Proceedings of IMECE
2007 ASME International Mechanical Engineering Congress and Exposition
Seattle, Washington, USA, November 11–15, 2007

IMECE2007-43009

CASE STUDY IN THE CHALLENGES OF INTEGRATING CNC PRODUCTION AND ENTERPRISE SYSTEMS

S. Venkatesh
Boeing Company
Seattle, WA

B. Sides
Okuma America
Charlotte, NC

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

ABSTRACT

Integration of factory floor Computer Numerical Control (CNC) information into Enterprise Resource Planning (ERP) subsystems has been difficult, if not impossible, as traditionally, factory floor machines have been "islands of automation." Boeing/NIST/Okuma jointly collaborated on a pilot project for using a CNC open architecture controller to collect real-time Boeing-specific part accounting data during the production of Boeing 737 Leading Edge (LE) Panels. The goal was to develop a practical and standardized approach in which to capture the real-time part data and then provide this information to an ERP subsystem. This paper presents the results from our Boeing/Okuma/NIST pilot project that evaluated OLE (Object Linking and Embedding) for Process Control (OPC) as an integration strategy for the LE Production Line part accounting at Boeing. Using OPC, automatic logging of the relevant part production statistics was done for each production line, which in turn was used to more accurately determine the total cost of making each LE production line.

Keywords

open-architecture, Computer Numerical Control (CNC), control, standard, manufacturing, Enterprise Resource Planning (ERP), OLE for Process Control (OPC)

1 INTRODUCTION

The manufacturing mantra "Design Anywhere, Build Anywhere, Support Anywhere" is predicated upon worldwide connectivity across all facets of design, manufacturing, distribution, and maintenance. To achieve this, information must flow seamlessly through the enterprise and requires extensive integration of the manufacturing elements. The increasing pressures on manufacturers to improve time to market and integrate the shop floor directly into enterprise business systems places a premium on better techniques to design, integrate, test, evaluate, and maintain control systems.

There have been many efforts to improve factory floor integration. An early attempt at control system integration was the MAP (Manufacturing Automation Protocol) standard, which is a communication standard for intelligent factory floors devices [1]. Because MAP is an all-encompassing standard, with requirements for physical/electrical interfaces, network protocols, data format and syntax, it could be prohibitive to implement and suffered from lack of support from vendors.

Another legacy factory floor integration standard is the Manufacturing Message Specification (MMS), which is a messaging system for exchanging data between control applications and networked devices [2, 3]. MMS is based on a client/server model and employs the concept of a Virtual Manufacturing Device (VMD) as its basis. A VMD is an

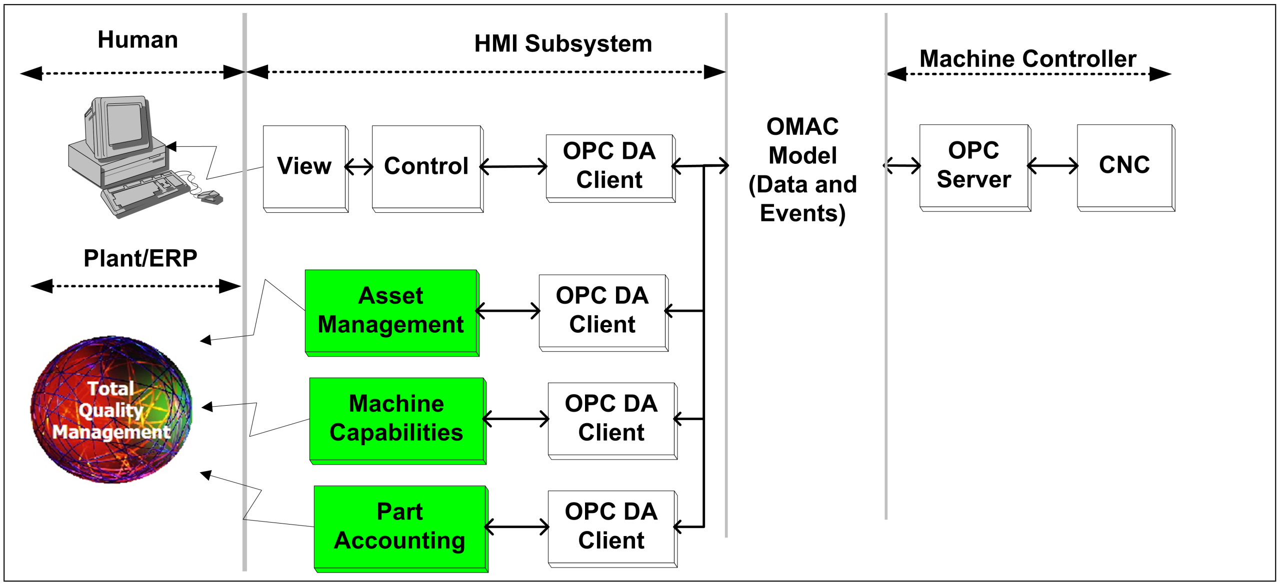

Figure 1. OMAC HMI Integration Architecture

object-oriented model of the externally visible behavior of a factory floor entity. A major drawback of MMS is the need for control vendors to provide presentation, session and transport communication functionality as well as the complementary message en/decoding.

Today, OLE for Process Control (OPC) has emerged as a leading worldwide specification in enabling connectivity and interoperability of factory floor equipment. OLE for Process Control (OPC) is an integration technology developed by the OPC Foundation that defines a standard interface to process control devices [4]. Based on commercial-off-the-shelf technology, OPC is a series of lightweight integration standards defined as interface specifications to support connectivity in industrial automation and the enterprise systems.

OPC promotes interoperability both horizontally and vertically in the enterprise so that it can cut integration costs, speed deployment and promote increased operating efficiency. OPC handles integration by creating a "software bus" so that applications need only know the data required from OPC data sources, not how to get it. This has benefits for both the application developers and the control vendors. On the application side, control device are easier to use since they provide a consistent interface so that applications are smaller and simpler to develop and maintain. On the control vendor side, device drivers are only required to provide data in a single format, according to the OPC specifications.

This paper will look at results of a Boeing 737 Leading Edge (LE) production pilot project to use OPC for CNC to enterprise connectivity. First, we will review issues related to the connectivity of the ERP and the factory floor. Following will be an examination of OPC and its role in the integration of real-time CNC part accounting data. The next section will discuss LE production line and will give a review of the system components used within in pilot production project. We will conclude with a discussion on the challenges encountered on the way to a integrating part accounting for Leading Edge production and our plans for the future.

2 ENTERPRISE CONNECTIVITY

Enterprise Resource Planning (ERP) is the broad term for the set of activities that help a manufacturer, including product planning, parts purchasing, maintaining inventories, interacting with suppliers, providing customer service, and tracking orders. ERP systems for managing the product life cycle have components for integrating factory floor information: parts produced, cycle times, machine and machinists performance. This factory floor ERP information can then be used to determine actual costs on a job-to-job basis, useful in bidding, accounting, and equipment utilization.

Integration of the factory floor consists of connecting the Enterprise to distributed intelligent devices operating concurrently and interacting in real time. The vision is to make real-time cost information available on-demand to an ERP and then accessible through Web portals. But the factory floor is inherently complex, involving machining, storage, and transport of material, tool and part program management, and recovery from faults and deadlocks. Further, the integration of factory floor information into ERP subsystems has been difficult, if not impossible, as traditionally, CNCs have been "islands of automation."

To overcome this isolation barrier, computer numerical controllers (CNC) need to provide open-architecture capabilities to allow access to machining information. Fortunately, the CNC marketplace has changed radically in the last decade from closed proprietary solutions to open products utilizing general-purpose off-the-shelf PC hardware and software technology wherever possible. CNCs based on open, desktop technology for the shop floor have been dubbed "open-architecture controllers". An open-architecture controller is one whose specifications are public by either officially approved standards or as privately designed architectures whose specifications are made public by the developers [5]. With an open-architecture, factory automation no longer simply controls machines; it now provides access to real-time data and information that can be used to optimize manufacturing processes.

The authors are members of the Open Modular Architecture Controllers (OMAC) Human Machine Interface (HMI) User Group, which is an industry working group under the auspices of the OMAC Users Group [6]. OMAC is an affiliate organization of the Instrumentation, Systems and Automation Society (ISA) working to derive common solutions for technical and non-technical issues in the development, implementation and commercialization of open, modular architecture control. A long standing objective of the OMAC HMI Working Group has been to define a series of CNC data specifications.

Figure 2 shows the scope of the OMAC HMI effort. The relationship of the HMI subsystems is best viewed as a generalization of the traditional Model-View-Controller (MVC) architecture, a well-known object-oriented design pattern for Graphical User Interfaces (GUI) [7]. The primary emphasis of the OMAC HMI work has been to define a MVC Model data model to allow the exchange of data and events between the HMI subsystem and the machine controller. Recently, the OMAC HMI workgroup realized that the HMI subsystem could also serve as a gateway to the Plant/Enterprise and broadened the scope accordingly.

A new objective of the OMAC HMI Working Group has been to promote best practices to exploit open-architecture through OPC data mining. The OMAC HMI group is endorsing the use of OPC as a CNC "best practices" integration standard. Although OPC is the largest integration standard used within the process control industry, acceptance has been slow within the CNC discrete parts industry. In order to increase the adoption of OPC within the CNC industry, we consider part accounting to be an important data mining application with impressive benefits. We define part accounting as ERP software that accumulates CNC process knowledge for calculating the actual machining cost of a part for bidding, determining profits, and other accounting functions. Part accounting should be considered an ERP function, even if it is run solely on the CNC, as it applies business analytics to enterprise operations.

3 OPC TECHNOLOGY

OPC leverages the Microsoft Component Object Model (COM) [8] to specify OPC COM objects and their interfaces. The control device object and interface are called OPC Servers. Applications, called OPC Clients, can connect to OPC Servers provided by one or more vendors. Before OPC, data access applications were required to develop completely different integration software for each control device. With the OPC standard, only one driver is needed to access data from any OPC-compliant process control device.

The OPC Foundation has defined several OPC specifications, including Data Access (DA), Event and Alarm Management, and Historical Data Access. The Data Access specification provides a standard mechanism for communicating to data sources on a factory floor. The Event and Alarm Management specification defines a means for transmitting alarm and event information between OPC servers and clients. The Historical Data Access specification allows OPC clients to access historical archives to retrieve and store data in a uniform manner.

Figure 2 shows the functionality of the OPC DA specification within the OMAC HMI System Architecture, which provides a standard model in which to exchange data between the real-time CNC and the non-real-time HMI

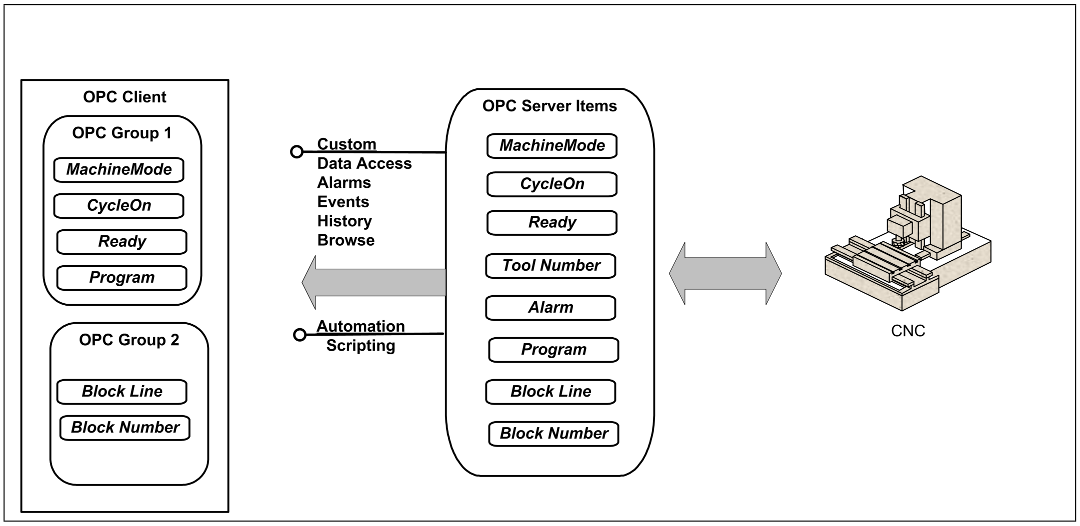

Figure 2. Sample OPC System Architecture

subsystem. The OPC specification describes a client/server object model to allow communication between client applications (OPC DA Clients) and control device servers (CNC OPC Server). All OPC client applications access data from any OPC control device in the same way.

The primary OPC specification is Data Access, which includes the following concepts:

OPC Server is a COM object to which the OPC client first connects. The OPC Server handles connectivity to automation hardware. Its responsibility is to manage OPC groups, translate errors, provide server status, and allow browsing of OPC items.

OPC Group is a COM object for logically organizing data items. OPC clients can pick and choose among the known OPC items on the OPC server in order to create groups. OPC Groups are managed by the OPC client who can activate or deactivate the group, change the group name or update rate among and subscribe for data change event notification. Reading and writing of OPC data is done through the OPC Group.

OPC Item is a single tag (or automation device data point) managed by the OPC server. OPC does not define any application item tag names, e.g., AxisLocation. Instead, OPC clients rely on the vendor to allow clients to browse for OPC data items, or provide a list of OPC data items.

Figure 2 shows OPC data management for a simple CNC OPC Server example. The OPC Client creates 2 OPC Groups, Group 1 containing the `MachineMode`, `CycleOn`, Ready and Program OPC Items, while Group 2 contains the `BlockLine` and `BlockNumber` OPC Items. Group 1 and 2 could run at different update rates if the data timeliness is an issue. To improve performance, if the CNC is in Manual mode, the Group 2 items `BlockLine` and `BlockNumber` could be deactivated. The COM "Custom Interface" uses early vtable binding to the interface methods, so is faster and used by C++ clients. Visual Basic or other scripting languages use the "Automation" interface, which does late binding and method lookup through a type library.

OPC offers additional technologies in which OPC clients and servers can communicate. In order to broaden the appeal of OPC, next generation OPC specifications are being based on Web Services specifications to ensure interoperability with non-Microsoft systems. However, we concentrated on the OPC specification based on Microsoft COM, because these newer OPC services will continue to support the existing technology.

4 CASE STUDY

A compelling use for OPC is for collecting real-time cost accounting data during manufacturing, which led a joint project between OMAC HMI members Boeing, Okuma and NIST. We wanted to evaluate the integration of CNC to ERP and determine if part accounting can be done with minimal integration effort using OPC/OMAC

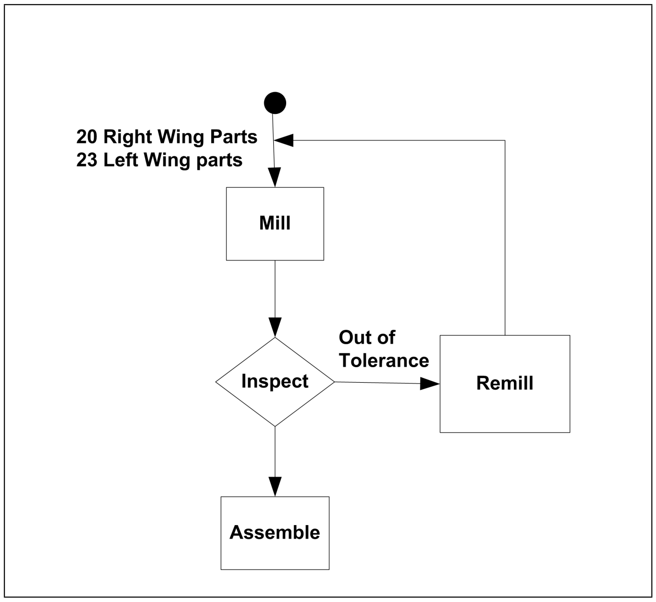

Figure 3. 737 Leading Edge Process

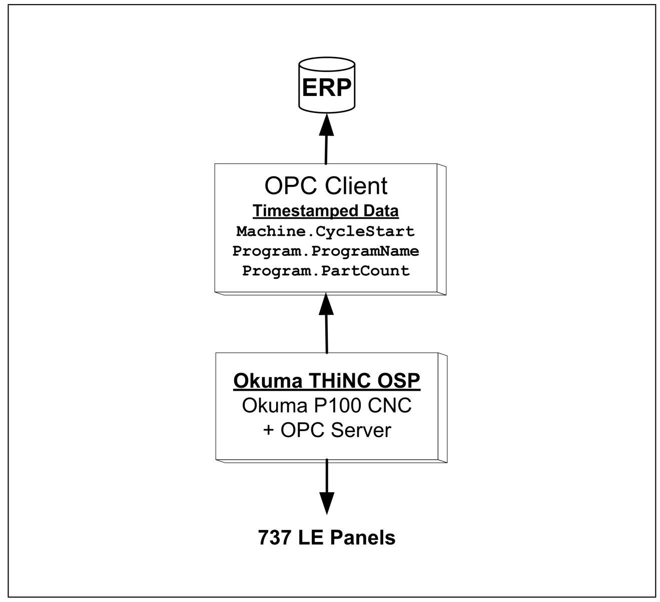

Figure 4. System Architecture

technologies. The project goal was to collect cycle times, setup and job times, part quantities and other vital information on machine and job performance to provide real-time part cost accounting to an ERP accounting subsystem.

This joint Boeing/NIST/Okuma work looked at integrating the production of Boeing 737 Leading Edge (LE) Panels with the enterprise to provide real-time cost data. Figure 3 shows the LE process within a production line, over 40 Leading Edge panels per plane are machined and then joined together in making the left and right aircraft wings. The Leading Edge panels are milled, inspected and assembled on the wing, but often after inspection out-of-tolerance panels are scrapped and new panels are remilled. Panels that need to be scrapped and then redone within a line add to the cost of the plane and are difficult to track. The real-time determination of scrap per production in a seamless integration model was our goal. Given real-time data, the ERP systems would then be able to establish cost of making LE panels more precisely.

Currently at Boeing, data is collected during the execution of assembly and installation jobs on the factory floor using machine readable bar codes to label various entities on the shop floor and to input data from the shop floor. At various stages within the manufacturing process, data from the device level is forwarded to the ERP System via a set of manual Web transactions. Browser-based data collection using HTML forms for data input is a common user interface configuration. The process is highly manual with large latencies between the ERP product costs view and the actual product costs view.

Figure 4 shows the pilot project System Architecture. The LE panels' production took place on an Okuma THiNC OSP-P100 CNC, an open-architecture controller. Okuma provided an OPC server that allows the collection of machine event data in a tagged format. Since we only had two months in order to be ready for production, we limited the scope of the data requirements. We had hoped to connect directly to the ERP but given the pilot time frame, this was not feasible. Instead, we conducted a series of tests with local and remote OPC clients that would eventually lead to data collection by the upstream ERP systems.

To gather the part accounting data, we developed an OPC client application that automatically logged the relevant part production statistics. The part accounting requirements consisted of part name, cycle time, setup time for each Leading Edge panel line. This led to the following OPC data item collection. First, to determine when a new part is being milled, the OPC Item `Program.ProgramName` was collected. Next would access the Okuma Run/Not Run variable to determine if actual machining was taking place, represented by the OPC Item `Machine.CycleStart`. Next we needed to deter-

Line Number	Start Time (Y)	End Time (Z)	Setup Time (min)	Machining Time(min)	Install Time	Total Time (min)	Part Count
1857-L01.MIN	Tue Aug 16 15:26:55 2005	Tue Aug 16 15:27:51 2005	0.58	0.35	3	3.93	1
1857-L02.MIN	Tue Aug 16 15:27:51 2005	Tue Aug 16 15:28:27 2005	0.24	0.35	3	3.59	1
1857-L03.MIN	Tue Aug 16 15:28:27 2005	Tue Aug 16 15:44:48 2005	16.01	0.35	3	19.36	1
1857-L04.MIN	Tue Aug 16 15:44:48 2005	Tue Aug 16 15:45:26 2005	0.28	0.35	3	3.63	1
1857-L05.MIN	Tue Aug 16 15:45:26 2005	Tue Aug 16 15:45:59 2005	0.19	0.35	3	3.54	1
1857-L06.MIN	Tue Aug 16 15:45:59 2005	Tue Aug 16 15:46:30 2005	0.17	0.35	3	3.52	1
1857-L07.MIN	Tue Aug 16 15:46:30 2005	Tue Aug 16 15:47:01 2005	0.16	0.35	3	3.51	1
1857-L08.MIN	Tue Aug 16 15:47:01 2005	Tue Aug 16 15:47:32 2005	0.17	0.35	3	3.52	1
1857-L09.MIN	Tue Aug 16 15:47:32 2005	Tue Aug 16 15:48:00 2005	0.12	0.35	3	3.47	1
1857-L10.MIN	Tue Aug 16 15:48:00 2005	Tue Aug 16 15:48:34 2005	0.21	0.35	3	3.56	1
1857-R17.MIN	Tue Aug 16 16:05:43 2005	Tue Aug 16 16:06:23 2005	0.3	0.37	3	3.67	1
1857-R18.MIN	Tue Aug 16 16:06:23 2005	Tue Aug 16 16:07:02 2005	0.28	0.37	3	3.65	1
1857-R19.MIN	Tue Aug 16 16:07:02 2005	Tue Aug 16 16:07:37 2005	0.24	0.35	3	3.59	1
1857-R20.MIN	Tue Aug 16 16:07:37 2005	Tue Aug 16 16:08:07 2005	0.14	0.35	3	3.49	1
1857-L09.MIN	Tue Aug 16 16:08:07 2005	Tue Aug 16 16:08:41 2005	0.23	0.35	3	3.58	1
1857-R14.MIN	Wed Aug 17 14:50:13 2005	Wed Aug 17 14:50:33 2005	0.33	0	3	3.33	1

Figure 5. Part Accounting Data Collection

mine how long it took to mill the entire part including setup and machining. We determined this by monitoring when a new part program was loaded and when it was unloaded. In addition, by using the OPC data cycle start on/off as indication of milling/not milling, we were able to determine when we were in setup and when we were milling. Finally, to determine how many pieces have been scrapped, we counted the number of repeated parts being machined within a single production line. For example, within production line 1857, if two L01.MIN parts were machined, we knew that one part had been scrapped.

Figure 5 displays an Excel spreadsheet showing the results from the CNC data collection omitting actual cost data. The following data was calculated either directly or indirectly from the data collected, or was given as fixed cost: **(1) Production Line Number, (2) Start Time, (3) End Time, (4) Setup Time (min), (5) Machining Time(min), (6) Install Time, (7) Total Time (min), (8) Part Count, (9) Burn Rate, (10) Shop Cost, (11) Panel Cost, (12) Total Cost**. The scrapped parts were highlighted in yellow to quickly identify the waste per production line.

5 DISCUSSION

In the end we successfully produced real-time part accounting data rather quickly, but along the way numerous data collection and integration challenges were encountered, which will be discussed.

The first implementation concern dealt with keeping track of finished parts, that is, when had the CNC completed a part program. To determine finished parts, we had initially hoped to poll the OPC Item current block to see if it contained the RS274 end of program codes `M02` or `M30`. If we found an end of program code, the part had completed. This approach was flawed as there is a racing issue to detect the end-of-program codes and a program rewind so the end-of-program codes were often missed. Instead, an OPC `PartCount` item was added by Okuma, which was not originally part of the Okuma OPC server.

Given the racing issues with polling, we relied upon the OPC asynchronous data change notification exclusively. However, if using asynchronous notification, OPC clients must be aware of networking issues. A networking problem can arise when the remote OPC Server attempts to connect to a callback interface on the client's PC. Most default client-side authentication security settings are intended to protect the client from perceived malicious attacks. OPC users must make sure proper security authentication is granted so that OPC clients and servers trust each other's identities.

We also encountered networking problems when doing actual connectivity of a remote OPC client and the Okuma OPC Server on a shop floor at Boeing. New more restrictive security measures made the ease of PC network connectivity more difficult, thus, we had problems establishing a connection to a remote OPC server. For our tests we simplified matters by, 1) making sure the remote user had an account and password on the Okuma and 2) insuring that the user and password were identical on remote/Okuma PCs. In the end because of a lack of time and complex security issues related to networking, the OPC client ran on the same PC platform as the Okuma OPC server.

Some care must be taken when manipulating OPC Item data, which is represented as a Microsoft "variant" - a universal data type that can represent double, integer, string, arrays, etc. The variant offers great flexibility, but without care can cause subtle errors. One variant issue was reading an alarm, with `EMPTY` versus `0` variant value both meaning no alarm. Another variant issue we encountered was not synchronizing data types during comparisons, so that an equality comparison of string "1" and an integer 1 was incorrectly determined.

Of note, the abundance of free OPC utilities and affordable commercial OPC toolkits certainly made the testing and debugging of the OPC client and server software easier and less troublesome.

6 SUMMARY

The focus of this paper has been on assessing the ability to integrate ERP subsystem and an open-architecture CNC using OPC. OPC has emerged as the leading worldwide standard in enabling manufacturing connectivity. Because OPC is based on commercial-off-the-shelf technology, it provides a cost-effective as well as widely-supported integration technology.

We presented results from our Boeing/Okuma/NIST pilot project that tested OPC integration to do part accounting for the Leading Edge Production line at Boeing. We have evaluated that the benefits of OPC integration, and found them compelling enough to expect that CNC integration into an ERP is possible with a reasonable amount of effort.

For the future, the OMAC HMI working group will seek to tighten the integration of ERP and CNC.

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 Boeing Aerospace, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

[1] Valenzano, C. D., and Ciminiera, L., 1992. *MAP and TOP Communications: Standards and Applications*. Addison Wesley Publishers, New York.

[2] International Organization for Standardization. *ISO/IEC 9506-1, Industrial Automation Systems - Manufacturing Message Specification - Part 1: Service Definition*.

[3] International Organization for Standardization. *ISO/IEC 9506-1, Industrial Automation Systems Manufacturing Message Specification - Part 2: Protocol Specification*.

[4] OPC Foundation. http://www.opcfoundation.org.

[5] Proctor, F., and Albus, J., 1997. "Open Architecture Controllers". IEEE Spectrum, **34** (6) June , pp. 60–64.

[6] OMAC Users Group. http://www.omac.org.

[7] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley Publishers, Reading, MA.

[8] Microsoft Corporation. *COM Specification*. http://www.microsoft.com/com.