Developing a Manufacturing Execution System (MES) using LabVIEW™

by
Vincent Demers
Projects Manager
Averna Technologies

Category:
Telecom

Products Used:
LabVIEW
SQL Toolkit

The Challenge:
Our client, a fast growing manufacturer of optic fiber components, is actually using a File Maker Pro system to record information during manufacturing. They want to increase productivity and to keep the fabrication history of every component delivered to their customers. Since our client is already using LabVIEW for many applications, they required a MES being developed using LabVIEW and MS-SQL 7.0.

The Solution:
A Tier-3 architecture approach has been used. LabVIEW was used to develop user interfaces and business services. The SQL module of LabVIEW was used to develop the data services. This approach was chosen to minimize rework when new modules will be integrated into the MES. Eventually, a stronger coupling between test sets, production control applications and the MES will be developed.

Abstract
Our MES application is tracking the fabrication of each component. All relevant data is recorded (Temperature, technician ID, work center ID, time, procedure, out-of-specs components, defects, etc…) and can be retrieved to analyze defects and improve quality. Reporting is done using Crystal Report.

Purchase orders from customers are transformed into work orders and the production is tracked through the MES. The products and their bill of material (BOM) are defined in the database, along with the work centers and the travelers. Change requests are also managed by the system.

The solution is expendable and the system may eventually include the Raw Material Treatment, Planning, Bidding, Invoicing, Procurement and Inventory Management modules.

Increasing productivity and enabling growth through a MES
The project initiated from the need of our client to track its components and to replace its paper-based tracking processes. They wanted to keep and retrieve the fabrication history of every component and assembly product. They also wanted to set the corner stones of a production system that could grow with the company and increase the overall plant productivity...

System features
Basically, the system tracks the components from ordering to wrapping. Once a client order is taken, the information is transferred into the MES. Production orders and work orders are created using the appropriate traveler (which is selected in the product definition phase). On the shop floor, there is usually a computer station for every work center (which is associated with a given production operation). When a technician begins a given operation, he scans the bar code of the current component. Automatically, this triggers the system to obtain from the database the information related to activities to be performed for that component for the actual operation. This way, every work order is tracked and production’s progress is monitored constantly.
Since products are often customized for a given customer, the MES has been developed with flexibility in mind. Consequently, users can easily modify or define new products, operations (i.e. sequences of activities), travelers (i.e. sequences of operations) and equipments in a timely matter. These actions are considered to be engineering changes and are tracked for later use. Request, consultation and approbation are performed using the MES.

Our client also required a high level of security in order to fulfill quality requirements and to pass ISO and BellCore auditing. Authorizations are required and performed using electronic signatures. Access to modules is controlled and user groups are defined. The system manages access and modification privileges in order to refrain unauthorized users from seeing sensible data or from modifying products and/or orders.

Future enhancements
The system has been designed with scalability and expendability in mind. Our client is a fast growing optical fiber manufacturer. Production is expected to increase exponentially over the next few years, and several plants will be opened. The new MES will therefore run in multiple plants and will be used by hundreds of users. Furthermore, many additional modules will be added over the next year and integrated within the system. These modules support processes such as Raw Material Treatment, Planning, Bidding, Invoicing, Procurement and Inventory Management. Depending on our client requirements, custom or off-the-shelf software will be integrated to provide the best solution.

Development methodology
The project started with a functional analysis that lasted several weeks. Then, system specifications were documented to provide software developers with a source of information to refer to. After acceptance of this document by our client, one of our system architects created the global design for the system, outlining the various system modules and their interrelations. Every system module was then subject to a detailed design. During that time, a database modeler was creating the database and was defining database procedures.

From there, a team of LabVIEW programmers started translating detailed designs into LabVIEW source code that followed our corporate coding standard. Each programmer was also responsible for defining unit tests for his modules. When these steps were completed, the system’s integrator performed unit tests. Upon success, each module was integrated in the next internal release of the system, which was then tested as a whole for performance and robustness.

At the beginning of the project, our project manager defined a development schedule and ensured during the length of the project that it was followed. This allowed us to react quickly to problems and to deliver the system on time. Activities such as quality assurance and client sign-offs were made regularly to ensure that the software never deviate from the client’s business objectives.

Technology and development tools
The system has been developed using a tier-3 architecture. This approach was used as a good software engineering practice to minimize rework and risks. The representation language used for the architecture was UML and the tool was Microsoft Visual Developer.

User Interfaces, Business Services and Data Services were coded in LabVIEW. Data services also made use of the SQL Toolkit. Since our client uses LabVIEW and other National Instruments products to create test bench, machine control and other specific applications, this was a natural selection. Since our client has many skillful and experienced LabVIEW developers, the maintenance of the system will be easier. Furthermore, the system will be more expendable because many resources are available to support and improve the system.

Our client already selected Microsoft as its general platform supplier. The selected development and deployment platform was then Windows NT 4.0/98 and MS SQL Server 7.0.
Conclusions
LabVIEW proved to be a valuable programming language for an IT project that was based on a structured software development approach. Since LabVIEW allows the usage of DLL and ActiveX components, we always felt that we could solve any potential performance issue by replacing critical code with performance and complexity-oriented languages.

Furthermore, we felt that a good LabVIEW programming standard was missing, so we created one and we initiated a discussion forum on the subject. Hopefully, this programming standard will grow and become a valued tool for developing LabVIEW applications in the future.