Abstract: This paper describes a framework for Manufacturing Execution System (MES), based on the concept of object orientation. The framework, OpenMES, supports an approach that reduces the complexity of constructing an MES, by supporting a step-by-step manner. The paper includes an evaluation of the framework based on experiences in a variety of applications that were developed with it. OpenMES is connected to a B2B framework and hence, agile global manufacturing such as sharing production data among sales division through the production division and rapid response over a supply chain can be realized.

1. Introduction

This paper describes a framework for Manufacturing Execution System (MES) in the assembly and machining industry. The framework, OpenMES, is developed on the basis of the concept of the object-oriented framework with considerations of issues in a system construction of MES.

This OpenMES framework in this paper is a MES framework for the discrete process developed by IBM Japan, and is based on “Open MES Application Framework” [KAWA99, HORI99, OKANO99] as a reference model which was developed and validated by Manufacturing Science and Technology Center (MSTC).

The framework allows a user to construct a target system with necessary customization and assembly of provided software components based on MES model, which is a template of the target system. It aims at constructing MES with shorter implementation time and lower cost.

The framework was applied to some real systems. One of them is installed at the Japan Society for Promotion of Machine Industry as a middleware for the research prototype known as “ASIST” of GLOBEMEN Work Package 1 (Sales and Service). GLOBEMEN is an international research consortium focusing on the global manufacturing and conducted by Australia, EU, Switzerland, and Japan.

2. Issues in MES Development

Issues in the construction of MES are discussed here from the viewpoint of a system development.
Effective Response to Unique Requirement
Development of MES in machining and assembly process often becomes one-of-a-kind production for correspondence to characteristics of production site. However, the resulting costs and long lead time in development are increasing becoming unacceptable to industry.

Consolidation of Connection among Applications
MES is a complex system consisting of various applications, and it is often introduced as divided multiple systems by application area. These applications may be introduced in different timing. In such a system, connection among applications tends to be insufficient.

Real-timeliness of Information
Recent production systems are required correspondence to complex production methodology such as various kind small production or variable kind variable production rather than small kind mass production. For this purpose, information of production site is required to be updated in minute order or second order. Traditional distribution of work orders and collection of work results based on paper sheet do not meet the requirements. Construction of an on-line system updating digitalized information in real time becomes indispensable.

Information Sharing and Collaboration
In order to drive optimization of a whole production system by introduction of MES, collaborative environment for sharing various kinds of information accumulated in production site and for solving problem by related organization are required. For this purpose, easy access method to the information as well as digitalization of information in production site should be realized.

3. Overview of OpenMES Framework

OpenMES framework is developed as a platform software supporting construction of MES for discrete production system. It enables construction of a target system with necessary customization and assembly of provided software components based on MES model, which is a template of the target system.

3.1. Framework

OpenMES framework is developed on the basis of a technology called “object-oriented framework” (hereafter, framework). Framework is a mechanism for making and re-using components of software and is “A framework is a set of prefabricated software building blocks that programmers can use, extend, or customize for specific computing solutions.” [Taligent].

Framework defines elements of target software system. Each element is realized and provided as a software component. Construction of highly reliable system in short time and lower costs can be realized by assembling software components on the basis of proper framework.

3.2. Adoption of Java/CORBA

Runtime environment of MES should be covered broad area including controllers of production equipment to host computer running ERP or SCM. Therefore, we adopted CORBA (Common Object Request Broker Architecture) [OMG] as the platform for implementation of OpenMES. CORBA is one of promising platforms for construction of distributed system over multiple computer systems. As for implementation language, we selected Java whose language processor is independent on platforms. Both Java and CORBA are available on many platforms and the combination of these two technologies is considered large advantage as implementation platform of MES assuming distributed environment and multiple platforms.

3.3. Construction of FactoryWeb
In the latest system development, system development based on Web is indispensable for sharing information and collaboration. In this kind of system, software called application server becomes to take principal role in a server. It is a middleware integrating environment including Servlet, JSP (Java Server Pages), EJB (Enterprise Java Beans), CORBA as well as Web server.

Implementation platform of components of OpenMES framework is Java and CORBA and it makes easy to be embedded in an environment of application server. As a result, environment where appropriate person can share information of production site in real time and can collaborate with each other for production optimization can be constructed. We call this kind of environment as FactoryWeb.

### 3.4. SubSystem Model

During the development of “Open MES Application Framework” in MSTC, we analyzed various activities in shop floor, and identified application domains to be included in MES. We divided MES into 11 subsystems. The subsystems are modeled to ensure that they work as independent systems and have minimum overlaps each other. At the same time, cooperation between subsystems is also considered. The subsystem model clarifies relationship between subsystems, and defines information that are exchanged or shared by the subsystems.

This approach reduces the complexity of MES, and makes possible to construct MES step-by-step manner.

### 4. Supported Subsystems

OpenMES framework has employed the following three subsystems from MSTC’s subsystem model.
- Process Management Subsystem
5. Process Management Subsystem

Process Management subsystem is responsible for dispatching process jobs, monitoring progress of manufacturing process in real-time, and tracking the process results. The following functions are provided:
- order management
- job management
- job dispatching
- collection of job result
- collection of work result

5.1. Order Manager Component

Order Manager component is an entity-manager that manages orders. It provides basic methods to manipulate an order object, which include add, remove, modify, and find method.

5.2. Job Manager Component

Job Manager component is an entity manager that manages jobs. It provides basic methods to manipulate a job object, which include add, remove, modify, and find method.

5.3. Dispatcher Component

Dispatcher component is responsible for dispatching a job to a resource, and sequencing jobs dispatched to a resource based on specific sequencing rule, and collecting job result and work result.

An assignment of job to resource is usually determined by the scheduler statically. However, there are cases that the assignment should be change dynamically at shop floor level. The following resource configurations require the dynamic assignment change. OpenMES framework has modeled these configurations and supports dynamic assignment change.

5.4. Production Controller Component

Production Controller component is responsible for operations that involve multiple components in Process Management subsystem. For instance, releasing process jobs to shop floor involves the following steps:
- ask Order Manager to set the released flag to the orders associated with the jobs being released.
- ask Job Manager to record release date/time into the jobs.
- ask Dispatcher to dispatch jobs.

Production Controller is responsible for executing above steps. The above steps should be executed as a single transaction. Production Controller ensures that they are committed / rollbacked in a single transaction.

6. Equipment Management Subsystem

Equipment Management subsystem provides interface functions between equipment controllers that control equipment devices and the MES management layer. It provides following functions:
- manage information on configurations of equipments
• monitor status of equipments
• query Jobs and notify Job Results
• query Work Instructions
• notify Work Results

6.1. Equipment Component / Process Equipment Component

Equipment component and Process Equipment component are connectors that connect an equipment controller to the MES management layer. Equipment component provides an interface between an equipment controller and an equipment monitoring system. Process Equipment component also provides an interface with a process management system on top of the interface provided by Equipment component.

- **Equipment Component**
  Equipment component provides an equipment controller with interface to notify generic status of equipment for monitoring equipments. Equipment component handles following information of equipment’s status.

<table>
<thead>
<tr>
<th>Status</th>
<th>overall status of an equipment: active, down, standby, attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm</td>
<td>alarms which are defined for each equipment, such as parts jam</td>
</tr>
<tr>
<td>Property</td>
<td>properties that indicate additional status of an equipment</td>
</tr>
<tr>
<td>Event</td>
<td>events that occur in an equipment and that should be notified to a monitoring system</td>
</tr>
</tbody>
</table>

Above information that an equipment controller generates is notified to the equipment monitor component when an equipment controller calls a notification API provided by Equipment component.

- **Process Equipment Component**
  Process Equipment component defines APIs to handle Jobs. By calling these APIs, an equipment controller can obtain Jobs dispatched to the equipment, and notify its progress to the MES management layer (i.e. Dispatcher component in Process Management subsystem). Process Equipment acts as just an interface; the equipment controller is responsible for keeping the status of the job being executed.

6.2. Equipment Monitor Component

Equipment Monitor component provides functions to monitor the status of equipments. The followings are major functions Equipment Monitor provides:
- receive notification of equipment’s status and alarms from equipment components.
- store the latest equipment’s status and history of the status into a database
- reply to query requests for equipment’s status from equipment monitoring applications

There are two options for monitoring equipment’s status: the monitor inquires the status to each piece of equipment repeatedly (pull model); each piece of equipment notifies its status to the monitor repeatedly (push model). The OpenMES framework selects the later option. Each piece of equipment sends status information to Equipment Monitor component and monitoring applications get equipment status from the component.

6.3. Equipment Configuration Manager Component

Equipment Configuration Manager component provides functions to manage information of equipment configurations. Equipment components and Equipment Monitor component read the equipment configuration information from Equipment Configuration Manager component for their initialization.

While Equipment Monitor component manages equipment’s attributes that dynamically change at runtime, Equipment Configuration Manager component manages static equipment’s attributes that are defined at build time.
Since the equipment configuration information differs for every system, it is one of customization points. The default equipment configuration information has following properties:

- equipment id
- model and type text string
- location of equipment
- ids of Process Unit contained by the equipment

### 6.4. Connection to the Equipment Controller

How to connect the MES to equipment control system layer is one of important subjects in building MES applications.

In the OpenMES framework, controller devices are connected to the MES by the equipment controller virtually. OpenMES framework does not define the substance of equipment controller, but it only defines abstract connector interface between Equipment component and the equipment controller. Supplier of controller devices or system integrator should implement this connector interface for each controller device.

Each controller device has unique interface. Therefore, equipment controllers are implemented for every controller devices.

Recently, many controller devices have network interfaces that support TCP/IP. In application programming interface, high level interfaces such as OPC (OLE for Process Control) are being provided. System integrator can develop equipment controllers efficiently by using such communication protocols and programming interfaces.

### 7. Schedule Management Subsystem

A scheduler is a tool for making daily schedule based on production order and process resource information. There are many scheduler products. Furthermore, some companies have unique scheduler as part of production planning system. Therefore, the OpenMES framework assumes the use of existing schedulers, and does not define the interface specification so far.

In current implementation of OpenMES based MES systems, the system integrators have to develop the interface module for each scheduler. However, the activity to standardize scheduler interface is making progress. Once such a standard interface becomes widely available and a common interface module is provided as a part of the framework, the system integrator will not need to develop new interface module for each system.

### 8. Evaluation of the Framework from System Integration Experience

#### 8.1. Integration Variety

MES applications we developed using the OpenMES include variety of scale from small-scale validations systems to large-scale mission critical systems. The overview of developed applications is summarized in the following sections.

- Type of Process
  - flexible manufacturing system that consists of NC machines
  - assembly line in which manufacturing robots and workers exist
  - assembly job shop in which workers assemble parts manually

We applied OpenMES framework to job shop type of systems in which work in process are routed freely
from an equipment to other equipment, and also applied it to flow shop type of systems in which work in process are routed through fixed route such as belt conveyor.

Most of client applications use a Web browser as front end GUI. There are two types of client application. One is Java applet that directly communicates with components on the MES server. The other is CGI/Servlet that work on the server and that return results as HTML document. The former is used for applications that need complicated interactive operations, and the latter is used for the display of reports.

- Communication Methods between an equipment controller and an equipment device
  - TCP/IP: An equipment controller communicates with the equipment device that has a network interface by using the protocol defined by the device on TCP/IP.
  - JNI (Java Native Interface): An equipment controller calls APIs in control programming library provided for the equipment device by using JNI.

8.2. Evaluation in Applied Cases

The principal merit of using a framework is considered as high productivity of development by re-using software component. However, it is not easy to measure quantitatively the degree of the improvement of productivity by using the framework because there is no case that the same system is developed by different methods. Unfortunately, we have not measured quantitative data about OpenMES so far also.

On the other hand, firms that take interest in adoption of OpenMES framework are often large enterprises that have multiple factories or production lines. These enterprises have often invested in duplicate or triple development for multiple factories or production lines because of the reasons described in the section “Issues in MES Development.” Therefore, OpenMES appeals to those enterprises because it enables set up or modification of production line in short time and can reduce investment of the development by a deployment of developed MES to other multiple factories and production lines.

Re-using of a framework is not limited only to re-using of the implementation. If anything, knowledge or know-how contained in the design are more important. OpenMES framework itself has been developed with feedback of the information of developed multiple MES in the past and receives many benefit from the knowledge and know-how in those MES. Furthermore, a user can extend it to a framework embedding knowledge and know-how that is specific to the enterprise and can deploy it to systems specific to each factory and production line. Actually, there are customers who are deploying a system to multiple factories with this approach.

9. Concluding Remarks

This paper studied construction of MES from the view point of system development and proposed a development methodology using a software framework for effective development.

Object-oriented technology, which is prerequisite of a framework, is rapidly spreading in software development domain. In accordance with it, the concept of framework will be spreading hereafter.

In the near future, this framework is connected to B2B framework and hence, agile global manufacturing such as sharing production data among sales division through production division and rapid response over a supply chain will be realized.

References


