Proceedings of ASME 2010 International Mechanical Engineering Congress & Exposition
IMECE 2010
Nov 12–18, 2010, Vancouver, British Columbia, Canada

IMECE2010–39147

PUSH–BUTTON DISCRETE EVENT SIMULATION FOR ANALYSIS OF FACTORY FLOOR OPERATIONS

J. Michaloski, B. Raverdy, B.E. Lee, and F. Proctor
National Institute of Standards
and Technology
Gaithersburg, Maryland

S. Venkatesh
Boeing Company
Seattle, Washington

N. Bengtsson
Production Modeling Corporation
Gothenburg, Sweden

A. Skoogh
Chalmers University of Technology
Gothenburg, Sweden

ABSTRACT

In manufacturing, Discrete Event Simulation (DES) can be effectively used to model production and provide sustainability analysis of equipment and system operation by measuring throughput, capacity, and bottlenecks. DES allows analysis under different scenarios and conditions that can then be used to forecast more optimal system performance. For discrete parts production, DES is rarely used because of the difficulty in attaining timely and accurate statistical modeling of the equipment and process operation. In this paper we look at the use of the plant floor interoperability standard, MTConnect, as a means to improve the access to machine tool data. Given data access, we develop a Finite State Model in order to streamline Machine Tool and DES integration. A case study of a prototype Capacity Planning system using DES as a modeling back–end will be described to help in understanding resource allocation of shop floor machines to a batch of aerospace parts. Direct machine tool statistical parameterization into a Web–based DES leads to "Push–button" automated simulations.

Keywords

MTConnect, device, machine tool, Kaizen, Lean Manufacturing, Computerized Numerical Control, open–architecture

Nomenclature

CNC	Computer Numerical Control
DES	Discrete Event Simulation
FSM	Finite State Machine
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
KPI	Key Performance Indicators
MTC	MT Connect
MTBF	Mean Time between Failure
MTTR	Mean Time to Repair
OEM	Original Equipment Manufacturers
SDK	Software Development Kit
XML	eXtensible Markup Language

INTRODUCTION

Simulation is a powerful analysis tool. Simulation can analyze the performance of a system, help understand proposed changes to the system, and assist in design decisions about a system. Simulation is especially popular where the complexity of real world makes analytical closed–form solutions impossible. Simulation that models a system as a chronological sequence of discrete events is known as Discrete Event Simulation (DES). DES is useful for modeling and analysis of manufacturing systems, such as, machinery operation, shop workflow and scheduling, and production lines.

In manufacturing, DES is a computer model of real or proposed production. DES is particularly effective because it is a controlled environment that assists in the study of the performance of the production system under different conditions. With DES, the model can be used to forecast throughput, bottlenecks and other performance metrics. Adjustments to parameters can be run through time sequences to predict the impact of potential changes. These prediction models can be developed without disturbing daily operations. DES is useful in decision–making for new product lines, changes of existing factory processes and sustainability impact of manufacturing operations.

World class manufacturing aims to increase throughput while simultaneously reducing inventory and operating costs. To perform detailed production analysis, manufacturers are turning to simulation as it is well suited to handle the complexity of large scale interaction of machines and processes. The level of simulation granularity is an important consideration as many simulations attempt to exactly replicate the machine being simulated. Of concern in this paper, is the process level where part cycle times, breakdowns, and inactivity are of interest.

Throughput, utilization, and cycle time are considered Key Performance Indicators (KPI) for DES manufacturing analysis but are not intrinsic to the machining operations. Machining inputs part programs, tool, fixtures, setup information and produces parts. Machining data model includes positions, velocity, program line numbers, etc. Since the machining static model is not sufficient to satisfy the DES KPI requirements directly, a mapping of the machining CNC model into the DES model must be done.

However, data acquisition for machine tools has been notoriously difficult. Historically, most machine tool Computer Numerical Controllers (CNC) are closed proprietary platforms. Closed platforms either do not readily provide access to the raw data or provide an expensive proprietary solution, so that characterizing the machining process is costly if not impossible. MTConnect is a new standard for data exchange on the manufacturing floor that helps in satisfying many of the DES data acquisition requirements. The open MTConnect standard for shop–floor integration should help end–users collect raw data and assist in understanding the nuances of their process. In this regard, MTConnect allows for raw data acquisition that can be used for continuous process improvements [1]. Formalization of raw machine data acquisition into a DES–ready format is the goal of this paper.

This paper will study the automation of DES data entry from shop–floor CNC machining data. Section 2 will give a formal overview of DES event simulation data modeling, finite state machine CNC logic, and describe the transformation from machining data in DES manufacturing KPI statistics. Section 3 will investigate the implementation details of automating the DES data processing and the issues involved. Section 4 will do a DES case study for Capacity Planning machine tools in a production workcell at a Boeing plant. The automation of machine tool data processing using MTConnect and some extensions will be discussed. Finally, a discussion on the results and future directions will be given.

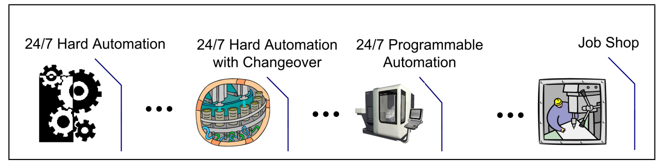

FIGURE 1: Spectrum of DES Application in Manufacturing

MAPPING MACHINING MODEL TO DES MODEL

DES models have a variety of application domains: analysis, training, experimentation, decision support, and design or system optimization. Within this realm, the direct applicability of DES to manufacturing depends on the deterministic nature of the system being modeled. Figure 1 shows the spectrum of manufacturing determinism that has a great impact on the ease of use and subsequent utility of DES. DES is especially useful when modeling deterministic hard automation systems, such as packaging lines, where the operation is virtually continuous, and there are series of machines connected by buffers, and there is relatively little variability in operation. Throughput, bottlenecks and failures are the basic measures of success or failure in this case and can be easily characterized by statistical models of the automation process. From this extreme form of hard automation, the determinism scale lessens into automation with changeover, to workflow between machines to the relatively random scenario of workflow in a job shop. The more variability in the system being modeled, the more difficult to garner accurate statistics since the range of potential data points would large, and statistical distribution would reflect this variability.

The type of manufacturing that will be studied for our DES analysis will be based on a quasi job shop facility – workflow that consists of jobs assigned to one of many machines, with little or no connectivity of process flow through between machines. We will consider the case of several machine tools operating in a workcell on similar parts, with similar tooling, cycle times that handle parts in a first–in, first–out sequence.

Hidergott et al [2] define a DES model simulation as the generation of a sequence $X(k)$ states, with initial state $X(0) = x_0$, in such a way that state $X(k+1)$ is obtained from $X(k)$ by a measurable state–transition mapping h that consumes a vector of mutually independent random input variable $Y(k) = (Y_1(k),..,Y_m(k))$, with $Y(k)$ independent of everything else, where m presents the number of input random variables needed in each transition;

$$X(k+1) = h(X(k), Y(k)), k \geq 0. \quad (1)$$

The goal of input to DES is to find unbiased estimators for the random variables $Y_i(k)$. Fortunately, the system dynamics of DES models for a machine tool is driven by relatively simple input distributions. For example, the machining depends on the timed intervals in various control states. Exponential, gamma, or Weibull distributions are used for modeling time variables, normal distributions for modeling noise, and Poisson distributions for modeling occurrence of certain events. The primary DES machining random variables of interest in this paper to be estimated by sample data are idle time, part cycle times and disturbance data (i.e., system breakdowns). These state random variables are primarily characterized by time, but can have secondary data elements such as energy consumed, or process characteristics (e.g., machine feeds and speed) associated to simulation time within the given state.

This DES model of the machine tool behavior can be equivalently described as a Finite State Machine (FSM), which is a more common paradigm to model machine CNC logic. A FSM model is a set of finite states together with a set of state transitions, where the machine controller is in one of a finite set of the possible states, known as the state–space, at any given time, and is defined by

$$G = (\sum, Q, \delta, q_0) \quad (2)$$

Here $\sum$ is the set of states, q_0 is the initial state, δ is a finite set of output symbols, and $\delta : \sum \times Q \rightarrow Q$ is the state transition function. For a machining process, G defines a function that starts in the state q_0 and generates a sequence of events, i.e. state transitions, subject to the range of transitions permitted by the function δ. Each event results in an output from the set $\sum$.

We will attempt to formalize the Machining component based on enumerating the states required for mapping to the DES KPI knowledge. Inside the Machining Process box shown in Figure 2, the basic FSM formalism is given as these states: OFF, DOWN, IDLE, MISC, and MACHINING.

OFF refers to the machine power being off due to inactivity.

DOWN refers to the machine being idle/off due to an alarm or fault.

IDLE refers to a non–machining condition, where the material removal process is either in manual model during setup and takedown or in automatic mode, but not paused.

MACHINING refers to the state where the material removal process is occurring.

MISC refers to CNC maintenance and other intermittent activities.

Adopting hierarchical state machine terminology, MISC is a superstate that contains nested substates for tool changes, lubrication cycles and other miscellaneous intermittent activities, and is unnecessary for our expected DES analysis. For our study, MISC will be consolidated into the IDLE state, for simplicity.

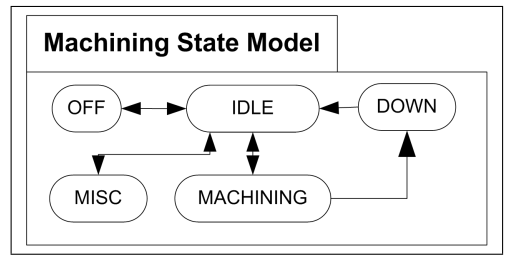

FIGURE 2: Machining State Model Overview

It is necessary to transform the machine state data into DES KPI event data. The main machining data requirement that is typically not part of raw machine data is the Part Count data item that signals cycle complete and triggers a cycle time event. There are other mechanisms to detect a completed program, for example, monitoring program blocks for an "M30", but have various flaws that draw reliability into question.

Table 1 gives an overview of the transformation from machining data into DES–ready events. Table 1 summarizes the logic used to calculate the DES model parameters, where $MTC_{item}\ E(x)$ means the reading of the MTConnect data value "item" and $t = T(a,b)$ means the elapsed time t from the beginning of event "a" until the occurrence of event "b" and $t = T(a)$ means time t spent in state "a".

The MTConnect Data row details the raw data available from the CNC machine tool. Using prevailing technology and providing free software development kits minimizes technical and economic barriers to MTConnect adoption. The basic MTConnect specification provides for: power (on/off), mode (automatic/manual), execution state (running/paused), program, position, feeds, speeds, alarms, and some tooling. For our work, the data item part count was added to the MTConnect data collection to assist in determining cycle time.

The KPI Parameters rows show the transformation of state data into DES events. This transformation entails accumulating the time spent in a current state until a state change occurs. Upon the state change, a new DES event is archived containing the specific event (e.g., cycle time), and the duration spent in that state. The Computed KPI Parameters rows create new DES events based on an existing event sequence.

TABLE 1: Mapping Machine State Data into DES KPI Data

MTConnect Data	Parameters
Machining Data	Timestamp(ts), Machine, Power, Mode, Execution, Program, Line, Sload, Xload, Yload, Zload, Aload, Bload, Cload, Toolnum, RPM, Alarm, AlarmState, AlarmSeverity, PartCount, Feedrate
KPI Parameters	**Data Mapping**
Cycle Time	$MTC_{mode} = Auto\ and\ MTC_{rpm} > 0\, and\, MTC_{feed} > 0$
Setup Time	$MTC_{program}(t) \neq MTC_{program}(t--1) \rightarrow T(MTC_{mode} = Manual)$ excluding palete shuttle program
Machining Time	Cycle Time
Off Time	$MTC_{power} = Off$
Down Time	$MTC_{alarm} = active$
Idle Time	$T(MTC_{execution} = Paused\, or\, MTC_{mode} = Manual)$
Misc Time	Not addressed in this analysis
Computed KPI Parameters	**Mapping**
Time Between Failure (TBF)	$t = T(MTC_{alarm} = active, MTC_{alarm} \neq active)$
Time To Repair (TTR)	Equivalent to Down Time

"PUSH–BUTTON" DES

"Push–button" automated DES would embed and streamline DES functionality within the machine tool providing DES–ready event data streams, statistical fitting to KPI, thus minimizing the time–consuming manual data operations. In this case, DES could be used continually and cost–effectively in order to evaluate machine operations. The major drawback to implementing "Push–button" DES is the time–consuming, error–prone, and costly aspect of data acquisition, cleaning and coding of the data into the DES [3, 4]. It is estimated that up to 31 % of the amount of time spent on building a DES model is spent on data collection [5]. Push–button DES requires automating the data collection as the statistical characterization of the machining using the MTConnect data.

Due to the costly nature of the DES data processing, the goal of automating the data processing for DES is undeniably critical and considerable effort has been applied in this area. NIST is collaborating with Chalmers University on the GDM–Tool [6], which is a middleware approach whose goal is to provide a completely generic solution that would allow any DES system to connect to any source of raw data and then use a computer–assisted approach to extract DES–ready data. Since the focus of our data sampling is constrained to the machining environment, a completely generic solution is beyond the scope of the project. A more directed approach will be taken by developing automated data processing that is restricted to DES simulations of machining operations.

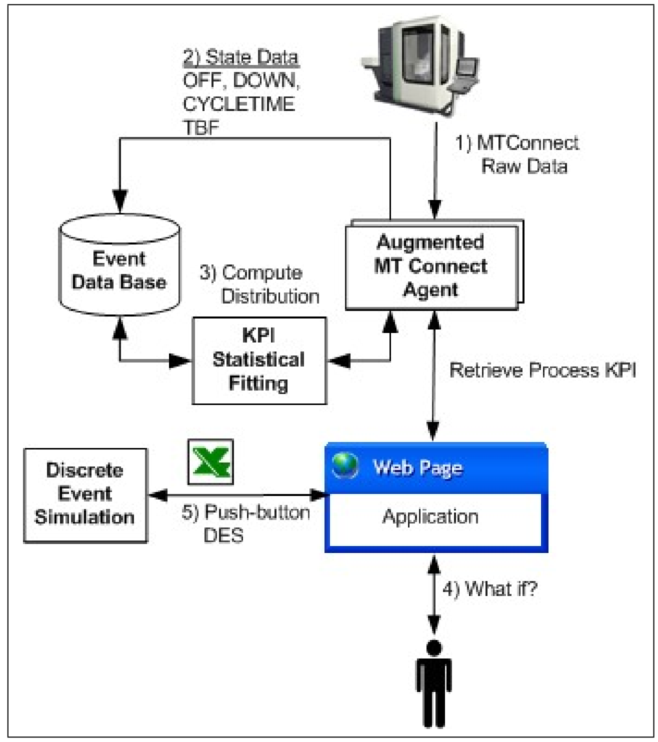

FIGURE 3: DES Push–button Implementation

Figure 3 shows the architecture for implementing "Push–button" DES. Step 1 concerns automating the raw data collection. The prototype data acquisition is based on using MTConnect technology [7–9], but other data acquisition technologies are used in manufacturing and could potentially have provided a similar solution [10–12]. MTConnect allows standard, automated data acquisition on the shop floor that is lightweight and flexible. MTConnect is a specification based upon prevalent Web technology including Extensible Markup Language (XML) [13] and Hypertext Transport Protocol (HTTP) [14].

Step 2 transforms machine state data into DES event data, which is based on the mappings defined in Table 1. The implementation was an augmented MTConnect Agent, based on the turnkey Microsoft C# .Net implementation of an MTConnect Agent [15]. Having successfully used the SDK, we felt that the MTConnect Agent could easily be augmented to handle more complex processing, while maintaining backward compatibility. Within the MTConnect agent, the raw data was filtered by machine states and then logged as DES events. DES naturally models the time in each state as a statistical distribution, and it is more efficient to do these statistical calculations on–machine as opposed to post–processing in Excel Macros [16] or through the use of data cleansing tools [5].

Step 3 computes DES KPI statistical data from the DES event data. DES systems use statistical distributions as input, so that statistical distribution fitting was applied to the DES event data. In previous work, the statistical distribution fitting was performed automatically on the event data by the DES statistical software [1]. In this implementation, some limited statistical software was embedded to do the statistical distribution fitting, but was aided by à priori understanding of the data and machine behavior. The decision to fit sample data to a smaller set of statistical lifetime distributions was deemed sufficient for a feasibility phase.

Step 4 involves using DES for various “What–if” applications. NIST and Boeing have been exploring various applications for exploiting the ease of integration of MTConnect data and DES modeling. Previously, shop floor MTConnect data was collected in order to study sustainability [16]. The sustainability analysis depended on manually–oriented DES input approach, and at that time, it was clearly evident that for DES to be used in a factory, it would need to be a more automated or “Push–button” technology. This overriding requirement prompted us to investigate more automated approaches of data integration within the DES software systems.

Step 5 is part of an ongoing demonstration of “Push–button” technology to test remote DES feasibility. Web–based software was developed that remotely communicates to the DES system. First, the Web software reads KPI parameters from the machines (such as cycle times and disturbance data). Next, KPI parameters are loaded into a Excel file across the internet. The DES simulation is remotely initiated using Excel interface. Finally, the DES simulation results are read back from the Excel file and displayed on the Web page.

CASE STUDY – CAPACITY PLANNING

The concept of “Push–button” DES as a way to perform shop–floor Capacity Planning was a compelling application for Boeing. Currently, production knowledge is often only gathered at a higher level of operation. Workorders enter the shop floor and then overall performance is measured upon completion. Typically, this shop floor knowledge may be augmented by manual signoffs at various stages and visual observation and manual documenting of the process. Intermediary analysis of the process steps and costs involved are then generally estimated. This data is entered into spreadsheets by industrial engineers who develop schedules and expected workflow based on a spreadsheet solution.

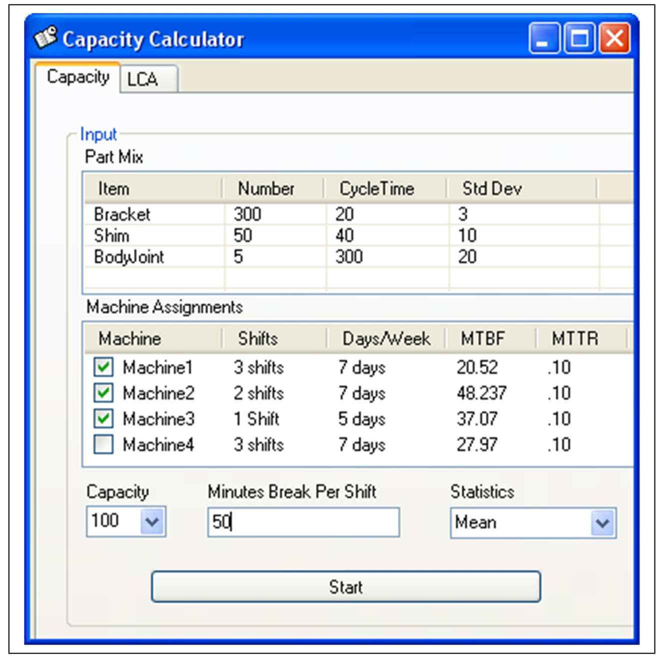

FIGURE 4: Capacity Planning User Interface

The Capacity Planner is used as a decision aide in determining the time required to satisfy a desired part throughput given a set of machine tools running at a given capacity. From a production perspective, Capacity Planning using actual shop–floor data could make scheduling for industrial engineers easier and more accurate. Better production flow could anticipate bottlenecks and adjust scheduling to prevent shortages or missed deadlines. Should the Capacity Planning systems determine beforehand that there are insufficient personnel and/or machine resources, it can react accordingly.

Workflow and throughput projections would be based on real factory floor data. For our initial Capacity Analysis, we concentrated on understanding the machining process within an integrated Workcell at Boeing that is primarily dedicated to making aluminum plane shims, brackets and body joints. The Workcell operates on batch lots of aluminum parts with part runs ranging from one shim to hundreds of brackets with assorted milling, drilling, facing and probing operations. Cycle times for these parts vary from twenty minutes for a bracket, to approximately five hours for a body joint. Each CNC features a high–speed spindle and other options for high–speed machining. Production volume varies, generally a little under 24/7 capacity, with most machines running 3 shifts a day.

Figure 4 shows the user interface of a prototype implementation of the Capacity Planning system. In this case, the scheduler is responsible for inputting the type and number of parts to be machined, and the desired machine capabilities: days/week, shifts/day, idle percentage, and break time/shift. The relevant KPI data is collected from the augmented MTConnect Agent for machine–specific KPI: MTTR, MTBF and for process–specific KPI data: part cycle time. The Capacity Planner stochastically picks from the part selection, calculates throughput and time, and quits when done.

The Capacity Planning methodology assumed shifts as the metric for planning production, as the machines utilization is based on staffing considerations. In order to assign machines, the total time must be considered in the context of shifts. The Capacity Planner must assign operators to run machines based on shifts (3/day or 2/day or 1/day), so that over time the unit of measure for machine availability is based on shifts.

Thus, the number of shifts is the fundamental measure of machine work. If we assume one machine is assigned to this workload, and further assume zero operator breaks, zero downtime, running at full 24/7 capacity, then the number of shifts divided by three is the number of days required to finish the work. Adding more machines will make the work go faster, but the amount of work as defined in shifts is constant. Adding idle time follows from this logic. Idle time, such as operator breaks, would need to be calculated per shift. We assume no unattended machining, so that any operator break time will be considered idle time.

The use of MTConnect, with augmented functionality, made the automation of DES event acquisition straightforward. Over the course of several months, MTConnect raw data was remotely accessed and transformed into DES events that were archived to a Microsoft Jet Database. Because the DES analysis was based on simple KPI, fault statistics only indicated a problem, not the type or nature of the problem.

We implemented two DES strategies for handling Capacity Planning. We use a stand alone closed–form solution as well as the previously described Web remote interface to an Excel front–end to the DES software. For the short–term, the closed–form solution was sufficient for understanding the capacity requirements of four machines, but as the number of machines and interaction among the equipment grows, the need for DES to handle the increasing complexity would be indispensable.

The major problem encountered in automating the Capacity Planning was the local program naming problem. When machine operators oversee a workorder, they download workorder part programs from a central repository to the CNC and then assign the program an ad–hoc local name. Thus, maintaining a tight correspondence between cycle times and the actual programs was challenging. In the part program, a header provides the actual program name and various setup details, but program source code was not accessible due to OEM MTConnect implementation issues that we hope to resolve in the future. Unfortunately, the program naming issue made the input of an expected part cycle time rely on a more empirical, quasi–manual approach, to interpreting program names. On–line streams of program, cycle times, and statistical summaries were available but in time, this manual operation should be completely automated with better access to the program header.

DISCUSSION

DES software has matured into a mainstream technology. Currently, most factories do not take advantage of DES technology because of various difficulties and obstacles associated with model development and data acquisition. On the surface, it would seem that once the DES model is in place, continuous operational analysis should be straightforward. However, this is not the case, mostly due to repetitive manual operations required for DES data entry. Our project was to determine the feasibility of "Push–button" DES, assuming an automated data stream were available.

Summarizing, a prototype automated DES event acquisition and statistical parameterization system was developed. The MTConnect data acquisition was straightforward and we were able to quickly derive the necessary event data. Statistical fitting of the events was more difficult and required developing goodness–of–fit algorithms for various statistical distributions we used.

In the future, we plan to integrate more functionality into the automated on–machine statistical analysis, such as trends, better fault data, abnormal deviations, and detection of excessive faults. We hope to report these findings as "Push–button" outcomes and only draw attention to trends that appear significant.

DISCLAIMER

REFERENCES

[1] Lee, B. E., Michaloski, J., Proctor, F., Venkatesh, S., and Bengtsson, N., 2010. "MtConnect-Based kaizen for machine tool processes". In Proceedings of the 2010 ASME International Design Engineering Technical Conferences, Proceedings of 2010 ASME International Design Engineering Technical Conferences, ASME.

[2] Heidergott, B., Vázquez-Abad, F. J., Pflug, G. C., and Farenhorst-Yuan, T., 2010. "Gradient estimation for discrete-event systems by measure-valued differentiation". *ACM Trans. Model. Comput. Simul.,* ***20***(1).

[3] Ingemansson, A., and Oscarsson, J., 2005. "Discrete-event simulation and automatic data collection improve performance in a manufacturing system". In WSC '05: Proceedings of the 37th conference on Winter simulation, pp. 1441–1445.

[4] Leemis, L. M., 2004. "Building credible input models". In WSC '04: Proceedings of the 36th conference on Winter simulation, pp. 29–40.

[5] Skoogh, A., and Johansson, B., 2008. "A methodology for input data management in discrete event simulation projects". In Proceedings of the 2008 Winter Simulation Conference.

[6] Bengtsson, N., Shao, G., Johansson, B., Lee, Y. T., Leong, S., Skoogh, A., and McLean, C., 2009. "Input data management methodology for discrete event simulation". In Proceedings of the 2009 Winter Simulation Conference.

[7] MTConnect Institute. MTConnect Standard Part 1: Protocol and Overview. `http://mtconnect.org`.

[8] MTConnect Institute. MTConnect Standard Part 2: Components and Data Items. `http://mtconnect.org`.

[9] MTConnect Institute. MTConnect Standard Part 3: Streams, Samples, and Events. `http://mtconnect.org`.

[10] International Organization for Standardization (ISO), 1990. "ISO/IEC 9506-1, industrial automation systems - manufacturing message specification - part 1: Service definition".

[11] International Organization for Standardization (ISO), 1990. "ISO/IEC 9506-1, industrial automation systems manufacturing message specification - part 2: Protocol specification".

[12] OPC Foundation, 1998. OPC data access 2.0 specification.

[13] The World Wide Web Consortium, 2006. *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. www.w3.org/XML.

[14] The Internet Society, 1999. *Hypertext Transfer Protocol - HTTP/1.1*.

[15] Georgia Institute of Technology, 2009. *MTConnect .NET Agent Software Development Kit*. www.mtconnect.org.

[16] Bengtsson, N., Michaloski, J., Proctor, F., Shao, G., and Venkatesh, S., 2010. "Discrete event simulation of factory floor operations based on MtConnect data". In Proceedings of the 2010 ASME International Conference on Manufacturing Science and Engineering, ASME.