Enterprise-wide Data Gathering and Reporting System

Michael Amos, Dr. Bruce Segee

University of Maine Department of Electrical and Computer Engineering
Instrumentation Research Laboratory

Abstract

Modern manufacturing methods require that products be assembled in the most cost efficient method available. In many cases, this includes separating the manufacturing process across several locations. Even though manufacturing takes place in several locations connections, it is necessary to provide a method to track work in process (WIP) and provide these reports to appropriate divisions of management without indicating the geographic diversity. The solution to this problem has three main parts. First, the data gathering equipment must be scalable, so as to allow virtually unlimited expansion without expensive upgrades. This is accomplished through the use of a network of barcode scanners and decoders. This barcode network communicates with a local PC that contains a database that holds new information. Unlimited scalability is achieved since each network only interfaces with one PC. Secondly, main data storage is contained on main database server. Each of the PCs running the barcode network connects periodically and adds new information from the local database to the main database. If the network is inoperable at any point, the information is not lost, since it is held in the local database. We have developed a system that provides an interactive World Wide Web (WWW) front end to the main database storage and provides management with the appropriate work tracking tools. The main database storage is refreshed at regular intervals to reflect the contents of each local database. The local database contains information from a small number of barcode decoders running on a dedicated network. Utilizing this system, it is possible to track product in real time at various locations around the world.

I. Introduction

Problem Description

A local company with several manufacturing locations separated by many miles needs a way to track work in process (WIP). The current system in use involves a sheet of peel-off labels that follows the product through the manufacturing process. As the employees complete each operation, it their responsibility to remove the label from the sheet and place it on their timecard for that week. This timecard is then read by an automatic reader to tally each employee's timecard and then input that data into the payroll system. There are several problems that arise from this method.

Since the automatic reader is particular about the placement of the labels for each operation, many timecards need to be entered by hand into the payroll system. The number of errors is

large enough to warrant two full time employees whose only responsibility is entering timecards into the system. Obviously, this is not optimal and those two employees could be utilized better in other areas of production.

Because the employees are aware of the problems with the automatic timecard reader, they take extra time to try and place the labels in a manner that will cause fewer errors. This is usually futile and causes little improvement in the number of errors. Because of the extra time taken for placement, production is slowed, causing lost revenue for the company, and the employee.

Since the employees hold their timecards until the end of the week, there is no real-time product tracking available. Until the timecards are entered into the payroll system, there is no way (short of actually going to the location and trying to find the product in question) to find out what stage of production that product is in. This, of course, causes problems for management, since it is difficult to see the performance of each employee and find where WIP is located.

The Solution

The solution being implemented replaces the peel and stick labels for each operation with a unique barcode on a printed work sheet. Each barcode represents a particular operation on a particular lot of product. These barcodes are read with barcode scanners attached to a stand alone barcode network. Traffic on the barcode network is coordinated by a network controller that is polled by custom software running on a PC on the production floor. When the software polls the network controller, it takes the data stored in the network controller and places it in a database contained locally. A separate piece of custom software takes each entry in the local database and copies it to a remote database server stored in a secure location. The database server is queried via a web server that serves a custom web application, allowing various users with different access levels to appropriately interact with the database server.

This solution will provide a return on investment, since the system eliminates the need for two full-time employees manually entering timecards into the payroll system.

Concerns

Since this system is running critical processes, such as payroll tracking, several concerns were expressed by the company. The implemented system must possess all of the following characteristics:

Security – since payroll and other critical operations were being tracked, it is necessary to have a secure system that is difficult, if not impossible to break into by unauthorized individuals.

Fail-safe protection – it is important to allow production to continue if problems occur, such as network failures and computer crashes.

Reliability – again, it is important to produce a system that works well under all possible circumstances without requiring extensive maintenance.

Speed – since one of the goals is to eliminate the non-productive/non-paid time that each employee must take placing labels on timecards, the new system must not take longer than the original label method. The current payroll system takes significant time to run reports and that must also be improved in the new system.

Scalability – this company is capable of expanding to other areas of the country and world with this new system. It is important that this system is able to grow to any size.

II. Implementation

This section focuses on the concerns in the previous sections and how they are being addressed.

Barcode Network

The barcode network consists of several off-the-shelf components from Welch-Allyn [Welch-Allyn, 1999]. These components are connected into a hierarchical system consisting of handheld barcode scanners, barcode decoders, and network controllers (see Figure 1). The network controller controls up to two networks of 31 barcode decoders. Each of the 31 barcode decoders can control two handheld barcode scanners. This means that up to 124 handheld scanners can be controlled by one network controller.

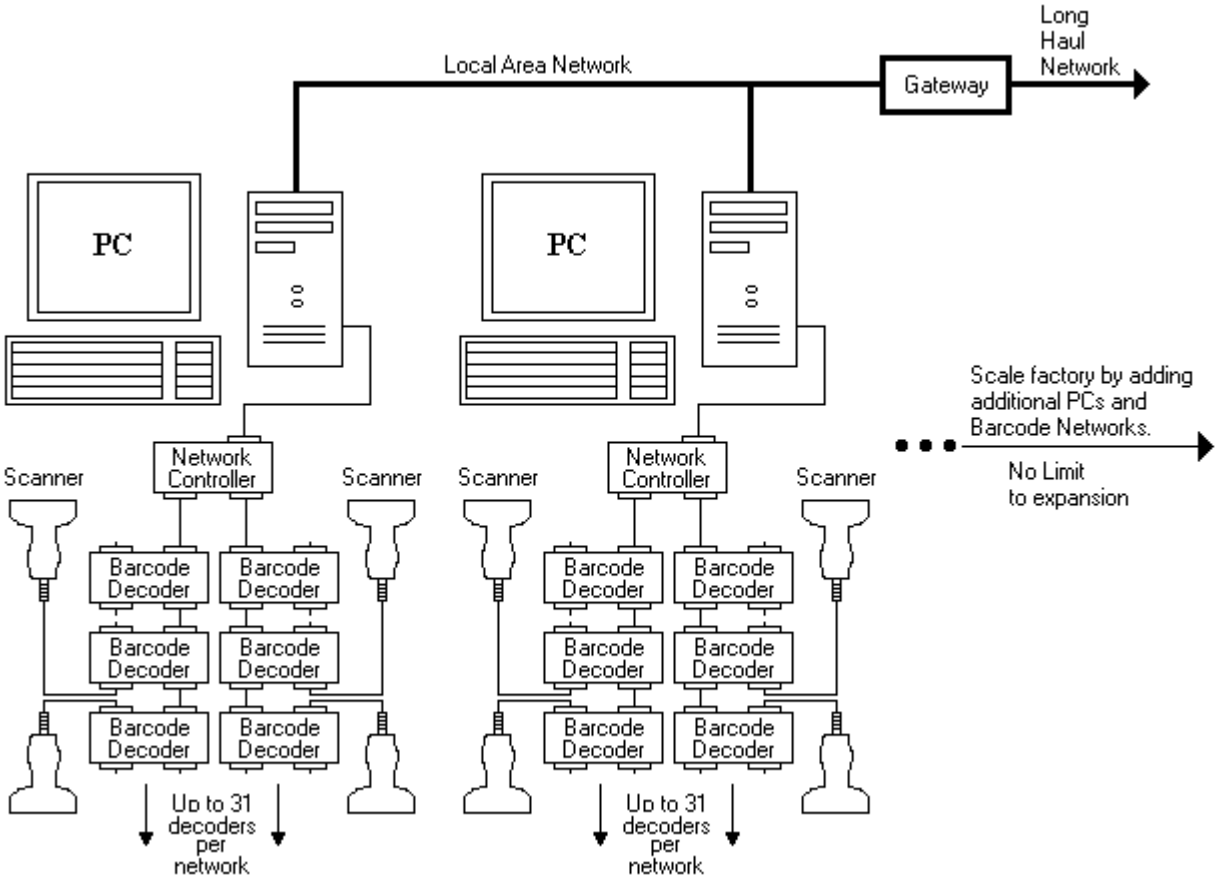

Figure 1. Barcode Network Structure

As a barcode is scanned by the handheld scanners, the data is transmitted from the scanner to the decoder. The barcode decoder interprets the barcode signal and stores it in local memory. The data remains in the decoder module until the network controller polls that decoder. When the

network controller polls the decoder, the data that is held in the local memory is transmitted via RS-485 to the network controller module, where it is held in local memory in the network controller. The data is held in the network controller until the controller is polled by the custom PC software and moved to another location.

In the event of a power failure after data has been stored in memory in either the barcode decoder or the network controller, but before being moved to a more secure location, a battery backup system will keep the memory intact. This is important for reliability and for fail-safe protection, both important topics to the client company.

Custom PC Software

The custom software developed for this project consists of two separate applications that simultaneously run on the production PC and control the gathering and relocation of the barcode scans accumulated by the barcode network. The first application, DataGet, controls the data gathering.

DataGet

After the barcode network controller collects the barcode scans accumulated by each barcode decoder, it is necessary for the PC to collect that data. DataGet performs this operation.

DataGet is a custom piece of software developed by the Instrumentation Research Laboratory at the University of Maine department of Electrical and Computer Engineering. It was developed using Microsoft ™ Visual Basic 6.0, since Visual Basic promotes rapid application development with features that simplify database accessibility [Harris, 1999].

The primary function of DataGet is to periodically poll each barcode network controller attached to that PC and retrieve several pieces of information. First, DataGet collects the barcode scans from each of the network controllers. This is effectively the data that was previously contained in the peel-and-stick labels. One big difference between the old and new systems can be realized here. Management previously needed to have to wait days or weeks before knowing that product had been moved through the production line. Now, there is only a small amount of system time (approximately 30 seconds) before each operation is entered into the database.

The information collected by DataGet is placed in a database stored locally on that machine. The information recorded here is the raw information reported by each network controller, consisting of an employee number, operation number, station number, lot number, and operation time.

Another piece of information collected by DataGet is the current status of every component on the barcode network. This is important to collect so that the foreman coordinating movement on the production floor can see when there are problems with the system and correct them before significant problems occur. The status of these components is converted into a set of custom operations that are handled in special ways. These custom operations are stored in the database locally in the same manner.

DataRelocate

As DataGet operates continuously, polling the network controllers at specified intervals, it is necessary to copy the data from the local database to the remote secure database server. DataRelocate performs this operation at a specified interval.

DataRelocate logs into both the local database and the remote database server using secure passwords. It then checks to see if there is local data that has not been copied to the database server. If there is new data, then DataRelocate tries to connect to the database server to copy the new data. If the network connection between the local PC and the database server is inoperable, then DataRelocate pauses and tries to connect to the database server at a later time.

When the connection to the server is successful, DataRelocate copies the new local data to the server by calling a stored procedure on the database server. Stored procedures are discussed in the next section. DataRelocate only copies a specified number of data entries at once, so as to not use up too many system resources. Once those specified number of data entries have been copied, DataRelocate waits for a time before checking to see if there are more new local database entries that are in need of relocating.

Database Server

The database server is the main storage unit for all compiled data from each PC collecting data. All data from each manufacturing location, whether different rooms in the same building or different countries in the world, is located in this database.

The method for accessing data from the database server is via stored procedures. Stored procedures are written in an SQL (Structured Query Language) statement and are able to be called like functions by other programs that access the database server [Wynkoop, 1999]. The stored procedures in place provide various functions, such as data entry, data modification, data deletion, and data reporting. Stored procedures were used for several reasons.

Stored procedures allow for most of the database access code to be stored in the database that they are accessing. This allows the database administrator to slightly change the way that the database is accessed without changing all code that accesses the database. For example, if one of the tables in the database were to change slightly, so that some of the field names were changed, the database administrator could change that one stored procedure that accesses that table and that is all the code that would need to be modified. If there were a dozen programs that used that stored procedure, then that would save another programmer from editing and recompiling a dozen programs and then redistributing them to all personnel that used them. It doesn't take long to realize that this can be a huge time and money savings for system administrators.

Stored procedures also allows the database administrator to allow access to the database without having to worry about harming the data integrity. For example, if an inexperienced person wanted to run some reports on the data, giving that person full access to all tables could prove disastrous. One wrong SQL statement and valuable data (including someone's payroll check) could be lost forever. Stored procedures give the database administrator the ability to control

exactly what someone does to the database, since they know the exact code that will be run with each procedure.

Web Server

The web server provides the user front-end for the entire system. Using the WWW as the front end provides several important features. It eliminates the need for custom client-server applications to be developed. Also, almost every PC in the world has a web browser on it. This means that a PC off the shelf probably already has the software installed that is necessary to access the system, run reports, and perform data maintenance.

Password protection is provided through the web server [Morrison, 1997]. If the person is trying to access a web page that is above their permission level, or they haven't given a successful user name and password, then the web application denies that person permission to view that web page. This helps to eliminate multiple points of entry into the front end and helps to keep private information secure from prying eyes.

The web server is given a username and password for permission to run all stored procedures on the database server. Again, this helps to keep all database access code in one central location for easy maintenance.

Some of the functionality that the front end provides is essential to management, and other functionality is useful to a production supervisor. For example, some of the management functions are in the form of reports. These reports are able to track individual lots as they move through the system. A manager is able to see exactly what operations have been performed on that lot of product, what station those operations were performed at, what employee performed the work, and what time that employee did that operation. This type of real-time information was not available with the previously mentioned peel-and-stick label system.

Other functions useful to management would include the following:

Payroll Report – a report that shows how much piecework payroll an employee has accumulated over the specified dates

Lot History Report – a report that indicates the history of operations of a certain lot

Employee History Report – a report that details the history of work performed by an employee

as well as a multitude of functions that allow an administrator to modify an employee's profile, station profile, or operations performed on various lots of product.

In addition to the management reports available to production supervisors, there are a multitude of other informative screens that are available via the web server. One of the most important is the status screen. This screen shows a virtual factory floor. The supervisor is able to see at a glance, what errors have occurred with the system recently as well as stations are currently up. If stations aren't responding, it is an indication to call maintenance and remedy the problem immediately so no data is lost.

Entire System

At this time, it is important to discuss the entire system as a whole and discuss some of the concerns of the client company and how those concerns were addressed in the system.

Security

One of the concerns mentioned earlier was that of security. Anytime confidential information is being handled, it is important to make sure that that information is secure. The level of security used in this project could be considered "good". The company is only trying to keep the curious employee from trying to investigate the database entries. This was achieved through the use of secure operating system and server software.

Each PC on the factory floor is running Windows NT Workstation. This provides an environment that is not easily accessed without proper administrative permission. The local database is saved on this machine in an area that is not directly accessible without the administrator permission. Furthermore, even if someone were able to get through the security of the operating system, there is no way to get access to that local database, because the database viewing software is not installed on these machines. The database file is too large to fit on a single disk, so it would not be able to be easily transported without special equipment.

Security for the web server and the database server is accomplished through both the operating system (Windows NT Server) and through physical security (locked in the administrator's office). The database server and web server both employ the security of the operating system, making them virtually impossible to tamper with without administrator permission.

Since each barcode is a unique combination of the lot number concatenated with the operation number, it is impossible for employees to rescan barcodes to get extra pay. If employees make fake barcodes, then the system alerts the supervisor that unknown lots are being processed.

Fail-Safe Protection

There are several levels of fail-safe protection provided in this system. The first is facilitated through the battery backed memory in each of the barcode decoders and network controllers. This battery backed memory protects vital information in the event of a power failure.

The databases also employ a method of fail-safe protection. When moving information between the local database and the database server, the data must be received by the server and acknowledged before moving to the next record in the local database. Even if there were some kind of computer problem, the data is never deleted from the local database, but simply marked as read.

Reliability

There are several aspects of this project that promote a reliable system. One of the most important is the choice of operating systems. The Windows NT operating is made so that one

application cannot crash the entire computer. This is extremely important for a system that needs to work all day long.

Another aspect of reliability comes from the modularity of the components of the system. The information gathering and processing has no point of failure that can result in total system failure. Each piece of the system can function even in the absence of any other. For example, the barcode terminals hold all scans until the network controller polls and acknowledges the transfer. Similarly, the network controller holds all barcode until the PC polls and acknowledges the transfer. Furthermore, the system recovers from transient problems (such as the loss of a long-haul network) without the need of user intervention.

Speed

Since this is going to be used in a factory setting, it is necessary to have a system that will operate at an acceptable speed. It is necessary to have computers that are able to run both the custom PC applications simultaneously with power to spare. Since both applications are written to be small and perform a specific purpose, the speed of the computers needed is well below the powerful desktop computers currently in production.

Scalability

Scalability is one of the most important issues facing this project. Great care has been taken to assure that this system can be scaled up without a significant loss in performance. The system uses a multi-tiered architecture [Vaughn, 1998] and each tier is scalable. The lowest tier, the barcode network, can be scaled up simply by adding another network controller with new networks of barcode decoders and scanners. The middle tier consists of the database server. The final tier consists of the web server. These tiers can be easily upgraded to allow multiple servers and virtually unlimited capability (see Figure 2).

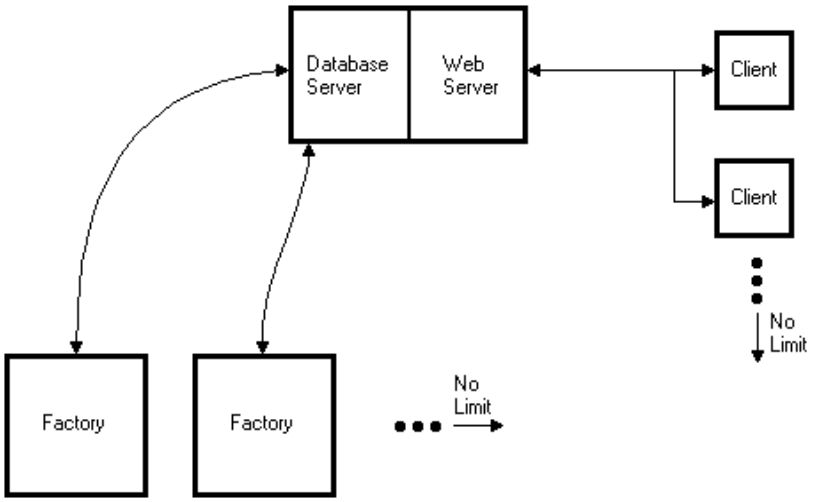

Figure 2. Block diagram of system demonstrating unlimited scalability

III. Conclusion

We have developed a multi-tiered system that provides management and production supervisors valuable tools to help manage a growing company. Through the use of modular components, this system is reliable, fast, and scalable. The user front-end is WWW based, simple and easy to operate, and provides a good level of security. The database server provides access to the data without giving all users permission to change or view critical values in the data.

This system was developed at the University of Maine with the help of Industrial sponsorship. It utilizes standard barcode equipment in a manner that is not common. This research has spawned numerous classroom-based activities involving multi-tiered architecture, barcode equipment gathering data, data storage through database applications, and web-based interface.

Bibliography

[Harris, 1999] G. Harris, "Neural network programming using rapid application development techniques:, SPIE Conference Proceedings, 1999.

[Morrison, 1997] M. Morrison, "Using Microsoft Visual Interdev", Que, Indianapolis, IN 1997.

[Vaughn, 1998] W. Vaughn, "Hitchhiker's Guide to Visual Basic and SQL Server", Microsoft Press, Redmond Washington, 1998.

[Welch-Allyn, 1999] Welch-Allyn Data Collection "Overview of Data Collection Technologies" http://dcd.welchallyn.com/techover/barcode.htm

[Wynkoop, 1999] S. Wynkoop, "Using SQL Server 7.0", Que, Indianapolis, IN, 1999.

MICHAEL AMOS

Michael D. Amos is currently an undergraduate student at the University of Maine pursuing a double major in Electrical and Computer Engineering. Mr. Amos received an Associates of Applied Science degree in Electromechanical Technology from Central Maine Technical College in 1991. Following that he worked for 3 years at Digital Equipment Corporation as an Electronics Technician.

BRUCE SEGEE

Bruce E. Segee is an Associate Professor of Electrical and Computer Engineering at the University of Maine. His research interests include Instrumentation, Automation, and Intelligent Systems. He is the Director of the Instrumentation Research Laboratory and a Member of the Intelligent Systems Group at the University of Maine. His work focuses on real-world deployable systems for use in manufacturing environments. Dr. Segee received his PhD from the Department of Electrical and Computer Engineering at University of New Hampshire in 1992.