*Proceedings of the 2010 Winter Simulation Conference*
*B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.*

# TOWARDS CONTINUOUSLY UPDATED SIMULATION MODELS
- Combining Automated Raw Data Collection and Automated Data Processing

Anders Skoogh

Product and Production Development
Chalmers University of Technology
Gothenburg, 412 96, SWEDEN

John Michaloski

National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899-8263, USA

Nils Bengtsson

Production Modeling Corporation
Gullbergs Strandgata 36D
Gothenburg, 411 04, SWEDEN

## ABSTRACT

Discrete Event Simulation (DES) is a powerful tool for efficiency improvements in production. However, instead of integrating the tool in the daily work of production engineers, companies apply it mostly in single-purpose studies such as major investment projects. One significant reason is the extensive time-consumption for input data management, which has to be performed for every simulation analysis to avoid making decisions based upon obsolete facts. This paper presents an approach that combines automated raw data collection and automated processing of raw data to simulation information. MTConnect is used for collection of raw data and the Generic Data Management Tool (GDM-Tool) is applied for data processing. The purpose is to enable efficient reuse of DES models by reducing the time-consumption for input data management. Furthermore, the approach is evaluated using production data from aerospace industry.

## 1 INTRODUCTION

Compared to desirable values presented by production researchers, the utilization of production resources is generally low in industry. For example, the Overall Equipment Efficiency (OEE) is often found to be 50 % -60 % in companies across various lines of business (Ingemansson 2004). Nakajima (1988) states that an OEE figure of 85 % should be possible to achieve and that 90 % is the ideal target.

Companies are well aware of the potential of reducing this gap and, thus, continuously work with system improvements in a systematic manner. However, despite the presence of powerful computer support, Information Technology (IT) tools are underutilized in support of continuous improvement processes. One example is Discrete Event Simulation (DES), which is applied in fewer projects than it should be when considering its capabilities (McLean and Leong 2001).

When correctly applied, DES works well in combination with continuous improvement philosophies like the Theory Of Constraints (TOC) (Goldratt 1990). During the philosophy's five iterative steps (Rahman 1998), simulation can be profitably applied in the following manner:

1. *Identify the system's constraint(s)*: DES results can be used to identify bottlenecks, for example by studying buffer sizes, resource utilization and machine states.
2. *Decide how to exploit the system's constraint(s)*: DES can be used to optimize buffer sizes around the bottleneck.
3. *Subordinate other resources to the constraint(s)*: DES can be used to evaluate what-if scenarios on how workers should prioritize the different production resources.
4. *Elevate the system's constraint(s)*: DES is a powerful tool for investment analysis.
5. *If in any of the previous steps a constraint is broken, go back to step 1.*

Based on the arguments above, and its capability of analyzing dynamic aspects of production flows, DES is an excellent tool for support of continuous improvement efforts. However, for various reasons, DES is often applied in well-delimited projects than on a daily basis (Williams 1996). One reason is the extensive time-consumption for raw data collection and for the processing of raw data to input information for simulation models. These steps, i.e. the input data management process, consume on average 31 % of the time for an entire simulation study (Skoogh and Johansson 2010). This problem becomes even more evident when DES is use on a daily basis, since the shop floor status changes continuously and therefore require repetitive efforts for input data management

Previous studies show that the main reason for the extensive time-consumption is that most companies collect raw data manually and convert it to information each time a simulation study is launched. One solution is to automate as many activities as possible and thereby enable automatic updates.

The purpose of this paper is to enable reuse of DES models by reducing the time-consumption for input data management. The aim is to evaluate the feasibility to combine automated raw data collection and automated data processing into a push-button solution for DES. In this study, MTConnect (MTConnect Institute 2010) is used to log time-stamps containing machine status directly from the machines. These raw data are then submitted to a tool for automated data processing, called the Generic Data Management Tool (GDM-Tool) (Skoogh 2009); see Figure 1. The GDM-Tool is a middleware solution that automates all data processing steps, including categorization, cleaning, calculation and condensation to statistical distributions. Using this solution, companies can apply DES in their daily work with continuous improvements, without spending time and efforts on repetitive work with input data management.

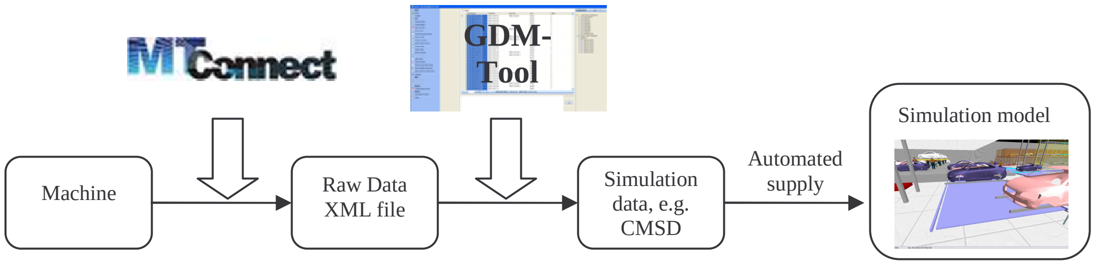


Figure 1: Schematic data flow from machine to simulation model.

Section 2 of the Paper list and describes tools that are currently available for data collection. In section 3 the methodology used in the Paper is presented. The Paper continues with the description over the project and the work that have been achieved in section 4. The Paper ends with the discussion in section 5 together with conclusions in section 6.

## 2 REVIEW OF INPUT DATA MANAGEMENT TOOLS

Previous research has examined the possible levels of automation in input data management for simulation. A good summary of four different approaches is provided by Robertson and Perera (2002):

1. *Tailor-made solution*: Data is primarily collected and processed by the project team, manually supplied to the model and resides in the simulation tool.
2. *Spreadsheet solution*: Data is primarily collected and processed by the project team, manually supplied to an intermediate spreadsheet interface and then automatically read by the simulation model. Data resides in the spreadsheet.
3. *Off-line database solution*: Data primarily derived from Corporate Business System (CBS), automatically supplied to an intermediate off-line database and then automatically read by the simulation model. Data resides in the database.
4. *On-line database solution*: Data primarily derived from CBS and automatically supplied to the simulation model without intermediate steps. Data resides in the CBS.

Robertson and Perera (2002) found that the manual solutions (1 and 2) were still most applied in industry, but there have been steady advances towards step three and four. However, integration of high level corporate databases such as Enterprise Resource Planning (ERP) systems has proved to be difficult because they contain too imprecise data for dynamic simulations (Moon and Phatak 2005).

As displayed in Figure 1, a data management system contains two main components: the collection of raw data and the processing of raw data into appropriate inputs for a simulation model. For collection, there are a few alternatives presented in previous publications. Ingemansson et al. (2005) presents a case study in an automotive company where an automated data collection system was used to log breakdown time-stamps from NC (Numerically Controlled) machines. The communication technology used in their case study and the format of the raw data is, however, unspecified. In addition to reduced time-consumption, their case study shows that automated collection of raw data also increases the data quality. See below for a brief review of other solutions for production data logging.

## 2.1 OPC for Raw Data Collection

One specific communication technology applicable for raw data collection is OPC (OLE (Object Linking and Embedding) for Process Control), which is a leading worldwide specification in enabling connectivity and interoperability of factory floor equipment. OPC is an integration technology developed by the OPC Foundation that defines a standard interface to control devices (OPC Foundation 2010). OPC promotes interoperability both horizontally and vertically in the enterprise so that it can cut integration costs, speed deployment and promote increased operating efficiency. The OPC software specification describes a client/server object model to allow communication between client applications (OPC Clients) and control device servers (OPC Servers).

OPC handles integration by creating a “software bus” so that applications need only know the data required from OPC data sources, not how to get it. OPC Foundation has defined several OPC interface specifications, including Data Access (DA), Event and Alarm Management, and Historical Data Access. The Data Access specification provides a standard mechanism for communicating to data sources on a factory floor. The Event and Alarm Management specification defines a means for transmitting alarm and event information between OPC servers and clients. The Historical Data Access specification allows OPC clients to access historical archives to retrieve and store data in a uniform manner. The major weakness of OPC is that it is primarily a standard IO communication mechanism, and does not offer any standard device information models. For example, OPC Clients are required to make assumptions that certain items would be available from the CNC (Computed Numerically Controlled) OPC Server, but there is not a guarantee to the tag availability, name, nor type of the data.

## 2.2 DPWS Combined with Automated Data Processing

A similar proposal as the one presented in this paper is provided by Aufenanger et al. (2010?). They used the Devices Profile for Web Services (DPWS) for collection of raw data from manufacturing resources such as conveyor belts and machines. DPWS enables collection of raw data, which is then stored in a database of an in-house developed application called Machine Data Acquisition (MDA). Furthermore the MDA application also includes algorithms for processing the raw data to serve as input to a simulation model. Raw data collection using DPWS is generic and automatically detects production resources connected to the local network, but the algorithms applied for data processing are specifically developed for each implementation. The paper states that their concept shows promising results in a test evaluation in laboratory environment.

## 3 METHOD

This paper combines the capabilities of two existing technologies, MTConnect and the GDM-Tool, in order to achieve a push-button solution for input data management in DES. The solution is designed and tested in a case study at a manufacturing company in order to ensure that it is applicable in a real world context. In addition to designing and demonstrating the solution, this study also compares it to alternative procedures of input data management. The comparison is performed by measuring the time-consumption for completing the input data management process using a completely manual approach and the proposed approach combining the capabilities of MTConnect and the GDM-Tool.

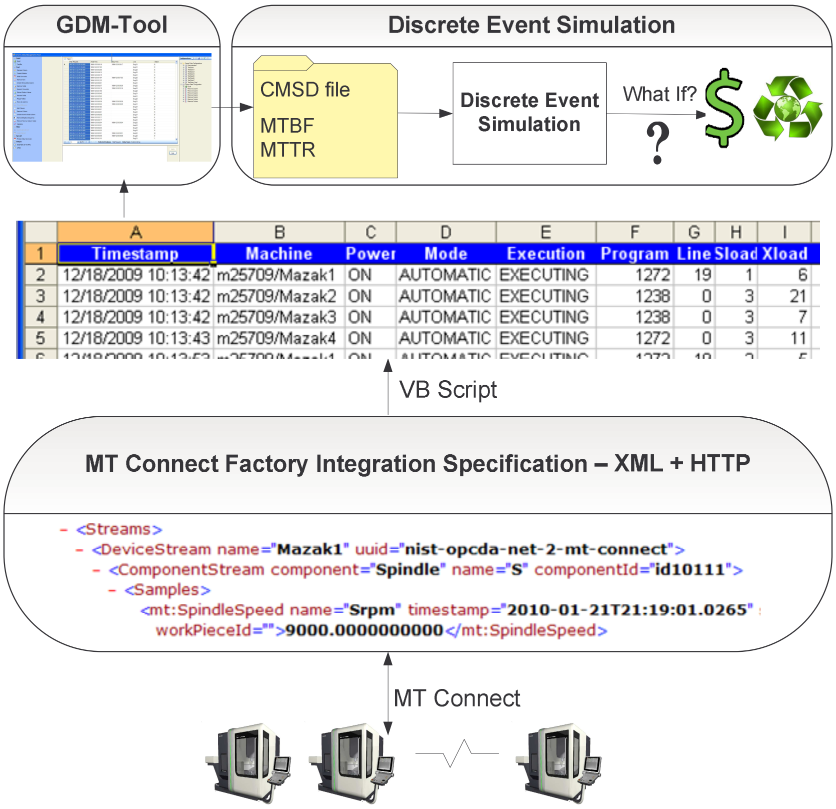
The manual approach is, according to the authors' experience, the most commonly applied in industry. It includes raw data collection from available data sources when present but also complimentary manual gathering, e.g. using a stop-watch. Necessary data cleaning such as elimination of outliers is performed using basic formulas in Microsoft Excel or similar applications. This is also the case for calculations, e.g. for the time between failures by finding the difference in start times of two failures. The data is finally condensed (usually to statistical distributions) using a distribution-fitting software or in worst case by assuming appropriate distribution families.

This case study is also a part of the validation of the GDM-Tool, which was originally designed and developed in a case study at a company in the automotive industry. The strategy is to test the tool in various case studies in different lines of business and thereby evaluate if its approach to automated input data management is feasible. Moreover, the different case studies will introduce further requirements on the tool and result in implementation of additional functionality. As a result, the tool will be increasingly robust and the vision is that it finally will be applicable for any DES study.

## 4 OUTLINE AND EVALUATION OF THE CONCEPT

A test implementation has been performed at a large aerospace manufacturing company in collaboration with National Institute of Standards and Technology (NIST) to explore what new opportunities that data from MTConnect offers for DES. A work cell in a job shop environment with four high-speed, five-axes CNC machines together with one pallet shuttle system was simulated. In the work cell a large variety of aluminum parts with individual cycle times are manufactured with high speed machining. The scope of the test implementation has been to show how MTConnect data can be used together with automated data processing and DES to improve production. Using data processing the Cycle Time, Mean Time Between Failure (MTBF), Mean Time To Repair (MTTR) and Energy usage could be calculated directly from the MTConnect data. Note however that the automated data processing evaluated in this paper is delimited to MTBF and MTTR. An overview of what have been done can be seen in Figure 2.

The goal for the entire research project was to study how energy usage data from MTConnect can be used for sustainability analysis in DES. It turned out that the data itself was useful for modeling but the savings on energy usage was unsatisfying. The data showed that the cost of running one of the machines in terms of electricity was around 50 cents. However, despite the low savings in this particular case the authors foresee an interesting future for combined DES and sustainability analysis.


GDM-Tool
Discrete Event Simulation
CMSD file
MTBF
MTTR
Discrete Event Simulation
What If?
?
VB Script
MT Connect Factory Integration Specification – XML + HTTP
MT Connect

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Timestamp | Machine | Power | Mode | Execution | Program | Line | Sload | Xload |
| 2 | 12/18/2009 10:13:42 | m25709/Mazak1 | ON | AUTOMATIC | EXECUTING | 1272 | 19 | 1 | 6 |
| 3 | 12/18/2009 10:13:42 | m25709/Mazak2 | ON | AUTOMATIC | EXECUTING | 1238 | 0 | 3 | 21 |
| 4 | 12/18/2009 10:13:42 | m25709/Mazak3 | ON | AUTOMATIC | EXECUTING | 1238 | 0 | 3 | 7 |
| 5 | 12/18/2009 10:13:43 | m25709/Mazak4 | ON | AUTOMATIC | EXECUTING | 1272 | 0 | 3 | 11 |

```
- <Streams>
  - <DeviceStream name="Mazak1" uuid="nist-opcda-net-2-mt-connect">
    - <ComponentStream component="Spindle" name="S" componentId="id10111">
      - <Samples>
          <mt:SpindleSpeed name="Srpm" timestamp="2010-01-21T21:19:01.0265"
            workPieceId="">9000.0000000000</mt:SpindleSpeed>
```

Figure 2: Overview of the test implementation.

### 4.1 MTConnect for Automated Raw Data Collection

MTConnect is a new standard for data exchange on the shop floor. MTConnect is a specification based upon prevalent Web technology including Extensible Markup Language (XML) (The World Wide Web Consortium 2010) and Hypertext Transport Protocol (HTTP) (The Internet Society 1999). Using prevailing technology and providing free software development kits minimizes technical and economic barriers to MTConnect adoption.

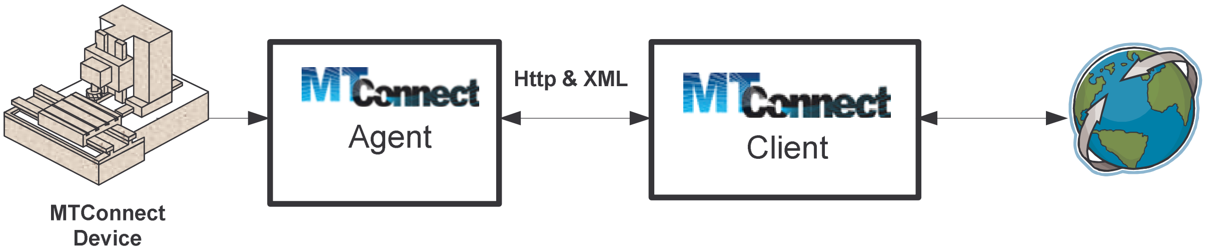


Figure 3: MTConnect architecture.

Figure 3 shows the MTConnect architecture, where an MTConnect Device is a piece of factory equipment organized as a set of components that provide data. An MTConnect Agent is a process that acts as a "bridge" between a device and a "Client Application". The MTConnect Agent receives and stores single or a time series of data samples or events that are made available to the Client Application. An MTConnect Client is typically a factory application, including shop floor dashboard visualization, Overall Equipment Effectiveness (OEE), and data mining of asset and process knowledge.

The basic MTConnect specification provides a CNC machine tool information models including definitions for position, feeds, speeds, program, control logic, and some tooling. The MTConnect Information

Model is not hardwired; rather users assemble an XML information model to match their devices. MTConnect further provides XML attributes in which to help refine the Device Information models. Such XML attributes include Category, Name, Type, Subtype and Units. New additions to the Information Model are backwards compatible, in that MTConnect Clients will not break when new information is supplied to them, but will only be unaware of its existence.

In previous work performed by the National Institute of Standards and Technology (NIST), data provided by MTConnect has been mapped according to the data requirements of DES. Table 1 shows how the raw data given by MTConnect can be converted to DES parameters. The table also includes sustainability metrics as a result of NIST's promising research on combining DES and sustainability analysis of production systems. Note however that this test implementation regards only the down time and the time between down times, which are the most important parameters to mimic the dynamics of production systems (Williams 1994).

Table 1: Mapping MTConnect data into possible DES parameters including sustainability metrics.

| MTConnect Data | Timestamp(ts), Machine, Power, Mode, Execution, Program, Line, Sload, Xload, Yload, Zload, Aload, Bload, Cload, Toolnum, RPM, Alarm, AlarmState, AlarmSeverity, PartCount, Feedrate |
|---|---|
| Cycle Time | $MTC_{mode}$=Auto & $MTC_{rpm}$ > 0 & $MTC_{feed}$ > 0 |
| Setup Time | $MTC_{program(t)}$ != $MTC_{program(t-1)}$ =>T($MTC_{mode}$ = Manual)<br>excluding pallete shuttle program |
| Machining Time | Cycle Time |
| Off Time | $MTC_{power}$ = Off |
| Down Time | $MTC_{alarm}$ = true |
| Idle Time | T($MTC_{execution}$=Paused \| $MTC_{mode}$=Manual) |
| Mean Time Between Failure | E(x) , where x = T($MTC_{alarm}$ !=active, $MTC_{alarm}$=active) |
| Mean Time To Repair | E(x) , where x = T($MTC_{alarm}$ =active, $MTC_{alarm}$!=active) |
| Coolant Energy | $MTC_{mode}$=Auto & $MTC_{rpm}$ > 0 => Coolant max rated kW load, else 0.0 |
| Power kW·h | = ($MTC_{spindle}$ ·15.0 + $MTC_{Xload}$ ·3.5 + $MTC_{Yload}$ ·3.5 + $MTC_{Zload}$ ·3.5+ $MTC_{Aload}$ ·1+ $MTC_{Cload}$ · 1)/110.0 + Baseline |
| $CO_2$ Emissions | 1.297 lbs $CO_2$ per kWh |
| $NO_x$ Emissions | 0.004 lbs $NO_x$/kWh |

### 4.2 The GDM-Tool for Automated Data Processing

The GDM-Tool is a computer application that demonstrates the concept of automated input data management. All steps in the input data management process are automated, except for the collection of raw data; see Figure 4. This exception is the reason for the potential of integrating the tool with MTConnect. The GDM-Tool is a MS Windows desktop application developed in C# (Visual Studio) and was originally developed for a case study in automotive industry as a part of a Swedish research project called FACTS (Skoogh, 2009). Its main data processing features include:

- *Data cleaning:* Removal of irrelevant data points. One common example is in collection of machine stop times, where several stop categories usually have to be removed to suit the needs of dynamic simulations.
- *Data calculations:* One example is the calculation of times between failures (TBF) as a subtraction of two different failure start-times.
- *Data condensation:* Distribution fitting. Files of DES raw data often contain several thousand data points, and that is generally considered too much to supply to the simulation model. Hence, practitioners often prefer to use statistical distributions.

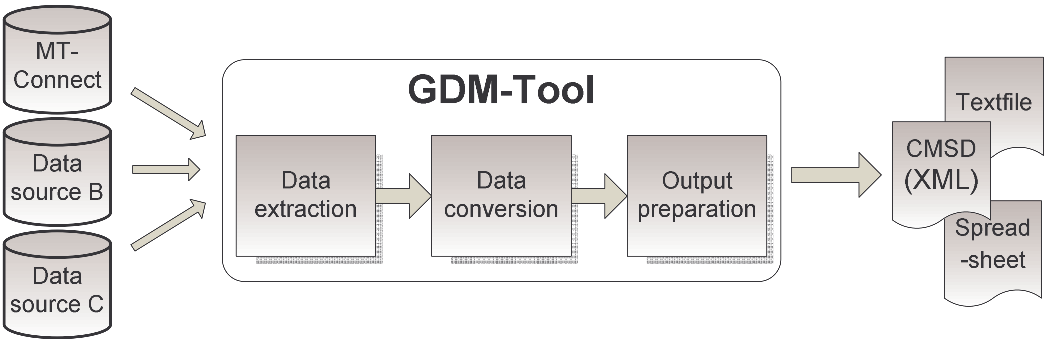


Figure 4: Outline of the GDM-Tool functionality.

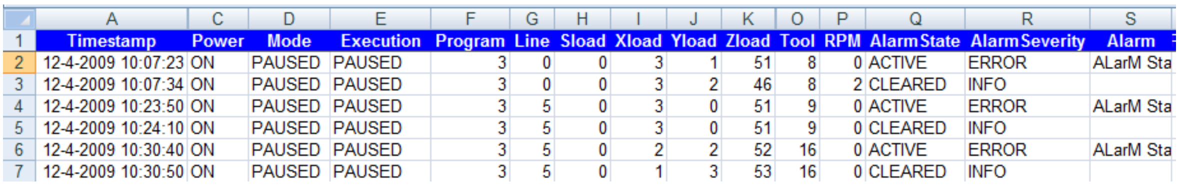
Notice that the GDM-Tool is a middleware solution that works similarly to solution 3 presented in chapter 2 (Robertson and Perera, 2002) with the difference that the GDM-Tool does not store data itself. Instead, raw data is still stored in the original data sources, e.g. MTConnect XML-files, CBS or regular spreadsheets. The simulation information, which is the output from the GDM-Tool, can reside in any application selected by the user. A standardized format such as a CMSD (Core Manufacturing Simulation Data) (SISO 2009) file is preferred for increased interoperability between applications.

An initial goal when designing the GDM-Tool was to make it completely generic and possible to automatically connect to any source of raw data. However, due to the diversity of data sources, with a huge amount of in-house legacy systems in industry, such a solution was not selected. Instead a user with good knowledge about the production process, the data sources and the simulation model has to set up a sequence of actions that the GDM-Tool will perform for a given simulation model. In other words a configuration is performed using tools for data extraction, data conversion and output preparation. These tools are implemented using a plug-in based architecture enabling convenient development and extension of the GDM-Tool functionality if required.

Later, a configuration can be executed in “automation mode” and fed with updated data sources. The automation mode is intended to be the most frequently used among the two user environments. Thus, the more times a configuration can be used without changes, the more beneficial is the approach of automated input data management. Finally, an old configuration can also be opened in the configuration mode for further development by selecting “edit mode” in the dialogue box.

### 4.3 Execution of Data Conversion Using the GDM-Tool

As previously stated, the production data used to evaluate the concept is taken from a real production system in the aerospace industry. The authors developed a script to poll raw data from the machines via MTConnect. Furthermore, a  Visual Basic (VB) script was developed to extract the relevant data (see Table 1) from the MTConnect XML file and present it as state changes over a sufficient period of time; see Figure 5.

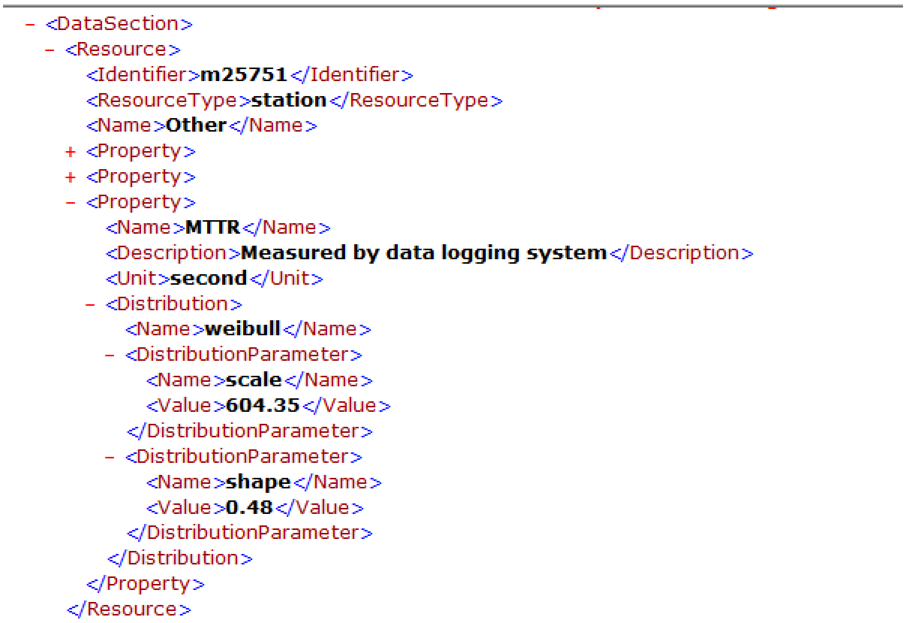


| | A | C | D | E | F | G | H | I | J | K | O | P | Q | R | S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Timestamp | Power | Mode | Execution | Program | Line | Sload | Xload | Yload | Zload | Tool | RPM | AlarmState | AlarmSeverity | Alarm |
| 2 | 12-4-2009 10:07:23 | ON | PAUSED | PAUSED | 3 | 0 | 0 | 3 | 1 | 51 | 8 | 0 | ACTIVE | ERROR | ALarM Sta |
| 3 | 12-4-2009 10:07:34 | ON | PAUSED | PAUSED | 3 | 0 | 0 | 3 | 2 | 46 | 8 | 2 | CLEARED | INFO | |
| 4 | 12-4-2009 10:23:50 | ON | PAUSED | PAUSED | 3 | 5 | 0 | 3 | 0 | 51 | 9 | 0 | ACTIVE | ERROR | ALarM Sta |
| 5 | 12-4-2009 10:24:10 | ON | PAUSED | PAUSED | 3 | 5 | 0 | 3 | 0 | 51 | 9 | 0 | CLEARED | INFO | |
| 6 | 12-4-2009 10:30:40 | ON | PAUSED | PAUSED | 3 | 5 | 0 | 2 | 2 | 52 | 16 | 0 | ACTIVE | ERROR | ALarM Sta |
| 7 | 12-4-2009 10:30:50 | ON | PAUSED | PAUSED | 3 | 5 | 0 | 1 | 3 | 53 | 16 | 0 | CLEARED | INFO | |

Figure 5: Example of list of state changes.

Thereafter, the GDM-Tool imports the list of state changes and processes it in order to go from raw data to simulation input information. To do this conversion the following steps are required, all corresponding to one plug-in in the GDM-Tool:

1. Import Spreadsheet file.
2. Remove all columns irrelevant to the specific analysis.
3. Remove all lines of data points containing state changes other than alarms.
4. Calculate the time difference between the start and stop time of each alarm to achieve the Time to Repair (TTR).
5. Store the calculated values in a new column.
6. Remove all stop times for the alarms.
7. Calculate the time difference between the start times of all alarms to obtain the Time Between Failure (TBF).
8. Store the calculated values in a new column.
9. Condense the data points in the two new columns via the plug-in for automated distribution fitting.
10. Export the data to a CMSD interface file; see example in Figure 6.



```
- <DataSection>
  - <Resource>
      <Identifier>m25751</Identifier>
      <ResourceType>station</ResourceType>
      <Name>Other</Name>
    + <Property>
    + <Property>
    - <Property>
        <Name>MTTR</Name>
        <Description>Measured by data logging system</Description>
        <Unit>second</Unit>
      - <Distribution>
          <Name>weibull</Name>
        - <DistributionParameter>
            <Name>scale</Name>
            <Value>604.35</Value>
          </DistributionParameter>
        - <DistributionParameter>
            <Name>shape</Name>
            <Value>0.48</Value>
          </DistributionParameter>
        </Distribution>
      </Property>
    </Resource>
```

Figure 6: Example of CMSD file.

### 4.3.1 The Difference Between States and Events

One very important difference between the raw data obtained by MTConnect and raw data from previous industrial test cases using the GDM-Tool is that MTConnect provides a list of states. Other data systems investigated by the authors have instead provided a list of events. Here, a *state* is defined as one row of data points describing the machine status at one given time; see the example in Figure 5. An *event* on the other hand is defined as a row of data points containing the start time and the duration of a specific state occurrence, in this case the start time and the duration of each specific alarm.

This difference requires some considerations for the automated data processing in order to identify the rows of data containing just start times and stop times of state occurrences, here called *state changes*. All other rows of data are irrelevant and must be disregarded from the data analysis. At present the identification of these rows containing state changes is performed by Visual Basic (VB) macros so the GDM-Tool imports and processes a list of state changes.

#### 4.3.2 Required Additions to the GDM-Tool

To streamline the data collection and processing even more, the GDM-Tool needs an addition of functionality for identifying the rows containing *state changes*. With such functionality the plug-in "Remove row by column value" can be applied to remove rows containing states and, thus, just keep a list of state changes. This enables parts of the VB script to be replaced by the GDM-Tool, which reduces the degree of customization.

## 5 DISCUSSION

This paper demonstrates complete automation of the input data management process in DES studies, by combining MTConnect and the GDM-Tool. The concept has the potential to significantly reduce the time-consumption for updating simulation models. As a result, simulation models can be used on a more frequent basis and be integrated in the daily work of industrial engineers. This provides support for iterative production development strategies such as TOC and would be of great advantage compared to the present situation when DES is mostly applied in single purpose projects.

MTConnect and similar technologies open several possibilities in collection of raw data for DES, but there are still a lot of challenges to meet. One challenge is to develop a solution to log MTConnect data over a longer time period. The current solution to log the data in a software spreadsheet works for now, but will face problems for longer time periods due to the limitation of rows. One solution is to set up a data-base and let the VB-script continuously store raw data samples. The GDM-Tool can then request and automatically import data from the data base when a simulation run is initialized.

Another challenge is to have a more detailed alarm handling in MTConnect. Currently the alarms from the machine itself are directly included in the MTConnect data. This causes a lot of problems since every machine vendor has its own sets of alarm codes and implementation of alarms. What needs to be put in place is a way of classifying how severe an alarm is to evaluate whether an alarm really corresponds to a complete machine breakdown. This and similar problems are very important to solve in order to maintain the user's trust in accurate data, which is a common criticism against automated data management.

The GDM-Tool has now been evaluated in three different industrial case studies, two in the automotive industry and one in the aerospace industry. The concept of automated data management still seems to be promising thanks to the relatively similar structures of raw data sources in industry. However, the tool has not yet reached the state where it can be directly applied at a new company without development of additional plug-ins. Thus, the goal of being generic cannot yet be claimed. More case studies are required to evaluate whether the middleware solution can be generic or if some customization always will be required. Regardless of these modifications, one common result for all case studies is a significant reduction of time-consumption compared to conventional methods in industry.

The approach demonstrated in this paper is one view of the third solution to input data management presented by Robertson and Perera (2002). Another similar approach is the one presented by Aufenanger et al. (2010), which combines Devices Profile for Web Services (DPWS) for raw data collection and a custom made application for data storage, data processing and simulation. A slight difference between this approach and the one presented in this paper is that the GDM-Tool is intended to strive towards a state where it can be considered generic to different production systems, while the alternative presented by Aufenanger et al. (2010) requires some customization. A third category in the same category is the use of VB macros for data cleaning and calculations. However, a separate application is often applied to condense data to a suitable format for simulation. Working with macros for input data management is considered relatively time-efficient given that the programmer is well experienced or/and has a library of similar macros for reuse of code. Otherwise the approach is too specific and the user runs the risk of spending more time on macro development than what was originally required to manually process the data.

## 6 CONCLUSIONS

This paper presents one solution of completely automated input data management for DES models including machining operations. The solution is demonstrated on the simulation input parameters MTBF and MTTR. Raw data is directly collected by the machines via MTConnect, processed and prepared for simulation in the GDM-Tool, and finally input to a DES model for production analysis. The main objective is to reduce the time-consumption for input data management in DES, which is the major part of most simulation studies. As a result, simulation models can be more frequently updated and be integrated in the daily work of production engineers.

The solution is tested in a case study at an aerospace manufacturing company with good results. Compared to complete manual data handling, the estimated time-reduction is around 75 % (from ½ work day to 1 hour) just for the processing of raw data. If the time for raw data collection is included, which normally takes several days or weeks, the reduction is of course even larger. Another option evaluated by the authors is to use VB macros instead of the GDM-Tool and this solution can be almost as efficient given that the user has built the macros in advance or that he/she has a well established library of suitable code sections. However, the advantage of the GDM-Tool is that it provides data processing operations applicable for any manufacturing company storing their raw data in some relational table format.

Both MTConnect and the GDM-Tool are quite new applications for input data management to DES and it is therefore necessary to evaluate and improve them using further industrial case studies. In this case study some experiences should be highlighted:

- Data provided by MTConnect is polled in intervals specified by the user and, thus, presented as a list of machine *states*. Similar systems in manufacturing industry typically stores raw data as *events* containing information of both start time and duration. This difference initiated the development of a new plug-in in the GDM-Tool converting a set of states into events.
- There are still some issues on how to interpret some data points provided by MTConnect. One example is that some down time samples seemed to be too short to be considered as machine breakdowns in a DES model. They rather appeared to be logged due to communication problems between the machine and the MTConnect agent. In any case, more studies are required before using the raw data in sharp industrial DES studies.
- Further development of the GDM-Tool would streamline the data flow even more by reading the raw data directly from the MTConnect XML-file and eliminate the need for intermediate spreadsheet storage.

## ACKNOWLEDGMENTS

The funding for this research is, in addition to the authors' employers, granted by the Swedish Foundation for Strategic Research (SSF) through a scholarship granted by their research program ProViking. Furthermore, the Swedish Agency for Innovation Systems (VINNOVA) provided financing via the research program FFI (Strategic Vehicle Research and Innovation) and more specifically via the research project Digital Factory Building Blocks (DFBB).

## DISCLAIMER

Commercial equipment and software, many of which are either registered or trademarked, are identified in order to adequately specify certain procedures. In no case does such identification imply recommendation or endorsement by Chalmers University of Technology or the National Institute of Standards and

Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

## REFERENCES

Aufenanger, M., A. Blecken, W. Dangelmaier, and C. Laroque. 2010, accepted for publication. Design and Implementation of an MDA Interface for Flexible Data Capturing. Accepted for publication in: *Journal of Simulation*.

Goldratt, E.M. 1990. *What is This Thing Called Theory of Constraints and How Should it be Implemented?* New York: North River Press.

Ingemansson, A. 2004. *On Reduction of Production Disturbances in Manufacturing Systems Based on Discrete-Event Simulation*. Doctorial dissertation, Department of Mechanical Engineering, Lund University, Lund, Sweden.

McLean, C. and S. Leong. 2001. The expanding role of simulation in future manufacturing. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1478-1486, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Moon, Y. B. and D. Phatak. 2005. Enhancing ERP system's functionality with discrete event simulation. *Industrial Management & Data Systems*. 105:1206-1224.

MTConnect Institute. 2010. <http://www.mtconnect.org> [accessed March, 2010]

Nakajima, S. 1988. *Introduction to TPM – Total Productive Maintenance*. Portland: Productivity Press.

OPC Foundation. 2010. <http://www.opcfoundation.org> [accessed March, 2010].

Rahman, S. 1998. Theory of Constraints – A Review of the Philosophy and its Applications. *International Journal of Production & Operations Management*. 18:336-355.

Robertson, N., and T. Perera. 2002. Automated data collection for simulation? *Simulation Practice and Theory*. 9:349-364.

SISO – Simulation Interoperability Standards Organization, CMSD Product Development Group. 2009. Standard for: Core Manufacturing Simulation Data - UML Model, May 11, 2009.

Skoogh, A. 2009. *Methods for Input Data Management - Reducing the Time-Consumption in Discrete Event Simulation*. Licentiate Thesis, ISSN 1652-9243, Department of Product and Production Development, Chalmers University of Technology, Gothenburg, Sweden.

Skoogh, A., and B. Johansson. 2010, accepted for publication. Mapping of Time-Consumption During Input Data Management Activities. Accepted for publication in: *Simulation News Europe*.

The Internet Society. 1999. Hypertext Transfer Protocol - HTTP/1.1. Available via <http://www.isoc.org> [accessed March, 2010].

The World Wide Web Consortium. 2006. *Extensible Markup Language (XML) 1.0*. 4th ed., ed. T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau.

Williams, E. J. 1994. Downtime Data -- its Collection, Analysis, and Importance. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 1040–1043. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Williams, E. J. 1996. Making Simulation a Corporate Norm. In *Proceedings of the 1996 Summer Computer Simulation Conference*, ed. V. W. Ingalls, J. Cynamon and A. V. Saylor, 627-632.

## AUTHOR BIOGRAPHIES

**ANDERS SKOOGH** is a Ph.D. student at the department of Product and Production Development, Chalmers University of Technology in Sweden. He received his M.Sc. degree in 2005 and the degree of Licentiate of Engineering in 2009. Anders also has industrial experience of discrete event simulation from an employment as logistics developer at Volvo Car Corporation. His research focuses on reducing the time-consumption for input data management in discrete event simulation. Currently, he develops and evaluates methodologies for automated data management. His e-mail address is

<anders.skoogh@chalmers.se>.

**JOHN MICHALOSKI** is a computer scientist in the Intelligent Systems Division at the National Institute of Standards and Technology in Gaithersburg, MD USA. His work focuses on promoting and developing machine control technology and standards. This work has lead him to chair and participate in numerous standards committees, as well as develop software and publish numerous papers related to manufacturing technology <john.michaloski@nist.gov>.

**NILS BENGTSSON** is an applications engineer at Production Modeling Corporation Europe AB. He received his Bachelor Degree in Automation and Mechatronics Engineering and Masters Degree in Production Engineering from Chalmers University of Technology in Gothenburg, Sweden. He is currently working as a guest researcher at the National Institute of Standards and Technology. His research interest includes discrete event simulation applicability and interoperability for industry.  His main focus is how discrete event simulation can be used to analyze and solve sustainability problems. His email is <nbengtsson@pmcorp.com>.