

A SIMULATION BASED DESIGN FRAMEWORK FOR LARGE SCALE INFRASTRUCTURE SYSTEMS DESIGN

Yilin Huang^(a), Mamadou D. Seck^(b), Michele Fumarola^(c)

^(a,b) Delft University of Technology
Faculty of Technology, Policy and Management
Systems Engineering Group
^(c) Macomi Systems Simulation

^(a) y.huang@tudelft.nl, ^(b) m.d.seck@tudelft.nl, ^(c) m.fumarola@macomi.nl

ABSTRACT

This paper discusses an enhanced use of Modeling and Simulation (M&S) in the design cycle of large scale infrastructure systems. After a short review of the use of M&S in the current systems design process, we identify some issues in multi-actor design environment. A Simulation Based Design (SBD) framework is proposed to tackle these issues using a multi-methodological approach. The framework is specified with formal description, and the tree-like structuring and refinement of design alternatives are explained. A case of applying the framework to container terminal design is presented and the evaluation is reported.

Keywords: simulation based design (SBD), systems engineering (SE)

1. INTRODUCTION

Infrastructure design is often complex and has critical implications at strategic, tactical and operational levels for organizations in public and private sectors. Design faults are difficult to foresee and can entail high costs when they emerge later at development and operational phases. A lot of modern infrastructure design processes have many actors involved. We can identify different actors, e.g., as problem owners, policy makers and clients. They may have diverging or even conflicting interests and opinions. For them, the decision making process is actually also a learning process wherein they are engaged in identifying shared problems and objectives, and understanding the others' positions and views.

In engineering, designing systems is a part of the field of Systems Engineering (SE). During the past decades, theories and methods in SE have brought many beneficial changes in how engineered systems are designed and developed. When we take a broad look at the complete engineering process, we can often identify a set of phases that follow an iterative and incremental path. The problem is first structured; then the solutions are formulated based on the selected criteria; and finally the best alternative is selected (Simon 1996). Along the iterative process, the problem definitions and system

requirements supposedly become more lucid and complete, contributing to solution finding.

In order to render SE processes more effective and efficient, we need to apply design and engineering methods with supporting tools. Following (Shannon 1975) and (Sol 1982), simulation can be seen as a method of inquiry that provides a possibility to study systems. It produces quantitative measures that allow users to study the dynamic behavior of the systems and make informed decisions of the design. Simulation Based Design (SBD) is promising in terms of providing designers and decision makers with insights into the engineered system at an early stage in the product lifecycle. This in turn increases the chance to correct design faults before development and operation.

Although the SBD concept holds a lot of potential, the realization in practice is hindered by the fact that Modeling and Simulation (M&S) is highly specialized and time-intensive. Given the current available M&S methodologies and tools, it is difficult to fully integrate M&S into the design process of complex systems. The reason is straight forward: modeling design alternatives of complex systems often takes long time and model building is expensive. Therefore, SBD methodologies and accompanying tools are required and should be developed if we are going to give SBD (currently merely a concept in theory) practical implications that could benefit designers and decision makers.

SBD methodologies and tools should allow designers, who often are not M&S experts, to simulate and evaluate their design without the concern of simulation model building. A SBD framework should offer easy-to-use tools with functionalities such as automatic model generation and calibration, version management, advanced data visualization, analysis and comparison.

In this paper, we present a SBD framework for large scale infrastructure design. We first give a short review of the use of M&S in current systems design processes, and identify some issues that often occur in multi-actor design environments. In Section 3, four constructs that are used in the design of the proposed framework are discussed. Section 4 specifies the

proposed SBD framework and explains the design process. A case of applying the framework to container terminal design is presented in Section 5 and the evaluation was conducted both with novice users (graduate students) and experts.

2. NEED FOR SIMULATION BASED DESIGN

2.1. Simulation in Traditional Design Processes

The traditional role of simulation models in the design process is in the analysis step of systems definition: evaluate design alternatives obtained in the conceptual design phase. Modeling often needs the following steps (Law 2007). First, the conceptual designs along with the system requirements and the definition of Key Performance Indicators (KPI) are provided to the M&S specialists. In the succeeding steps, the corresponding conceptual models are built and translated into M&S specifications. After model verification and validation with the domain experts, experiments are designed and performed to evaluate the design alternatives. This typical use of M&S has some weaknesses as stated below (Fumarola 2011).

Because of the time and high cost entailed by modeling, the number of design alternatives actually assessed through simulation is not high. M&S tasks are often conducted by professionals outside of the design team. They need some time to acquire proficiency in the problem domain in order to develop reasonable models. In many cases, when the simulation models are domain, organization or case specific, they need to be developed from scratch; hence these models have poor reusability, especially outside of an organization.

The acquisition of domain knowledge may be alleviated by choosing M&S consultants specialized in the relevant field. However, the transfer of concepts from designers to modelers is not without risk because they do not necessarily share the same mental model of the problem situation, the current state of the system and the envisioned solution.

Outsourced M&S consultants generally only deliver specific answers to the initial questions that are requested by the problem owners. The models, if they are a part of the deliverables at all, are often unable to answer new questions that possibly emerge at a later stage in the design process. Extending the model needs professional modeling expertise so that the designers have to commission the task again.

Although the compartmentalization of conceptual design and detailed design phases helps structure and reduce design complexity, it also forces designers to move back and forth the two phases. The designers often do so in an unstructured way. Blending these two phases may save design and modeling resources. But in doing so, supporting tools are required to reduce the complexity.

(Saanen 2003) advocates for the use of simulation throughout the design process. However, the current practice suggests that little has been changed due to applicatory difficulties. Some are mentioned above.

2.2. Multi-Methodological Approach

M&S has been considered as a methodology within the hard systems approach, a traditional school in systems thinking. Many studies show that the hard systems approach has some limitations, for its basic assumption of the existence of an optimal or quasi-optimal solution which can be solved by algorithmic methods without much consideration for the human actors involved in the system (Ackoff 1979, Simon 1996). Some attempts hence ensued to mitigate the perceived limitations of the hard systems design processes. This resulted in a tendency to pay more attention to the diverging views of different actors involved in the process. This school of thought is referred to as soft systems methodology (Checkland 1999). (Robinson 2001) advocates for a “hard” to “soft” continuum instead of bipolar extremes. To benefit from both, multi-methodological approaches have been introduced.

Many survey results reported a tendency towards combining hard and soft systems methods, e.g., in (Munro and Minger 2002). Multi-methodology has been used in problematic situations to support the model building phase of simulation studies (Mingers and Rosenhead 2004). Qualitative methods and techniques are suggested to be embedded into multi-actor design processes that leave more room for negotiation and mutual learning. They have the potential to support design processes (1) in a multi-actor environment with diverging stakes and (2) using simulation technology to foster discussion based on explicit knowledge.

2.3. Requirements for Simulation Based Design

The multi-methodological view (Robinson 2001, den Hengst et al. 2007) could benefit the SBD process in which designers and stakeholders can learn the others' perspectives by participating in the steps of the process. This feature is not yet present in the current design or modeling frameworks, e.g., (Shannon 1975, Zeigler et al. 2000). Some issues identified in the current systems design processes include (but may not be limited to) the following (Fumarola 2011):

1. **Unstructured Design Process:** without a well-defined structure, the design process may not be able to support designers' abstract and divergent thinking.
2. **Design Management:** without the ability to manage different designs in the design process, designers may not be able to compare them.
3. **Documentation:** without the ability to document assumptions, decisions and argumentations, actors may not be able to follow and audit the design process.
4. **Compare Designs:** without the ability to compare different designs in the design process, actors may not be able to identify the preference.
5. **Consistency:** Actors may have difficulties in keeping the conceptual designs and the detailed designs mutually consistent.

6. **Involvement:** not every actor is involved in both the conceptual design process and the detailed design process.
7. **Prediction:** without simulation models, actors may not be able to predict the design decisions.
8. **Understandability:** the design is often presented in a way that not all actors can understand.
9. **Process Flow:** The system design process is often neither interactive nor iterative.

These issues are identified as relevant in many systems design projects. They reflect the requirements for the design of a SBD framework and tools. A design process that involves multiple actors requires a SBD framework that facilitates collaboration to enhance shared understanding. The framework should support automatic model generation so that a design can be transformed into executable simulation models without advanced modeling knowledge. The models should be available at different levels of completion and abstraction reflecting the nature of the design process. The model experiments should be manageable by the designers, and the simulation outputs generated should be presented in a way that they allow non-simulation specialists to compare different alternatives. And the different design alternatives (and models) need version management so that the designers can trace different versions and their logical relations systemically.

3. A SYSTEMS ENGINEERING APPROACH FOR SIMULATION BASED DESIGN

SE has evolved from being underpinned by a rather mathematical and hard systems approach of design towards an approach of transforming users' needs into an operational system via inter-disciplinary efforts. SBD influences systems development models, e.g., Royce's Waterfall model, Boehm's Spiral model and Forsberg and Moog's V-model. The former two models have many applications in software engineering, while the V-model remains dominant in SE. In current systems design process, some distinct phases can be recognized, e.g., requirements analysis, conceptual design and detailed design.

We posit that SBD could cover the three phases by blending them into a participative design phase. Using SBD, the design process on its own is both interactive and iterative such that all actors can be involved in discussions to exchange their opinions, and they should be able to explore the system requirements and the designs using the supporting tool. The conceptual design and detailed design phases are currently often separated because the details needed in the latter are considered as a burden in the former. However, in a SBD process, the model building blocks should be able to encapsulate the complexity in detailed designs and expose only the relevant information that is necessary to the conceptual design.

Based on these concepts and the issues addressed in Section 2, we propose four constructs that can be used for the design of a SBD framework. (We use the term

construct to refer to the related methods or models that can be used to tackle a set of the issues.) Each construct tackles some issues addressed.

1. Component-based Modeling (issues 2, 4, 7, 9)
2. Different Levels of Specification (issues 1, 5, 6)
3. Structure Alternatives (issues 1, 2, 4, 9)
4. Participatory Design (issues 3, 6, 8, 9)

A SBD framework requires domain specific tool support. An extensible model component library is needed to allow the modeling of design alternatives. Pre-defined simulation components let users build new simulation models by focusing on the composite structure without the concern of the inner details of the components. Components support model modularity by using port-based communication (Paredis et al. 2001). Once this approach is adopted, model reusability is also enhanced. Therefore, we propose a component-based architecture as shown in Figure 1.

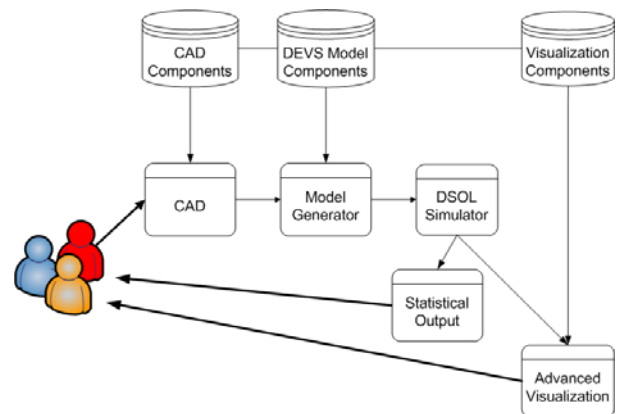


Figure 1: A Component-based Architecture.

Three libraries are in place: the design components (CAD), the simulation model components and the visualization components. The CAD components are used by designers in a design environment. The design is passed on to the Model Generator which transforms it automatically into a simulation model (Fumarola et al. 2010, Huang et al. 2011). This step should be transparent to the designers. The model will be executed and visualized. The results form an instant feedback loop to the designers and the other actors involved in the design process.

A typical design process starts with an incomplete picture of the design problem. Therefore, it is infeasible to start directly with a detailed solution; instead, the design is refined in a couple of steps to deal with uncertainty, complexity, and the cognitive limitations of designers (Hoover and Rinderle 1991, Goel 1995, Ullman et al. 1998). Correspondingly, along the design process, simulation models should be constructed to fit the pre-defined abstraction levels. This allows designers analyze simulation results throughout the process, even in the initial phases characterized by rough and incomplete designs. Further down the process, abstract models are replaced by more concrete

implementations until the end where detailed models are used to gather precise simulation results.

Creating alternatives is one of the fundamental steps in design methods. In a multi-actor design environment, the amount of alternatives in the solution space could become very high. They are often reduced to a possible minimum at an early stage of the design process due to the high cost of maintaining and comparing the alternatives. This unfortunately induced designers to systematically explore the solution space using trees, matrices, rankings and charts. Structured experimentation (Dwarakanatha and Wallacea 1995) can bring more insight into the way designers think. With it, designers can follow the paths in the tree-like structure to assess alternatives. Whenever a solution seems undesirable, the decision branch can be removed. Structured experimentation guides designers towards an iterative process to specify alternatives, evaluate them (formal or informal) and make selections.

A system design process is seldom performed by a single actor or even a single person. In a multi-actor design process, each actor tries to achieve his/her own goals. Hence, the design process should in general support a certain convergence of interests of all actors. The conceptual designs are the outcomes of negotiation and agreement between the actors. The major decisions (in the conceptual design) are made based on informal assessments because the simulation studies are often performed at the detailed design process. By bringing formal methods closer to the conceptual design process, the reasoning for choices can be more grounded and shared understanding can be enhanced. This leads to an approach that is “soft with a hard centre” (Robinson 2001). Moreover, explicit documentation ensures the consistent provision of rationale behind the decisions and provides insight into the design making process (Girod et al. 2003) in a multi-actor decision making environment.

4. THE PROPOSED FRAMEWORK

Based on the four constructs discussed in the previous section, this section proposes a SBD framework. The framework is specified with a formal description that assumes the use of the DEVS formalism (Zeigler et al. 2000) to model the system. The SBD framework

$$\mathcal{F} = \langle P, A, \mathcal{L}, \zeta, \eta, Z, \rho \rangle$$

where

P is the set of triples (N, L, t) where

N is a coupled model, as defined in (Zeigler et al. 2000)

L is the set of components of N , $L \in \mathcal{L}$

t is the creation time of N , $t \in \mathbb{R}^+$

A is a tree defined as $(P, <_{t,z,L})$ where

$<_{t,z,L}$ is the ordering relation of the creation time t , the applied decisions z , and the set of components L ,

$z \in Z$ and $L \in \mathcal{L}$

$\mathcal{L} = \{L_0, L_1, \dots, L_k\}$ is the set of component sets where

$\forall L \in \mathcal{L}$ and $\forall c \in L$, c is a coupled model

$\zeta: L_i \rightarrow L_j, L_i, L_j \in \mathcal{L}$ is the component transformation

function where $\zeta(c_{Li}) = c_{Lj}$, $c_{Li} \in L_i$, $c_{Lj} \in L_j$

$\eta: S \times R \rightarrow S$ is the initialization selection function that selects the set of states such that $\eta(s, r) = s_{initial}$ with $s, s_{initial} \in S$, $r \in R$ and R is the set of sequences for which each sequence selects a specific state

$Z = \{A_{components,r}, R_{components,r}, A_{coupling}, R_{coupling}\}$ is the set of decisions where

$A_{components,r}$ adds the component $\eta(s, r)$

$R_{components,r}$ removes a component

$A_{coupling}$ adds a coupling

$R_{coupling}$ removes a coupling

Z is the set of all k -tuples (a_1, a_2, \dots, a_k) with $a_1, a_2, \dots, a_k \in Z$

$\rho: P \times Z \rightarrow P$ is the branching function

$\rho(p_i, b_j) = p_h$ where

$(p_i, p_h \in P) \wedge (p_i <_{t,z,L} p_h), b \in Z$

The structure \mathcal{F} consists of eight elements: system models P , a tree A , a set of sets of components \mathcal{L} , a component transformation function ζ , an initialization function η , a set of decisions Z , a set of all sequences of decisions Z , and a branching function ρ . To follow the design process, we need to have the ability to branch a model into a couple of new models that contain the possibilities one wants to assess using a predefined set of components.

A set of sets of components \mathcal{L} is provided during the design process. Each node, from the set of possible nodes P , has a model constructed using components from a set of components. A component transformation function ζ transforms a model into another, each of which contains a set of components.

The branching function ρ constructs a new model from a (existing) model and a sequence of decisions in Z . The set of decisions applies to the model structure. Structural changes need four operations: add or remove components, and add or remove couplings. Adding components needs the initialization function η that sets the initial state of the model from the set of states S .

The models generate by the branching function are organized in a tree structure A . The root represents the initial model. Each node contains an alternative design (or model) which is derived from its parent node by applying some change. The leaves of the tree contain models with no further change. The simulation of each model (in the tree) generates outputs, i.e., the KPIs, of a design alternative. The range of each KPI is related to the model's position in the tree; namely, a less detailed model has outputs with a broader range, and a more detailed model has outputs with a narrower range.

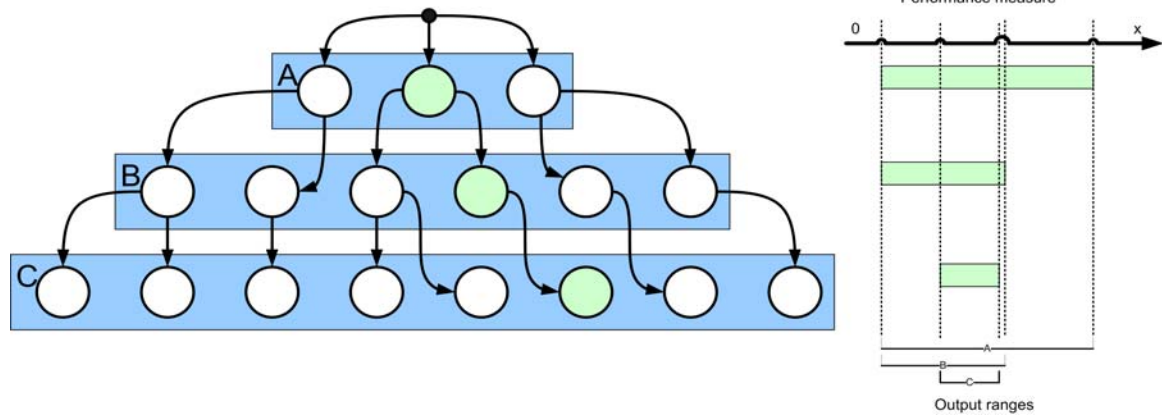


Figure 2: Output Ranges of the Model in the Tree Structure (Fumarola 2011)

The designs (or models) closer to the root of the tree contain more uncertainties designers have during the early design phases. Further down the tree, more details are added so that they reduce the solution space. This relation is exemplified in Figure 2 which has designs of three levels.

The tree guides actors or designers through the design process. At each level of the tree, an appropriate class of decisions needs to be made to refine the current design. Each decision leads the designers to a lower level node that contains a refined design that is derived from a higher level node. This helps the comparison of the designs at every level. The designers can steer the design process by choosing decisions that can reduce the KPI range of their interest. The design process will in turn gradually refine the design and narrow the outputs to a desired combination of KPI ranges.

5. APPLYING THE FRAMEWORK TO CONTAINER TERMINAL DESIGN

The SDB framework is applied to studies in automated container terminal design (Fumarola et al. 2010, Fumarola 2011). In the studies, simulation component libraries are integrated with a CAD design environment so that simulation models can be directly generated and experimented from the design alternatives. Using the tree structure to organize and guide the design process showed promising results.

5.1. Simulation Components

The model library contains a collection of DEVS components that represent a large variety of material handling equipments, such as quay cranes, automated guided vehicles and rail mounted gantries that are commonly used in automated container terminals. The components can be divided into two groups: the high level components and the low level components. They specify the equipments at high or low abstraction level. Relations are defined between the high level and the low level models. A dynamic build-up mechanism can add and remove sub-models to or from a model, and establish and remove couplings correspondingly. The simulation outputs are collected for visualizations and statistics as design evaluation criteria.

5.2. The Design Process

The design process, as shown in Figure 3, starts with the design requirements specification and the collection of documentations about the equipment, terminal plot, demand forecasts, etc. These documents define the amount of investment, the physical constraints and the required throughput. With the information, designers can identify a list of evaluation criteria. It is needed at a later design stage to compare alternative designs using multi-criteria decision analysis.

Once the criteria list is prepared, the first high level design can be constructed. At this level, the models are relative simple so the simulation run time is in general short. Designers can create a high number of designs with relative low experimentation cost. Using the results of the high level models, the designs can be refined by using the low level models, allowing for more detailed analysis. The low level models can be used (in the container terminal design studies) to assess the terminal layout and the choice of equipments. The alternatives can differ in cost, performance and environmental footprint. The position, number and type of the equipments are decided at separate design levels. This allows the designers to concentrate on one decision factor at a time.

The tree structure used in the framework offers users the possibility to return to the higher design levels, alter their decisions, experiment with them, and discover the interactions between them. At each level, designers can evaluate the designs using multi-criteria decision analysis. When they had assessed different design alternatives based on their individual and collective preferences, they can focus on a set of final designs. That may conclude the design process.

5.3. Evaluation with Students and Experts

5.3.1. Usability Test with Students

The evaluation was first conducted with nine graduate students (at Delft University of Technology) for the usability tests of the tools of the framework. The students were divided into three groups, and each group participated in a design workshop where they were given a design task.

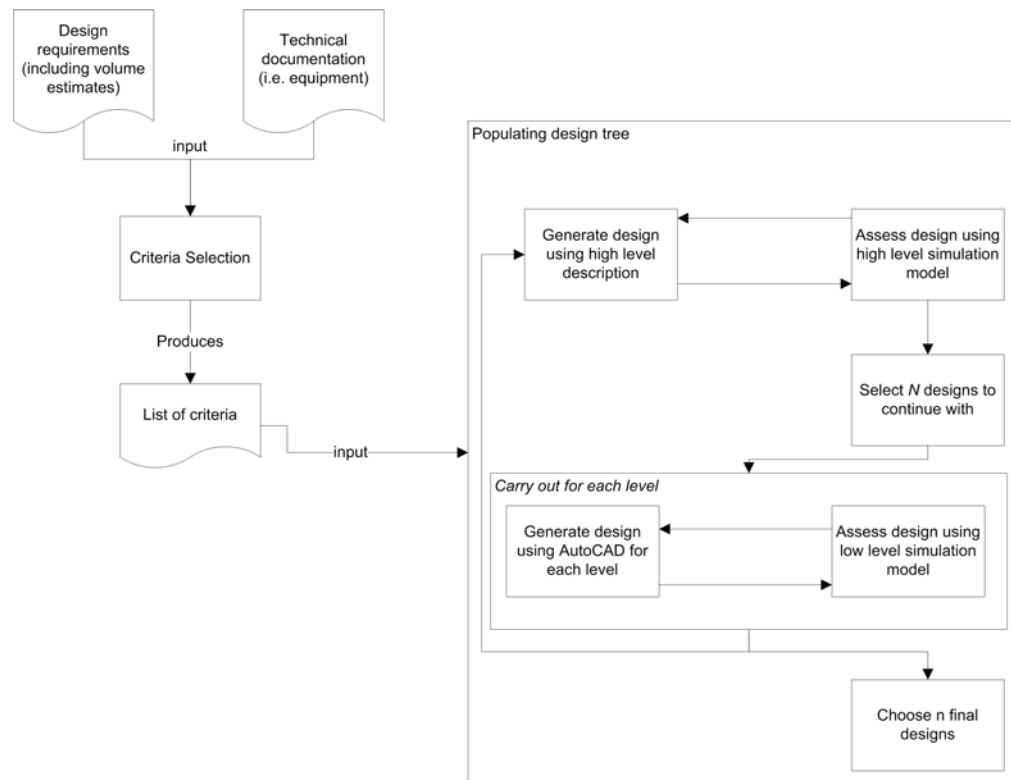


Figure 3: The Design process Using the SBD Framework (Fumarola 2011)

Each student was assigned a different role with specific design goals. Semi-structured group interviews were given to the participants after each workshop.

The students indicated that the tool guided them in their design effort and helped them in structuring design decisions. They gave a higher score to the support by statistics than by animation. As novice users, they had problems in using the AutoCAD design environment. Because conventions are strictly required for the CAD designs in order to transform them into simulation models, the errors had to be corrected first to further proceed with the design process. To solve this problem, formal CAD model checking mechanisms have to be added into the simulation environment. The students also suggested that a more extensive set of statistics could be added to provide more detailed views on the performances of the designs.

The students' design processes were given to an expert (in automated container terminals) to ask for opinions. The expert noticed that the students' design process complied with good design process practices in which the first level designs were extensive, and from those one or two designs were chosen to be continued. The expert further commented that in current practices, unfortunately, limited by resources and costs, only a few design alternatives (typically two) were compared. The comparison of a larger number of high level designs is appreciated by the expert as it allows the exploration for non-trivial designs.

5.3.2. Effectiveness Test with Experts

Semi-structured interviews were opted for the expert evaluation. Eight domain and methodological experts

participated in the interviews. The questions focused on several aspects: the interviewee's level of expertise; the perceived problems they have encountered in their organization; the proposed framework, and possible adoption of the framework in their current organization.

The experts stated that simulation models are frequently used to support decision making in logistics management. However, in the design process, because designers do not always understand the model (partially or fully), they are reluctant to use them. When the designers are presented with models that do not conform to their expectations, they will disregard the models and continue the design process without simulation. Several interviewees pointed out that the introduction of SBD (that uses simulation models) *would* therefore meet some resistance. The resistance is due to "fear of the unknown" instead of any intrinsic quality of a design method.

On the other hand, they also pointed out that simulation models *should* be used earlier and oftener in the design process. An earlier feedback loop is mostly missing in current design approaches. And modern logistics systems are often too complex to comprehend without the support of simulation models. The proposed framework has the potential to solve these problems.

They acknowledged that the SDB framework, particularly the tree structure in organizing different designs, can help to structure the design process and to alter among different design abstractions. However, the actors in the design process should be fully aware of the type of questions that can be answered in every level of the process.

Issues	Usability Workshop	Expert Evaluation
1 <i>Need a structured design process</i>	The approach helped participants follow a structured design process.	The approach is perceived useful in collaboration and comparing results in a structure way.
2 <i>Manage different system designs</i>	Participants were capable of comparing different designs.	Quantitative comparison of alternatives helps collaboration between different actors.
3 <i>Document the process</i>	(not evaluated)	Storing a design process can be very useful for future reference.
4 <i>Compare different designs</i>	Participants used the comparison to make design decisions.	Quantitative comparison of alternatives helps collaboration between different actors.
5 <i>Mutually consistent conceptual and detailed designs</i>	(not evaluated)	Attention should be given to the choice between analytical models (high level) and simulation models (detailed design).
6 <i>Involvement of actors throughout the process</i>	Each participant was involved in the design process.	Collaboration is required throughout a design process.
7 <i>Predict the influence of design decisions</i>	The results from the simulation models were used to make comparisons.	KPIs help streamline the discussion between several actors.
8 <i>Understandable presentation</i>	The 3D and statistical output were used to compare designs.	(not evaluated)
9 <i>Interactive and iterative process</i>	Participants were able to design collaboratively.	Iteration is required in a design process.

Table 1: Evaluation of the Method with Issues Identified in Section 2.3

They acknowledged that the SDB framework, particularly the tree structure in organizing different designs, can help to structure the design process and to alter among different design abstractions. However, the actors in the design process should be fully aware of the type of questions that can be answered in every level of the process.

Using simulation to guide the design process also has benefits from a collaboration perspective. The proposed framework could support a collaborative design process. One domain expert phrased this as such: when you start to consider various dimensions of a design, finding a common communication mechanism among the actors is very important; this framework could achieve this goal. Many projects are not about finding the optimal solution but to let the actors understand the problem and make decisions that are good enough to suffice the agreed-upon requirements. A documented and structured design process can serve as a knowledge management system. The proposed framework has the potential to be used for large-scale design project in other domain, particularly engineering projects. To provide an overview, a brief summary of the evaluation is listed in Table 1.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a SBD framework. In a multi-actor design environment, pure hard systems approaches showed limitations in involving different actors. Some common issues were discussed. The proposed framework is composed of four constructs that address the identified issues. It guides users to follow a tree-like design refinement that specifies and evaluates the designs in more detail at each level. The evaluation showed that the framework and the

supporting tools suit well in a multi-actor environment, and they allow interactive and structured exploration of the design solution space.

Further research can be conducted to post-process the simulation results to meet the needs of individual actors from multiple perspectives, and to extend the simulation component library accordingly.

REFERENCES

- Ackoff, R. L. (1979). The future of operational research is past. *The Journal of the Operational Research Society* 30, 93–104.
- Checkland, P. (1999). *Systems Thinking, Systems Practice: Includes a 30-Year Retrospective*. John Wiley & Sons.
- den Hengst, M., G.-J. de Vreede, and R. Maghnooui (2007). Using soft OR principles for collaborative simulation: a case study in the Dutch airline industry. *Journal of the Operational Research Society* 58, 669–682.
- Dwarakanatha, S. and K. M. Wallacea (1995). Decision-making in engineering design: observations from design experiments. *Journal of Engineering Design* 6(3), 191–206.
- Fumarola, M. (2011). *Multiple Worlds: A multi-actor simulation-based design method for logistics systems*. Ph. D. thesis, Delft University of Technology, The Netherlands.
- Fumarola, M., Y. Huang, C. Tekinay, and M. D. Seck (2010). Simulation-based design for infrastructure system simulation. In G. K. Janssen, K. Ramaekers, and A. Caris (Eds.), *Proceedings of The 2010 European Simulation and Modelling Conference*, Belgium, pp. 288–293. Eurosis-ETI.

- Fumarola, M., M. D. Seck, and A. Verbraeck (2010). A DEVS component library for simulation-based design of automated container terminals. In *SIMUTools '10: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, Belgium, pp. 1–7. ICST.
- Girod, M., A. C. Elliott, N. D. Burns, and I. C. Wright (2003). Decision making in conceptual engineering design: An empirical investigation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 217(9), 1215–1228.
- Goel, V. (1995). *Sketches of Thought*. MIT Press.
- Hoover, S. P. and J. R. Rinderle (1991). Models and abstractions in design. *Design Studies* 12, 237–245.
- Huang, Y., M. D. Seck, and A. Verbraeck (2011). From data to simulation models: Component-based model generation with a data-driven approach. In S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu (Eds.), *Proceedings of the 2011 Winter Simulation Conference*, Phoenix, AZ, pp. 3724–3734. IEEE, Piscataway, NJ.
- Law, A. M. (2007). *Simulation Modeling and Analysis* (4th ed.). McGraw-Hill.
- Mingers, J. and J. Rosenhead (2004). Problem structuring methods in action. *European Journal of Operational Research* 152, 530–554.
- Munro, I. and J. Minger (2002). The use of multimethodology in practice: results of a survey of practitioners. *Journal of the Operational Research Society* 53, 369–378.
- Paredis, C., A. Diaz-Calderon, R. Sinha, and P. Khosla (2001). Composable models for simulation-based design. *Engineering with Computers* 17(2), 112–128.
- Robinson, S. (2001). Soft with a hard centre: Discrete-event simulation in facilitation. *The Journal of the Operational Research Society* 52, 905–915.
- Saenen, Y. A. (2003). *An approach for designing robotized marine container terminals*. Ph. D. thesis, Delft University of Technology, The Netherlands.
- Shannon, R. E. (1975). *Systems simulation: the art and science*. Prentice Hall, Inc.
- Simon, H. A. (1996). *The Sciences of the Artificial* (3rd ed.). MIT Press.
- Sol, H. G. (1982). *Simulation in Information Systems Development*. Ph. D. thesis, University of Groningen, The Netherlands.
- Ullman, D. G., T. G. Dietterich, and L. A. Stauffer. (1998). A model of the mechanical design process based on empirical data. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing* 2(1), 33–52.
- Zeigler, B. P., H. Praehofer, and T. G. Kim (2000). *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems* (2nd ed.). Elsevier/Academic Press.