Proceedings of ASME 2013 International Mechanical Engineering Congress & Exposition
IMECE 2013
Nov 15–21, 2013, San Diego, CA, U.S.A.

IMECE2013–63546

TOWARD THE IDEAL OF AUTOMATING PRODUCTION OPTIMIZATION

J. Michaloski, F. Proctor
National Institute of Standards and Technology
Gaithersburg, Maryland

J. Arinez
General Motors
Warren, Michigan

J. Berglund
Chalmers University of Technology
Chalmers, Sweden

ABSTRACT

The advent of improved factory data collection offers a prime opportunity to continuously study and optimize factory operations. Although manufacturing optimization tools can be considered mainstream technology, most manufacturers do not take full advantage of such technology because of the time-intensive procedures required to manually develop models, deal with factory data acquisition problems, and resolve the incompatibility of factory and optimization data representations. Therefore, automated data acquisition, automated generation of production models, and the automated integration of data into the production models are required for any optimization analysis to be timely and cost effective. In this paper, we develop a system methodology and software framework for the optimization of production systems in a more efficient manner towards the goal of fully automated optimization. The case study of an automotive casting operation shows that a highly integrated approach enables the modeling and simulation of the complex casting operation in a responsive, cost-effective, and exacting nature.

Keywords

Optimization, discrete event simulation, modeling, automation, CMSD, key performance indicators

Nomenclature

ANSI	American National Standards Institute
B2MML	Business To Manufacturing Markup Language
CAEX	Computer Aided Engineering Exchange
CAD	Computer Aided Design
COM	Microsoft's Component Object Model
CMSD	Core Manufacturing Simulation Data
CNC	Computer Numerical Control
DES	Discrete Event Simulation
ERP	Enterprise Resource Planning
ISA	International Society of Automation
HTML	Hypertext Markup Language
KPI	Key Performance Indicators
PLC	Programmable Logic Controller
MES	Manufacturing Execution Systems
MTBF	Mean Time between Failure
MTTR	Mean Time to Repair
MTTP	Mean Time to Processing
OEE	Overall Equipment Effectiveness
OEM	Original Equipment Manufacturers
PLC	Programmable Logic Controller
SDK	Software Development Kit
UML	Unified Modeling Language
WBF	World Batch Forum
XML	eXtensible Markup Language
XSD	XML Schema Definition

INTRODUCTION

Although manufacturing optimization tools can be considered mainstream technology, most U.S. manufacturers do not take full advantage of such technology because of time-intensive deployment and the incompatibility of factory and optimization tool data representation. To achieve automated analysis of production data, all aspects of the manufacturing operation must be

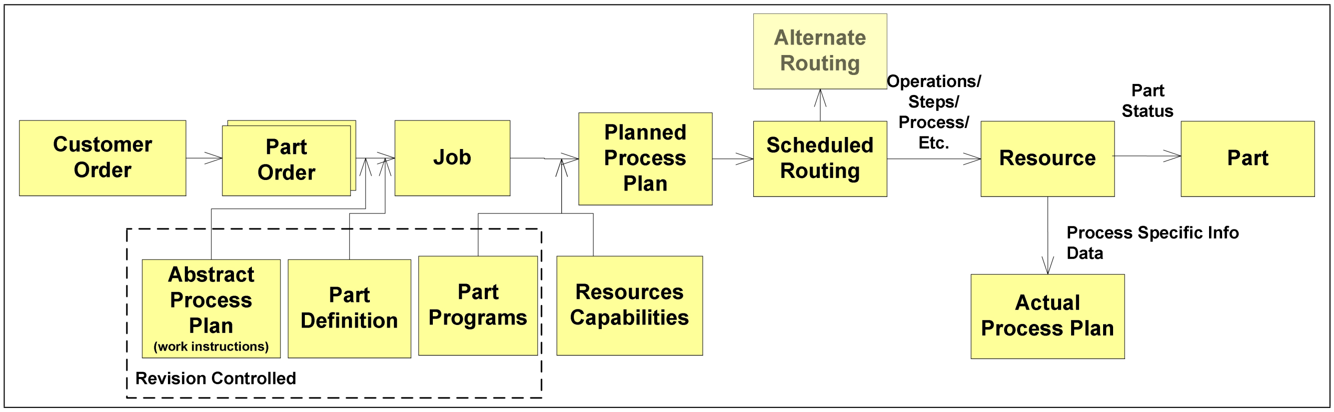

FIGURE 1: Job Life Cycle

included: design, production, and maintenance. Simulation offers a controlled environment to study the large scale interaction of machines and processes under different conditions. Simple parameters adjustments can be run through simulation time sequences to predict the impact of potential changes. Yet, there is a lack of decision-making strategies for optimizing manufacturing using simulation.

Fundamental to a smarter understanding of a process is the ability to measure it. Currently, prescribed methods in industry are often related to lean manufacturing concepts and include treasure hunts, value stream mapping, Six Sigma, and Kaizen events [1]. Most of these methods rely on empirical observation and basic analysis. However, informative, accurate, and timely shop-floor production data should be considered vital to understand a process. Only with accurate and timely data from the shop-floor can analysis be suitably done to eliminate waste and inefficiencies.

Though many companies cannot afford sophisticated factory data collection, the decreasing cost of networks and computers is continually lowering the financial threshold of acquiring plant information systems that can perform real-time data collection and archive the operational behavior of their PLCs, automation, and other equipment. Increasingly, companies collect process data from the various control and supervisory systems on the plant floor and store the data in databases. This work seeks to use DES to build, test, and optimize an integrated production system.

Core Manufacturing Simulation Data (CMSD) [2] provides the information model in which to collect data from one or more different manufacturing domains such as process planning, scheduling, inventory management, production management, or supply chain management The goal of this paper is to study the current state of the production operation and then propose an approach to improve the production operation by quickly modeling the process, ascertaining and mapping different elements of the production data, and incorporating the modeling results in production data to improve manufacturing.

Section 2 analyzes the purely manufacturing problem of modeling “job-driven” production operation. Section 3 discusses the manufacturing data in this “job-driven” production operation that are covered by the CMSD coverage of “job-driven” production specifications and then separates the CMSD production operation into manufacturing operation, shop floor data, and job components to streamline operation and reusability. Section 4 introduces the concept of CMSD optimization constructs, their methodology, and their application to for analysis. Section 5 investigates a case study of a casting production operation at a General Motors plant that uses CMSD and its optimization extensions. Finally, a discussion on the results and future directions will be given.

PROBLEM STATEMENT

The enterprise domain is responsible for processing customer orders and deciding whether to make or buy parts. The processing of customer orders triggers the creation of a unique part order (or workorder) within the manufacturing execution system (or production system). The creation of a part order (e.g., 10 front bumpers, 12 side panels) is incumbent on the knowledge of how to build the parts on some set of equipment (abstract process plan), the PART definition (revision, part and quality), and part programs (how to make the part). All these are combined into jobs to make the parts (and may be scheduled). Of course, a job could describe not only production, but quality, inventory, or maintenance tasks. But in our case, we first want a complete model of production operation so we will initially focus on production.

In Figure 1, a PART is assumed to be a finished product that was produced or that can be used in production activities as raw material or a work in–progress component. Many different kinds of information can be specified for a PART, such as, information about the production status of a PART, the named category of

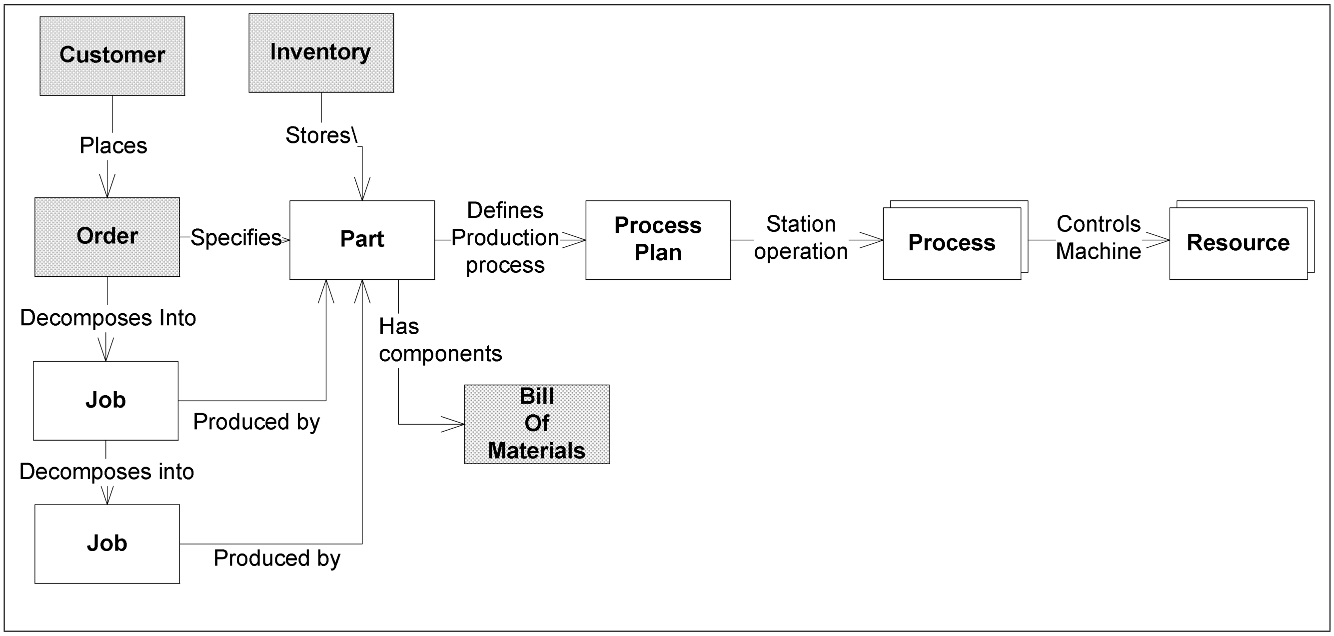

FIGURE 2: Overview CMSD Information Flow

parts that a specific PART belongs to, the sub-component parts used to create this part, the process that can be used to create the part, and some basic characteristics of this part. For a PART, there is also a PROCESSPLAN, which holds a list of PROCESS objects in some sequence. This sequence can be used to describe the routing of a PART object. Each PROCESS class holds a list of the PART Type that it requires for processing, as well as the products it creates. The PROCESS class also holds a list of the resources required for processing. These resources can include physical locations of machines and mobile resources (such as laborers or work fixtures).

In the scenario, a customer order enters the system and triggers a part order. To match our case study discussed later, we will assume that the part order will contain only one type of PART, a fixed number of parts will be made per day, and that this PART will be made based on an abstract process plan, a PART definition (revision and associated resources and part programs) and part programs (or recipes) to make PART on a series of equipment. After all this information is collated, a job will be generated containing the PART and a process plan to describe the potential sequence through a set of resource types. This process plan will serve as the scheduled routing of the raw material to become a PART. The alternate routing serves in case to reroute if a broken equipment arises - however, in our case study, this is not a concern as the primary equipment will be used and fixed if broken. Using shop-floor data, the simulation will then make the PART and buffer the finished PART based on the data.

In Figure 1, the concept of a resource breaking down is based on the production task – even though maintenance information is necessary to differentiate the equipment and the mechanic must replace or fix the broken equipment even though from a production standpoint, the piece of equipment and time to break down and time to repair are the important performance indicators.

There is an overloading of terminology. The concepts of routing, alternative process, step, and sequence are but a few of the overloaded terms.

CMSD MODELING

Currently in manufacturing, it can be quite difficult to extract knowledge into a common format [3]. Digital CAD "drawings" are used for the facilities and plant layout, process plans are contained in data bases, and workflow data is contained in spreadsheets, so that various pieces of production knowledge may be distributed throughout the enterprise. Often, storage of the production knowledge is tailored for human comprehension, i.e., spreadsheets, that are not as conducive for digital sharing.

A neutral format to represent this data is desirable. CMSD is a freely available standard specification that would allow the translation of different data formats from numerous related domains into a manufacturing domain-specific representation suitable for analysis. The primary use of CMSD is to generate the simulation model by using a suitable model representation of the physical system. In this approach, all CMSD information required must be acquired at the point of creating the simulation model.

Figure 2 addresses the problem using the CMSD specification to address issues related to information management and manufacturing simulation development. Please note, shaded CMSD boxes representing manufacturing applications that are part of the production scenario boxes are out of scope for this paper. The CMSD entities defined in this framework represent a core set of the manufacturing entities and relationships needed for manufacturing simulation. CMSD offers representations for many categories of manufacturing information, but in our case we were most interested in:

Resource - describes equipment that performs manufacturing activities. Resources in the CMSD are used to represent stations, machines, cranes, employees, tools, and fixtures. (for this iteration we assumed no trained personnel were required.)

Part - provides a means to specify the characteristics of the materials and subcomponents that are used to make end products.

Process plan - specifies the set of production activities needed to transform materials and subcomponents into finished products. Each process plan is built from process steps (with associated resource(s)) that must be executed for the part to be finished.

Process - defines a manufacturing activity or group or manufacturing activities that encompass a detailed strategy for creating a part. The process will most likely contain information that describes the resources that will be used, the parts that will be consumed and produced, the sequence in which resources will be used, and the sequence of activities within a group of activities.

Job - defines normal, maintenance or repair operation, but in our case the job represents normal manufacturing and is the central construct of the system. Each job (assuming it came from customer order as described earlier), would generate an appropriate number of parts into "spawned" jobs (type of job) and under each spawned job contains the part knowledge exhibited within the job e.g., process plan, the resources, etc. The spawned job would contain a copy of the initial job that described all the parts and quantities.

Jobs typically define complex production work items and can involve activities at multiple stations that ultimately produce parts. Processes are lower level work items that are typically performed at a single workstation or area within the shop. The basic fulfillment of a "spawned" job is to know its process plan, its current process step within the process plan (at what process), and its processing status.

The goal of CMSD is to provide a neutral framework that facilitates the creation of collections of related manufacturing information suitable for use in the creation or enhancement of manufacturing simulations and other manufacturing applications. We found that CMSD would be better served if supplemented by a 1) more incremental approach to file development, 2) more feedback and separation of manufacturing operation, and 3) intrinsic language to describe optimality in the system. Figure 3 shows the

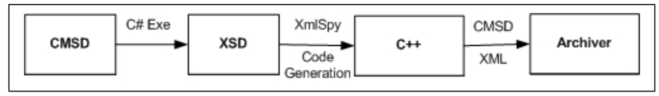

FIGURE 3: Overview CMSD Archiving and Code Generation

National Institute of Standards and Technology (NIST) sequence of operations to turn CMSD information model into an achievable entity. First, although designed in UML, CMSD has a C# or .Net Framework mapping in which to read CMSD files. Using the EXE, the xsd.exe software tool from Microsoft generates an XSD. This XSD gave a schema for the CMSD information model (although CMSD had Schematron and other representations, no XSD was available.) Next, the commercial tool XMLSpy was used since it provides facilities to load XSD documents, validate the XSD files, and then generate C++ archival (read and writing from files) code based on a XML parser. For the XMLSpy approach, we generated code for XML reading and validation using Microsoft MSXML technology. One area that was troublesome is the mating of XML to some C++ internal representations. To this end, we maintained CMSD definitions in a simple reflection C++ list that maintained the relationship between XMLSchemas and the model for CMSD and MySQL archiving.

Now with the incremental loading of manufacturing information in XML, the CMSD can be used to incrementally grow and develop manufacturing information models. The XML in CMSD is not endemic to the specification, as such, explicit enumeration of this feature is desirable. For example, we used this incremental feature to separate the production measurement from the production operation. Thus, one CMSD file was used for describing a PART and its defining process plan. Another CMSD file was developed to describe the resource operation with KPI to describe the length of buffers, the failure rate of the equipment, and the time for processing a unit. The goal of the combined CMSD was to replicate the original data output from of the manufacturing system (via SQL queries on a database.) As pointed out, we modified CMSD to allow the merging of factory information from multiple files before simulating a production line. Figure 4 shows use of a CMSD resource linked to an existing CMSD reference, which could be extended to merge the CMSD manufacturing model, and allow modularization of data. Thus, manufacturing operations, manufacturing data, and the job could all be separated and then input simultaneously to create a Factory Model in an incremental mode.

Within our ProcessPlan we included CMSD Resource to describe equipment or groups of equipment that perform manufacturing activities. A CSMD resource may be processed on a particular layout for one manufacturing configuration for a certain

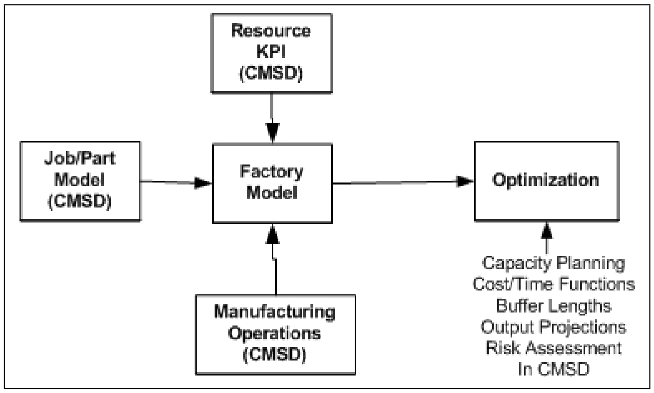

FIGURE 4: Manufacturing Operations represented with incremental CMSD

amount of time, and then used in a different layout for another manufacturing configuration.

```
<CMSDDocument>
 <DataSection>
  <Resource>
   <Identifier>SMCO:LINE1_PS_CAST1_ELV1</Identifier>
   <Name>LINE1_PS_CAST1_ELV1</Name>
   <ResourceType>elevactor:</ResourceType>
   <Description>Elevator1</Description>
  </Resource>...
```

Although a CMSD job has the ability to reprogram the sequence of operations of the manufacturing equipment, this reconfigurability requires a different CMSD job strategy and a more dynamic layout of the resources in the manufacturing operation. Before delving into CMSD optimization of resource allocation, we will assume that part/jobs define a static layout of resource. Each resource can then add or subtract parameters to attempt to optimize the manufacturing operation.

```
<CMSDDocument>
 <DataSection>
  <Resource>
   <Identifier>SMCO:LINE1_PS_CAST1_ELV1</Identifier>
   <Property><Name>InQueue</Name><Value>1</Value></Property>
   <Property><Name>Mtbf</Name> <Value>394</Value></Property>
   <Property><Name>Mttr</Name<Value>85.8</Value></Property>
   <Property><Name>Mttp</Name> <Value>64.3</Value></Property>
  </Resource>
 </DataSection>
</CMSDDocument>
```

Development of a DES model is a large undertaking, but with the incremental CMSD approach, deployment can be handled in phases so that one can incorporate increasingly detailed parameterization. At first, DES manufacturing operation can start with the basic manufacturing operations to build parts, process plans, processes, and resources. Next, a CMSD file (possibly generated from live data sources) can add key performance indicators (KPI) such as cycle time, breakdown, and buffer sizes. Later we will discuss an approach to add optimization criteria as part of the CMSD framework.

RELATED WORK

Discrete Event Simulation (DES) has mainly been used as a production system analysis tool to evaluate new production system concepts, layout, and control logic [4]. For the determination of productivity, the use of DES is considered critical to developing a production and benchmarking methodology. In manufacturing, DES simulates a real or virtual model of production based on statistical characterization of a manufacturing process, such as cycle time, idle time, and failure rates. Once developed, the DES model can then be used to predict outcomes given different parameterization scenarios. DES can also be used in the design of new facilities using historical production data to ensure modeling accuracy.

The structures currently available in CMSD are a continuation of earlier work done building an information model that describes a job shop. The long–term objective of the CMSD information modeling effort is to develop a standardized representation that allows for exchange of information in a machine shop environment. From this perspective an information model must satisfy the following needs: support data requirements for the entire manufacturing life cycle, enable data exchange between simulation and other manufacturing software for machine shops, provide for the construction of machine shop simulators, and support testing and evaluation of machine shop manufacturing software.

CMSD originates in the effort known as National Institute of Standards and Technology (NIST) Shop Data Model [5]. Evaluating and testing of the CMSD information model with real world production scenarios were done in order to further develop and validate the CMSD standard development efforts. CMSD has been used within a variety of applications: standard modular simulation in semiconductor wafer fabrication systems [6]; generic simulation of automotive assembly for interoperability testing [7]; homeland security modeling and simulation [8]; incident management simulation and gaming [9]. NIST has applied CMSD in an automotive assembly plant model in order to create data-driven simulators across the manufacturing hierarchy, extending from the supply chain network level to a process on the production floor [7]. Fournier discusses representing operations from a shop floor and retrieving real–time data from the shops, and then using CMSD as a neutral front-end platform in which to develop DES back-ends for Delmia QUEST, Rockwell Arena, ProModel simulation tool, and Flexsim simulation tool [10].

Existing XML work overlaps the overall CMSD manufacturing work, and these standards efforts can be used to complement the CMSD work - as XML is reputed to be more neutral than other representations. B2MML is a freely available XML implementation of the ANSI/ISA 95 family of standards [11], which integrates ERP to MES systems using XML schemas standards. AutomationML (Automation Markup Language) is an XML- based open standard for the storage and exchange of plant engineering information [12,13] geared for deployment from the moment an automation system is conceived. AutomationML uses a full complement of standards: CAEX topology with IEC 62424 [14], geometry and kinematics with COLLADA [15], programming logic with PLCopen XML [16]. The Digital Factory is a widely supported initiative that has chosen AutomationML as the intermediate format.

As prescribed by the CMSD effort, factory data collection is one critical part of the DES modeling and to subsequent factory optimizing operations. Several challenges must be addressed for DES to become a completely autonomous endeavor. Today, over 30 % of the cost of developing a Discrete Event Simulation model is associated with data input [17]. Nils Bengtsson et al. point out that DES is a time consuming and costly process and that it requires a methodology to identify and collect data, and then use sophisticated software to extract and process the data [18]. The Factory Analyses in Conceptual Phases Using Simulation (FACTS) has focused on developing new and modified production systems, with the results of their experiences used to enhance and evaluate the CMSD standardization process [19].

The authors have investigated the automated integration of factory data with automatically generated models of operations that are required for optimized analysis. From our research, it is clear that this integration must be accurate, timely, and cost effective. Previously, related work on integrating manufacturing process and energy data has been done, and [20] discusses the difficulties in integrating and simulating process energy. Similar work was done to facilitate simulation models to combine automated raw data collection and automated data processing [21–23].

OPTIMIZATION WORK

A large number of factors are critical in effectively modeling a production system. Manufacturing systems involve a number of interrelated elements, including equipment strategy, number of product options, material handling systems, system size, process flow configuration, processing time of the operations, system and workstation capacity, and space utilization. The model must be combined with other constraints such as unpredictable machine breakdowns, varying operational requirements, schedule variation, and different production demands. The following lists some of the basic optimization parameters one would find in a manufacturing system:

- Improve uptime, availability
- Minimize waiting time constraints
- Raise system performance and reduce production costs
- Hedge against the risk of a shortage or sudden price increase
- Anticipate expected output from current level of resources
- Increase income, lower cost, and reduce use of tied-up capital

The following lists some of the other optimization parameters that one would find in a manufacturing system, but are out of scope (and can be found in [20]): improve product delivery performance, product quality, OEE; providing better information how resources should be used ; minimize resource contention and process dedication; improve supply chain/inventory or other non-covered manufacturing areas; among others.

In industry, a primary optimization criterion is to anticipate whether the resources will be in place to handle an increasing number of parts as the number of customer orders increase. Capacity planning is the process of determining the production capacity needed by an organization to meet changing demands for its products. A disparity between the capacity of an organization and the demands of its customer's unplanned changed request is detrimental, either in under-utilized resources or unfulfilled customers.

A desired increase in production outputs from the current baseline will require capacity planning in order to determine if there are the necessary resources, operators, and schedule in order to fulfill the additional output. In CMSD, so far reconfigurability of the system by allowing the addition/removal of a new/existing resource has not been covered. CMSD provides for a ResourceGroup, which we call a Cell that provides one level of abstraction with a CMSD Process Plan. One can add or remove Resources (already described with operational data from CMSD) to a Cell to change to responsiveness of a cell within the Process Plan. Clearly, one cannot remove all the Resources from a Cell or that Cell would not be able to carry out its mission – for example, turn a PART. Thus, it would be expected that removing slower Resources and adding faster Resource to a CMSD Cell would improve performance.

Thus, the definition of a cell and its parameterization is necessary to optimize performance and is captured below in the sketch of CMSD providing for a Resource and a Cell (e.g., ResourceGroup) containing the Resource.

```
<CMSDDocument>
 <DataSection>
  <Resource>
   <Identifier>SMCO:LINE1_PS_CAST1_ELV1</Identifier>
   <Name>LINE1_PS_CAST1_ELV1</Name>
   <ResourceType>elevactor:</ResourceType>
   <Description>Elevator1</Description>
  </Resource>
  ...
  <Resource>
   <Identifier>Plant1:Cell1</Identifier>
   <Name>Cell1</Name>
   <ResourceType>Station</ResourceType>
   <GroupDefinition>
    <ResourceGroupMember>
     <ResourceIdentifier>SMCO:LINE1_PS_CAST1_ELV1
     </ResourceIdentifier>
    </ResourceGroupMember>
   </GroupDefinition>
  </Resource>
...
```

Currently, CMSD is a literal information modeling language, and once a definition is in place, it is assumed to be statically defined and hence immutable. One of the features that would make CMSD more powerful is the ability to dynamically change how a PART is made and to select from the best alternative. This would require CMSD support for changing a PART and its process plan and all its processes to add, merge, or subtract resource from a cell (which is a container of CMSD Resources with equal likelihood of processing the PART.) As such, CMSD does not directly support the concept of providing an objective function in which to streamline DES searching. Without limiting searching time, optimization could run endlessly and provide impractical answers. At this time, only the following simple functions are introduced: MAXIMIZE, MINIMIZE, TREND, RANGE, ADD, REMOVE, etc. The authors are working on a complete vocabulary to simplify optimization.

```
<Job>
 <Identifier>Job1</Identifier>
 <PlannedEffort>
  <PartsProduced>
   <PartType>
    <PartTypeIdentifier>Part1-12345</PartTypeIdentifier>
   </PartType>
   <PartQuantity>MAXIMIZE</PartQuantity>
   <EffectiveEndDate>2013-04-15T00:00:00</EffectiveEndDate>
  </PartsProduced>
 </PlannedEffort>
 <Resource>
  <Identifier>Plant1:Cell1</Identifier>
  <Name>Cell1</Name>
  <ResourceType>Station</ResourceType>
  <GroupDefinition>
   <ResourceGroupMember>
    <ResourceIdentifier>
    <REMOVE>
    SMCO:LINE1_PS_CAST1_ELV1
    </REMOVE></ResourceIdentifier>
    <ResourceIdentifier>
    <ADD>
    SMCO:LINE1_PS_CAST1_ELV2
    </ADD>
    </ResourceIdentifier>
   </ResourceGroupMember>
  </GroupDefinition>
 </Resource>
</Job>
```

In this example, all the optimization parameters are grouped under the <JOB> CMSD parameter, since we are performing Capacity Planning and would like to study whether the system has sufficient resources to satisfy the customer. Under the <JOB> setting, "2013-04-15T00:00:00" is a timestamp indicating the end of Capacity Planning given the current Cell configuration with the current set of Resource(s). MAXIMIZE is a CMSD optimization function to indicate that the Capacity Planning is to run and compute the maximum number of parts that would be created. <REMOVE> is a CMSD function that removes a resource from a Cell and <ADD> is the complementary add of a resource to a CMSD Cell.

From a simple Capacity Planning analysis, the questions to answer include, "Can we achieve the goals with our current setup? (yes or no) What do we need to do if we need to ramp up production? Are there any optimization strategies we can incorporate to increase output without additional cost?" Clearly, a separate <JOB> CMSD description could be developed to add more resources to a Cell if we are not achieving the goals as outlined. Below a modification to MTBF is analyzed to see the effect on Capacity Planning.

```
<Job>
 <Identifier>Job1</Identifier>
 <PlannedEffort>
  <PartsProduced>
   <PartType>
    <PartTypeIdentifier>Part1-12345</PartTypeIdentifier>
   </PartType>
   <PartQuantity>MAXIMIZE</PartQuantity>
   <EffectiveEndDate>2013-04-15T00:00:00</EffectiveEndDate>
  </PartsProduced>
 </PlannedEffort>
 <Resource>
  <Identifier>SMCO:LINE1_PS_CAST1_ELV1</Identifier>
  <Name> Mtbf </Name> <Value><High>394</High><Low>194</Low>
  </Value>
 </Resource>
</Job>
```

To determine any optimization strategies to increase output, we could look at improving the performance of the resource under the cells by improving the reliability of the machines, selecting best buffer sizes or improving the cycle time. In this CMSD <JOB> example, the MTBF has been modified with a <HIGH> to <LOW> range to study the trending of MTBF on the capacity performance of `SMCO:LINE1_PS_CAST1_ELV1`. The CMSD concept of <LOW> should match expectations of performance or the analyst has wasted their time.

Not covered in this optimization analysis, is the concept of shifts, which forms the fundamental metric for planning production, as the machine utilization is based on capacity planning. In order to assign machines, the total time must be considered in the context of shifts. Thus, 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. Also out of scope for the optimization computation is the transportation time and cost (moving of material) and any inventory costs (includes all finished product not being processed). Further, we will assume that for much of our case study, the operation is limited to buffering and not to moving resources to better perform the processing.

CASE STUDY

DES analysis was applied to a case study of an automotive precision casting production facility. Figure 5 shows a high-level overview of the precision casting process. The molten aluminum process is responsible for melting the aluminum, refining the melt, and adjusting the molten chemistry. Once molten, the aluminum is degassed, leveled, and laundered to remove deleterious gases before being tapped to flow into cores. Cores are made of sand which is poured into molding machines to create the contours of the casting, pressed and heated to bind the sand. Since the sand casting process is an expendable mold metal casting process, the core process builds a new sand core for each casting. Overall, core parts are molded from sand and binding elements, assembled into the engine block core, and then dried before casting. The casting and finishing process is where the molten aluminum flows into the sand cast core, after which, the casting is cooled and then casting sand is removed from around the now solidified aluminum engine block by shakeout, trim, and degating operations.

Some observations are in order. Because of intellectual property issues, representative data will be given, not actual performance data.

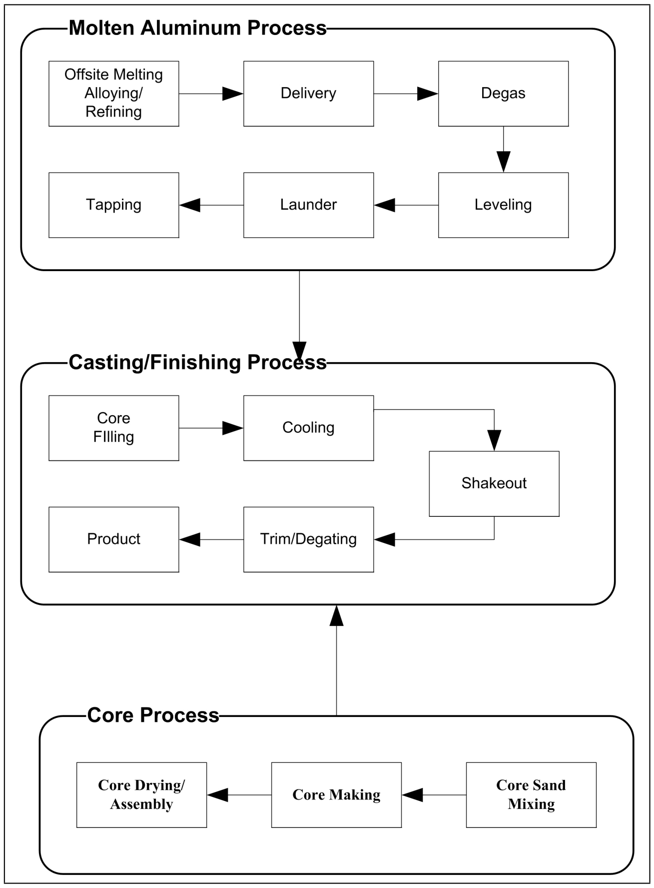

FIGURE 5: Overview Precision Casting

However, to ensure applicability to manufacturing problems, real data from the shop-floor was used as input to any DES analysis and optimization. The analysis was limited to data already being collected by the plant's production system, so no new data was available for calculation. Within a CMSD Process step, cycle time and equipment fault data was collected and easily adapted into CMSD Process KPI parameters (CMSD property values) that were incrementally added as operational data via a CMSD XML file.

The goal of the GM/NIST work was to analyze the manufacturing operations of precision sand casting and use DES modeling to derive manufacturing cost projections based on real factory floor data. The studied General Motors sand casting production is a large process, with hundreds of pieces of electrical equipment being controlled – robots, conveyors, elevators, sand core making machines, saws, etc. The eventual goal of the work is to completely model the casting production although its size necessitated narrowing the initial analysis scope to finishing.

Using a commercial DES software package, a model was developed to correlate the production activity with the process energy consumption. This was not straightforward as the DES package did not inherently support manufacturing sustainability concepts, but correlation of the data by separating the integration into production and process energy submodels was possible. Berglund et al. presents a cleaner, but less portable, DES analysis of the General Motors precision sand casting operation [24]. Although clearly helpful, it would be preferable if the DES development was easier, timelier, and more automated. The cost and manual effort in DES development would be more beneficial if it was a remunerative effort, or it is not worth doing in the first place. Hence, the motivation to automate the DES development, deployment, and analysis process was seen as crucial to success in the project.

In our recent experiments, the model characterization was as follows. The manufacturing operation was a given by the layout of the equipment through the CMSD Process Plan and in each step a Process had a CMSD Cell with one or more resources. In the sand casting facility, Cells were limited to one resource to match the expectations of the facility. The DES model required buffers and sizes and we assumed each buffer was part of a resource, so that buffer sizes could change (and each resource could have a growing input or output buffer), but was bounded by the buffer size as would be expected by the shops (as were reflected with marginal change captured by our CMSD optimization routines). Finally, our manufacturing operation assumed only one part was produced within the factory, although the number of parts could vary from day to day. Because of buffering, N parts could in fact be active. We found that one part implied a static configuration of equipment in production, and therefore, CMSD Process Plans and accompanying Process steps (and layout) were fixed. Tests to validate the automatically generated sand casting data were done before proceeding to the CMSD optimization exercises.

DISCUSSION

The primary reason for building DES simulations is to provide support tools that aid the manufacturing decision-making process. It would be unreasonable to expect a large car company to change its steady-state production based upon the findings of a DES system. As would be expected, DES simulations are developed to be a part of a case study commissioned by the manufacturing management to address throughput and related factory performance issues. Again, it would be unreasonable to expect a large car company not to have undergone some optimization of buffers, equipment layout, etc. before assembling the production line. Further, day-to-day matters will become routine and change itself can be difficult [25]. Machines cannot be swapped out, production line buffers are relatively fixed, and overall only minor changes can be undertaken.

Manufacturing operations revolve around the production of parts, i.e., the fabrication of parts from raw materials such as metal or plastic. Undeniably, the need to speed up the DES modeling process and reduce the level of effort required in the construction of a simulation model is imperative to success for any manufacturer. Today simulation analysts typically code their models from scratch and build custom data translators to import required data. From our discussion, CMSD as augmented with optimization parameterization could improve DES turnaround and on the whole improve the applicability of simulation technology to the manufacturing industry. Standard interfaces such as CMSD (especially CMSD open source solutions) could help reduce the costs associated with simulation model construction – and thus make simulation technology more affordable and accessible to a wide range of potential industrial users [26].

NIST has developed a Virtual Factory Testbed with a stated goal to automate the generation of DES models from CMSD and then run simulations based on the factory described in CMSD with automated data acquisition. The first mapping of CMSD was to Rockwell Arena and was facilitated by using the Microsoft COM Automation feature of Arena – to generate equivalent Arena objects found in the CMSD and then using COM to run the simulation replications. Arena provides some standard DES features but any additional modeling for part generation, statistics collection, and resource sharing/scheduling must be done manually. Given the known portability of CMSD [10], we have elected to study the use of CMSD as a backbone for manufacturing simulations, and since CMSD is based on XML we will continue to perform such analysis with modified versions of CMSD. Given that NIST itself only has a small manufacturing job shop, the use of CMSD is applicable beyond this scope and will greatly assist in the ability to measure and optimize enumerable manufacturing scenarios.

In summary, this paper has presented an approach to develop CMSD optimization models which can be used to evaluate the performance of a given manufacturing system. We have assumed many constraints are inherent from the start, but that the development of a time-responsive DES system facilitated on CMSD and its optimization criteria will assist plant personnel in understanding their shop activity. Where applicable to our case study, the purely coded DES analytic results from our CMSD backbone and CMSD optimization extensions can be found at our code repository mentioned in the Introduction.

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 the National Institute of Standards and Technology or General Motors, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

[1] U.S. Environmental Protection Agency. Lean energy toolkit. http://www.epa.gov/lean/toolkit/LeanEnergyToolkit.pdf.

[2] Simulation Interoperability Standards Organization (SISO), 2010. Core manufacturing simulation data (CMSD) – siso-std-008-2010. Accessed December 3, 2011. `http://www.sisostds.org`.

[3] Lanz, M., and Tuokko, R., 2009. "Generic reference architecture for digital, virtual, and real representations of manufacturing systems". In Proceedings of the Indo-US Workshop on Designing Sustainable Products, Services, and Manufacturing Systems.

[4] Heilala, J., Montonen, J., Jarvinen, P., Kivikunnas, S., Maantila, M., Sillanpaa, J., and Jokinen, T., 2010. "Developing simulation-based decision support systems for customer-driven manufacturing operation planning". In Proceedings - Winter Simulation Conference.

[5] McLean, C. R., Riddick, F., and Lee, Y. T., 2005. "An architecture and interfaces for distributed manufacturing simulation". *Simulation,* ***81***(1).

[6] Ramirez-Hernandez, J., Li, H., Fernandez, E., McLean, C., and Leong, S., 2005. "A framework for standard modular simulation in semiconductor wafer fabrication systems". In Proceedings - Winter Simulation Conference.

[7] Kibira, D., and McLean, C. R., 2007. "Generic simulation of automotive assembly for interoperability testing". In Proceedings - Winter Simulation Conference.

[8] Jain, S., and McLean, C. R., 2009. "Recommended practices for homeland security modeling and simulation". In Proceedings - Winter Simulation Conference.

[9] Jain, S., and McLean, C. R., 2008. "Components of an incident management simulation and gaming framework and related developments". *Simulation,* ***84***(1).

[10] Fournier, J., 2011. "Model building with Core Manufacturing Simulation Data". In Proceedings - Winter Simulation Conference.

[11] Boulonne, A., Johansson, B., Skoogh, A., and Aufenanger, M., 2010. "Simulation data architecture for sustainable development". In Winter Simulation Conference.

[12] Drath, R., Lüder, A., Peschke, J., and Hundt, L., 2008. "AutomationML - the glue for seamless automation engineering". In ETFA.

[13] Schleipen, M., and Drath, R., 2009. "Three-view-concept for modeling process or manufacturing plants with AutomationML". In IEEE International Conference on Emerging Technologies and Factory Automation.

[14] International Electrotechnical Commission, 2008. IEC 62424 representation of process control engineering - requests in P&I diagrams and data exchange between P&ID tools and PCE–CAE tools.

[15] Arnaud, R., and Barnes, M. C., 2006. *Collada: Sailing the Gulf of 3D Digital Content Creation*. AK Peters Ltd.

[16] Estevez, E., Marcos, M., Lüder, A., and Hundt, L., 2010. "PLCopen for achieving interoperability between development phases". In IEEE International Conference on Emerging Technologies and Factory Automation.

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

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

[19] Johansson, M., Johansson, B., Skoogh, A., Leong, S., Riddick, F., Lee, Y. T., Shao, G., and Klingstam, P., 2007. "A test implementation of the core manufacturing simulation data specification". In Proceedings - Winter Simulation Conference.

[20] Arinez, J., Biller, S., Lyons, K., Leong, S., Shao, G., Lee, B. E., and Michaloski, J., 2010. "Benchmarking production system, process energy, and facility energy performance using a systems approach". In Performance Metrics for Intelligent Systems Workshop (PerMIS '10), IEEE.

[21] Skoogh, A., Michaloski, J., and Bengtsson, N., 2010. "Towards continuously updated simulation models: combining automated raw data collection and automated data processing". In Winter Simulation Conference.

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

Engineering Technical Conferences, ASME.

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

[24] Berglund, J., Michaloski, J., Leong, S., Shao, G., Riddick, F., Arinez, J., and Biller, S., 2011. "Energy efficiency analysis for a casting production system". In Proceedings - Winter Simulation Conference.

[25] Goldratt, E., Cox, J., and Whitford, D., 2004. *The Goal: A Process of Ongoing Improvement*. Gower.

[26] McLean, C., Leong, S., Zimmerman, P., Harrell, C., and Lu, R., 2003. "Simulation standards: current status, needs, and future directions". In Proceedings - Winter Simulation Conference, Vol. 2.