From data to simulation models : component-based model generation with a data-driven approach

Huang, Y; Seck, MD; Verbraeck, A

DOI
10.1109/WSC.2011.6148065

Publication date
2011

Document Version
Accepted author manuscript

Published in
Proceedings of the 2011 Winter Simulation Conference

Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

FROM DATA TO SIMULATION MODELS: COMPONENT-BASED MODEL GENERATION WITH A DATA-DRIVEN APPROACH

Yilin Huang
Mamadou D. Seck
Alexander Verbraeck

Systems Engineering Group
Faculty of Technology, Policy and Management
Delft University of Technology
PO Box 5015, NL-2600GA Delft
The Netherlands

ABSTRACT

Model building is time-consuming and requires expertise in different areas. In this paper, we propose a data-driven approach for automatic model generation using pre-built and validated model components (or building blocks). We view this approach as an automated reuse of model components. Issues such as modularity and composability of model components are addressed. Models can be generated by automatically selecting, structuring and configuring the model components. The formulated rules can be structural and behavioral, by which a relational representation of the desired model composite structure is incrementally constructed. An example of generating a rail network model is given to demonstrate the steps.

1 Introduction

Building a valid simulation model is time intensive. It is a highly specialized activity which often necessitates expertise in different areas. It is challenging to build valid and efficient models given the complexity and scale of systems to be modeled nowadays. Having enriched the knowledge base of modeling and simulation (M&S), simulationists start to consider issues related to automatic model generation (AMG). AMG renders simulation studies more efficient and effective. This does not only improve the process of simulation itself but also better integrates simulation with its applications, such as simulation-based design, engineering, training, etc.

Many AMG attempts used formal model specifications as the basis for simulation model (or code) generation (Balci et al. 1990, Kang 1997, Son et al. 2000, Son et al. 2003, Posse and Bolduc 2003, Foeken and Voskuijl 2010). The formal specification could be in the form of *inter alia* a specification language, a meta-model or their graphical representations.

Some recent works are inclined to data integration (Randell 2001, Shephard et al. 2004, Jeong and Allan 2004, Lucko et al. 2010). The goal is set to generate simulation models directly from external data sources using data structuring and analysis algorithms for creating and configuring the model. Hence such an approach is also referred to as being data-driven (Huang and Verbraeck 2009, Bergmann and Strassburger 2010). According to Bergmann and Strassburger (2010), approaches of (semi-) AMG can be classified into three main categories:

1. Parametric approaches: Models are generated based on existing simulation building blocks (stored in simulation libraries) which are selected and configured based on parameters.

2. Structural approaches: Model generation is based on data describing the structure and layout of a system, typically from relevant computer aided design (CAD) and computer aided engineering (CAE) systems.
3. Hybrid-knowledge-based approaches: These approaches combine artificial intelligence (AI) methods (e.g. expert systems and neuronal nets) with both of the above approaches.

Component-based modeling (or component-based software in general) and its advantages are extensively discussed in literature. It is founded on a paradigm which is common to all engineering disciplines: complex systems can be obtained by assembling components (building blocks), e.g. see Simon (1996)'s parable of the two watchmakers; composition is used to build complex components from simpler ones (Gössler and Sifakis 2005). A complex system can be recursively decomposed into sub-systems until an elementary level is reached, where a sub-system can no longer be reduced; the criterion for reduction is functionality (Simon 1996). Such a decomposition (or composition if we observe it bottom-up rather than top-down) forms the system and the model of the system into a hierarchal structure.

The behavior of a model component is determined by the collective behavior of its sub-components, and varying the composite structure of a component can alter the behavior of that component. In other words, component-based modeling requires not only the information about what components the model is composed of, but also the information about the model composite structure. If both types of information can be obtained directly or indirectly from available data sources, it is feasible to develop a fully automated model generator. With this approach, simulation models are generated by the composite of pre-built and validated model components using data structuring and analysis methods and techniques. In this case, sufficient data sources are required but not formal model specifications.

The authors argue that using data is preferable in many cases than model specifications for the following reasons. First, defining model specification takes considerable time for large-scale models and requires personnel that have proficiency in using the specification. Second, as CAD systems are more frequently employed in recent years, design data are often available for use; and similarly, the advances in data collection and storage technology enabled many organizations and institutions to accumulate a large amount of data. Third, large-scale models are often built for systems that have long life cycles. Although developing a model component library inevitably introduces great complexity and cost, it is still beneficial in terms of model reuse and modification in a long term perspective. Additionally, fully automated model generation using pre-built and validated model components can minimize errors caused during model specification. If successfully designed and implemented, AMG could make it easier to integrate M&S into the design, operation, and control processes of a system.

This paper discusses such a data-driven approach for component-based model generation. It addresses the principles of model component design and how model generation may be approached by analyzing the data. An example of generating a simulation model of a rail transport network is given to illustrate how each part/step of the AMG is completed.

2 Principles of Model Component Design

Parasuraman and Riley (1997) defines automation as the execution by a machine agent (usually a computer) of a function that was previously carried out by a human. The AMG described in this paper is the automated reuse of model components. This approach proposes to generate models (re-)using a library of (pre-build and validated) model components based on the analysis of the available data sources. Therefore, the first step is to develop a model component library that is suitable for automated reuse. Developing simulation models that are intended for reuse shows many similarities with software engineering development (Pidd and Robinson 2007). The goals and qualities of software design (Sommerville 1996, Braude and Bernstein 2010), such as sufficiency, understandability, modularity, cohesion, coupling, robustness, flexibility, reusability, information hiding, efficiency and reliability, can be equally applied to simulation model design. However, model component design has its own focuses.

Model components are sometimes termed as model building blocks (MBBs) (Valentin and Verbraeck 2002, Verbraeck and Valentin 2008). According to Verbraeck et al. (2002), a building block is a self-contained, interoperable, reusable and replaceable unit that encapsulates its internal structure and provides useful services to its environment through precisely defined interfaces. Model components or MBBs are the basic units in constructing simulation models. Therefore, if a MBB could represent a distinct physical phenomena, a logical process, a method, etc., as how it is distinguished in real life, the modelers and domain experts could easily identify and allocate the components to populate larger models. In this way, domain specificity and knowledge are integrated into the model by the identification, terminology and behavior of MBBs (Valentin and Verbraeck 2002).

Over-sizing or under-sizing MBBs can be avoided by dividing them according to core functionalities. The criteria used to do so are in some cases obvious or intuitive, in others domain expertise has to be consulted. Questions such as "is it likely that model users need to change some functionalities of the model" shall be asked during the design process.

The essence in designing components is composability. Composability is the capability to select and assemble simulation components in various combinations into valid simulation systems to satisfy specific user requirements (Petty and Weisel 2003). According to Simon (1996)'s near decomposability concept (a property of complex systems), (1) the interactions between sub-systems in a complex system are weaker than the interactions within them, and (2) each sub-system in a decomposed system is almost autonomous, i.e., each has independent functionality but still provides value to the overall system by maintaining a weak connection with it. From a modeling perspective, near decomposability or composability of model can be achieved through the modularity of its components.

Port-based modeling is proposed in many M&S literature for its modular principles, e.g, Zeigler et al. (2000), Paredis et al. (2001), Liang and Paredis (2004), Breedveld (2009). The interface of a port-based model (PBM) consists of input and output ports, through which all interactions with the environment occurs. The interactive relations are represented by couplings (sometimes also called connections or bonds). The model can place a value on one of its ports, but the actual destination of the output is not determined until the model becomes a component in a larger model and the coupling relation is specified; therefore, a PBM can (1) be developed as a stand-alone unit, (2) be placed in a model repository and reactivated at will, and (3) be reused in any context in which its behavior is appropriate and coupling to other components makes sense (Zeigler et al. 2000). The model interacts with its environment only via its ports; thereby, direct dependencies between components are eliminated (Röhl and Morgenstern 2007). For these reasons, we consider port-based modeling and its principles adequate for model component design, as they fullfill Simon (1996)'s near decomposability concept.

In our research, the Discrete Event System Specification (DEVS), a formalism for port-based modeling defined by Zeigler et al. (2000), is used to specify the model components. In the DEVS formalism, sub-systems at an elementary level (that can no longer be reduced according to certain criteria) are represented by atomic models and sub-systems at higher levels are represented by coupled models; see Zeigler et al. (2000). Take rail network modeling, for instance, the atomic models can be rail vehicles, track segments, sensors, signals, etc. Each has dedicated ports for information exchange. The coupled models are stations, intersections, block systems, etc., each of which is composed of atomic models or smaller coupled models.

In general, the model component development process follows the conversional M&S process. However, as the development of model component is intended for reuse during design and test, the modeler shall place more emphasis on model decomposition strategies, and the modularity and composability of individual component to enable the automation of the reuse process.

3 Model Composite and Hierarchy

Structural composition as a model generation strategy is powerful because it links to the essence of design as understood in system theoretical terms. In Zeigler et al. (2000), system design is understood as the process of going from a lower to a higher level of system epistemology, namely from the level of the generative (or lower order structure) systems to that of a higher order structure system. The designer uses components,

with the knowledge of their inner structure and behavior as his or her basic vocabulary, and connects them within new structures, encoding new knowledge that was not available in the previous step. The result of this design activity can in turn be analyzed by simulating it, which corresponds to automatically translating the structure system to a data system. Following the concept of closure under coupling, defined in Zeigler et al. (2000), the resultant of the coupling of DEVS models is also a DEVS model, making it possible to incrementally develop a hierarchy of more and more complex model components, whose patterns of state changes could not have been specified all at once. Furthermore, the hierarchical composition of models makes it possible to separate the different levels of organization which are relevant for system analysis. This makes model understanding easier as long as the world view guiding the component library definition coincides with the world view of the end users. There is, however, a challenge in hierarchical composition of models. First, a model library must be developed and this development process follows a top-down hierarchical decomposition of the system, in order to identify the different entities relevant in the domain of application. Such a decomposition is done according to a certain functional perspective, which per definition excludes many aspects of the system of interest. This will result in a component that cannot be used to assemble a useful model in some other context. This must be strongly communicated to the end users of the model library. An advantage of a data driven AMG is that it ensure the model components will be used in accordance with the pragmatic frame that led to their development, since the process of component selection and structuring is automated and beyond the reach of the user.

4 Automatic Model Generation

As mentioned earlier, once developed and validated, a model component library integrates domain specificity and knowledge by the identification, terminology and behavior of the model components. When we, as human modeler, want to construct a larger model from model components, we need to know (1) what components to use, and (2) how to configure and structure them together. For the automated process, the same information is required.

AMG is centered on a model generation algorithm (MGA) that constructs models from selecting, configuring and structuring the components. Model selection heuristics, that represent domain knowledge of whether the component is relevant to a modeling goal, are used to guide the search and composite of the applicable components (Lee and Zobel 1996). They are useful for defining the inferential rules of the model composite algorithm. Applicability conditions are the formulated rules of selecting, configuring and eventually structuring a model component. There are two kinds of applicability conditions as stated by Gruber (1993): structural preconditions and behavioral preconditions.

1. Structural preconditions are used to associate model components with objects that are given in the initial structure of the modeled system.In the simplest cases, model components can be directly associated with real (existing) components. Structural preconditions are also used for identifying structural abstractions, i.e., model composite relations.
2. Behavioral preconditions state the required dynamic conditions under which a model component applies or shall be configured. They are used to determine which or whether certain model behavior will occur. Behavioral preconditions also can be used for identifying dynamic structural abstractions. During simulation experiments, behavioral preconditions may change so that some model components need to be removed, replaced, reconfigured or restructured.

In principle, if the applicability conditions of a model component hold, then the component will be selected, configured and structured accordingly. The information needed by the MGA to infer sufficient applicability conditions is derived from the available data sources. In some cases, if the modeled system has a simple structure, the model can be directly generated. However, in many cases, the system is complex and the data sources that describe the system do not contain a structure and relational logic that can be directly applied to the desired model structure and to create the coupling relations. Therefore, intermediate data structures are often employed to incrementally construct a relational representation of the model (component) structure.

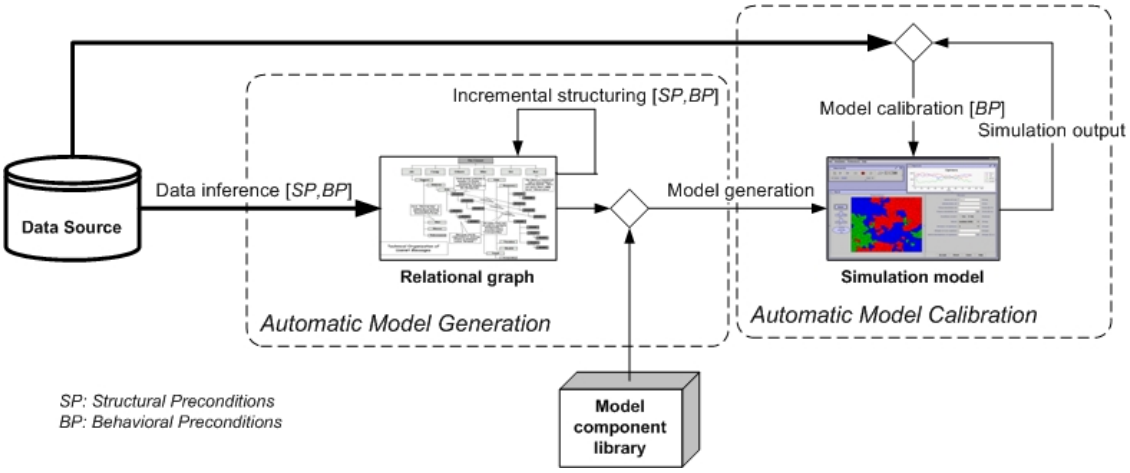

Figure 1: Component-based model generation with a data-driven approach.

Figure 1 shows how data are utilized in the context of AMG. The scope of our research covers two parts. The first is to use data and model components for AMG. The second part is to analyze the simulation output in comparison with the available data in order to perform automatic model calibration (AMC). The data of these two parts are of different nature. For AMG, the data analysis results shall reflect the structural and behavioral preconditions that are the basis for constructing the (initial) model structure and initializing the model state. These data come from design, process planning, operational scheduling, resource allocation, etc. In AMC, the model is adjusted in an iterative process where the data from the real system and the simulation output data are compared and analyzed. The analysis may detect divergences that are caused by behavioral preconditions which are not possible to discover during the AMG process, because the source of the behavioral preconditions is not observed or is not observable from the real system so that there are not direct data sources that could reflect such behavioral preconditions. The data used in AMC is typically collected during the operation of the real system. In this paper we focus on AMG but not AMC.

5 Data Sources and Knowledge Discovery for Model Generation

The advance of computer aided technologies (such as computer aided design and engineering, enterprise resource planning, manufacturing process management) together with the rise of data collection technology and inexpensive storage media in recent years have enabled organizations to accumulate a vast amount of data (Tan et al. 2005). For AMG, data are useful if they contain information about model component selection, initialization and structuring so that a complete model can be built which satisfies user requirements. Relevant data sources typically contain information with the following categories.

- Data describing geomatics, geometry or topology of a system or sub-systems.
- Data describing orders or demands.
- Data describing resources.
- Data describing operations or processes.
- Data describing products or services.

Depending on the application domain, different types of data can be of value for AMG. They can be gathered during the design phase (such as CAD data), planning phase (such as crew and resource scheduling), operation phase (such as data collected by sensors), etc. Simulation data (from other models) may be an additional source if the results are credible. In public transport simulation, for example, the results from road traffic simulation (of private road transport) are often used as inputs to study the mutual influence between them. Because the data are not produced for the purpose of model building, often a few steps need to be completed in order to generate models from these data sources. In this regard, the concepts

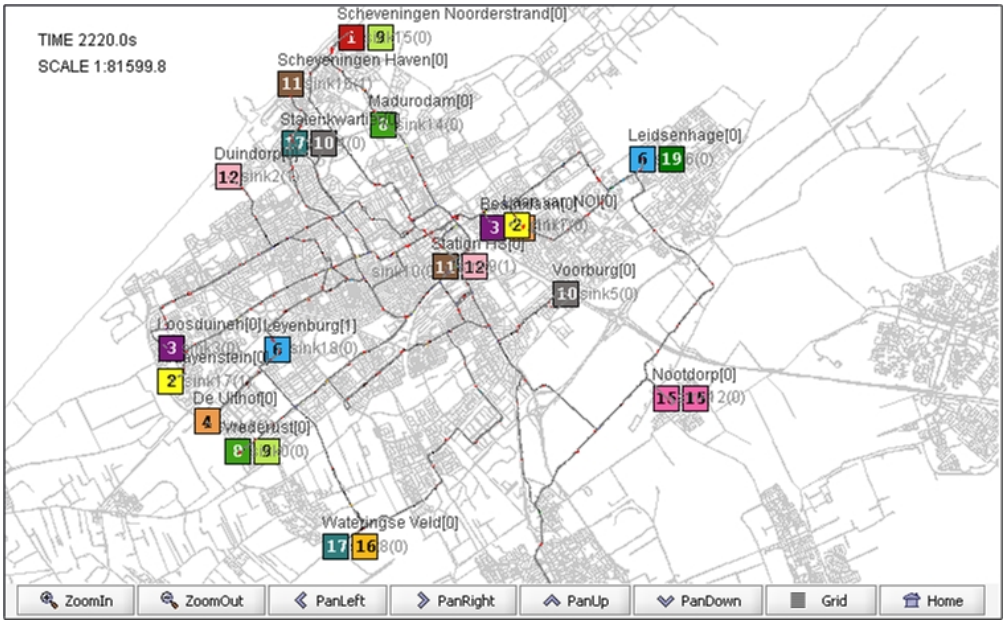

Figure 2: The Hague tram network model.

and techniques of data mining are applicable, as the "Automatic Model Generation" box in Figure 1 can be seen as knowledge discovery, i.e., data⇒knowledge⇒model. The knowledge needed is a composite structure graph by which the available model components can be directly mapped and coupled.

The terms of data mining (DM) and knowledge discovery in database (KDD) are sometimes used interchangeably. Nonetheless, the former is a step of the latter. KDD is the overall process of converting data into knowledge. The process includes data selection, preprocessing, data mining and data pattern postprocessing (Fayyad et al. 1996). Data preprocessing includes activities such as data cleaning, normalization, aggregation, subsetting, and transformation (Tan et al. 2005, Kononenko and Kukar 2007). The Industrial Standard Process for Data Mining (Chapman et al. 2000 or www.crisp-dm.org) provides a reference of different phases in data mining projects, their respective tasks and relationships. And many literatures discuse the theory, tasks, methods and techniques of DM, e.g., Kleinberg et al. (1998), Hand et al. (2001), Coppi (2002), Han and Kamber (2005), Tan et al. (2005), Sumathi and Sivanandam (2006), Kononenko and Kukar (2007). In the next section we will explain by means of an example how information for AMG may be extracted from the data.

6 An AMG Example

Figure 2 shows a generated simulation model of the tram network in The Hague, The Netherlands. The network covers over 150 km^2 with 14 tram lines, 135 km tracks and 539 stops. A rail simulation model component library called LIBROS (Huang et al. 2010, Huang et al. 2011) is first developed according to the principles and concepts discussed in Section 2 and 3. Based on our experience, manually constructing such a large model could be problematic because of the long time needed (5∼6 weeks per 10 km) and the risk of errors. The data used for AGM in this example are: (1) CAD data which describe the blueprint of the rail tracks and stop locations, (2) timetables that schedule the tram services, (3) service configurations that define the routes of the tram lines, (4) vehicle types that are used for the tram lines, and (5) line transformation data that specify the locations where one line may change to another.

CAD data can provide useful structural information. However, they only contain a list of geometric primitives that describe each object (Flynn and Jain 1991). Geometric inference is needed to identify and recognize the objects and object relations and represent them in data structures that could be easily translated into model structures. This step is often complex and needs to be divided it into a number of sub-steps that incrementally construct the desired (hierarchical) model structure. Figure 3 shows an

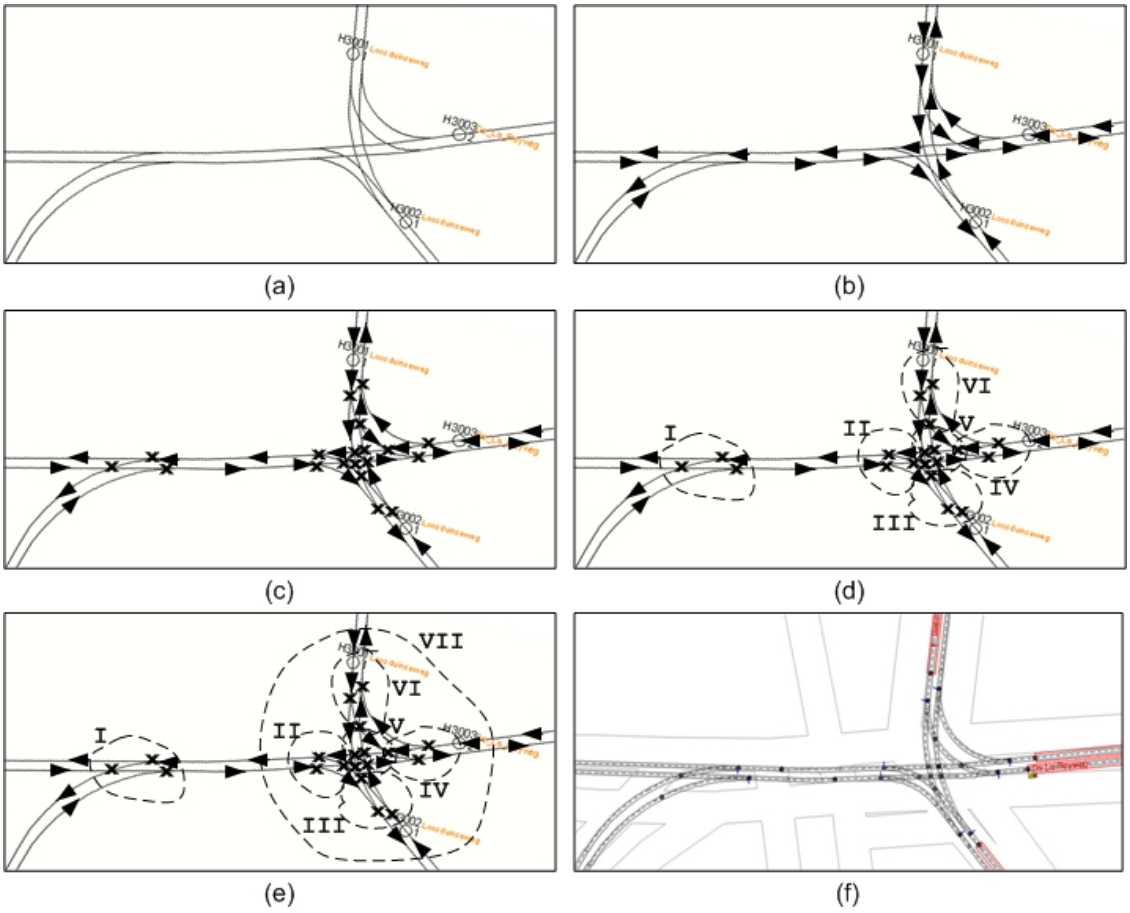

Figure 3: From CAD objects to model structure: an example.

example of the steps. (a) is a fraction of the rail infrastructure representing the infrastructure blueprint by means of CAD data. Each line and arc object represents a rail *track* segment that contains information about its geometry, but there is no orientation or connectivity information. (By orientation we mean that the track segments in the model shall indicate the direction of the traffic current.) These information are obtained first through traversing the whole rail network (graph traversal) starting from the sources where the vehicles are generated, i.e., at the terminal stations. (b) shows the result of this step. Each track segment becomes a vectorial object being indexed by another and represented in a directed graph data structure. In (c) the *point* locations (indicated by "**x**" in the figure) are detected where more than two track segments connect. At the same time, the point type is determined (diverging points are the ones with 1 in track and 2 out tracks; diverging contrariwise; crossing points are the ones with 2 in tracks and 2 out tracks but one direction does not interfere with the other). These points are inserted into the graph structure. (They are not objects in the CAD data but are a part of the simulation model.)

Based on the type of the points, their position and connectivity, we can determine different patterns of intersections. In the example, each of I∼IV and VI consists of 1 converging point, 1 crossing point and 1 diverging point. These points are positioned in a specific way and together with the tracks they form a "Y" shape. V has a square shape where 4 crossing points are connected next to each other. These points and the related track segments are aggregated, as shown in (d), and node transformations in the graph structure are performed at this step. The nodes of (c) are the points, whereas the nodes of (d) are the groups of I∼VI each of which contains a sub-graph. In (d) a second-order node transformation is performed by which II∼VI are aggregated to VII, forming a "butterfly" shape.

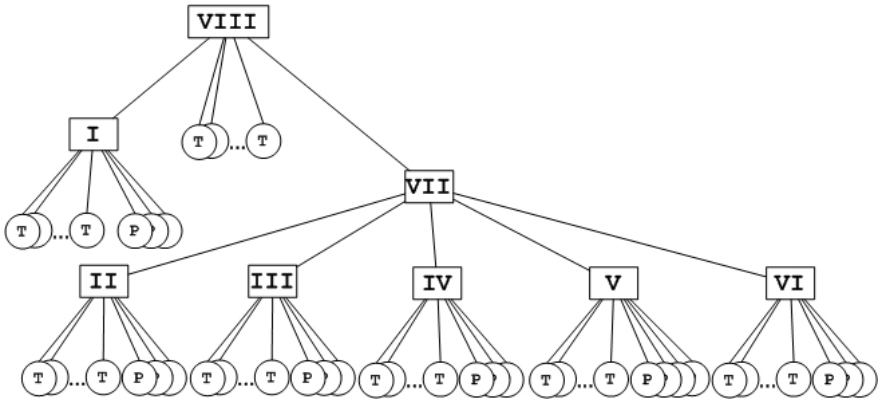

Figure 4: The model composite structure tree: an example.

The model composite structure insofar is illustrated in Figure 4. Under the top level infrastructure (the root VIII) as in the example, there are 2 aggregation I and VII, the "Y" intersection and the "butterfly" intersection. Both have corresponding (coupled) model components in the LIBROS library. The composite members are used to configure and couple the model components at model generation stage. Additionally, control units are instantiated in coupled models when appropriate. In the example, each intersection has a control unit that monitors the correct operation of that intersection. (This is a part of the model component itself.) Service configuration data are used to configure the routes so that the points can switch to the correct position for the vehicles moving along the right route. The generated infrastructure model is shown in (f) which is a fraction of the model in Figure 2.

Stations, block sections and the other types of intersections are generated in a similar way. For example, a station occupies a certain length of track whose starting point is indicated in the CAD data. The timetable data of that station are also loaded into the model of that station for modeling and statistics purposes. Vehicles are generated at terminal stations according to the scheduled services.

7 Conclusions and Future Work

This paper discussed component-based model generation with a data-driven approach. As many organizations possess a large amount of data, using them more efficiently and effectively in simulation studies could not only improve the quality of simulation itself but also better integrate simulation with its applications, such as simulation-based design, simulation-based engineering, simulation-based training, etc. Based on the experience in developing simulation model building blocks and using different data sources in generating large-scale rail network models, we high-lighted and clarified a couple of issues in model component design and automatic model generation, such as the modularity, the composability and the incremental construction of model composite structure. We also gave an example of automated generation of a rail network model using available data sources. Future research will focus on automatic model calibration. Different data analysis methods and techniques will be investigated to estimate the model parameters.

REFERENCES

Balci, O., R. E. Nance, E. J. Derrick, E. H. Page, and J. L. Bishop. 1990. "Model generation issues in a simulation support environment". In *Proceedings of the 1990 Winter Simulation Conference Proceedings*, 257–263.

Bergmann, S., and S. Strassburger. 2010. "Challenges for the Automatic Generation of Simulation Models for Production Systems". In *Proceedings of the 2010 Summer Simulation Multiconference*, 545–549. Ottawa, Canada.

Braude, E. J., and M. E. Bernstein. 2010. *Software engineering: Modern approaches*. 2nd ed. John Wiley & Sons.

Breedveld, P. 2009. *Modeling and control of complex physical systems - the port-hamiltonian approach*, Chapter Port-Based Modeling of Dynamic Systems, 1–52. Springer Verlag.

Chapman, P., J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. 2000. *Crisp-dm 1.0 step-by-step data mining guide*. CRISP-DM consortium.

Coppi, R. 2002. "A theoretical framework for Data Mining: the "Informational Paradigm"". *Computational Statistics & Data Analysis* 38 (4): 501–515.

Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth. 1996. "From data mining to knowledge discovery in databases". *American Association for Artificial Intelligence Magazine* 17 (3): 37–53.

Flynn, P. J., and A. K. Jain. 1991. "CAD-Based Computer Vision: From CAD Models to Relational Graphs". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (2): 114–132.

Foeken, M., and M. Voskuijl. 2010. "Knowledge-based simulation model generation for control law design applied to a quadrotor UAV". *Mathematical and Computer Modelling of Dynamical Systems* 16 (3): 241–256.

Gössler, G., and J. Sifakis. 2005. "Composition for component-based modeling". *Science of Computer Programming* 55 (1-3): 161–183.

Gruber, T. R. 1993. "Model formulation as a problem-solving task: computer-assisted engineering modeling". *International Journal of Intelligent Systems* 8 (1): 105–127.

Han, J., and M. Kamber. 2005. *Data mining: Concepts and techniques*. 2nd ed. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers.

Hand, D., H. Mannila, and P. Smyth. 2001. *Principles of data mining*. Cambridge: MIT Press.

Huang, Y., M. D. Seck, and A. Verbraeck. 2010. "LIBROS-II: Railway Modelling with DEVS". In *Proceedings of The 2010 Winter Simulation Conference*, Edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan. Baltimore, MD, USA: IEEE.

Huang, Y., M. D. Seck, and A. Verbraeck. 2011. "A DEVS Library for Rail Operations Simulation". In *Proceedings of The 2011 Spring Simulation Multiconference*. Boston, MA, USA: SCS.

Huang, Y., and A. Verbraeck. 2009. "A Dynamic Data-Driven Approach For Rail Transport System Simulation". In *Proceedings of The 2009 Winter Simulation Conference*, Edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 2553–2562. Austin, TX, USA: IEEE.

Jeong, K.-Y., and D. Allan. 2004. "Integrated system design, analysis and database-driven simulation model generation". In *Proceedings of the IEEE Annual Simulation Symposium*, 80–85.

Kang, S. 1997. "Knowledge based automatic simulation model generation system". *IEE Proceedings: Circuits, Devices and Systems* 144 (2): 88–96.

Kleinberg, J., C. Papadimitriou, and P. Raghavan. 1998. "A Microeconomic View of Data Mining". *Data Mining and Knowledge Discovery* 2 (4): 311–324.

Kononenko, I., and M. Kukar. 2007. *Machine learning and data mining: Introduction to principles and algorithms*. UK: Horwood Publishing.

Lee, C., and R. Zobel. 1996. "Representation of simulation model components for model generation and a model library". In *Proceedings of the IEEE Annual Simulation Symposium*, 193–201.

Liang, V.-C., and C. Paredis. 2004. "A port ontology for conceptual design of systems". *Journal of Computing and Information Science in Engineering* 4 (3): 206–217.

Lucko, G., P. C. Benjamin, K. Swaminathan, and M. G. Madden. 2010. "Comparison of manual and automated simulation generation approaches and their use for construction applications". In *Proceedings of the 2010 Winter Simulation Conference*, 3132–3144.

Parasuraman, R., and V. Riley. 1997. "Humans and automation: Use, misuse, disuse, abuse". *Human Factors* 39 (2): 230–253.

Paredis, C., A. Diaz-Calderon, R. Sinha, and P. Khosla. 2001. "Composable models for simulation-based design". *Engineering with Computers* 17 (2): 112–128.

Petty, M. D., and E. W. Weisel. 2003. "A composability lexico". In *Proceedings of the Spring 2003 Simulation Interoperability Workshop*, 181–187. Simulation Interoperability Standards Organization.

Pidd, M., and S. Robinson. 2007. "Organising insights into simulation practice". In *Proceedings of the 2007 Winter Simulation Conference*, 771–775. Piscataway, NJ, USA: IEEE Press.

Posse, E., and J.-S. Bolduc. 2003. "Generation of DEVS modelling and simulation environments". In *Proceedings of the 2003 Summer Computer Simulation Conference*, Edited by A. Bruzzone and M. Itmi, 139–146. Montral, Canada.

Randell, L.G., B. G. 2001. "Database driven factory simulation: A proof-of-concept demonstrator". In *Proceedings of the 2001 Winter Simulation Conference*, Volume 2, 977–983.

Röhl, M., and S. Morgenstern. 2007. "Composing simulation models using interface definitions based on web service descriptions". In *Proceedings of the 2007 Winter Simulation Conference*, 815–822.

Shephard, M. S., M. W. Beall, R. M. O'Bara, and B. E. Webster. 2004. "Toward simulation-based design". *Finite Elements in Analysis and Design* 40 (12): 1575–1598.

Simon, H. A. 1996. *The sciences of the artificial*. 3rd ed. MIT Press.

Sommerville, I. 1996. *Software engineering*. 5th ed. Addison-Wesley.

Son, Y., R. Wysk, and A. Jones. 2003. "Simulation-based shop floor control: Formal model, model generation and control interface". *IIE Transactions* 35 (1): 29–48.

Son, Y. J., A. T. Jones, and R. A. Wysk. 2000. "Automatic generation of simulation models from neutral libraries: An example". In *Proceedings of the 2000 Winter Simulation Conference*, Volume 2, 1558–1567.

Sumathi, S., and S. Sivanandam. 2006. *Introduction to data mining and its applications*, Volume 29 of *Studies in Computational Intelligence*. Berline Heidelberg: Springer-Verlag.

Tan, P. N., M. Steinbach, and V. Kumer. 2005. *Introduction to data mining*. Addison-Wesley.

Valentin, E., and A. Verbraeck. 2002. "Guidelines for designing simulation building blocks". In *Proceedings of the 2002 Winter Simulation Conference*, Edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. Charnes, 563–571. San Diego, CA.: IEEE.

Verbraeck, A., Y. Saanen, Z. Stojanovic, E. Valentin, K. van der Meer, and A. M. B. Shishkov. 2002. *Building blocks for effective telematics application development and evaluation*, Chapter What are building blocks?, 8–21. TU Delft.

Verbraeck, A., and E. Valentin. 2008. "Design guidelines for simulation building blocks". In *Proceedings of the 2008 Winter Simulation Conference*, 923–932.

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.

AUTHOR BIOGRAPHIES

YILIN HUANG is a Ph.D. Candidate in the Systems Engineering Group of the Faculty of Technology, Policy and Management of Delft University of Technology. Her research interests include dynamic data-driven simulation, on-line data analysis, and transportation systems simulation. Her email address is y.huang@tudelft.nl.

MAMADOU D. 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. from the Paul Cézanne University of Marseille, France. His research interests include modeling and simulation formalisms, dynamic data-driven simulation, human behavior simulation, and agent directed simulation. His email address is m.seck@tudelft.nl.

ALEXANDER VERBRAECK is a Full Professor 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 focuses on development of object-oriented simulation building blocks, participative modeling, serious gaming using virtual reality, and agent technology in simulation. His email address is a.verbraeck@tudelft.nl.