Proceedings of the 6th WSEAS International Conference on Robotics, Control and Manufacturing Technology, Hangzhou, China, April 16-18, 2006 (pp81-86)

Realizing Manufacturing Execution Systems with Supervisory Control Features in a C# and .NET Environment

HAMIDULLAH KHAN NIAZI, SUN HOUFANG, GONG LIN, RIAZ AHMAD▲

(Department of Manufacturing and Automation, Beijing Institute of Technology 100081, China)

(▲ Aerospace Manufacturing Department, Beijing University of Aeronautics and Astronautics, 100083 China)

Abstract: The information gap between the enterprise resource planning (ERP) and shop floor has always been a critical problem in a manufacturing organization. Keeping in view this integration problem, a Manufacturing Execution System (MES) complexity has been presented in a MES and ERP framework and then the issue has been tackled using an open .NET framework with Microsoft SQL Server 2000, C# and ASP.NET technologies in a web based application. Using any web browser and intranet/internet, the management of manufacturing control process and its coupling with ERP application could be realized with full support from database for controlling the live and real phenomenon taking place at a various manufacturing plants distributed globally at different geographical locations in distributed manufacturing control (DMC) environment. In the last part, a new supervisory Control paradigm has been proposed with ongoing future work to bring all such tasks to reality. SQL 2000 acting as a relational database plays a critical role between the ERP and shop floor, which keep on updating the management through controlled stored procedures, triggers and alerts. The programming interfacing of C#, ASP.NET and ADO.NET with a blend of delegate and events makes the real time tracking practically happen on the shop floo**r.**

Keywords: Manufacturing Execution System, .NET Framework, Supervisory Control, ERP, ASP.NET

1 Introduction

In the last couple of decades, the manufacturing is getting into a phase which has been called collaborative manufacturing. Due to increased automation process, the real benefits of automation between the top layer and bottom layer still needs to be explored due to information gap between these two layers. Historically, manufacturing planning activities have been covered under various names like Material Requirements Planning (MRP), Manufacturing Resources Planning (MRPII), Enterprise Requirements Planning (ERP), and Manufacturing Control Systems (MCS). However, the MES is more than a planning tool like ERP or MRPII. MES is an on-line extension of the planning system with an emphasis on execution or carrying out the plan. The MES is a manufacturing tool designed and built for manufacturing. Execution means: making products, turning machines on and off, making and measuring parts, moving inventory to and from, workstations, changing order priorities, setting and reading measuring controls, assigning and reassigning personnel and inventory, scheduling and rescheduling equipment [1]. The most important goal of the MES is to provide right information at the right time at a right place. The management needs timely and handy information required for various sections responsible for the management of manufacturing affairs at an enterprise level. Recently, one of the cost reduction strategies for the firms has been to shift their manufacturing facilities to overseas markets to gain a marginal advantage on labor costs [2] which is ultimate goal towards DMC system. MES coupled with the ERP platform can realize DMC and digitized automation of manufacturing management in more practical manners with the help of modern DOT.NET environment. MES incorporates powerful and automatic inherent inbuilt documentation features for product traceability and document management as demanded by the international organization for standardization (ISO). These are valuable features in quality management affairs without involvement of any paper based work. By definition, a MES consists of a set of integrated software and hardware components that provide functions for managing production activities from job order launch to finished products. Using current and accurate data, a

MES initiates, guides, responds to, and reports on production activities as they occur. A MES provides production activity information to other engineering and business activities in the enterprise and its supply chain via bidirectional communication [3]. This paper is organized as follows. The integration complexity of MES and ERP is presented in section 2. Section 3 deals with the implementation of MES in .NET with C# language. Section 4 presents the supervisory control functional architecture in a distributed shop floor control system (SFCS). The conclusion is presented in section 5

2 Integration Complexity of MES and ERP

Due to lack of understanding between ERP and shop floor and also due to pressing user requirements, the need for the development of shop information system has been become more intense. The issue of integration between ERP and MES became more important due to the intense market competition among entrepreneurs and growing demand of customized products from customers. With the advancement in IT and strong desire to fill the gap, effort by different vendors have been intensified. When ERP applications were initially tried in the manufacturing organizations, they either lacked the integration phenomena or were too process centric. It is no doubt that the real benefits could never be reaped unless the involvement of the operational layer is materialized in seamless manners. Some of the real nature issues are as follows: What are the job statuses for the current production order and its work in progress (WIP)? What is the equipment status, its capability regarding order change or it's splitting along with repercussion for overall production plan? What is the effect of scheduling, rescheduling, sequencing on the overall makespan of the order? What are the current performance indicators with respect to product-process matrix? What are the bottlenecks in the production management systems? The integration complexity grows exponentially as the planning horizon gets narrower. Figure 1 shows the complexity relationship with respect to planning period. The highest level of complexity is at the bottom level of manufacturing i.e. at the shop floor level as compared with the overall enterprise planning level. The complexity paradigm over the functional architecture of ERP and MES is presented in figure 2. In this architecture, the MES core as well support functions have also been listed along with main ERP functions. The complexity may have many facets but some of them are as following:

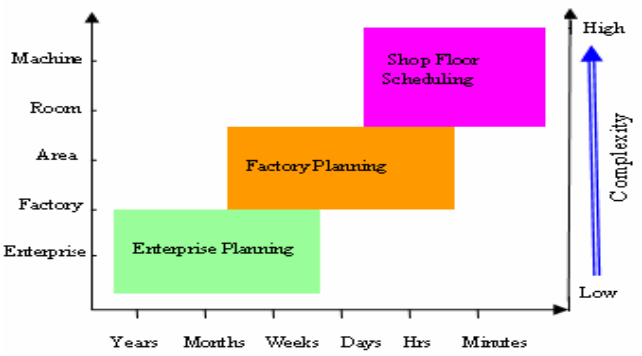

Fig: 1 Planning sequence and their horizons

2.1 Application Diversity

The application diversity varies from the vendor to vendor depending upon the application area. Application's usefulness for one area could not be the same for another user due to configuration complexity inside the design of an application. This is why one successful application fails with adverse results in another application. ERP vendors emphasize on generic application which fail to encompass the usefulness of ERP solution in another one.

2.2 Vendors Diversity

In figure 2, there are about 13 ERP functions where each one may have many facets. Vendors specialized in certain areas produce narrow solution but brand them as a versatile. Each vendor deals the issue with its own approach with its focus on specialized expertise; this gives a rise in great diversity with distracting point solutions.

2.3 Data Diversity

Due to different ERP vendors have been trying to integrate the MES using upward and down approaches at various levels. Various types of data and format during communication have been the main source of diversity between MES and ERP integration. The use of different operating systems (OS) and platforms have further added to the problem for data communication between these two systems

on internet or intranet in the distributed manufacturing environment.

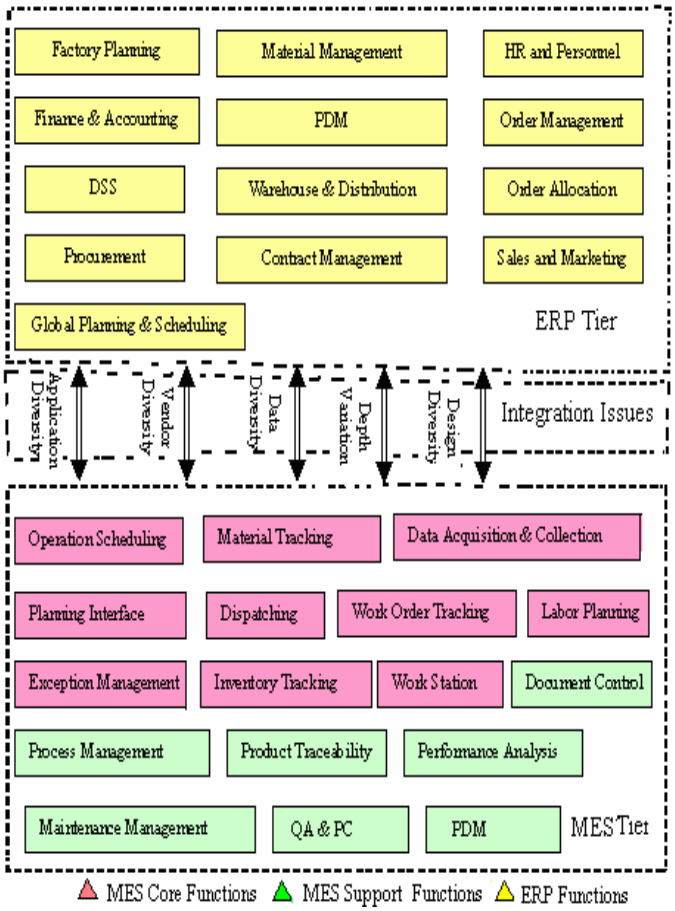

Fig 2: Key Components and Integration Complexity between ERP and MES

2.4 Depth Diversity

This integration diversity is significant where vendors of ERP have tried to expand some ERP functions beyond their domain. In this extended effort, some functions of ERP got overlapped over the MES domain and proved a simple duplication of functions in both layers. Consequently, the integration complexity became more intense instead of transparent and coherent information flow.

2.5 Design Complexities of MES and ERP

The design parameters of ERP mostly depend upon the long planning of corporate business functions. Their interfacing with MES whose function planning time is short and some time real time capturing of data gives rise to complexity in the usage of such information in the ERP domain. The key design parameters of both layer come from the need / requirement analysis in their respective domain. The design complexity is more intense if both systems come from different operating systems along with different working platforms.

The integration of a manufacturing execution system (MES) with related manufacturing applications, such as computer-aided design (CAD), computer-aided process planning (CAPP), computer aided manufacturing (CAM), scheduling, ERP, and equipment control systems, is a relatively new practice. Many Industries currently use a MES for managing factory floor information and activities to increase productivity and improve quality. Examples of factory floor information and activities are resource allocation, dispatching production units, quality management, operation planning, detailed scheduling, labor management, product tracking, and keeping records of product genealogy. Because CAD, CAM, CAPP, ERP, control, and scheduling systems support linked sets of activities and information flow, it is beneficial to industrial users if these manufacturing software systems can exchange data and messages (function calls) with each other in a diverse computing environment. This means that the MES has to be an integrated component of a manufacturing system [4].

3 Implementation of MES System In .NET Environment with C# Language

In this paper, a new approach for implementing MES module using Microsoft integrated development environment in the Visual Studio.Net with C# language. The most valuable facility which the .NET environment provides is that you can select language of your choice out of many choices like Microsoft Visual Basic .NET, Microsoft Visual C++ .NET, Microsoft Visual C#.NET and Microsoft Visual J#.NET. Microsoft doesn't specifically claim that C# and .NET are intended for real-time systems, but many of the platform's general purpose features support such features with the help of ADO.NET and ASP.NET. We have developed a real time application in this new environment. The functional framework of .NET web technology is given in figure 3.

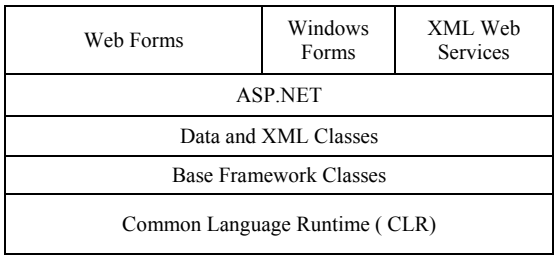

Fig. 3 Functional Framework of Web Technology

We have chosen Microsoft Visual C# as a suitable choice for this MES application. Our choice for .NET platform was due to the fact that it provides the user a free choice of language and also the application can be expanded on the requirement basis. One may select a choice to develop a choice for Windows applications, Web application, Web service, ASP.NET applications and many other ones according to your suitability choice. All of these can be run in Common Language Run (CLR) time environment without much difficulty. In this prototype, any kind of service including web services for the users can be materialized in the emerging standard of Extensible Markup Language (XML). We have developed this application by using ASP.NET Web application with Microsoft SQL server 2000 for data base services. The SQL server acts as an important repository for providing timely support in data manipulation. Other different kind business services can also be linked via internet. The Hyper Text Markup Language (HTML) based services can be materialized for best of your choice just by one click. First of all .NET framework abstracts the hardware peripherals like user terminal, digital cameras, and other visual devices linked with MES system which helps to materialize all business services in the execution layer. The client sends back data typically using Hyper Text Transfer Protocol (HTTP) over an internet or intranet and can receives back the response as standard HTML page. The XML is used for communicating among web services using specially formatted XML, also called Simple Object Access Protocol (SOAP) while HTML communicates the data between the browser of the user/client and web services. The data could also be received in HTML form by the user by filling a web form which is sent back to the web server [5]. The web server after processing user request with the help of web services can send back the results to user via a standard HTML page. These are open standards and Microsoft does not own any of these standards. Our MES prototype model has been built using Microsoft built Internet Information Services (IIS), and http://localhost as our local website using Microsoft Internet Explorer as our web browser for running our application. However, the application can be run using remote website with any computer supporting this application without any running problem. Thus, it is an open choice for a user to run his application on localhost on his computer or any remote website for launching any Windows based application or Web based dynamic Web application in any browser of your choice. Being open system, the computer using any operating system can listen to these SOAP requests on a TCP/IP socket port with read / write capability. On the client or application side, an operating system can read/write to a socket port to issue service requests. The client computer must also be capable of supporting whatever features the user's application desires. If the user application wants a window application, the operating system must provide this functionality or the application developer must give support to implement it on any machine. Actually Microsoft.NET initiative has made this computing world to be platform independent for developing and launching business logic, applications and services in an open manners.

To materialize the real time data capturing and creating useful management alerts in the module, in MES prototype, an extensive use of trigger, delegate/events, and SQL and ADO.NET stored procedures to implement business rules. For example during following situations like completion of a work order or if the bottleneck is created in the system or inventory falls down a certain minimum level or if the number of waiting component to be processed exceed a certain level, then management is to be alerted automatically by system alerts and triggers designed in the software program. The event handler methods or stored procedure handle the business queries with improved performance by accepting the user data values with customized set of transact SQL (T-SQL) commands to handle all user needs. However, to call a certain store procedure saved on the server by any client, first a query should be written only once by the system designer. Thus the correct SQL query gets executed by all clients. And if the business logic changes, one can update once on the server and all clients get informed with automatic change. The stored procedure can be run any number

of times, thus providing an edge over T-SQL statements. An example of a small stored procedure is given in figure 4 for displaying the sales statuses by a required time frame from current database status where data type is 'nvarchar' in the database.

```
An Example of Simple Stored Procedure for sales
   CREATE PROCEDURE SalesByProdTypeAndCateg
@CategName nvarchar(20), @OrdYear nvarchar(10) = '2005'
AS
IF @OrdYear != '2003' AND @OrdYear != '2004' AND @OrdYear != '2005'
BEGIN
SELECT @OrdYear = '2005'
END
SELECT ProductName,
TotPurch=ROUND(SUM(CONVERT(decimal(18, 2), OD.Quantity
 * (1-OD.Discount) * OD.UnitPrice)), 0)
FROM [Order Details] OD, Orders O, Prod P, Categ C
WHERE OD.OrderID = O.OrderID
AND OD.ProductID = P.ProdID
AND P.CategID = C.CategID
AND C.CategName = @CategName
AND SUBSTRING(CONVERT(nvarchar(22), O.OrderDate, 215), 1, 4) = @OrdYear
GROUP BY ProdName
ORDER BY ProdName
GO
```

Fig. 4 A stored procedures for sales by date

Another example of implementing events handler has implemented in figure 5 to authenticate the user login in the database of MES.

```
An Example of Event Handler in C#
private void Button1_Click(object sender,
System.EventArgs e)
  {
      String sr1 = TextBox1.Text;
      String password = "";
      DataSet2 ds2 = new DataSet2();
      DataRow dr = ds2.USER.NewUSERRow();
      Component1 co = new Component1();
      ds2 = co.Funct1(sr1);
      if (ds2.USER.Rows.Count > 0)
    {
      dr = ds2.USER.Rows[0];
      password = dr["pw"].ToString();
      if (password == TextBox2.Text)
      Response.Redirect("WebForm1.aspx");
    }
  }
```

Fig. 5 An event handler for login security

This system is a blend of such kinds of stored procedures, delegates and events handler to realize the real time data analysis and materlise the supervisory control features. We have limited our application to only web application but it can be expanded to Windows application and for other system devices with extensive XML web service standards to convert into professional software. Visual Studio .NET 2003 Professional enables you to rapidly build a broad range of applications for Microsoft Windows, the Web, and mobile devices. It is easier to run application on any web server remotely or localhost at its own machine for testing purpose before fully launching on the internet for public use. The localhost could also provide all kind of services with full data security measures at the disposal of application developer. The real time link between the operator and manufacturing system has been built up by a human machine interface (HMI) where an operator can get the job input and other process information related with the manufacturing activity. Figure 6 gives a snapshot of a simple HMI designed for the shop floor operator. This HMI equipped with other user friendly environment enable the worker to select the input parameters from drop list.

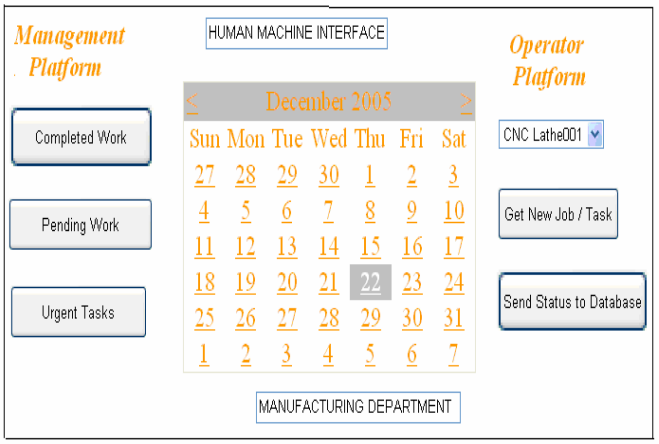

Fig. 6 Human Machine Interface of Manufacturing department

4 Supervisory control realization in MES Solutions

One of the key problem in the present day DMS is that the ERP applications could only be fully realized if proper manufacturing integration is successfully realized using supervisory control and data acquisition (SCADA). Historically, the ERP has been designed primarily for issues which are concerned with planning of production and supply issues at a higher level but it fails to focus the issues directly related with operational level at the bottom line. However, the implementation of ERP in a manufacturing enterprise could reap the full benefits of E-manufacturing if the whole shop floor information system is fully integrated with ERP system. However, this integration could realize with the help of suitable framework and mechanism to handle the numerous other issues of operational

system. If the system fails to integrate the above problem, there would be an information gap between the ERP and shop floor in a manufacturing organization. In this research, a focus has been made on a fully integrated manufacturing system to fulfill the following requirements:

- Interfacing with the ERP system for filtering the required data from ERP system and send it to required place with real time features.
- To perform scheduling of jobs keeping in view the system constraints.
- To display high degree of pro-activeness in handling messages, alerts regarding changes or split in orders or any updates on production order, emergencies, starting / finishing of order completion, splitting events reporting and machine breakdowns. It must handle the two way traffic with ERP in reporting such events.

Following are some supervisory features being materialized in our prototype integrated software.

- WIP tracking upto a machine level.
- Equipment control and generating monitoring reports.
- Product traceability
- Data Capturing
- Line fill bottleneck reports by worker /supervisor/management if any
- Real Time Scheduling and dispatching Handling automated material handling,
- Fault detection and classification
- Handling advanced process control features
- To be fully integrated with warehouse management system with full support for spare parts management.
- To provide a decision support tools to management in production control and in quality management systems with ISO 9000 like features in real time fashion.
- To be able to integrate with engineering analysis systems, and fully E-diagnostics in maintenance systems.

To incorporate statistical process control module with full support from database.

Some of these main automation supervisory features are listed in figure 7.

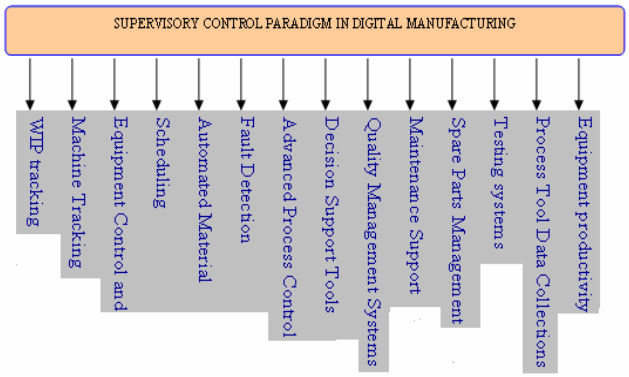

Fig. 7 Automation supervisory control features

5 Conclusion

Due to pressing requirement of in the recent manufacturing control system, MES for real time tracking has become an important issue for manufacturing management. In this paper, the problem of integration between MES and ERP were studied and a .NET based new architecture has been presented with a solution module realized in an open Microsoft.NET framework using C#, ASP.NET and Web based technology with the help of Microsoft SQL Server 2000 database. A new supervisory control framework for detailed management control features has been realized in this module with the use of ASP.NET, SQL stored procedure, events and delegates in C# programming interface. This framework could lead to realize the manufacturing in a complete digitalized environment.

References

[1] Michael Mcclellan, Introduction to Manufacturing Execution Systems, MES Conference & Exposition June 12-14, 2000 Phoenix, Arizona.

[2] A.W.L. Yao, "Design and implementation of Web-based diagnosis and management system for an FMS", Int J Adv Manuf Technology (2005) 26: 1379–1387.

[3] MESA International, "MES Explained: A High Level Vision," White Paper 6 (Pittsburgh: Manufacturing Execution Systems Assoc., 1997.

[4] Feng. S., "Manufacturing Planning and Execution Objects Foundation Interfaces," Journal of Manufacturing Systems, Vol. 19, No.1, 2000, pp. 1-17.

[5] Jeffrey Richter. " Applied Microsoft .NET Framework Programming", Microsoft press, Redmond Washington 98052-6399, 2002.