automated container terminals. We opted for a component based approach to facilitate the construction of various instantiations for varying container terminals. Designers need therefore only to construct the structure in which these components reside, thus following Klir's systems engineering's understanding of design (Klir 1985). The component library is limited to seaside and yard operations, as these are considered the critical points of container terminal design (Saanen 2004). Nevertheless, the design of the library allows easy extensibility which therefore reduces the efforts required in the implementation of new useful components, such as trucks and trains for land transportation. The complete library is presented in the system entity structure in Figure 1.

Again in Figure 1, we see the distinction made between high definition (HD) and low definition (LD) components. This confirms our interest in supporting the design process by providing abstract and concrete components that can be used throughout the design process. The most concrete components result from a physical and functional decomposition. We have a structural component for each piece of equipment, which on its turn is decomposed in its physical components, such as frame, trolley and spreader for the cranes, and its functional components, that are in charge of steering the physical components. To go to higher abstraction levels, therefore means to abstract decompositions at specified levels.

To further clarify how our ideas on providing multi-resolutional modeling environment can be integrated into a large-scale complex system, we give details about our Java implementation and model design processes using the quay crane component inside the DEVS model for automated container terminals.

4.1 Enabling the View Approach using Quay Cranes

4.1.1 Initial High-Resolution Quay Crane Model

In the initial container terminal model (Fumarola, Seck, and Verbraeck 2010a), quay cranes are present to load and unload containers from and to the vessels. The quay cranes are designed to have both behavioral and physical submodels to have a modular approach that makes it flexible enough to change the behavior of the crane. The behavioral submodels of a quay crane are named "controllers" and the physical submodels are the trolley, the moving frame and the spreader. High resolution quay crane model with its behavioral and physical subcomponents and the detailed moving frame model are given in Figure 2.

A controller of a quay crane is responsible for receiving an order and translating it into a set of physical movement commands for the related physical models of the quay crane. In order to satisfy an order, certain physical movements are required to be performed in a certain order by the moving frame, trolley and spreader. In the container terminal model, those translations from an incoming order to a set of physical movements are handled by the sub-controllers which are in fact submodels of the quay crane controller. However, it has to be noted that having a quay crane controller model with three sub-controllers, like the one in our initial quay crane model, is not a design restriction for further uses. The initial quay crane controller model only gives guidelines to the designer. The name of these three sub-controllers, which are responsible for the three similarly named phases, are ‘‘preparing’’, ‘‘positioning’’ and ‘‘transporting’’.

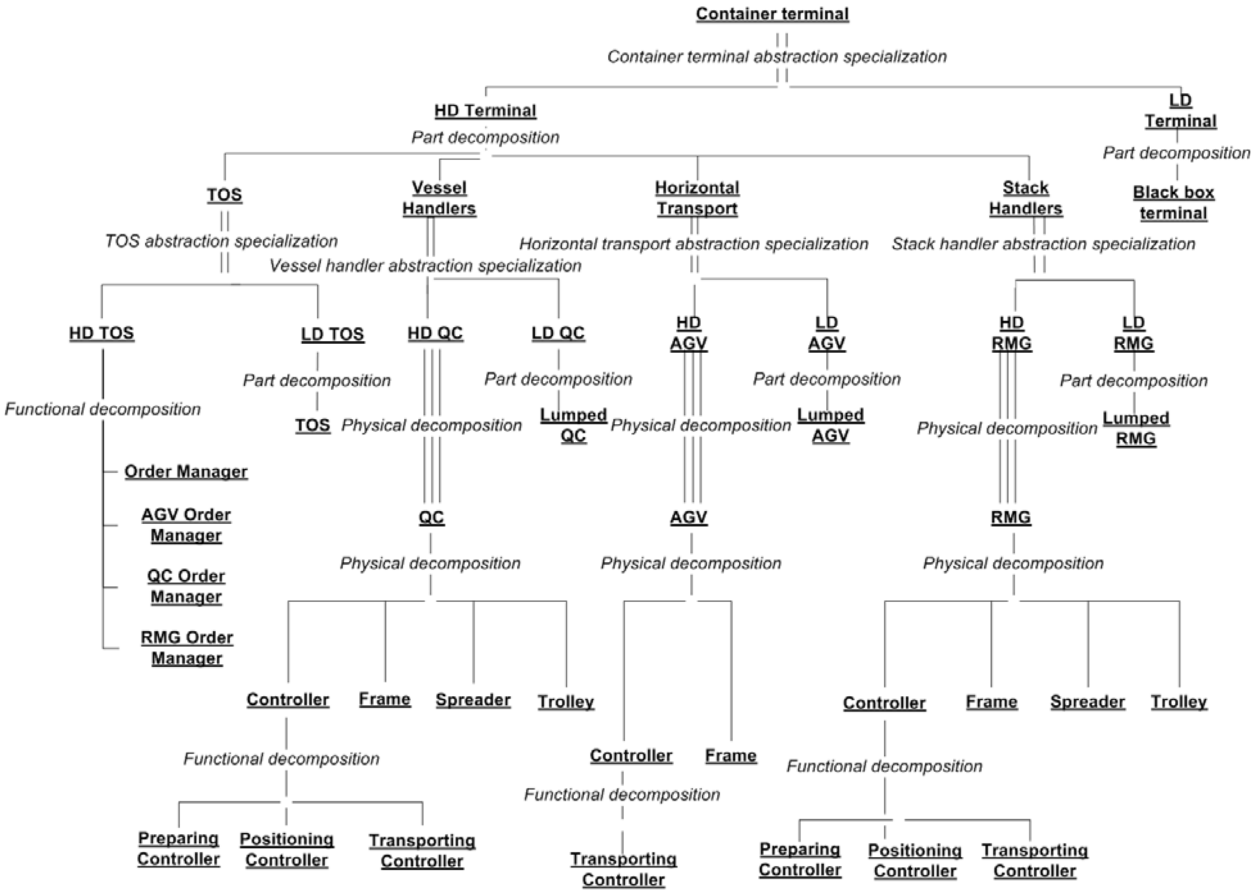

Figure 1: The system entity structure of the automated container terminal component library.

The preparation phase of a quay crane has a set of physical movements of the trolley and the spreader to make the quay crane ready for the positioning phase. The positioning phase contains the combination of movement commands for the moving frame, the trolley and the spreader to set the quay crane to the exact position to be able to pick up a container. Finally, the transportation phase consists of picking up the container and delivering it to its final destination. At the same abstraction level as the controller model of the quay crane, the physical models in the initial design are responsible for providing the detailed calculations (e.g. speed and position) for the mechanical activities (e.g. every kind of movement of the moving frame, trolley and spreader).

However, when we are only interested in a rough outline of the movements of cranes, we don’t necessitate having the complete trace of trolleys and spreaders, but only of the position of the complete crane. Furthermore, in the initial design stages, designers can completely ignore most of the behavior of equipment, and are satisfied with an approximation of the performance rate given a specified number of equipment.

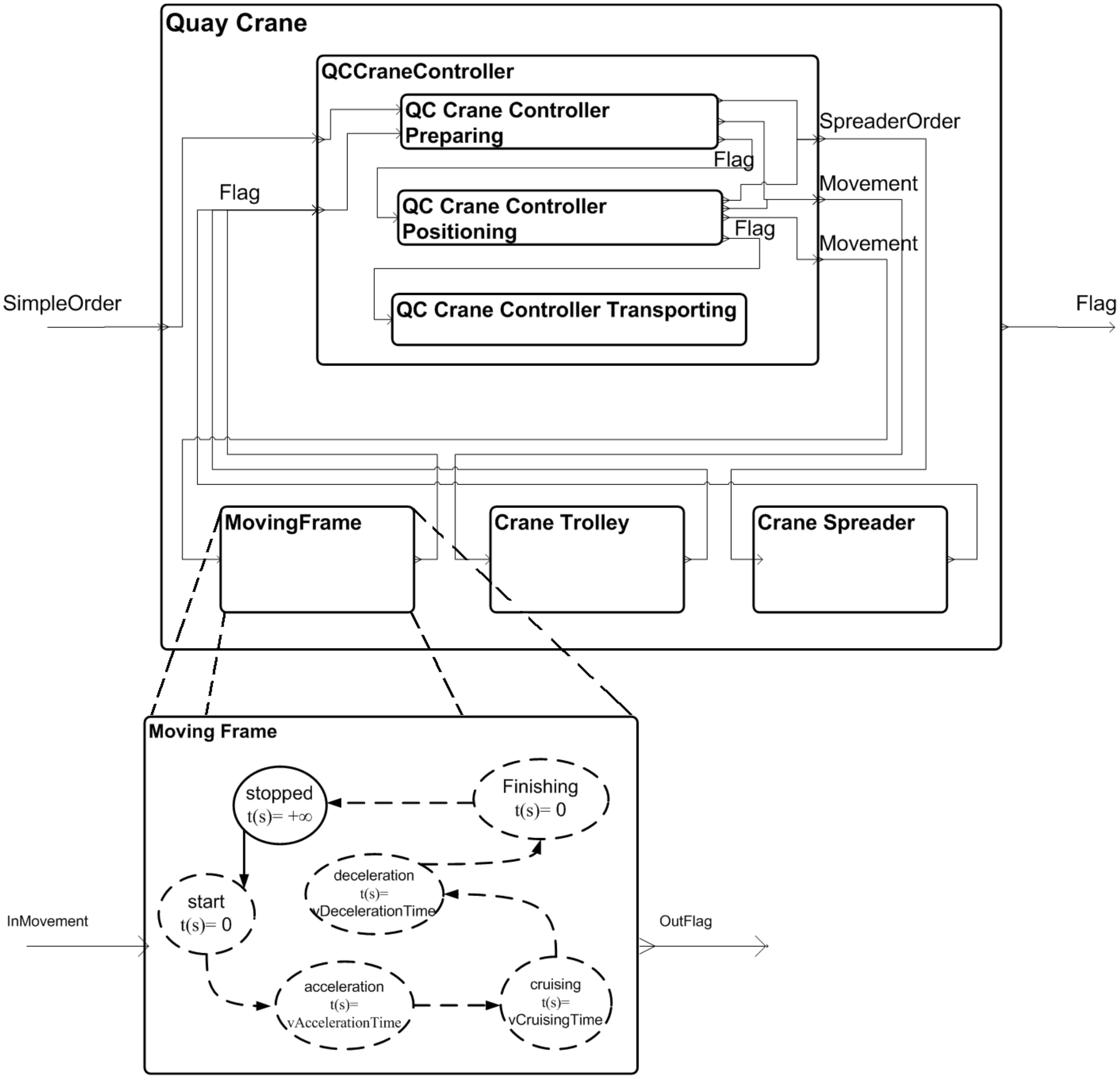

Figure 2 : Model of the high resolution quay crane with the controllers and physical components and a zoomed view of the moving frame model.

4.1.2 Deriving a Low-Resolution Quay Crane Model

The initial quay crane model is a DEVS coupled model containing atomic models like spreader, trolley, moving frame and a coupled model for the quay crane controller. However, because one of our concerns is to provide a multiresolutional model description of quay crane, at this stage, we need to obtain a low-resolution representation of quay crane from the high-resolution model. To do that, we followed a manual model abstraction process. First of all, we assume that the user treats the whole quay crane as one big atomic model. Therefore, he does not have to look at the detailed action of a trolley and spreader whenever he only needs summary information about the quay crane's movement status. The low resolution quay crane model is given in **Figure 3**.

In our model, the abstracted representation of quay crane is designed to receive the same input using the identical input port configuration of the initial model. Additionally, the abstracted quay crane is also designed to produce an accurate approximation of the initial quay crane model's output using the same output port configuration. The reason why we insist on keeping the same input and output port signature is to support modularity. Therefore, whenever the user wants to switch between the DEVS models of quay crane, the selected representation at a selected abstraction level is replaced with the old model. Such approach also preserves the reusability of the container terminal model.

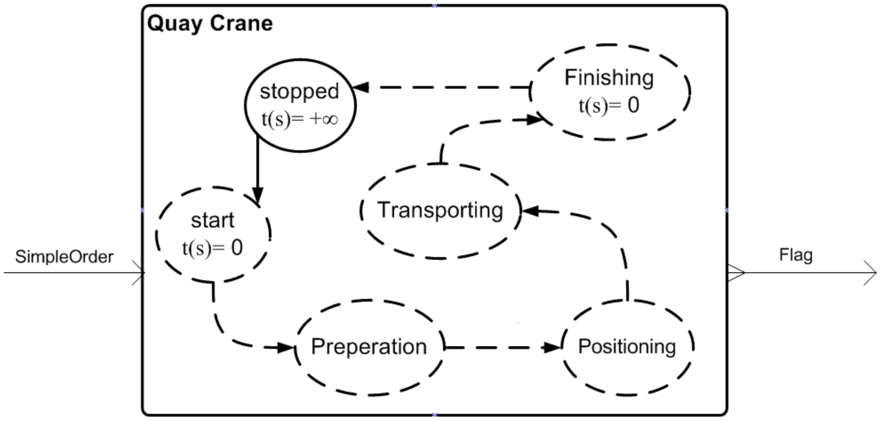

Figure 3: Model of the low resolution quay crane with its phases.

During the initial stages of a design process, designers are only interested in an approximation of the expected productivity instead of detailed simulation runs. Therefore we do not necessarily need the detailed physical components like the trolley, spreader and the moving frame. However, in order to generate an output of movement for the quay crane, we still need to find the approximate time for the whole movement that we can obtain by producing event logs from the initial terminal model. This log file keeps track of the phases for physical and behavioral components of the high level quay crane. Given log file in Table 1 only covers the state traces of preparation and positioning phases. The rows given in bold and italic are representing the phase changes for the quay crane. By tracing the log files and running the model several times, we created a distribution of times for each phase. The abstracted model therefore omit the physical details of the quay crane and regard the quay crane as a single atomic model characterized by a single position coordinate. To calculate the position we only need the controlling phases that influence the positions. We therefore decided to import the three previous phases (preparing, positioning and transporting) into our new abstract quay crane model.

Table 1: Automatically generated log file for the high-resolution quay crane model.

Preparing	Positioning	Transporting	Moving Frame	Trolley	Spreader
init	init	init	stopped	Stopped	stopped
control	***init***	***init***	***stopped***	***Stopped***	***stopped***
init	init	init	stopped	Stopped	Start
init	init	init	stopped	Stopped	running
init	init	init	stopped	Stopped	picking
init	init	init	stopped	Stopped	finishing
control	init	init	stopped	Stopped	stopped
init	init	init	stopped	Stopped	stopped
init	init	init	stopped	Start	stopped

init	init	init	stopped	Running	stopped
init	init	init	stopped	Finishing	stopped
init	init	init	stopped	Stopped	stopped
init	***control***	***init***	***start***	***Stopped***	***stopped***
init	init	init	acceleration	Stopped	stopped
init	init	init	cruising	Stopped	stopped
init	init	init	deceleration	Stopped	stopped
init	init	init	finishing	Stopped	stopped
init	control	init	stopped	Stopped	stopped
init	init	init	stopped	start	stopped
init	init	init	stopped	running	stopped
init	init	init	stopped	finishing	stopped
init	control	init	stopped	stopped	stopped
init	init	init	stopped	stopped	Start
init	init	init	stopped	stopped	running
init	init	init	stopped	stopped	picking
init	init	init	stopped	stopped	finishing
init	init	init	stopped	stopped	stopped

4.1.3 Introducing the View Concept to the Quay Crane

Our initial container terminal model is designed to observe the operational perspective (e.g. delivering/receiving orders etc.) of the real system. However, as we mentioned earlier, it is also possible that a user might be interested in looking at the energy consumption, safety issues, financial issues or the emission rate of the container terminal (or at least a certain part of the system like a single quay crane). Each of these system perspectives constitute the views of a container terminal model. In order to provide a better understanding about the concept, we can adapt the energy consumption example, which we discussed in Section 3, for our container terminal model.

In Section 3, we gave a simple example about a system user who wants to know about the energy consumption of a certain type of vehicle or machine during the simulation run time. Such data can be useful for several kinds of departments inside the organization. As an example, energy consumption data of a single or a group of vehicles can be used by the finance department when making a financial analysis about the energy costs of those vehicles.

To make our example more solid, let's assume that the vehicle is a quay crane. However, we know that neither our initial high-resolution DEVS model of quay crane, nor the low-resolution DEVS model can helps us to observe the system perspectives outside the operational one. For that reason, no predictions are possible about the energy consumption of a single or a certain group of quay cranes inside the terminal without building additional models or transforming the output data.

Although we know that a ''view'' of the system is a multiresolution model by itself, one should never forget that the view concept uses the multiresolution modeling approach as a tool to represent contextual phenomena inside the system. Therefore, one of the main difference between a view and a multiresolution model lies in the fact that a view provides extra contextual information to describe a system perspective (e.g. high - low energy consumption, high - low risk etc.). For further explanation, let's consider the phases of the moving frame that we presented in the detailed model in Figure 2 . If we examine the phases based on the energy consumption criteria, we can mark the ''acceleration'' phase as **high energy consumption.** Similar to acceleration phase, we can set the ''cruising'' phase as **normal energy consumption** and the ''deceleration'' phase as **low energy consumption**.

One of the important things to remember is that the energy consumption for these phases are completely context dependent. Different complex systems with almost identical phases can have an entirely different energy consumptions. We can add these tuples (e.g. acceleration phase – energy – high consumption, cruising phase – pollution – high gas emission etc.) inside the DEVS model definition of the moving frame, or we can create a totally new model inside our quay crane model description in the initial phase. In the implementation phase, using these lists of phase-perspective matches, whenever a user has a query about an perspective of the system, we can dynamically reconstruct the relevant perspective.

On the other hand, providing such view-specific information means that the designer must have prior knowledge about every possible perspectives of the system. In multiresolution modeling, having such a context-based approach produces better decomposition trees as mentioned in Davis and Tolk (2007). However, this modeling approach requires expertise on the relevant subject and it doesn't sound like a feasible way to find out every possible perspectives before the model design. Therefore, automatic system reconstruction would be desirable. Supervised learning and data mining are promising avenues to explore.

5 CONCLUSIONS AND FUTURE WORKS

In this paper we presented the multiresolution representation and the Distributed Simulation Object Library (DSOL) implementation of quay crane equipment inside an automated container terminal model to discuss the challenges of designing a multiresolution modeling and simulation framework. We introduced the ''view'' concept to support multi-perspective modeling and simulation environment. We discussed the pros and cons of our conceptualization.

As a future work, we will explore a new theory to combine the existing studies about the multiresolution and multi-perspective modeling with our own ideas to form a consistent modeling framework. We will try to find a more feasible way to reconstruct the systems instead of hand-driven methods. Therefore, methodologies like supervised learning and data mining will be considered as possible approaches to identify views in a more automatic fashion. Once a consistent and satisfying design has been identified, the validation and refinements of our model can be done to solidify our research.

REFERENCES

Bigelow, J. H., and P. K. Davis. 2003. *Implications for model validation of multi-resolution, multi-perspective modeling (MRMPM) and exploratory analysis*. RAND.

Davis, P. K., and A. Tolk. 2007. Observations on new developments in composability and multi-resolution modeling. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 859-870. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Davis, P. K., and R. Hillestad. 1993. Families of models that cross levels of resolution: issues for design, calibration and management. In *Proceedings of the Winter Simulation Conference*, ed. G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles, 1003-1013. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.

Davis, P. K., and R. K. Huber. 1992. *Variable-resolution combat modeling: motivations, issues, and principles.* RAND Notes.

Davis, P. K., and J. H. Bigelow. 1998. *Experiments in MRM.* RAND MR-100-DARPA.

Fishwick, P.A. 1986. *Hierarchical Reasoning: simulating complex processes over multiple levels of abstraction.* Ph.D. dissertation, published by University of Pennsylvania.

Fishwick, P.A., and Zeigler, B.P. 1992. A multimodal methodology for qualitative model engineering. *ACM Trans. Modeling Comput. Simulation*, 2(1):52-81.

Fumarola, M., Seck, M., and Verbraeck, A. (2010a). A DEVS component library for simulation-based design of automated container terminals. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering.

Fumarola, M., Seck, M., and Verbraeck, A. (2010b). Supporting The Design Of Automated Container Terminals With The Multiple Worlds Concept. In *Proceedings of SCS 2010 Summer Computer Simulation Conference.* The Society for Computer Simulation International.

Klir, G. 1985. *Architecture of Systems Complexity.* Sauders, New York, USA.

Lane, D. 2006. *Hierarchy, Complexity, Society*, in Pumain, D. ed, *Hierarchy in Natural and Social Sciences*. New York, New York: Springer-Verlag. 81–120.

National Research Council. 2006. *Defense modeling, simulation, and analysis: meeting the challenge.* Committee on Modeling and Simulation for Defense Transformation. National Academies Press.

Pegden, C.D., R.E. Shannon, and R.P. Sadowski. 1995. *Introduction to Simulation Using Siman.* 2nd ed. The McGraw-Hill Companies, Inc., London.

Reynolds Jr., P. F., A. Natrajan, and S. Srinivasan. 1997. Consistency maintenance in multi-resolution simulations. *ACM Transactions on Modeling and Computer Simulation* 7(3): 368–392.

Roberts, N., D. F. Andersen, R. M. Deal, M. S. Garet, and W. A. Shaffer. 1983. *An Introduction to Computer Simulation: A System Dynamics Modeling Approach*, Reading MA, Addison Wesley.

Saanen, Y. 2004. *An approach for robotized marine container terminals*. PhD thesis, Delft University of Technology.

Schmidt, J. W., and R. E. Taylor. 1970. *Simulation and Analysis of Industrial Systems.* Richard D. Irwin, Inc., Homewood, Illinois.

Simon, H. A. 1962. The Architecture of Complexity. In *Proceedings of the American Philosophical Society*, 467-482.

Simon, H. A. 1977. *The New Science of Management Decision.* Prentice Hall PTR, Upper Saddle River, NJ, USA.

Takahara, Y., J. Iijima, and Q. Zhao. 1988. A Hierarchical View of Complex Systems. International Journal of General Systems. 14:3, 201-237.

Van Gigch, J.P. 1991. *System Design Modeling and Metamodeling.* Plenum Press, New York.

Yılmaz, L. and Ören, T.I. 2004. Dynamic model updating in simulation with multimodels: A taxonomy and a generic agent-based architecture. In *Proceedings of SCSC 2004 - Summer Computer Simulation Conference,* July 25-29, San Jose, CA., pp. 3-8.

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.

Zeigler, B.P. and T.I. Ören. 1986. Multifaceted, Multiparadigm Modelling Perspectives: Tools for the 90's. In: *Proceedings of 1986 Winter Simulation Conference*, ed. J. Wilson, J. Henriksen, and S. Roberts, 708-712. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers, Inc.

AUTHOR BIOGRAPHIES

ÇAĞRI TEKİNAY is a Ph.D. researcher in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. He received his B.Sc. in computer engineering from Baskent University and his M.Sc. in Information Systems from Middle East Technical University in Turkey. His current research interests are simulation and serious gaming. His email address is <c.tekinay@tudelft.nl> and his website is <http://www.tudelft.nl/ctekinay>.

MAMADOU SECK is an assistant professor in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. He received his Ph.D. degree and his M.Sc. from the Paul Cézanne University of Marseille, and his M.Eng. from Ecole Polytechnique Universitaire de Marseille, France. His research interests include modeling and simulation formalisms, dynamic data driven simulation, human behavior representation and social simulation, and agent directed simulation. His e-mail address is <m.seck@tudelft.nl> and his website is <http://www.tudelft.nl/mseck>.

MICHELE FUMAROLA is a Ph.D. researcher in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. He received his M.Sc. in computer science from Hasselt University in Belgium. His current research interests are simulation and serious gaming. His email address is `<m.fumarola@tudelft.nl>` and his website is `<http://www.tudelft.nl/mfumarola>`.

ALEXANDER VERBRAECK is a full professor in Systems and Simulation in 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 open and generic libraries of object oriented simulation building blocks in Java. His e-mail address is `<a.verbraeck@tudelft.nl>` and his website is `<http://www.tudelft.nl/averbraeck>`.