

## A TEST IMPLEMENTATION OF THE CORE MANUFACTURING SIMULATION DATA SPECIFICATION

Marcus Johansson  
Björn Johansson  
Anders Skoogh

Swee Leong  
Frank Riddick  
Y. Tina Lee  
Guodong Shao

Pär Klingstam

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

Manufacturing Systems  
Integration Division  
National Institute of Standards and Technology  
Gaithersburg, MD 20889-8261, USA

Manufacturing Engineering Office  
Dept 81021, PVÖS 31  
Volvo Car Corporation  
Gothenburg, SE-405 31, SWEDEN

### ABSTRACT

This paper describes an effort of testing the Core Manufacturing Simulation Data (CMSD) information model as a neutral data interface for a discrete event simulation model developed using Enterprise Dynamics. The implementation is based upon a model of a paint shop at a Volvo Car Corporation plant in Sweden. The model is built for a Swedish research project (FACTS), which focuses on the work procedure of developing new and modified production systems. FACTS has found standardized simulation data structures to be of high interest to achieve efficient data collection in conceptual stages of production development programs. For the CMSD-development team, implementations serve as an approach to validate the structures in CMSD and to gather requirements for future enhancements. CMSD was originally developed to support job shops, but the results of this implementation indicate a good possibility to extend CMSD to also support flow shops.

### 1 INTRODUCTION

A global market where companies are active in many countries with many sub-suppliers, customers, software and hardware-providers sets high demands on standardized work procedures. These are needed to enable efficient sharing and communication within the business.

Discrete event simulation (DES) has evolved over a few decades now (Banks et al. 2005) and is moving into a mature state in which standardization is a relevant part (Banks et al. 2003). A Core Manufacturing Simulation Data (CMSD) information model has been developed at the National Institute of Standards and Technology (NIST) as a part of a system interoperability effort to enable integration of DES and other manufacturing software applications.

*Factory Analyses in Conceptual Phases Using Simulation* (FACTS), <http://www.his.se/facts>, is a

Swedish research project involving the Swedish automotive industry and research institutions. FACTS focuses on the development of work procedures for conceptual plant design. Case studies are planned for FACTS to test and verify research results. The first case study is based on a paint shop operation at a Volvo Car Corporation plant. Two different models are built for this case study using different approaches: abstract and detailed modeling. This paper aims at describing the research project and case study in general, and to describe a test implementation of the CMSD information model for that case study in particular. The implementation objective is to provide manufacturing data for a model built in the DES package Enterprise Dynamics (ED) (Incontrol 2007), based on a shop floor data collection and user interface created in Microsoft Excel. The implementation covers data for workstation cycle time, staffing and breakdowns, and also the speed of material handling resources.

A brief introduction of the CMSD effort is provided in section 2. An overview of the total research project is given in section 3. The case study with the CMSD information model implementation is presented in section 4. Discussion and recommendations are provided for further development, extension, and use of the CMSD specification, in order to take one more step towards interoperability between simulation and other manufacturing software applications.

In this paper, the abbreviation *CMSD* will be used both to address the CMSD standardization effort, and to address the CMSD information model.

Demonstration material for this paper is available at <http://www.tfd.chalmers.se/~johamarc/wsc07/>

### 2 INTEROPERABILITY

While simulation technology has been demonstrated to be an effective tool for reducing costs, improving quality, and

shortening the time to market for manufactured goods, there are a number of technical and economic barriers that hinder the use of this technology in the manufacturing industry. The cost of developing, implementing, and using simulation technology is high. The costs of integrating simulation systems with other manufacturing applications are even higher. There is always a need to transfer, exchange, and share data between simulation and other manufacturing applications.

Developing custom-built proprietary interfaces is costly and makes using simulation technology prohibitive. The development of a reusable, neutral, standardized interface could help reduce the costs associated with simulation model construction, data exchange, and integration between simulation and other manufacturing applications. NIST researchers in collaboration with industrial and university partners have been engaged in a standards development effort in the Core Manufacturing Simulation Data Product Development Group under the guidelines, policies, and procedures of the Simulation Interoperability Standards Organization, <<http://www.sisostds.org/>> (SISO). A brief introduction of the CMSD effort is provided in the next section. A draft CMSD information model specification is available online (SISO 2007).

## 2.1 Core Manufacturing Simulation Data

Interoperability between manufacturing software applications and simulation is currently extremely limited. Achieving interoperability involves addressing a number of issues including model development, data translation, and system integration. The CMSD effort was organized to address these issues. The CMSD information model defines a data specification for efficient exchange of manufacturing data in a simulation environment. The specification provides a neutral data format for integrating manufacturing software applications with simulation systems. The initial effort focuses on machine shop data definitions (McLean et al. 2005). The plan is to extend the data specification to support automotive vehicle assembly operations, aerospace assembly operations, plant layout, supply chain, and other relevant manufacturing and simulation information.

CMSD, when completed, will satisfy the following goals: (1) to enable data exchange between simulation systems, other software applications, and databases, (2) to support the construction of manufacturing simulators, (3) to support testing and evaluation of manufacturing software, and (4) to support manufacturing software application interoperability.

The primary approach of CMSD is to develop neutral data structures for exchanging manufacturing data between various manufacturing software applications, such as DES, Enterprise Resource Planning (ERP), Master Production Schedule (MPS), and Manufacturing Execution System (MES), see Figure 1. The approach is to use the same data

structures for managing actual production operations and for simulating the manufacturing shop. The rationale is that if one structure can serve both purposes, the need for translation and abstraction of the real data would be minimized when simulations are constructed. It is also recognized that maintaining data integrity and minimizing the duplication of data are important requirements. For this reason, each unique piece of information appears in only one place in the model. Cross-reference links are used to avoid the creation of redundant copies of data.

CMSD describes the essential entities in the manufacturing domain and the relationships between those entities that are necessary to create manufacturing simulations. This information model will facilitate the exchange of information between simulation and other manufacturing software applications. The major categories of manufacturing information that are defined in this model include: *organization, calendar, resource, skill definition, setup definition, operation definition, maintenance definition, part, bill-of-materials, inventory, process plan, work, schedule, revision, distribution definition, reference, and unit defaults*. A description of these categories is included in Lee and Luo (2007).

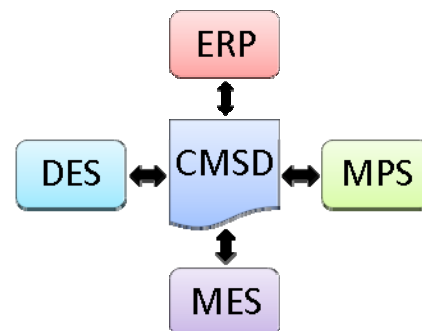


Figure 1: CMSD is intended to serve as a neutral format among common production software applications.

## 3 INDUSTRY NEEDS

A call for standardization of manufacturing data in simulation environments is taking place in the Swedish automotive industry. There is also a need for rapid and precise methods and information technology tools for the analysis of plant systems in early program stages, e.g., the introduction of new products or major investments in production equipment. The manufacturing industry continuously faces challenges that drive the need for new concepts for production systems, and the time available for design and development is continuously shortened to obtain competitive market positions through fast product introductions (Wheelwright and Clark 1992). An example of this challenge from the automotive industry is the need to develop new engines designed to have little or no environmental

impact. This kind of new product concept will drive requirements for the design of new production concepts.

It is in the early stages of product and process development that costs are determined and fixed, i.e., the decisions have high financial impact (Thomke and Fujimoto 2000). However, simulation is often used in late phases when costs are fixed, and thereby opportunities to influence solutions are very limited (Ericsson 2001). Since input data management often represents a proportionally large effort of a simulation project (Umeda and Jones 1997), projects using analysis tools like DES consume a large amount of time (Skoogh and Johansson 2007). Consequently, organizations are tempted to use less precise methods, which lead to difficulties in predicting the consequences of decisions that are required for robust, flexible and cost effective production. Methods and standards for efficient model building and data exchange would reduce the cost and time needed for simulation studies, and thereby opening doors for the wider use of simulation in the early operation design phases.

The Swedish research project FACTS focuses on the work procedure of developing new or modified production systems. The objective is to support the concept definition phase and to "frontload" the use of virtual methods to analyze complete automotive plants. To overcome the barriers that complicate the use of DES in early phases, efforts are needed to support development of (1) abstract modeling, (2) data handling and (3) optimization. Therefore, FACTS is divided into three subprojects:

1. Abstraction: Test, develop, and implement analytic methods for abstraction of complex manufacturing systems in DES models.
2. Input data management: Create prerequisites for the analysis of different manufacturing concepts through effective shop floor data gathering and organizing. Use of standardized data structures is of high interest, since it would increase data availability by enabling automatic data exchange between manufacturing applications, databases, and enterprise information systems.
3. Optimization: Reduce the manual simulation experimentation, traditionally performed to forecast the most cost effective manufacturing solution; evaluate the optimal settings of a considerable amount of manufacturing parameters, e.g., buffer sizes, number of operators and batch sizes.

The industrial partners in FACTS are Volvo Car Corporation (project management), Volvo Technology, Volvo Powertrain, and Volvo Trucks. Research partners are University of Skövde, Chalmers University of Technology, and IVF (Industrial Research and Development Corp.) The FACTS project also collaborates with NIST on standardization development for DES model representations.

#### **4 CASE STUDY AT VOLVO CARS PAINT SHOP**

Four case studies serve as the basis for the FACTS-project. The first one is carried out at Volvo Cars in Gothenburg, Sweden, and is discussed in this section. The industrial needs of Volvo Cars are used as the starting point for this case study. There are two basic issues regarding the analysis of complete plants at Volvo Cars:

- Modeling of complete plants takes too long and requires simulation expertise.
- Input data management consumes too much time and requires a lot of manual work.

Therefore, there is a need for rapid creation of easily understood abstracted models that can be used by manufacturing program managers; they are usually the ones that make important decisions in early stages.

There is also a need for standardizing and automating the collection and preparation of data for input into relevant analysis tools. There is usually a vast amount of crude production data available at Volvo Cars, but this data is not suitable for direct input into the analysis tools. Furthermore, the abstract model approach implies manipulation of input data to fit the chosen level of abstraction. Volvo Cars has a clear need for a standardization work, with clear definitions for the representation and application exchange of all shop floor data that is needed to carry out simulation at all abstraction levels.

The Volvo Cars paint shop case study is split up in two major tasks: First, create an abstract model of the factory. Second, create a detailed model of the same factory. The model of the latter task is meant to be used as a point of reference (will be explained below). The work conducted in the second task is the topic for this paper.

##### **4.1 Purpose of Detailed Modeling Task**

The main purpose of the detailed modeling task is to provide a comparison instrument for evaluating the robustness and confidence of the applied abstraction techniques used in its sister model.

Another purpose is to develop reusable objects and generic solutions for the simulation engineers at Volvo Cars. This will be the first attempt to make a detailed model of a Swedish Volvo factory by using ED. The extensible and object oriented structure of ED makes it possible to create tailored modeling objects for common resources used by Volvo Cars. Tailored objects and reusable logic functions would speed up the modeling of the paint shop for experimenting purposes as well as modeling of other factories within the Volvo Car Corporation.

There is also a goal of making it possible for factory management to use the model for planning on a weekly basis. They are interested in a tool that can help them forecast necessary shop floor staffing. Because of this goal, a model data interface is needed that is easy to use and preferably

made in Microsoft Excel, which is a well-known tool to the presumed user.

The fact that two models will be built of the same process, sets the input data in focus. Both models will use data created from shop floor logs and cycle time estimations. However, since one of the models will be built by using abstraction methods, it will need “abstracted” input data, such as Effective Process Time (EPT) (Jacobs et al. 2003). EPT recalculates process time and disturbance data from a detailed level to fit as input to a more abstract simulation model (Hopp and Spearman 1996). The input data *sources* will still be the same for both models and hence, the use of a common data structure is still interesting. As explained earlier, Volvo Cars and the FACTS-project have a high interest in a standard for input data structures. To enable standardized input data management, CMSD (see section 2.1) is used as a basis for representing data related to resources and work processes.

This paper focuses on the test implementation of CMSD. The results of the detailed modeling case study are presented in Johansson and Zachrisson (2006).

## 4.2 Method and Realization

To accomplish the detailed modeling task, the paint shop manufacturing process is mapped in detail, a simulation model is built, and a user interface based on Microsoft Excel is created. Input for the modeling effort is solicited from personnel involved in Volvo Cars operations. This was undertaken to provide a high level of realism and validation for the model, and to foster buy-in for the personnel who might use the model.

The methodology used during the model-building phase is derived from the one developed by Banks et al. (2005). The main difference is that Banks proposes complete conceptualization before simulation modeling, whereas Johansson and Zachrisson (2006) use iterations of gradual conceptualization and simulation modeling by dividing the whole factory in smaller areas modeled one at a time.

### 4.2.1 CMSD Work Procedure

Since one of the purposes of this task is to create a useful tool for factory managers, an Excel user interface was created to facilitate easy data input. In the interface it is possible to set cycle time, disturbance data, speed of material handling equipment and resource availability. The Excel interface can then serve as a data collection and as the application for the creation of CMSD data files.

The CMSD implementation was carried out at the end, due to resource constraints. Also, for the FACTS-project a detailed, valid, and useable simulation system that can be presented to Volvo Cars is the top priority. An implementation of CMSD is not explicitly necessary for that.

With the simulation model at hand, XML (W3C 2007) instance documents based on the CMSD were created containing information associated with the resources and work processes defined in the simulation model. A script to interpret and transfer the XML data into the simulation model was written based on the XML instance documents.

## 4.3 The Manufacturing Process

The factory contains several processes associated with the painting of a car body: sealing, washing, painting, hardening and controlling. A simple flow diagram of the factory can be seen in Figure 3. A thorough flow diagram and a conceptual model is included in Johansson and Zachrisson (2006).

Beside *continuously driven lines* like washing, painting processes, and hardening ovens, the work carried out in the factory can be divided in two resource types: *manual workstation* and *automatic workstation*. Cycle times for those stations are easily available, but cycle times for individual tasks on a workstation are not. Therefore workstations provide the lowest level of operation that can be modeled. Process logs are available in a shop floor database from which disturbance data for workstations and other factory resources can be retrieved.

The car bodies are carried on skids through the factory. The skids are moved on special skid-conveyors (see Figure 2). The number of conveyor resources is extensive compared to the work-related resources. There are also several elevators in the factory. Disturbance data logged for conveyors and elevators are scarce. Therefore, those are modeled as reliable resources and no breakdown data can be set for them in the model.

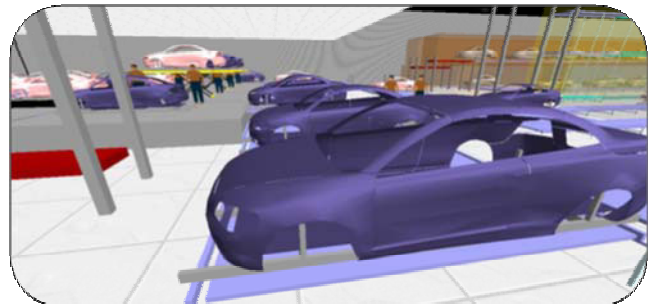


Figure 2: Car bodies on skids, moving on skid-conveyors after primer-paint control stations.

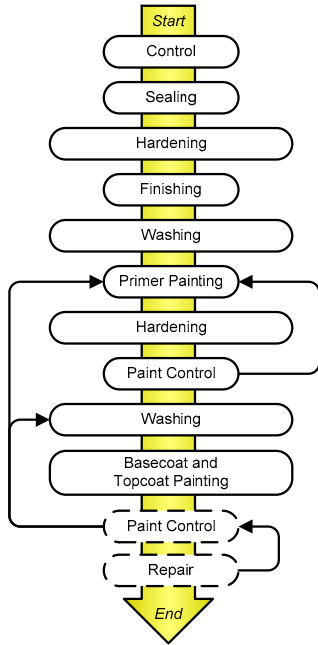


Figure 3: A simple flow chart of the factory. Steps with dashed lines are present in the factory but not included in this case study. Steps with wide symbols (hardening, washing, etc.) are *continuously driven lines*.

#### 4.4 The Input Data Collection

The contents and structure of the resources contained in the Excel interface are described here to support the upcoming explanation of how this information was represented in CMSD.

Separate worksheets are provided for defining conveyor and elevator data. For conveyors, only the speed can be set. For elevators, two different speeds can be set, indicating the speed for an empty elevator and the speed of a loaded elevator.

For workstations, several worksheets are used to provide a means to specify staffing, cycle time, and disturbance data. In the staffing sheet, the number of employees assigned to each workstation on a certain shift can be set. In the cycle time sheets, the distribution type and its parameters can be set for each combination of workstation, product, and staffing (workstation staffing is only a parameter for manual stations). In the disturbance sheet the user can select distribution type and parameters for *Mean Time To Failure* and *Mean Down Time*.

#### 4.5 The CMSD Implementation

Since the Excel interface contains all necessary resource data, the CMSD-based XML document was created based on that data. Discrepancies exist between the way some of the data is modeled in CMSD and how it is modeled in the

ED simulation model. These discrepancies and the methods used to resolve them are presented in this section. In general, discrepancies were resolved by creating tentative extensions to the CMSD specification rather than changing the simulation model. This approach was undertaken because recreating and re-verifying the finished simulation model would be extremely burdensome, creating tentative extensions to the CMSD would not. One of the interests of the CMSD development team is to use CMSD in integration projects and case studies such as this to validate the structures already present in CMSD and to gather requirements for future enhancements. The extensions to the CMSD were created solely to carry out the integration efforts associated with this project, and may or may not be permanently included as a part of CMSD. Working with real world data from companies such as Volvo Cars provides a means to evaluate the CMSD specification's usefulness in providing a framework for fostering the exchange of data between manufacturing applications.

An XML instance document was created containing the data that was modified through the Excel user interface and that was to be read into the ED simulation. To automate this process, an Excel macro was written using Microsoft Visual Basic for Applications commands and the Microsoft XML Core Services Library (MSXML) (MSDN 2007). The complete data flow is visualized in Figure 4. ED provides a basic set of functions to read and write XML documents, and these were used to import the data into the ED simulation. The ED functions provide a means to access specific XML elements in a document directly by node name, and this capability was used to create a translation script in ED that worked irrespective of the order of the data in the XML document.

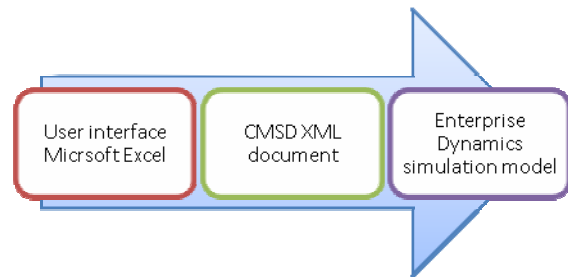


Figure 4: The simulation input data flow.

##### 4.5.1 Conveyors in CMSD

CMSD does not directly support material handling equipment such as conveyors and elevators, but is currently being extended with such support. Only small changes in the CMSD specification were needed to make the implementation of such data possible. The proposed way to define input data for a conveyor in a CMSD XML document is shown below:



```
<Resource identifier="Part_13:C_341.05">
  <Description>ALONG</Description>
  <ResourceType>conveyor</ResourceType>
  <Speed>
    <Value unit="meterPerSecond">0.33</Value>
  </Speed>
</Resource>
```

This way of defining conveyor information is completely in line with how other resources are defined in CMSD. The only extension is to recognize conveyor as a valid resource type. A goal of CMSD is to enable a detailed definition of the core attributes for all entities included. For a conveyor that could be its length, position, design, curves, elevation, etc. Direct support for attributes related to conveyors is planned for the next release of CMSD.

#### 4.5.2 Elevators in CMSD

Elevators need data similar to conveyors, but they have two speeds defined instead of one. In CMSD this can be modeled as:

```
<Resource identifier="E_52-11-341.30">
  <ResourceType>elevator</ResourceType>
  <Speed>
    <Description>unloadedSpeed</Description>
    <Value unit="meterPerSecond">0.06</Value>
  </Speed>
  <Speed>
    <Description>loadedSpeed</Description>
    <Value unit="meterPerSecond">0.15</Value>
  </Speed>
</Resource>
```

As with conveyors, the list of acceptable resource types in CMSD needed to be extended to accept elevator as a valid type. The Description child element of each Speed element is used to distinguish the different speeds associated with an elevator.

#### 4.5.3 Workstation Staffing in CMSD

In the model, each resource must be associated with data for staffing, cycle time, and disturbance. CMSD provides a means to: define shifts, associate resources with shifts, and to associate the minimum and maximum number of employees associated with a resource. But there is not a way to associate a specific number of employees for each resource for each shift. Hence, a special CMSD structure, called *shift staffing plan*, is built for this particular implementation to support the special data association used in the model. The shift staffing plan contains *shift staffing plan items*. Each item includes staffing data for a certain workstation on a certain shift. The data structure for three workstations in the staffing worksheet and the proposed XML-structure defining the staffing of the first workstation in that worksheet are presented below and in Figure 5.

```
<ShiftStaffingPlan identifier="AllResources">
  <ShiftStaffingPlanItem>
    <Resource
      resourceIdentifier="WM_11-140"/>
    <Shift shiftIdentifier="AShift"/>
    <NumberOfEmployees>3</NumberOfEmployees>
  </ShiftStaffingPlanItem>
  <ShiftStaffingPlanItem>
    <Resource
      resourceIdentifier="WM_11-140"/>
    <Shift shiftIdentifier="BShift"/>
    <NumberOfEmployees>3</NumberOfEmployees>
  </ShiftStaffingPlanItem>
  <ShiftStaffingPlanItem>
    <Resource
      resourceIdentifier="WM_11-140"/>
    <Shift shiftIdentifier="CShift"/>
    <NumberOfEmployees>2</NumberOfEmployees>
  </ShiftStaffingPlanItem>
</ShiftStaffingPlan>
```

Workstation	A-shift	B-shift	C-shift
WM_11-140	3	3	2
WM_11-150	0	0	0
WM_11-160	2	2	2

Figure 5: The staffing worksheet showing data for three different manual workstations. The CMSD staffing structure for the uppermost workstation is shown in the XML-code above.

#### 4.5.4 Workstation Cycle Time in CMSD

The CMSD specification is built upon a foundation that separates cycle time data from resource definitions. Cycle time is normally not a natural attribute of a resource, but rather an attribute of a *process plan* or *work* definition. But the simulation model has no central entity where products and resources can be “connected” to define cycle times according to a process plan or a work definition. Instead, cycle times are handled as workstation specific data and are stored in tables included in each workstation object. In combination with the fact that the model uses current staffing as a parameter to determine cycle time, it was decided to include cycle time, as an extension, in the resource definition and not use CMSD’s process plan or work classes.

The XML code below represents the data for the first workstation in Figure 6 and Figure 7. For this station, cycle time is defined for one and two operators. There are three different products. Thus, a total of six different cycle times are defined. To save space, only one of these cycle time nodes is shown.

```
<Resource identifier="WM_GRIND1">
  <Description>Finish_grinding</Description>
  <ResourceType>station</ResourceType>
  <CycleTime product="Body1" numberOfOp="1">
    <Distribution>
```

```

<Name>normal</Name>
  <DistributionParameter>
    <Name>mean</Name>
    <Value>412.8</Value>
  </DistributionParameter>
  <DistributionParameter>
    <Name>standardDeviation</Name>
    <Value>17</Value>
  </DistributionParameter>
</Distribution>
</CycleTime>
.
.
.
</Resource>

</MTTF>
<MDT unit="second">
  <Distribution>
    <Name>lognormal</Name>
    <DistributionParameter>
      <Name>mean</Name>
      <Value>1130</Value>
    </DistributionParameter>
    <DistributionParameter>
      <Name>standardDeviation</Name>
      <Value>2024</Value>
    </DistributionParameter>
  </Distribution>
</MDT>
</Resource>

```

Workstation Name	Distri- bution	1 Op.			2 Op.		
		Body 1	Body2	Body 3	Body 1	Body2	Body 3
WM_GRIND1	Normal	412.80	412.40	436.00	206.40	206.20	218.00
WM_GRIND2	Normal	412.80	412.40	436.00	206.40	206.20	218.00
WM_GRIND3	Normal	412.80	412.40	436.00	206.40	206.20	218.00

Figure 6: The cycle time sheet with mean values.

Workstation Name	Distri- bution	1 Op.			2 Op.		
		Body 1	Body2	Body 3	Body 1	Body2	Body 3
WM_GRIND1	Normal	17.00	25.00	15.00	11.00	15.00	10.00
WM_GRIND2	Normal	17.00	25.00	15.00	11.00	15.00	10.00
WM_GRIND3	Normal	17.00	25.00	15.00	11.00	15.00	10.00

Figure 7: The cycle time sheet with standard deviations.

#### 4.5.5 Workstation Disturbance in CMSD

Every resource instance in CMSD can have zero or more *resource characteristics*. Resource characteristics include *Mean Time Between Failure* (MTBF) and *Mean Time To Repair* (MTTR). The measures used in this case are however *Mean Time To Failure* (MTTF) and *Mean Down Time* (MDT). The reason for the use of MDT instead of MTTR is that MTTR is thought of as a measure of how long the actual repair work takes, while MDT also includes the waiting time for the repairman. The reason to use MTTF instead of MTBF is that the native data is of the former format.

Since the data content of these different measures are the same, new characteristics were added to CMSD to also support MTTF and MDT. In the XML-code below, the CMSD representation of the disturbance data used in this implementation is presented. All cycle time nodes are removed to shorten the example.

```

<Resource identifier="WM_11-160">
  <Description>Manual_sealing</Description>
  <ResourceType>station</ResourceType>
  <MTTF unit="second">
    <Distribution>
      <Name>exponential</Name>
      <DistributionParameter>
        <Name>mean</Name>
        <Value>20688</Value>
      </DistributionParameter>
    </Distribution>

```

## 5 DISCUSSION

The structures currently available in CMSD were designed primarily as parts of an information model that describe a job shop. CMSD originates in the effort known as *NIST Shop Data Model* (McLean et al. 2005). Hence, CMSD fulfills certain needs that are obvious when modeling a job shop, but sometimes are not sufficient or appropriate for modeling a flow shop, which is what this case study is about. However, the extensions made to CMSD in this implementation indicate that CMSD was created in such a way, that it should be straight-forward to extend it to support the modeling of flow shop operations in the future.

CMSD defines a set of related abstract structures, referred to as context definition structures, which define the high level relationships between all information in the information model. All other structures in CMSD either directly or indirectly specialize the context definition structures. All proposed XML-node structures presented in this paper were made according to the CMSD context definition. Only slight changes had to be made to support input data for conveyors and elevators. For workstations, on the other hand, considerable changes were made by associating cycle time with the resource definition instead of using the process plan class. This definition change is not an indication of a weakness in CMSD, but rather an indication that understanding of CMSD is essential before building an implementation model. In this case study, it would definitely have been easier to use the *process plan* structure to define cycle time, if a more suitable modeling approach had been chosen from the beginning. The completed model has evolved into a sophisticated, detailed, and extensive model that would have taken long time to modify to support the process plan structure.

While opening an XML document in ED, the whole content is parsed into memory. As documents grow big, the parsing seems to become very memory intensive. We recommend DES software developers to follow the development of XML tools and continuously upgrade their simulation packages to support effective XML data handling. The fact of memory consuming XML parsers also addresses a need for the CMSD information model to pro-

vide a compact but yet comprehensive representation of simulation data.

Anyone with modest experience with ED, basic understanding of XML, and CMSD knowledge would be able to write a script for a CMSD implementation. This finding is positive, since it will enable more people to understand and use the standard. However, a more useable solution would be if manufacturing application vendors could provide translators and APIs for writing and reading data to/from CMSD models.

Even though this case study had many purposes, none of the other purposes interfered with the CMSD implementation. This indicates that modelers who want to make a test implementation of CMSD do not necessarily have to set the CMSD implementation itself as the primary purpose, though it helps if the modeler has an understanding of the CMSD specification at project start.

## 6 CONCLUSION

There are often several equally valid ways to represent manufacturing information. If CMSD will be used as an exchange/input mechanism for model data, it is better to consider CMSD approach for defining manufacturing information in the early phases of simulation model development to avoid discrepancies. The development of the draft standard is still ongoing. Frequent update meetings are held within SISO and also at the Winter Simulation Conference. Participants are encouraged to provide input, requirements, and recommendations to further the CMSD effort. Please contact the authors for further discussion or information on the development of CMSD into a standard for DES and other manufacturing applications.

Volvo Car Corporation has an interface connecting the CMSD XML document to ED, which enables standardized model building for future factories. This also provides opportunity for non simulation experts to understand the input data structure, as well as automatic input data updates from other systems due to the standardized representation of data in CMSD.

## ACKNOWLEDGMENT

The funding for this research is granted by VINNOVA (Swedish Agency for Innovation Systems, integrates research and development in technology, transport and working life.) through the MERA-program.

## DISCLAIMER

The work described was also funded by the United States Government and is not subject to copyright. No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied.

## REFERENCES

- Banks, J., J. C. Hagan, P. Lendermann, C. McLean, E. H. Page, C. D. Pedgen, O. Ulgen, and J. R. Wilson. 2003. The future of the simulation industry. In *Proceeding of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 2033-2043. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2005. *Discrete-event system simulation*. 4th ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.
- Ericsson, U. 2001. *Discrete Event Simulation, The Truth*. Ph.D. thesis, Department of Production Engineering, Chalmers University of Technology, Sweden.
- Hopp, W. J., and M. L. Spearman. 1996. *Factory Physics*, Irwin, Chicago, Illinois.
- Incontrol 2007. The Enterprise Dynamics simulation software, Incontrol. Available via <http://www.incontrol.nl> [accessed July 1, 2007]
- Jacobs, J. H., L. F. P. Etman, E. J. J. van Campen, and J. E. Rooda, 2003. Characterization of Operational Time Variability Using Effective Process Times. *IEEE Transactions on Semiconductor Manufacturing* 16(3): 511-520.
- Johansson, M., and R. Zachrisson, 2006. *Modeling automotive manufacturing process*. Master's thesis. Department of Product and Production Development, Chalmers University of Technology, Gothenburg. Available via <http://www.tfd.chalmers.se/~johamarc/Report.pdf> [accessed July 1, 2007].
- Lee, Y. T., and Y. Luo. 2007. *Machine Shop Information Model Application, Next Step*. Technical report NISTIR 7388, National Institute of Standards and Technology, Gaithersburg, Maryland.
- McLean, C., Y. T. Lee, G. Shao, and F. Riddick. 2005. *Shop Data Model and Interface Specification*. Technical report NISTIR 7198, National Institute of Standards and Technology, Gaithersburg, Maryland.
- MSDN 2007. *Microsoft XML Core Services*. Microsoft Developer Network, Microsoft Corporation. Available via <http://msdn2.microsoft.com/en-us/library/ms763742.aspx> [Accessed July 1 2007]
- SISO 2007. *Draft core manufacturing simulation data information model part 1: UML model*. CMSD Product Development Group, Simulation Interoperability Standards Organization. Available via <http://discussions.sisostds.org/default.asp?action=9&boardid=2&read=39532&fid=24> [Accessed July 1, 2007]
- Skoogh, A., and B. Johansson, 2007. Time-consumption Analysis of Input Data Activities in Discrete Event



Simulation Projects, In *Proceedings of the Swedish Production Symposium 2007*, Gothenburg.

Thomke, S., and T. Fujimoto, The effect of "Front-Loading" problem-solving on product development performance, *Journal of Product Innovation Management* 17:128-142, 2000

Umeda, S., and A. Jones. 1997. *Simulation in Japan: State-of-the-art update*, Technical Report NISTIR 6040, National Institute of Standards and Technology, Gaithersburg, Maryland

W3C 2007. *Extensible Markup Language*, World Wide Web Consortium. Available via <http://www.w3.org> [accessed July 1, 2007]

Wheelwright, S. C., and K. B. Clark. 1992. *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. City of New York, New York: Free Press

## AUTHOR BIOGRAPHIES

**MARCUS JOHANSSON** is from Gothenburg in Sweden where he is employed as a project assistant at the Department of Product and Production Development at Chalmers University of Technology. He is currently a guest researcher in the Manufacturing Simulation and Modeling Group at the National Institute of Standards and Technology where he works with implementation and development of Core Manufacturing Simulation Data. He has a M.Sc. degree in the specialized discipline Automation and Mechatronics Engineering. His e-mail address is [johamarc@chalmers.se](mailto:johamarc@chalmers.se).

**SWEE LEONG** is a senior manufacturing engineer in the Manufacturing Simulation and Modeling Group at the National Institute of Standards and Technology (NIST) Manufacturing System Integration Division since 1994. Prior to joining NIST, he worked at Ford Motor Company, John Deere, and IBM on different factory automation projects. His research interests are in modeling and simulation activities for the manufacturing industries and engineering tools integration. Currently, Swee manages the Simulation Standards Consortium at NIST. He is Chairman of the Core Manufacturing Simulation Data Product Development Group. He received a Bachelor and Master Degrees in Industrial Engineering from Purdue University in West Lafayette, Indiana. He is a senior member of the Society of Manufacturing Engineers. His e-mail address is [leong@nist.gov](mailto:leong@nist.gov).

**ANDERS SKOOGH** was born in Floby, Sweden, 1980. He attended Chalmers University of Technology at Automation and Mechatronics Engineering, where he obtained his M.Sc. degree in 2005. Anders has industrial experience of Discrete Event Simulation from his former employment as logistics developer at Volvo Car Corporation. He is now

working as a PhD student in the field of Discrete Event Simulation at the Department of Product and Production Development, Chalmers University of Technology, Sweden. His email is [anders.skoogh@chalmers.se](mailto:anders.skoogh@chalmers.se).

**BJÖRN JOHANSSON** is an assistant professor at Product and Production Development, Chalmers University of Technology. His research interest lies in the area of discrete event simulation for manufacturing industries. He is interested in the specifics of modular modeling methodology for effective and swift model-building, which is accompanied by software development, user interfaces, and input data architectures. His email address is [bjorn.johansson@chalmers.se](mailto:bjorn.johansson@chalmers.se).

**FRANK RIDDICK** is a computer scientist in the Manufacturing Simulation and Modeling Group in The National Institute of Standards and Technology (NIST) Manufacturing Systems Integration Division. He has participated in research and authored several papers relating to manufacturing simulation integration and product data modeling. He holds a Master's Degree in Mathematics from Purdue. His email address is [riddick@nist.gov](mailto:riddick@nist.gov).

**PÄR KLINGSTAM** Ph.D. employed by Volvo Car Corporation, for the development and implementation of the Manufacturing part of the new Global Product Development System. He is also the project leader for the research project Conceptual Plant Development - FACTS. He received his Ph.D. from the Department of Production Engineering at Chalmers University of Technology. His main interests include DES and operational development. His email address is [pklingst@volvocars.com](mailto:pklingst@volvocars.com).

**Y. TINA LEE** is a computer scientist in the Manufacturing Simulation and Modeling Group at NIST. She joined NIST in 1986. Her major responsibility in recent years has been to develop information models to support various manufacturing application areas. Previously she worked at Contel Federal Systems and Sperry Corporation. She received her BS in Mathematics from Providence College and MS in Applied Science from the College of William and Mary. Her e-mail address is [leet@cme.nist.gov](mailto:leet@cme.nist.gov).

**GUODONG SHAO** is a computer scientist in the Manufacturing Simulation and Modeling Group in The National Institute of Standards and Technology (NIST) Manufacturing Systems Integration Division. He has participated in research relating to FMS, CIMS, and manufacturing simulation integration for many years. He holds a Master's Degree from University of Maryland at College Park. He is a Ph.D. student at George Mason University. His e-mail address is [gshao@cme.nist.gov](mailto:gshao@cme.nist.gov).