Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

STANDARDS BASED GENERATION OF A VIRTUAL FACTORY MODEL

Sanjay Jain

Department of Decision Sciences
School of Business
The George Washington University
2201 G Street NW, Suite #415
Washington, DC 20052, USA

David Lechevalier

Le2i – Laboratoire Electronique,
Informatique et Image
Université de Bourgogne,
allée Alain Savary
21000 Dijon, FRANCE

ABSTRACT

Developing manufacturing simulation models usually requires experts with knowledge of multiple areas including manufacturing, modeling, and simulation software. The expertise requirements increase for virtual factory models that include representations of manufacturing at multiple resolution levels. This paper reports on an initial effort to automatically generate virtual factory models using manufacturing configuration data in standard formats as the primary input. The execution of the virtual factory generates time series data in standard formats mimicking a real factory. Steps are described for auto-generation of model components in a software environment primarily oriented for model development via a graphic user interface. Advantages and limitations of the approach and the software environment used are discussed. The paper concludes with a discussion of challenges in verification and validation of the virtual factory prototype model with its multiple hierarchical models and future directions.

1 INTRODUCTION

A number of past and recent initiatives, such as smart manufacturing (SMLC 2012) and Industrie 4.0 (Mario, Tobias, and Boris 2015) have identified modeling and simulation as key to the advancement of manufacturing. Some have proposed the use of simulation at multiple levels within manufacturing, ranging from physics-based models of the manufacturing process at a very detailed level to high level supply chain models and everything in between.

The development of simulation models of manufacturing systems requires expertise in multiple areas including manufacturing operations, conceptual modeling of manufacturing systems at the right level of abstraction, and implementation of the conceptual model using appropriate simulation software. The expertise requirement goes up substantially if multiple levels of manufacturing are to be modeled. The expertise requirement can be a deterrence to wider use of simulation in manufacturing. It could be a roadblock for the move towards smart manufacturing and Industrie 4.0.

A virtual factory has been presented as a multi-resolution simulation model of a corresponding real factory with the capability to model with high fidelity if desired (Jain et al. 2015). Such a virtual factory model can provide the modeling and simulation capabilities envisaged in smart manufacturing and Industrie 4.0. However, at present developing a virtual factory model may only be an option for large corporations with substantial budgets given the high expertise requirement.

This paper proposes automatic generation of virtual factory models based on data as a way to reduce the expertise requirement and thus facilitate the increased use of simulation. This is admittedly not a new idea. There have been solutions from commercial vendors of discrete event simulation (DES) software in the past that generated factory models when provided with a data file in their proprietary data formats. This

978-1-5090-4486-3/16/$31.00 ©2016 IEEE

effort proposes going beyond the prior offerings in two key ways. First, the effort proposes the use of standard formats for input data describing the subject manufacturing system. Second, the generated model is proposed to be a virtual factory model as defined with multi-resolution capabilities rather than the single level models available in commercial offerings. The current implementation of the concept is a first step with limited scope with use of only a couple standards towards achieving the proposed capability.

The next section discusses *relevant* efforts reported in the literature over the last year. Section 3 discusses the proposed overall approach to achieve the vision of virtual factory and provides an overview of the approach for the current implementation. The implementation using AnyLogic, the selected software for this initial step, is then described in detail in section 4. Section 5 concludes the paper with a discussion of future work to continue implementation of the virtual factory concept.

2 LITERATURE REVIEW

Publications relevant to this effort that have appeared over the last year are briefly reviewed in this section. The relevant areas are virtual factory and the key enabling capabilities of multi-resolution modeling, hybrid simulation, and automatic model generation. Readers are referred to Jain and Shao (2014) and Jain et al. (2015) for reviews of relevant literature prior to last year.

2.1 Virtual Factory

Jain and Shao (2014) presented four different connotations of virtual factory in the extant literature: as a high fidelity simulation as used in this paper, as a virtual organization, as a virtual reality (VR) representation, and as an emulation facility. Over the last year, publications using the four different connotations have continued to appear, but a much higher proportion of papers use the definition of virtual factory as a high fidelity simulation, with papers using the virtual reality representation definition in second place. There are multiple papers that combine the high fidelity simulation and virtual reality representation connotation and some even extend to include the emulation aspect. A representative sample of relevant publications, that use the first definition of virtual factory are primarily reviewed here.

Terkaj, Tolio, and Urgo (2015) present an ontology-based virtual factory that is continuously updated to reflect the events of the corresponding real factory. They used ARENA as the DES software together with a 3D animation. The virtual factory can be used for predictive analysis, for example evaluating management decisions on production plans and maintenance. While a capability for a multi-scale representation is mentioned, it does not appear to cover the process-machine-cell/line-factory hierarchy. Oyekan et al. (2015) utilize WITNESS simulation software connected to Visionary Renderer VR software package to create a high fidelity simulation of the factory floor. It does not appear to allow multiple resolution modeling but it does appear to have some of the core capabilities of a virtual factory as defined here.

Ayadi et al. (2015) present an information system developed using WinDev for supporting a digital factory. They place an emphasis on the product development process with the supporting information system providing access to the product and process information. The production simulation uses an unnamed software and appears to provide the capability to support multiple resolution levels including plant, line, station, and task. While the authors recognize the need for interoperability and base the concepts on standards, the data interfaces for the current system are custom developments. Constantinescu, Francalanzab, and Matarazzoc (2015) propose to develop the virtual factory using Systems Modeling Language (SysML) and TOPCASED open source software and emphasize the knowledge capturing and human system interface aspects to support decision makers using the results from modeling and simulation. The use of open source software may require a large effort to reach to the level of capabilities available in commercial simulation software, but it is a worthwhile effort for making the developments available to a wider community.

Some publications focus on the "digital factory" concept, suggesting the extensive use of modeling and simulation in a manufacturing context but with a product development perspective. Gregor, Hercko, and Grznar (2015) suggest that a digital factory uses data from simulation while a virtual factory uses data from a real factory, though such a distinction was not found in other publications. Kádár, Terkaj, and Sacco (2013) present an alternate view and define digital factory tools as modules of virtual factory for data communication including supporting synchronization of real and virtual representations. Matsuda and Kimura (2015) use a digital factory for evaluating sustainability of manufacturing systems and production plans. The system is implemented using the commercial multi-agent simulator "artisoc" and provides the levels of granularity for the simulation at the factory, machine, and product levels. Matsuda and Kimura (2015) appear to use the terms "digital factory" and "virtual factory" interchangeably though most literature differentiates between the two concepts.

While the efforts focusing on a virtual factory connotation other than as a high fidelity simulation are not discussed, a good source for papers identifying virtual factory as a VR representation is the review provided by Choi, Jung, and Do Noh (2015) encompassing 154 papers over last couple of decades discussing the application of VR to manufacturing.

2.2 Multi-Resolution Modeling

Schönemann et al. (2016) present a multi-level framework for manufacturing system simulation and identify the data exchanges among the disparate simulation models at different levels. The framework presented includes a few other models beyond manufacturing, such as product, building, and technical building services (TBS), to provide a holistic approach. The example is quite similar to that presented in Jain et al. (2015) including the use of AnyLogic software, Java code for the process model, agent based simulation (ABS) to model the machine level, and DES models for the cell/process chain level. They appear to use custom interfaces for data, which can lead to a large effort when using the approach for a new scenario.

Alvandi et al. (2015) use a similar approach with an example implementation of their proposed hierarchical modeling framework for the analysis of material and energy flows in manufacturing systems. They also use AnyLogic software, an ABS model for the unit process level (similar to machine level), and a DES model for the process chain level. Their factory level model includes the TBS, production planning & control, and transport modules. They access data for the models directly from enterprise resource planning (ERP) and supervisory control and data acquisition (SCADA) systems. Their approach may require substantial effort to apply in scenarios with different ERP and SCADA systems.

The independent use of similar set-ups in the two efforts discussed above provides support for the approach used in Jain et al. (2015). The use of standards in such efforts will facilitate collaboration and implementation.

2.3 Hybrid Simulation

The use of hybrid simulation, that is multiple simulation paradigms in the virtual factory models, leads to the challenge of their effective integration. In this work, the use of AnyLogic software has enabled hybrid simulation to be used without having to use a distributed simulation mechanism such as that used by Rabelo et al. (2015) for a hierarchical model of a supply chain with the higher level modeled using system dynamics paradigm in Vensim and the functional details modeled using DES in ARENA.

Abduaziz et al. (2015) also utilize hybrid simulation for a green logistics assessment. However, in their work the supply chain system dynamics model developed in iThink doesn't run concurrently with the factory DES model in Arena. The two models are executed iteratively with information exchanges among the two models. The work supports the idea of using simulation paradigms appropriate to the level of detail.

2.4 Automatic Model Generation

A number of efforts over the past couple decades have implemented mechanisms for automatic generation of simulation models corresponding to manufacturing systems. Barlas and Heavey (2016) provide a recent review of efforts to automate input data to DES for manufacturing. They define five categories for the purpose: intermediary database, Programmable Logic Controllers (PLC) programs, developed applications, data interfaces/standard translators, and direct integration. The current implementation for the reported effort in this paper primarily falls in the data interface/standard translators category particularly for the cell level model. The use of standards based interfaces is anticipated to allow for wider use of the proposed virtual factory capability.

Overall, the efforts reported in the literature in the four relevant areas over the last year support the direction of the work reported here.

3 APPROACH

3.1 Objectives

The objective of this part of the work is to automatically generate a virtual factory model, using data from the real factory in applicable standard formats, with the capability of generating output data streams based on other applicable standards formats. The automatic generation of the virtual factory model is intended to go beyond the previous efforts involving automatic generation of single level factory simulations by generating a multi-resolution model and using standard formats of input files.

The target multi-resolution model currently has three levels: process, machine, and cell. The machine and cell levels align with work unit and work center objects respectively defined as part of the equipment hierarchy in International Society for Automation's (ISA) ISA-95 and ISA-88 standards (MESA 2010). The process level aligns with the actual production process defined as Level 0 of ISA-95 Functional Hierarchy Model in MESA (2010). The cell level can also represent the production part of a small factory. However, per the definition provided in Jain et al. (2015), the factory level model should also include functions other than production, and hence it is being referred to as the cell level. The definitions of levels used in this paper are summarized in Table 1. The target standard for reading in cell level configuration data is the Core Manufacturing Simulation Data (CMSD) standard from Simulation Interoperability Standards Organization (SISO 2012). Custom files are being used for machine information pending identification of an appropriate standard. The process level behavior information is currently embodied in the model components themselves while the operation information is provided using STEP-NC standard. Again, standards appropriate for representing process level information will be identified in near future. The output data stream with detailed time-stamped information of events such as part and tool movements at the machine level is being generated based on MTConnect standard (MTConnect 2012).

Table 1: Resolution levels modeled and their characteristics.

Level	*ISA-95/ISA-88 objects/ term*	*Modeled phenomena*	*Key entity*	*Time-scales modeled*	*Simulation paradigm*
Cell	Work Center	Production of batches of parts through a multi-step process plan	Part batch	Minutes to hours	DES
Machine	Work Unit	Production of parts at a single machine including batch and part setup and ejection	Part	Seconds to minutes	DES/ ABS
Process	Actual Production Process	Physics of the transformation processes such as turning and milling for individual parts	Part features	Milliseconds to seconds	Continuous based on equations

Levels above the cell level such as department/line and factory (corresponding to area and site respectively in ISA-95) are anticipated to be represented using DES. Levels higher than factory such as enterprise and supply chain may be represented using system dynamics simulation. The higher levels will be addressed in future.

3.2 Approach for Current Implementation

Automated model generation at factory level has been done in the past as mentioned earlier. Past implementations modeled each process plan step using generic representations such as serial or batch processes with defined times for set-up and operations. The generated models were generally presented as logic networks without the benefit of an animated layout based on the scenario. While such as approach can work for factory level models, the generic representation will not work at the process level given the wide variety of manufacturing processes and their unique parameters. For example, material removal operations require modeling of cutting forces based on cutting speed and feed rate, while an injection molding process may require modeling of pressure on the molten material and cooling rates. The physics at the process level is thus widely different and will require a unique meta-model or equations to represent them. One would thus need a library of machine level models with associated process level models as suggested by Matsuda and Kimura (2015). Selected machine level models can be included in the virtual factory model as needed based on the data from the real factory to be modeled.

The proposed automatic generation approach can broadly be described as below:

- Read the manufacturing system configuration data via an interface supporting in a standard format
- Read in machine parameter and process level data
- Assemble the factory or cell level logic network based on the input process plans data
- Link the factory or cell level logic network to individual machine and corresponding process models using the corresponding models available in the library
- Render the layout of the facility based on the information from the configuration data with links to the logic network
- Execute the model with selected parameters such as resolution level and output formats selected by users via run-time interaction

The current approach is summarized in Figure 1. At present this approach makes a number of assumptions on the material flow control. For example, dispatching and priority rules at different cells are set by default, and a common queue is used for all machines in the cell. These will be addressed in future.

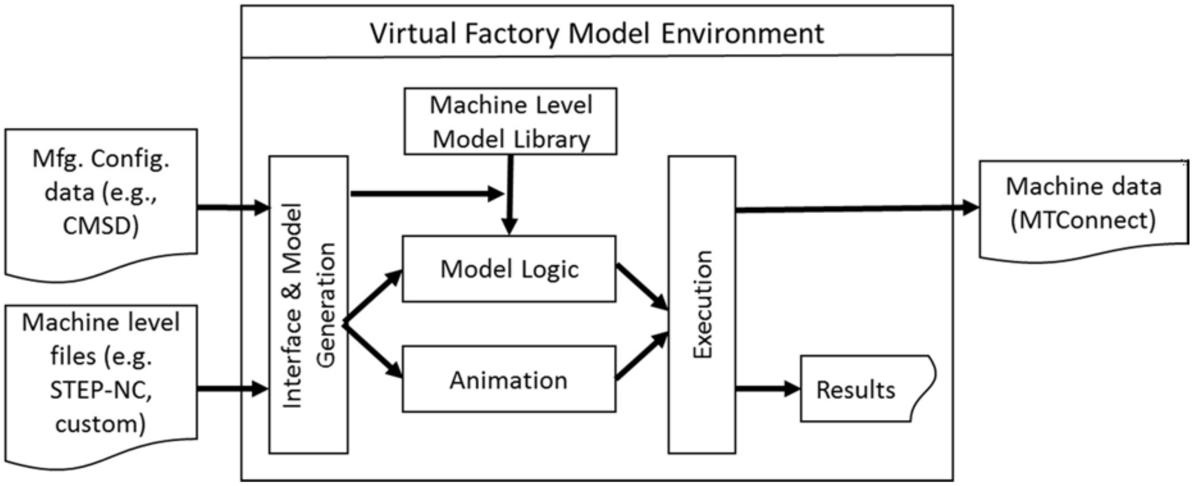

Figure 1: Current approach for automated model generation.

The targeted use of multiple paradigms to implement the multi-resolution models prompted the use of AnyLogic software to avoid the complexity of distributed simulation at this stage as was discussed in Jain et al. (2015). Like many other current commercial simulation software packages, AnyLogic appears to emphasize interactive model building via the graphical user interface (GUI). The coding of automated generation hence turned out to be challenging, and at times some aspects of automated generation appeared impossible to implement. The lessons learned from the efforts to implement the automatic generation capabilities are detailed out in the next section to aid other research teams who may be attempting similar implementations.

4 IMPLEMENTATION

In this section, an example of a machine shop is used to illustrate the implementation process for the corresponding virtual factory. The first sub-section presents the example followed by discussion in subsequent sub-sections of an algorithm to automatically generate a virtual factory model primarily from a CMSD file. The algorithm is capable of building models representing machine shop scenarios described via CMSD files as long as each cell is not composed of more than 5 machines. The current limitation of 5 machines per cell is due to the use of AnyLogic *SelectOutput5* block in the auto generation code. This limitation may be removed using a hierarchy of *SelectOutput* and *SelectOutput5* blocks in the generated code based on the number of machines per cell.

4.1 Model Example

Figure 2 shows the logic network in an AnyLogic model that represents a machine shop. This machine shop is composed of a turning machine cell and a milling machine cell. The parts processed in this workshop have two operations. They are machined first in the turning cell and then in the milling cell. The parts are processed in batches of varying sizes.

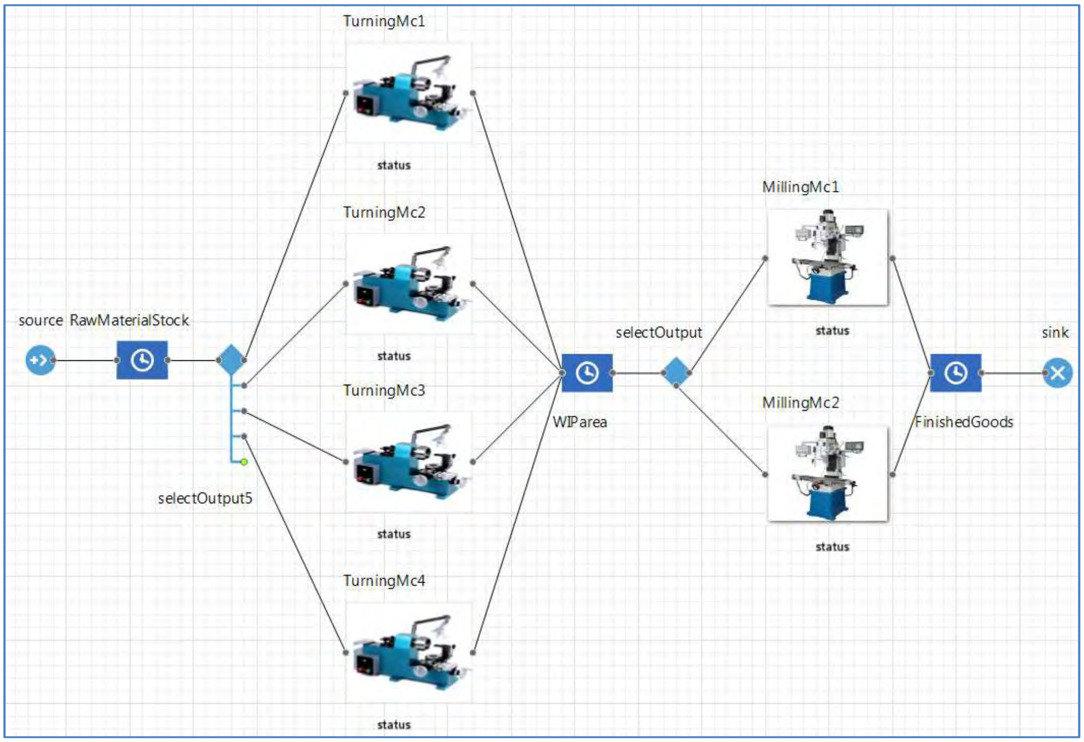

Figure 2: Example of a machine shop.

The logic network representation of the workshop starts with a *Source* block to model arrivals of batches. Upon arrival the batches wait in the raw material storage area represented by a *Delay* block. This *Delay* is used as a queue and holds the batch until a turning machine is available. Since the turning machine cell is composed of four machines, a *SelectOutput5* block (an output selector with 5 conditional outputs) is used to decide to which machine the batch should go. After the turning machine cell, the batch moves to the work in progress area represented by another *Delay* block, where it stays until a milling machine is available. Again, since the milling machine cell is composed of 2 machines, a *SelectOutput* block (an output selector with two conditional outputs) is used to select the available machine. After completion of the second operation, the batch moves to the finished goods area where batches are waiting for pick up. This area is represented with yet another *Delay* block. The shipping of batches from the workshop to customers is represented by batches leaving the model via a *Sink* block.

4.2 Data Input Interface

The interface has the capability to parse a CMSD file that describes the machine shop. The interface collects the required information to build the model such as the part characteristics, information on the cells, and specifications of the machines that compose each cell. Since CMSD is Extensible Markup Language (XML) based, parsing a CMSD file requires the ability to go through XML documents. A Java parser has been developed to go through a CMSD file for the machine shop and collect the information required to automatically build the corresponding virtual factory model.

4.3 Machine Level Model Library

Per the virtual factory definition used in this work the model should allow execution at the factory level as represented in Figure 2, at the individual machine level, and at the associated process level. The representation at machine level requires provision of a machine model library that allows selection and use of the machine level model corresponding to that defined in the data. At this prototype stage, only two machine level models are included in the library that represent turning machines and milling machines. These objects can be included in the logic network. Jain et al. (2015) introduced a turning machine model. Lechevalier et al. (2015) described a milling machine model that was developed following the same approach as the turning model. These two models are the first entries in the library. The machine level models follow the simple statechart represented in Figure 3.

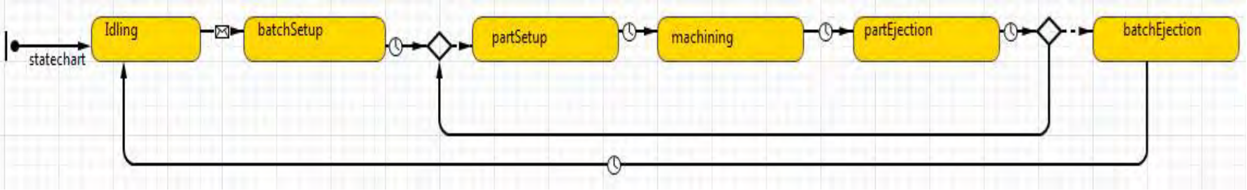

Figure 3: Statechart of the machine level models.

The transition times for batch and part setups and ejection are populated via the CMSD file. The machining time per part can be populated via the CMSD file for modeling at the machine level or generated via the associated process level model. Separate files including the STEP-NC instruction file and machine specification file with data such as feed rate, cutting speed, and depth of cut, are used by the process level model where algorithms simulate the physics of the process (turning or milling) and compute machining time and power consumption. The time generated at the process level can be used to update the part machining time at the machine level, and in turn the part times at machine level can be used to update the batch processing time at the factory level. The model library can be easily extended with additional models to include additional types of machines and associated processes.

4.4 Auto Generation of Logic Network

The logic network to represent the machine shop is automatically generated based on the data parsed from the CMSD file. First, the algorithm builds and customizes the batch *Source* block. Once the source is built, the algorithms build the last *Delay* and the *Sink* of the model. The last *Delay* represent the finished goods area where a batch is waiting for pick up. The sink represents the end of the flow in the factory level model.

After building these objects the algorithm develops representations of the cells of the machine workshop. Cells can be composed of different numbers of machines. This implementation offers the capability to build a cell having up to five machines as mentioned earlier. Depending on the number of machines, the algorithm chooses the appropriate components in the process modeling library available in AnyLogic. The cell is represented with a selected set of objects including a *Delay* block, an output selector (that can be *SelectOutput* or *SelectOutput5* block), and machine level agents from the library to represent the machines in the cell. First, the algorithm creates the appropriate number of machines using the available agent-based models. The agent based models are initially created with no parameters. The parameters are populated later at the beginning of the simulation execution. This was done to work around the AnyLogic procedure that initialized all the automatically generated agents at the start of simulation execution. The procedure does not appear to affect the agents generated through the GUI in the same manner. This is another example of code manipulation required for automatic generation of the model.

Once the machines are generated, the algorithm builds associated *Delay* blocks. The *Delay* is customized to hold the batch as long as all machines of the cell are busy. As soon as one machine is available, the *Delay* block lets a batch go. If a cell is composed of one or two machine, the algorithm builds a *SelectOutput* which provides two conditional paths as outputs. The condition to decide which way the batch should go consists of checking if the first machine of the cell is available. If not, the batch goes to the second machine. If both machines are busy, the batch stays in the related *Delay* block. If the cell is composed of more than two machines, the algorithm builds a *SelectOutput5* that provides five conditional paths as outputs. After the implementation of objects and the required customization to include the conditions, the *Delay* and the output selector within each cell model are connected to each other. The algorithm also connects the output selector and the machines that compose the cell.

Next, the cell models are connected as required by the part flows. The algorithm collects the parts and associated process plans from the CMSD file to determine the connections between the cells. It establishes connections between the batch *Source* block and the *Delay* block (if the cell is the first one in the flow) or the machines from the previous cell and the *Delay* block (if the cell is not the first one). If this cell is the last cell of the line, the output paths of the machines are connected to the last *Delay* block that represents the finished goods area. The approach allows for simple configurations currently. The algorithm will be enhanced iteratively in the future to support more complex logic networks.

4.5 Auto Generation of Layout

The algorithm builds a 2D representation of the machine shop using the information in the layout area of the CMSD file. The algorithm defines shapes to represent the different areas in the factory. The algorithm creates the shapes based on the location and size of the area defined in the CMSD file and adds them to the simulation animation window. To allow the real time visualization of the model execution, the algorithm associates every shape with the corresponding object defined during the automatic generation of the model. Figure 4 shows the rendered layout corresponding to the logic network shown in Figure 2.

The algorithm also creates buttons in the layout to allow going to the next level of resolution. The next level of resolution for the machines is the statechart of the agent based representation of the machine. The next level of resolution for storage areas is the corresponding *Delay* block. Using these buttons, a user can examine the animation at the next resolution level and check the current state of a machine. The addition of this capability in AnyLogic required some custom coding. AnyLogic runs an internal routine to display the blocks that have been built using the GUI. This routine cannot be overwritten to provide capabilities to display agents and blocks that have been automatically built. Since the software does allow overwriting the code of the button function, the button function code was overwritten to specify the agent that should be displayed using each button.

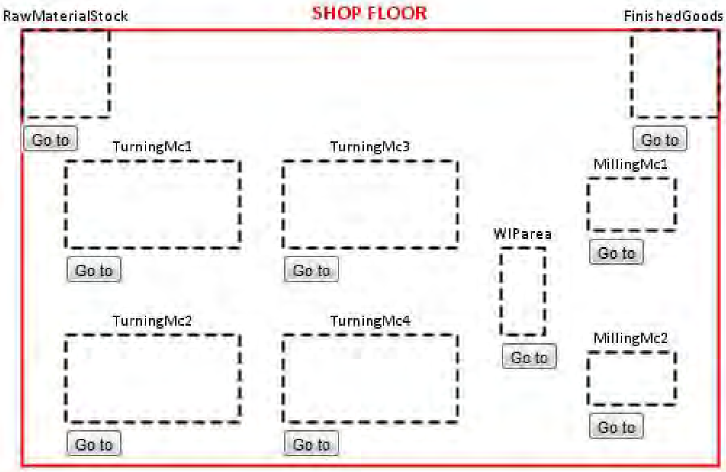

Figure 4: 2D representation of the machine shop.

4.6 Data Output Interface

The execution of the machine shop model with the machine level models activated generates MTConnect data streams on demand. MTConnect is an XML-based standard to represent monitoring data collected from machines. The MTConnect files contain information such as the tool position and the power demand values for every 100 milliseconds. Users can use a provided checkbox to generate the MTConnect files at desired times or over desired durations. An MTConnect file can be generated for each part processed in each machine.

4.7 Execution of the algorithm

The execution of the model runs the algorithm to automatically build the logic network, the required machine level models, and the 2D representation of the scenario. Once the model and the representation of the scenario are built, the *source* block starts creating the batches. The generation of blocks, layout, and the animation can be watched in real time. The animation can be visualized using the layout or the logic network. Figure 5 shows a screenshot of the execution of the model using the layout. In the figure, each batch shown as a rectangle has ten parts represented by colored dots. As a part is processed it turns from red to blue. As may be noticed from the figure, a machine can have only one batch at any time while the storage areas can have multiple batches. The checkbox for initiating generation of MTConnect data can be seen on the top left of the figure.

5 VALIDATION CHALLENGES

Simulation models need to be validated before their results can be used. The challenges of validation of simulation models have been discussed extensively in literature. A large part of the validation literature has focused on quantifying the intrinsic uncertainty of the model, that is, uncertainty based on the variabilities defined for the input data for modeled stochastic phenomena. Barton and Schruben (1993) pointed out the need to consider the impact of extrinsic uncertainty, introduced due to the inability to capture the input data variabilities correctly. There has been a lot of work to model input uncertainty since the 1993 paper. Barton (2012) discussed the leading methods for quantifying input uncertainty.

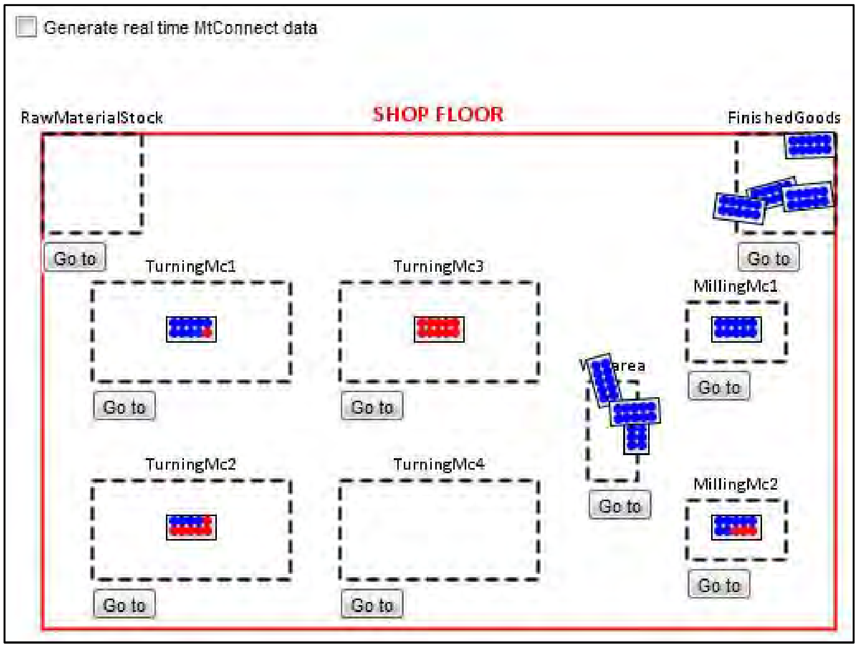

Figure 5: Animation of the model using the factory layout.

Each simulation model has to be validated carefully including the impact of intrinsic and extrinsic uncertainties. All the physics-based process models have to be validated against real machine processes and their ranges of applicability defined. The current two process level models for turning and milling have been validated to varying degrees as defined in Jain et al. (2015) and Lechevalier et al. (2015) respectively. Similarly, machine level and factory level models have to be validated and their ranges of applicability defined. The multi-resolution model results in stacking of uncertainties across the multiple levels of resolution. The outputs of the process level model have uncertainties. The output of the process level model becomes one of the inputs for the machine level model. The outputs of machine level model thus include uncertainties emanating from all its inputs including the machining time provided by the process level model, and the variabilities defined in the batch and part set up and ejection times. This continues further up to the outputs of the cell level model that include uncertainties due to operation times from the machine level coming as inputs. The impact of stacking of uncertainties needs to be understood and quantified before the virtual factory and other multi-resolution models can be used to support decision making in industry.

6 CONCLUSION

The paper described the next step in a long term effort to build virtual factories corresponding to real factories. Previous reports of the work described the ad hoc implementation of a multi-resolution model of a virtual factory corresponding to a small machining shop. This step of the work reported on the automatic generation of the virtual factory model for the machining shop using inputs based on standards. The automated generation capability needs to be enhanced to handle larger more complex scenarios. When fully implemented the automated generation capability should help increase the use of virtual factory simulations and in turn help the move toward realization of initiatives such as smart manufacturing and Industrie 4.0.

Future work includes enhancement in the model, inputs, and outputs. The model and the interface need to be enhanced for handling more complex scenarios. The machine level model library needs to grow to include more manufacturing machines and processes. The capability to read in the current status possibly based on ISA-95 standard from a warm start to the model is being contemplated. Additional capabilities for generating output data particularly at the cell and factory level possibly based on ISA-95 standard are planned. Once the concept is well developed and issues understood in the single software environment provided by AnyLogic, integration with component models in other software via a distributed simulation mechanism may also be explored.

DISCLAIMER

No approval or endorsement of any commercial product by the National Institute of Standards and Technology (NIST) is intended or implied. Certain commercial software systems are identified in this paper to facilitate understanding. Such identification does not imply that these software systems are necessarily the best available for the purpose.

ACKNOWLEDGMENTS

Sanjay Jain's work on this effort has been sponsored under the cooperative agreement no. 70NANB13H158 between NIST and George Washington University. David Lechevalier's work on this effort was supported by National Institute of Standards and Technology's Foreign Guest Researcher Program. The work described was funded by the United States Government and is not subject to copyright.

REFERENCES

Abduaziz, O., J. K. Cheng, R. M. Tahar, and R. Varma. 2015 "A Hybrid Simulation model for Green Logistics Assessment in Automotive Industry." Procedia Engineering 100: 960-969.

Alvandi, S., G. Bienert, W. Li, and S. Kara. 2015. "Hierarchical Modelling of Complex Material and Energy Flow in Manufacturing Systems." *Procedia CIRP* 29: 92-97.

Ayadi, M., R. C. Affonso, V. Cheutet, and M. Haddar. 2015. "Info Sim: Prototyping an Information System for Digital Factory Management." *Concurrent Engineering* 23(4): 355-364.

Barlas, P., and C. Heavey. 2016. "Automation of Input Data to Discrete Event Simulation for Manufacturing: A Review." *International Journal of Modeling, Simulation, and Scientific Computing* 7(01), 1630001, http://dx.doi.org/10.1142/S1793962316300016.

Barton, R. 2012. "Tutorial: Input Uncertainty In Output Analysis," In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, 67-78. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Barton, R. R., and L. W. Schruben. 1993. "Uniform and Bootstrap Resampling of Empirical Distributions." In *Proceedings of the 1993 Winter Simulation Conference*, edited by G. W. Evans, M. Mollaghasemi, E. C. Russell, W. E. Biles, 503-508. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Choi, S., K. Jung, and S. Do Noh. 2015. "Virtual reality applications in manufacturing industries: Past research, present findings, and future directions." *Concurrent Engineering* 23(1): 40-63.

Constantinescu, C. L., E. Francalanzab, and D. Matarazzoc. 2015. "Towards Knowledge Capturing and Innovative Human-system Interface in an Open-source Factory Modelling and Simulation Environment." *Procedia CIRP* 33: 23–28.

Gregor, M., J. Hercko, and P. Grznar. 2015. "The factory of the future production system research." In Proceedings of the 21st International Conference on Automation and Computing (ICAC), 1-6. IEEE.

Jain, S., and G. Shao. 2014. "Virtual Factory Revisited for Manufacturing Data Analytics." In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 887-898. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Jain, S., D. Lechevalier, J. Woo, and S.-J. Shin. 2015. "Towards a Virtual Factory Prototype." In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon,

T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2207-2218. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kádár, B., Terkaj, W., & Sacco, M. 2013. "Semantic Virtual Factory Supporting Interoperable Modelling and Evaluation of Production Systems." *CIRP Annals-Manufacturing Technology* 62(1): 443-446.

Lechevalier, D., S.-J. Shin, J. Woo, S. Rachuri, and S. Foufou. 2015 "A Virtual Milling Machine Model to Generate Machine-Monitoring Data for Predictive Analytics." At Product Lifecycle Management 2015. Doha, Qatar.

Mario, H., P. Tobias, and O. Boris. 2015. "Design Principles for Industrie 4.0 Scenarios: A Literature Review." Working Paper No. 01 / 2015, Technische Universität Dortmund, Dortmund, Germany.

Matsuda, M., and F. Kimura. 2015. "Usage of a digital eco-factory for sustainable manufacturing." *CIRP Journal of Manufacturing Science and Technology* 9: 97-106.

MESA. 2010. *Building a Manufacturing Transformation Strategy with ISA-95 Methods*. White paper #38. Chandler, AZ, USA: MESA International.

MTConnect. 2012. *Part 1-Overview and protocol, Version 1.2.0*. MTConnect Institute.

Oyekan, J., W. Hutabarat, C. Turner, A. Tiwari, N. Prajapat, N. Ince, X.-P. Gan, and T. Waller. 2015. "A 3D Immersive Discrete Event Simulator for Enabling Prototyping of Factory Layouts." *Procedia CIRP* 38: 63-67.

Rabelo, L., A. T. Sarmiento, M. Helal, and A. Jones. 2015. "Supply Chain and Hybrid Simulation in the Hierarchical Enterprise." *International Journal of Computer Integrated Manufacturing* 28(5): 488-500.

Schönemann, M., C. Schmidt, C. Herrmann, and S. Thiede. 2016. "Multi-level Modeling and Simulation of Manufacturing Systems for Lightweight Automotive Components." *Procedia CIRP* 41: 1049-1054.

SISO 2012. *SISO-STD-008-01-2012: Standard for Core Manufacturing Simulation Data – XML Representation*. Simulation Interoperability Standards Organization, Orlando, FL.

SMLC 2012. *SMLC Forum: Priorities, Infrastructure, and Collaboration for Implementation of Smart Manufacturing: Workshop Summary Report*. Smart Manufacturing Leadership Coalition (SMLC), Washington, DC, USA. Oct. 2-3. Accessed April 6, 2014: https://smartmanufacturingcoalition.org/sites/default/files/smlc_forum_report_vf_0.pdf.

Terkaj, W., T. Tolio, and M. Urgo. 2015. "A Virtual Factory Approach for In Situ Simulation to Support Production and Maintenance Planning." *CIRP Annals-Manufacturing Technology* 64(1): 451-454.

AUTHOR BIOGRAPHIES

SANJAY JAIN is an Associate Professor in the Department of Decision Sciences, School of Business at the George Washington University. Before moving to academia, he accumulated over a dozen years of industrial R&D and consulting experience working at Accenture in Reston, VA, USA, Singapore Institute of Manufacturing Technology, Singapore and General Motors North American Operations Technical Center in Warren, MI, USA. He served as an associate editor of the *International Journal of Simulation and Process Modelling* for 2004-2014 and continues to serve as a member of the editorial boards of the *International Journal of Industrial Engineering* and *Smart and Sustainable Manufacturing Systems*. His research interests are in application of modeling and simulation to complex scenarios including smart manufacturing systems and project management. His email address is jain@email.gwu.edu.

DAVID LECHEVALIER is an international guest associate in NIST's Systems Integration Division of the Engineering Laboratory. He is a PhD student in the laboratory Le2i at the University of Burgundy. His current research topics include advanced analytics and modeling for Smart Manufacturing. He received a computer science Master's degree in Database and Artificial Intelligence from University of Burgundy in 2012. His email address is david_lechevalier@etu.u-bourgogne.fr.