A WEB-BASED MANUFACTURING EXECUTION SYSTEM FOR DISCRETE MANUFACTURING

Olivér Hornyák[1], ***Károly Nehéz***[2]
PhD[1,2], *Production Information Engineering Research Team (PIERT) of the Hungarian Academy of Sciences*[1]
Department of Information Engineering, University of Miskolc[1,2]

ABSTRACT

The modern manufacturing requires collaboration of production staff and machines in a structured way. The responsibility of each member of the team includes the members primary function, but also for being an active member of the communication network — collecting, filtering, and passing along information. Human oversight and error can cause failure in the system, which is why a lot of scientific effort is focused on capturing information.

This paper overviews the Manufacturing Execution Systems (MES) which plays an important role in managing and controlling discrete production activities on the shopfloor. The major MES functionalities are also discussed.

In the second part of the paper a proposal is presented on the applicable IT solutions to implement a WEB-based Manufacturing Execution Systems. The development framework and environment will be discussed.

KEYWORDS: Manufacturing Execution System, J2EE, JBOSS, XDoclet

INTRODUCTION

The modern manufacturing requires collaboration of production staff and machines in a structured way. The responsibility of each member of the team includes the member's primary function, but also for being an active member of the communication network — collecting, filtering, and passing along information. Human oversight and error can cause failure in the system, which is why a lot of effort is focused on capturing information. Manufacturing Execution Systems have been playing important role on managing real time processes on a manufacturing shopfloor. There is a great scientific effort focusing on the modelling and developing such systems. Some MES definitions of the related literature are as follows:

- MES is more than just manufacturing data collection. It is the core framework of enabling the real time enterprise [1].
- A Manufacturing Execution System is a collection of hardware/software components that enables the management and optimization of production activities from order launch to finished goods. While maintaining current and accurate data an MES guides, initiates responds to and reports on plant

activities as they occur. An MES provides mission-critical information about production activities to decision support processes across the enterprise [2].

- By definition, the functions of an MES range from operation scheduling to product genealogy, to labour and maintenance management, to performance analysis, and to other functions in between. If it happens on the factory floor, MES is supposed to manage it. MES is intended to be much like what Enterprise Resource Planning (ERP) is to the business side of the house. [4].

It is noticeable that MES systems are defined around the functionalities rather than applications. Figure 1. b) shows the major functions of a MES system. The scope of this paper and the related research work is limited to the discrete part manufacturing. The core functions to be detailed and implemented are

- the process management,
- the operations, batches (re)scheduling,
- and the quality management aspects.

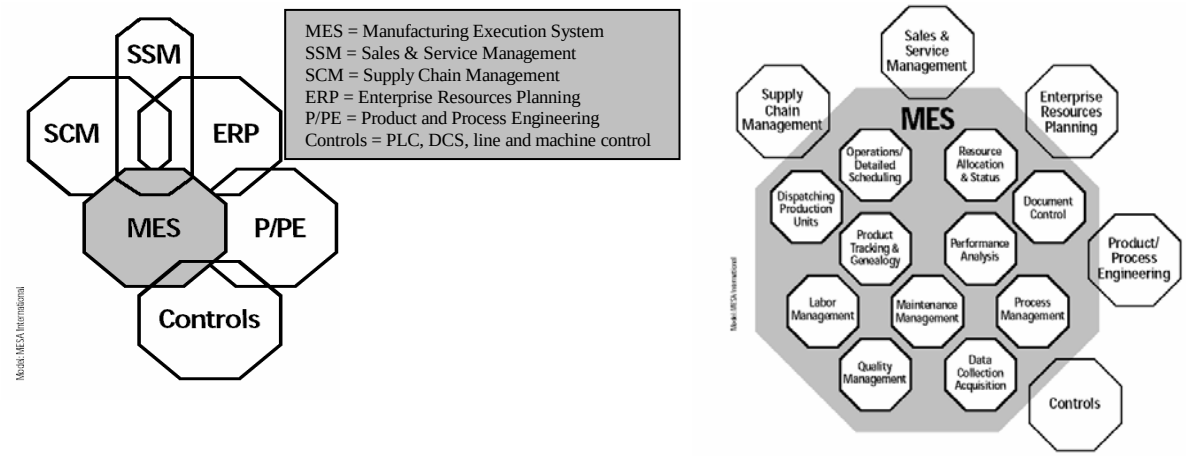

Figure 1
a) MES context mode l and b) MES functional model [3]

DEVELOPING MANUFACTURING EXECUTION SYSTEM AS A DISTRIBUTED, SCALABLE APPLICATION

Applying J2EE technology

J2EE (Java 2 Platform, Enterprise Edition) is a Java platform for multi-tier server-oriented enterprise applications. Sun Microsystems (together with determinant industry partners such as IBM) designed J2EE to simplify application development in a *thin client* environment. A *thin client* is a low-cost, centrally-managed computer devoid of disk drives, and expansion slots and widely used as a synonym for both the NetPC and the *Network Computer* (NC). J2EE model generally includes the following three tiers: client- , middle -, and Enterprise Information System tier (see Figure 2). The *client tier* is a set of applications or browsers on an NC or a desktop computer. The J2EE platform takes place in the

middle tier and consists of a Web server and an *Enterprise JavaBeans* (EJB) server. These servers are also called "containers." The middle tier can be divided into additional sub-tiers. The Enterprise Information System tier has the existing applications, files, and databases.

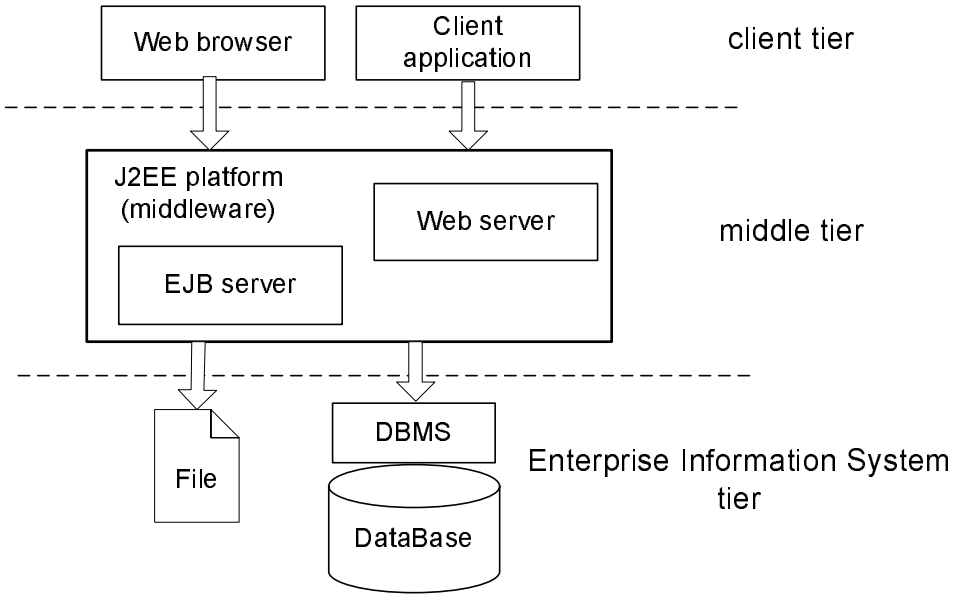

Figure 2: Three-tier software architecture.

An application server (*middleware system*) is an implementation of the J2EE platform specification that resides in the middle-tier of a three-tier architecture. It provides *middleware* services for security and state maintenance, along with data access and persistence. We can declare a *middleware* as software layer between client and server processes. J2EE includes a number of components and services, such as:

- Support for Enterprise JavaBeans. EJB is a server-based technology for the delivery of software components in an enterprise environment. It supports the Extensible Markup Language (XML) and has enhanced deployment and security features.
- The *Java servlet* API (Application Programming Interface) enhances consistency for developers without requiring a Graphical User Interface (GUI).
- *Java Server Pages* (JSP) which is used for dynamic Web-enabled data access and manipulation.

EJB-s plays the most important role in the MES development. J2EE specification defines three different types of enterprise beans: *Session*, *Entity* and *Message-driven* beans.

- *Session bean*: as its name suggests, a session bean is similar to a (not shared) interactive session. It drives user interactions.
- An *Entity bean* represents a persistent business object. In MES systems, examples of these business objects are: *product*, *machine*, *task*, *batch*.

- A *Message-driven bean* is an enterprise bean that allows J2EE applications to process messages asynchronously. It acts as a message listener. The messages may be sent by any J2EE component - a client application, another enterprise bean, or a Web component. This method enables to notify the MES management system of the status about sensors or other low-level devices.

Applying JBoss server as a MES development framework

JBoss application server is a component-based, open source application which fully implements J2EE. JBoss provides *JBossServer*, as the basic EJB container, and Java Management Extension (JMX) infrastructure [6]. It also provides several components (see Figure 3). Therewith, JBoss enables to mix these components through JMX by replacing any component with other JMX compliant implementation for the same APIs.

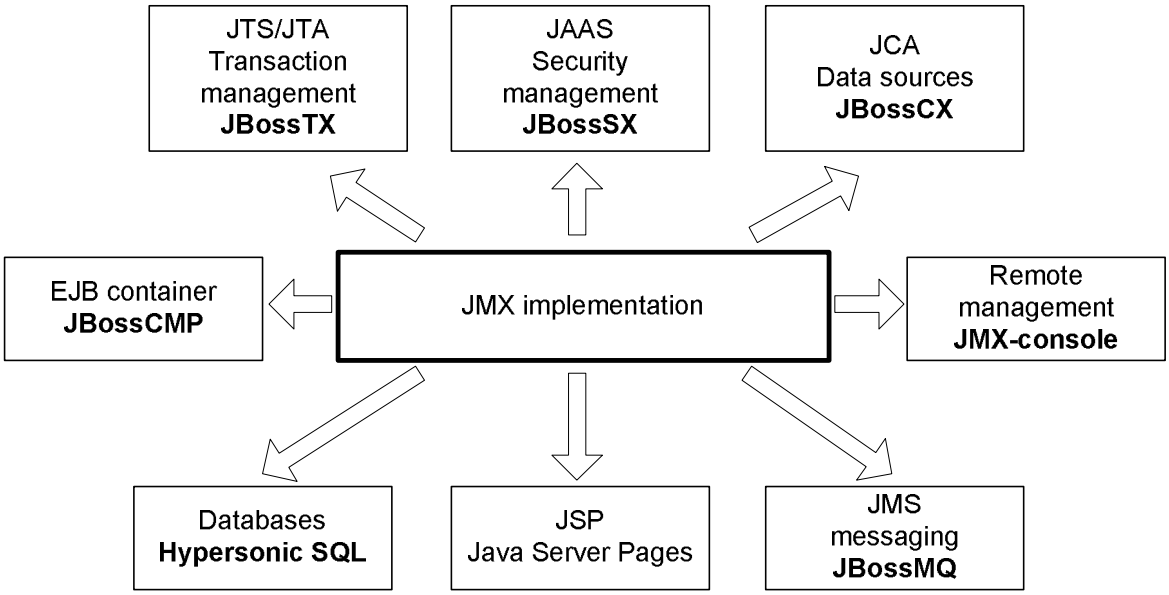

Figure 3: JBoss JMX integration bus and the standard JBoss components.

MES DEVELOPMENT PROCESS

This paper focuses on presenting a modern application development strategy using open source software applications. Basically EJB development process is a complicated and complex subject, application developers have to create many stub classes and XML deployment descriptors.

Nowadays, there are numerous tools which make this process more natural and automatic. One of them is the XDoclet [8] code generation engine which enables Attribute-Oriented Programming for Java. XDoclet provides continuous integration of the deployment meta-data whenever source code changes. Working with only one

file per component gives the developer a better overview of code. Additionally XDoclet provides security-, transaction-, persistency-, database relation management functions. This technology reduces development time developers can concentrate better on business logic.

In our early development stage a comprehensive development strategy have to be defined. The proposed set of tools in J2EE development can be summarized as follows [11]:

- *Analysis and design tool* - A visual modelling environment in which a model can be developed, through various UML diagrams.
- *Development tool* - Also known as an *integrated development environment* (IDE). IDE can speed up development time greatly. E.g. NetBeans [9]
- *Build tools* - A set of utilities to manage your development configuration and enable auto-deployment of your components to the J2EE environment. [10]
- *Source code control tool* - A shared repository for code base in various versions of development.
- *Testing tools* - Utilities to perform various types of testing. [9]
- *Problem tracking tool* - This component integrates with source code control environment to track problems from identification to resolution.

The proposed development strategy is based on the previously mentioned ideal environment:

1. Producing UML static, sequence and activity diagrams,
2. Exporting source files into NetBeans IDE [9] and adding XDoclet tags,
3. Deploying application into JBoss application server with Ant building tool [10].

Based on this strategy, an open source project (OpenMES) [12] was started on Sourceforge software code repository at the end of 2003. Sourceforge is a collaborative development site for open source projects with high-level services.

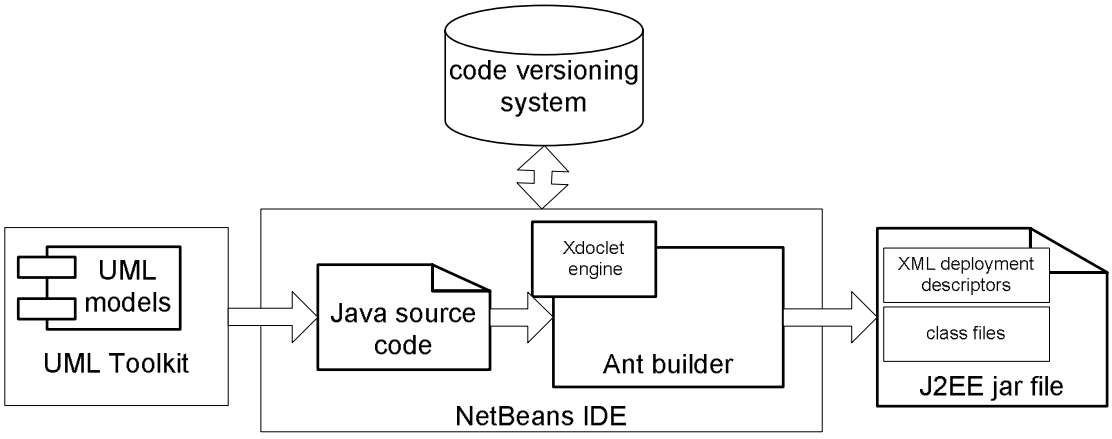

Figure 4: Proposed development process of J2EE based MES application.

CONCLUSION

A new approach to develop a Manufacturing Execution System as a distributed application based upon J2EE and EJB technologies has been presented. Continuous work will focus on creating detiled component specifications and implementing of the proposed software modules. The presented OpenMES project is being developed on Sourceforge.net software repository.

ACKNOWLEDGEMENTS

The research and development summarized in this paper has been carried out by the Production Information Engineering and Research Team (PIERT) established at the Department of Information Engineering and supported by the Hungarian Academy of Sciences. The financial support of the research by the aforementioned source is gratefully acknowledged.

REFERENCES

[1] Johnson, C.: **Don't Overlook Manufacturing Execution in your Product Lifecycle Management Strategy**. www.visiprise.com/pdf/dont_overlook_manufacturing.pdf, 1998.

[2] Barkmeyer, E., Denno, P., Feng, S., Jones, A., Wallace, E.: **NIST Response to MES Request for Information.** NISTIR 6397, National Institute of Standards and Technology, Gaithersburg, MD, 1999.

[3] **MES Explained: A High Level Vision**, MESA International White Paper Number 6, September 1997.

[4] Neil, S.: **MES Meets the Supply Chain**, Managing Automation Magazine, Vol.16, No. 12, pp.18-22, December 2001.

[5] Tóth, T., Erdélyi, F.: **The Role of Optimization and Robustness in Planning and Control of Discrete Manufacturing Processes**. Proceeding of the 2nd World Congress on Intelligent Manufacturing Processes & Systems. June 10-13. pp. 205-210, Budapest, Hungary, Springer, 1997.

[6] **JBoss Administration and Development** – Open Source J2EE based application server. www.jboss.org, 2004.

[7] Roman, E.: **Mastering Enterprise JavaBeans – Java 2 Platform Enterprise Edition**. Wiley Computer Publishing, 2000.

[8] **XDoclet documentation**, xdoclet.sourceforge.net, 2004.

[9] **NetBeans Integrated Development Environment Documentation**, www.netbeans.org, 2004.

[10] **Ant documentation**, ant.apache.org, 2004.

[11] Gabrick, K.A., Weiss, D.B.: **Java 2EE and XML development,** Manning Publications Co. 2003.

[12] http://sourceforge.net/projects/openmes/