WORK FLOW MANAGER FOR LARGE MAILROOM AND PRINT-ON-DEMAND APPLICATIONS

Anthony F. Stuart*

Keywords: Architecture, Automation, Control, Model, Monitor, Open Systems, Work Flow

Abstract: An object oriented approach has been defined to model the hardware and software processes involved in high volume, customized document printing and finishing work flow applications. Based on the open system architecture of the Desktop Management Interface, a set of standard groups and attributes are described using the Management Information Format (MIF) language. A Graphical User Interface (GUI) interprets the MIF structures, extracts object data, and maps the manageable characteristics of the work flow processes to windows on a display to monitor and control the operations. This paper describes the architecture of the system and demonstrates the capabilities provided through the GUI to manage work flow in a large scale document printing and finishing environment.

Introduction

In recent years, manufacturers in the printing and finishing industry have made significant progress in the development of computer controlled devices. These advances in computer control technology have appeared in several distinct generations. First, dedicated microcontrollers were developed that featured simple control panels with switches and alphanumeric displays. Next came general purpose microprocessors with LCD panel or CRT displays. These were followed by high performance CISC and RISC microprocessors with touch screen displays and graphical user interfaces.

Along with these advances in microprocessor and user interface technology came advances in networking technology, and manufacturers began to incorporate network support into their devices using proprietary network protocols. With network support, one operator at a terminal could monitor and control a number of devices.

---

*IBM Printing Systems Company, 6300 Diagonal Highway, Boulder, CO 80301. Internet: tony@vnet.ibm.com

By the early 1990's, it was common for each machine in the printing and finishing process to have its own dedicated microprocessor. Groups of these machines were connected together by high performance networks. These computer controlled devices offered many improvements over their electro-mechanically controlled predecessors: they were easier to set up and operate; they provided a wider range of processing capabilities; and they were faster and more efficient. These aspects of computer control resulted in reduced personnel expenses, shorter turnaround times, and a higher quality product.

Despite the advances in technology, there was still a significant problem: there was no cooperation between manufacturers. This problem manifested itself in several ways. Each manufacturer had its own network technology and its own user interface technology. Each manufacturer concentrated on a single step in the process. One manufacturer's devices could not communicate with another manufacturer's devices. With these proprietary systems, customers were locked into a single vendor, and could not mix and match components to solve specific problems. There was no way to apply the advantages of computer control across the entire operation.

To address these issues, the IBM Printing Systems Company convened the Large Mailing Operations (LMO) Advisory Council in September 1992. The council, consisting of customers and manufacturers, was formed to discuss common concerns in the printing and finishing industry. (See Appendix A for a list of the members of the LMO Advisory Council.)

One of the top priorities identified by the customers on the LMO Advisory Council was to define a set of standards that would tie together this heterogenous environment and provide a system for the overall management of the entire printing and finishing operation. This system would define common interfaces, provide for the sharing of information, and reduce expenses by giving operators and managers a common view of the system. Customers were convinced that such a system could provide valuable insight into their operation and allow them to identify new ways to save time, materials and money.

The manufacturers on the LMO Advisory Council agreed to work together to build this system. The resulting effort lead to the selection of a standard management protocol, the development of standard device definitions, the identification of a mechanism to extend standard device definitions to accomodate product differentiation, the development of a set of network interfaces to facilitate vendor support, and the development of a graphical user interface to provide a common view of the system.

The system is called the LMO Systems Manager, or LMO. The LMO provides functions to monitor and control devices and to track jobs. It is now in use at a number of locations and customers are beginning to realize the benefits of total process control of their document printing and finishing operation.

The LMO Environment

In a typical large mailing operation, the printing process includes an MVS host system, one or more high speed printers with preprocessing devices such as a roll unwinds, and postprocessing devices such as an output folders, one or more inserters, and a number of other devices that support these major devices. The devices are manufactured by a variety of vendors. Many of the newer devices can be attached to a Local Area Network (LAN). See Figure 1.

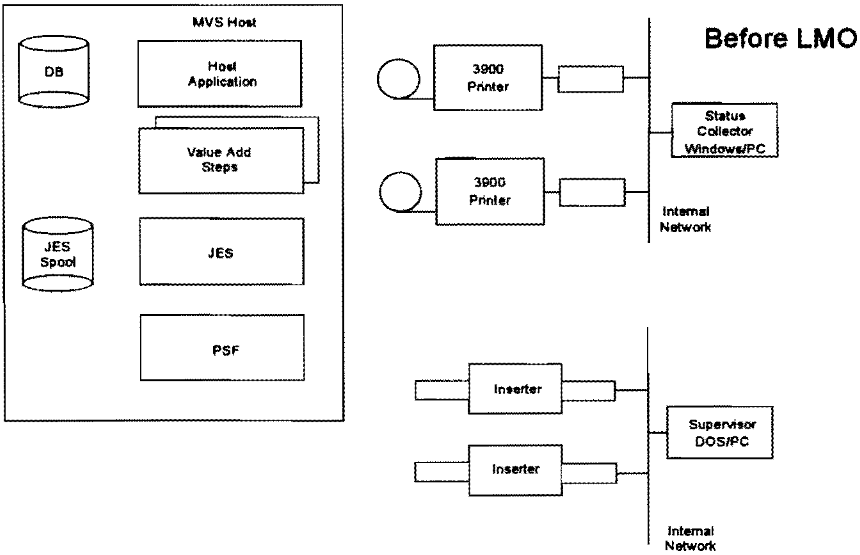

Figure 1. Large Mailing Operations - Before LMO Systems Manager

The print data is generated by a commercial application on an MVS host system. These applications are usually written in COBOL and run in batch mode. They are often "legacy" applications, that is, applications that have been around for a long time and are not easily modified. A typical host application would extract customer account information from a host database and generate a formatted statement for each customer account.

The print data is either generated on the large mailing operation MVS system or supplied to the system on magnetic tape or via a communications subsystem. The print data is then passed through one or more job steps that process the data. These processes vary, but often include address validation, zip code presorting, bar code generation, addition of inserter instructions, and segmentation. These processes are called "value add" steps. Each value add step may be supplied by a different vendor. Some are customer provided.

The print data produced by the commercial application is divided into smaller units for ease of handling. This process is called segmentation and is often done by one of the value add steps. Some definitions are useful here: the print data produced by the commericial application is called the product. The product is divided up into manageable chunks called mailjobs. Each mailjob is composed of pages that go into a single envelope. These are called mailpieces.

The printing and mailing process for a single large host application can generate millions of envelopes for delivery to the post office. A typical product consists of 150 mailjobs. Each mailjob consists of 10,000 mailpieces. Each mailpiece consists of 2 pages. In this example there are 1,500,000 mailpieces and 3,000,000 pages. Every month, a typical large mailing operation handles 2 to 5 products of this scale, and 10 to 15 smaller products.

The print data is spooled to an MVS Print Services Facility (PSF) printer using the Job Entry Subsystem (JES). An operator or administrator releases the print data from the JES spool to PSF and the data is printed. PSF, the printer, and intelligently coupled preprocessing or postprocessing devices are able to recover from paper jams or other hardware problems and reprint mutilated pages.

After printing, the mailjob is deleted from the JES spool. If any pages are lost after this point they must be regenerated and sent back through the process.

The printed, trimmed and folded pages are boxed or stacked on pallets and brought to the inserter, where they are inserted into envelopes with other materials. (Some inserters can keep track of mailpieces that they mutilate and display the information on their console.) The envelopes are then placed in trays and delivered to postal processing. From there, the envelopes are sent to the post office or directly to the airport for inclusion on specific flights to regional postal centers.

The final step in the process is reconciliation. In this step, the input of the process, print data, is correlated with the output of the process, finished envelopes, and any discrepancies are recorded. These discrepancies are usually used to generate a reprint job which then goes through the same process as the original mailjob.

The remainder of this paper will discuss the technical aspects of the design, implementation, and use of the LMO.

System Model

The printing and finishing process is much like other manufacturing processes. Raw materials, such as paper, toner, inks, inserts, glue, and wire, are converted into a finished product. As is the case in other manufacturing processes, the goal of this process is to produce a high quality finished product as quickly and as inexpensively as possible. Initially, the LMO Advisory Council believed that it might be possible to adapt existing manufacturing process control systems to the printing and finishing environment. After some research, however, the group came to the conclusion that a new system, tailored to the unique requirements of the printing and finishing process, was needed.

The first task in developing the LMO was to accurately model the components in the system. These components included not only hardware components, such as printers and finishers, but also software components that control devices and provide additional services. Two types of components were identified in the model: those that needed to manage other components, and those that needed to be managed by other components (see Figure 2). The LMO would implement an interface between the two types of components, and this interface became the basis for the LMO architecture.

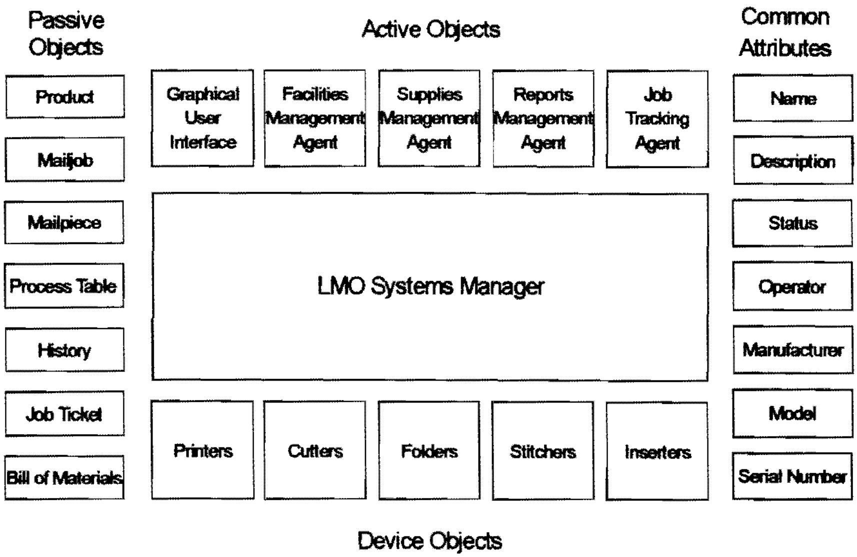

Figure 2. LMO Objects

The next task was to determine whether an existing management protocol could provide the necessary interface. Several standard management protocols were considered, including the Internet Engineering Task Force (IETF) Simple Network Management Protocol (SNMP), the International Standards Organization (ISO) Common Management Interface Protocol (CMIP), and the ISO Document Printing Application (DPA) standard. In addition, there was discussion of using a distributed database management protocol, such as the Open Database Connectiviy (ODBC) standard, and a suggestion that a new management protocol be defined for the system.

Desktop Management Interface

After much consideration, a relatively new interface standard, the Desktop Management Task Force (DMTF) Desktop Management Interface (DMI) protocol, was selected.

The DMI consists of four major components (see Figure 3). These four components are:

1) The Management Information Format (MIF) database that describes the manageable characteristics of the hardware and software components in the system.

2) The Management Interface through which management applications can request information from the MIF database or get and/or set attributes in components.

3) The Component Interface through which hardware and software components can respond to requests from management applications or generate asynchronous events.

4) The Service Layer, a program that provides both the Management Interface and the Component Interface.

In the DMI model, the Service Layer acts as an abstraction layer between management applications and components. With this abstraction layer, a management application does not need to know the details of how to communicate with a component and a component does not need to know the details of how to communicate with a management application.

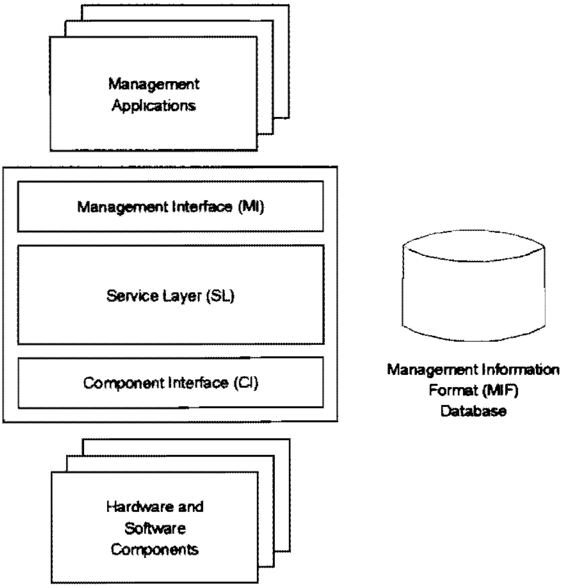

Figure 3. DMI Functional Block Diagram

The typical operation of the Service Layer is as follows: The Service Layer receives a request from a management application via the Management Interface, determines which component can service the request using the information in the MIF database, and forwards the request to the component via the Component Interface. It then waits for a response from the component and returns the response to the management application.

In addition to handling all interface processing, the Service Layer handles management application and component registration, component installation, arbitration, serialization, and multiplexing/demultiplexing of multiple commands. The Service Layer also receives asynchronous events from components and distributes them to management applications.

Objects, such as hardware and software components, are defined in the MIF using a hierarchy that consists of components, groups, attributes and values.

Note that a device can use both the Management Interface to discover and control other components, and the Component Interface to allow itself to be discovered and controlled by other management applications. The DMI interfaces are dynamic. Objects can start and stop as necessary without disturbing the rest of the system.

Applicability of DMI to LMO

The DMI standard was selected for use in the LMO system for the following reasons:

1) The DMI architecture provides interfaces for management applications, which are objects that manage other objects, and components, which are objects that are managed by other objects. This is precisely the model that was developed for the LMO.

2) The DMI defines an easy to use, flat file based, object definition language called the Management Information Format (MIF). Standard objects and standard interfaces can be defined using the MIF language. Manufacturers can build DMI objects from standard groups and manufacturer supplied groups. This provides support for product differentiation.

3) The MIF metadata (the data that describes the structure of the hardware and software components in the system) is available independently of the attribute values. This enables the development of extensible (table driven) management applications, such as Graphical User Interfaces (GUI).

4) The DMI "key" mechanism allows direct access to rows in tables based on an arbitrary number of keys of arbitrary type. The key mechanism can be overloaded to pass parameters to commands. Thus, it can be used as a general purpose command line interface.

5) The DMI does not specify a particular network protocol. A layered network interface can be defined that provides different levels of capability at different levels of ease of use. The network support can be made transport independent and can be built on top of a reliable, sequenced, nonduplicating protocol such as TCP/IP.

Despite the benefits of the DMI, there were also some problem areas that had to be addressed. Most of these were due to the use of a desktop management paradigm in a device management environment, and had to do with network access and performance issues in a large scale production printing system.

The network access issue was solved by defining two levels of network interface, both built on the industry standard TCP/IP network protocol.

The low level network interface uses a Remote Procedure Call (RPC) mechanism to provide an interface that is identical to the interfaces provided by the DMI. Once a management application or component on a client node has connected to an LMO server, it behaves exactly as if it is a logical extension of the server machine.

In addition to this low level interface, an easier to use, higher level interface is provided. The higher level interface provides network support using a simple get, set, list protocol. The higher level interface greatly simplifies the burden of providing an enabled device. This results in broader industry support.

The performance issue is based on the job tracking function provided by the system. In job tracking, a potentially large number of job objects is tracked by the system. Rather than store these objects in the DMI MIF database, the LMO provides a DMI component interface to a standard relational database. This enables the LMO to handle large numbers of objects efficiently and access the objects using standard DMI operations. See DMTF (1994) for a complete specification of the Desktop Management Interface.

Standards

Once the advisory council had agreed on a management protocol, it began the critical task of defining standard components. There were two aspects to this task. The first was to standardize the common attributes of each class of device, to meet the device control and monitoring requirement. The second was to standardize the attributes and processing instructions for the printed material, which constitute the manufactured goods produced by the overall system, to meet the job tracking requirement.

The advisory council spent approximately nine months working on standard definitions for a wide range of devices, including input devices, marking devices, output devices, finishers, and inserters. Examples of marking devices include laser printers or web offset presses. Examples of finishers include machines that stitch, fold, bind, trim, die cut, punch, perforate, slit, imprint, wrap, and band. Examples of inserting devices include machines that accumulate, collate, and combine materials and insert them in envelopes.

The intent of the standards effort was to standardize the common attributes of a device, that is, the attributes that were common across a broad range of similar devices from many vendors, but not to standardize every single attribute of every device. This resulted in an intersection of standard attributes rather than a union. This process created a much more manageable standard that provides support for a very important aspect of the industry: competition. Because a DMI component is made up of groups, and groups can be either standard groups or vendor extensions, it is possible to standardize the common attributes of a device and still provide support for competitive features or product differentiators. Many standards efforts have failed because they tried to create a superset of all existing devices or because they limited the capabilities of a device to a least common denominator. The advisory council standards enjoy broad support because they reflect the realities of the industry.

The standard device definitions define methods by which a device can report its physical processing capabilities, its current supplies status, and any alerts or errors. The standard also provides a mechanism by which a device can be controlled by another component, which could be either a device or a program. See IBM (1994) for a complete specification of the printing and finishing standards. The standards were formally approved by the DMTF in December, 1994.

System Architecture

The LMO is a client/server based architecture (see figure 4). The server consists of the DMI Service Layer, the Network Interface, and the Agents.

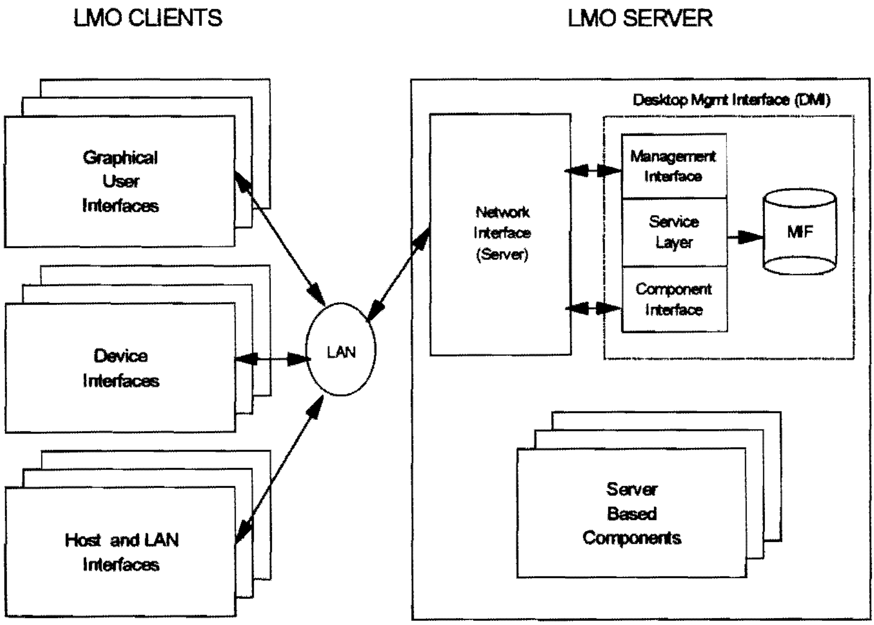

Figure 4. LMO System Architecture

The DMI Service Layer provides the Management Interface, the Component Interface, and access to the MIF database as described above.

The Network Interface distributes the low level DMI interfaces and the high level LMO interfaces across the network. The Network Interface uses RPC and Socket level services over the industry standard TCP/IP protocol. The server can be connected to the network using any network interface hardware supported by TCP/IP, for example, token ring and ethernet. One part of the Network Interface executes on the server and another part executes on the client.

The Agents are an open ended set of applications that provide value added services within the system. In the LMO environment, there are Agents for job tracking, job reconciliation, history, and alert monitoring. Any vendor can develop and install new Agents to run on the server.

Attached to the network are a variety of clients. These include devices that access the system using either the Management Interface or the Component Interface. The Graphical User Interface, which accesses the system using the network Management Interface, is one example of a client. Device controller code, which accesses the system using the network Component Interface, is another example of a client. The device controller code runs on the device controller, that is, the control hardware for the device. The clients are connected to the server via a TCP/IP capable network.

In addition to GUI's and Devices, there is also a client interface to a data processing host. In the LMO environment, this host is the source of the printed data. The system currently supports MVS hosts, with VM and AIX (UNIX) host support planned as future enhancements.

The server software is written in the 'C' programming language and runs on a variety of platforms. The Graphical User Interface operates in a WIN/OS2 or Windows environment.

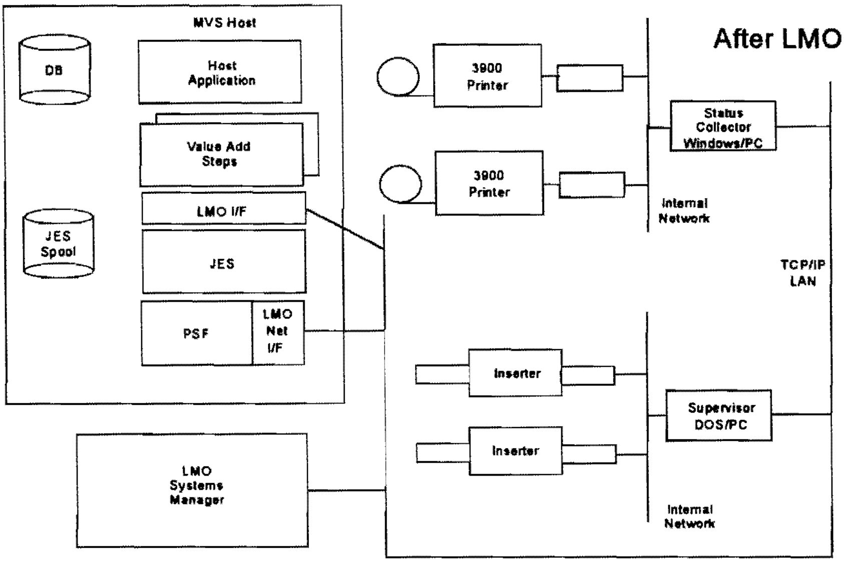

Figure 5. Large Mailing Operations--After LMO Systems Manager

Contrast Figure 5, Large Mailing Operations--Ater LMO Systems Manager with Figure 1, Large Mailing Operations--Before LMO Systems Manager.

Graphical User Interface

The GUI, an LMO client, provides an interface between the user and the network enabled DMI Management Interface. It features account level security that associates access permissions with a user or group of users. During the login process, the GUI requests the MIF metadata from the server. With this information, the GUI has an internal representation of the complete structure of the system, including the number and type of devices, their characteristics, the number and type of jobs in progress, as well as historical information.

Once logged in, the GUI displays system reports that were marked "display-on-login" in the account file. The reports are defined using an easy-to-use report definition language. The reports can display any information that is defined in the MIF database and are updated in real time. The report definition includes a configurable alert facility that allows an action to be taken when a value in a report matches a particular pattern or enters a particular range. These actions include displaying the value in an alternate color, displaying a bitmap graphic indicating a problem, or chaining to an external program, which could perform any user defined function, such as dialing a pager and sending an alphanumeric problem code.

A GUI user can double click on an item in a report and chain to another report, for example, one that gives more detail, or the user can chain to an update dialog box. This gives the user the opportunity to interact directly with a device, using a standard interface. The GUI also supports a configurable menu facility, so that common monitor or control operations can be defined as menu items.

The GUI supports an ad-hoc charting capability. Any information in any report can be displayed in chart format. The charting facility supports the standard chart types: bar, 3-D bar, stacked, percent, pie. The chart facility does automatic scaling, including a logarithmic scale. Charts can also be included as part of the report definition. These associated charts will be displayed automatically when the base report is opened.

A user can navigate directly through the objects defined in the MIF database using the GUI browser. The browser uses a hierarchical list box to display the objects in the MIF. Double clicking on a component displays the component's groups; double clicking on a group displays the group's attributes; and double clicking on an attribute displays the attribute's value. Using the browser, a device can be accessed directly by invoking an update dialog box.

Installation

A typical system consists of an LMO server, thirty or more devices, a host data processing facility such as MVS, ten GUI's, and a token ring or ethernet network.

System installation and configuration is a coordinated effort between the device manufacturers and the manager of the printing and finishing center. The steps are:

Install the hardware
Install the software
Configure the network
Install the MIF definitions
Customize the system
Document the customizations
Train users
Test the system

Installing the hardware and the software, and configuring the network are standard installation activities that will not be addressed here. Once this is complete, the next step is to install the MIF definitions. The device manufacturer provides a MIF file that describes the manageable characteristics of the device in terms of both standard groups and vendor extensions. The manufacturer also provides software on the manufacturer's device controller that supports the functions defined in the MIF file. The LMO system administrator invokes an installation program to install the MIF file into the MIF database. The installation program converts the human readable MIF file (see Figure 6) into a more efficient binary format. Once the MIF file is installed, the device is known to the system and any other management application in the system can get MIF metainformation on the device and can get or set attributes.

```
Start Component
    Name = "IBM 3900 Printer"
    Start Group
        Name = "ComponentID"
        Id = 1
        Class = "DMTF|ComponentID|1.0"
        Start Attribute
            Name = "Manufacturer"
            Id = 1
            Access = Read-Only
            Storage = Common
            Type = String(64)
            Value = "IBM Corp."
        End Attribute
        // More attribute definitions...
    End Group
    // More group definitions...
End Component
```

Figure 6. Sample MIF Source File

After the MIF definitions are installed in the MIF database on the server, devices in the system can register with the server, and the DMI Service Layer will pass requests from management applications to the device.

The next step in the installation process is customization. This involves tailoring the Graphical User Interface to meet specific end user requirements. The system is installed with a default set of reports that meet 80% to 90% of requirements. The installation team works with the operations staff to further customize the content and method by which information is displayed on the GUI.

With the system installed and customized and the users trained, the final step is to run an operational test. This is done by selecting a production job and running it through the system during off hours. Any problems that may be encountered are fixed.

With the completion of the test phase, the system is ready for production use.

Monitoring Functions

The LMO provides both device monitoring and job tracking functions. In this section, the device monitoring functions will be further explained.

What kind of device monitoring functions does the LMO support? Due to the table driven, open systems nature of the LMO, the answer is virtually anything the vendor can supply. At a minimum, this generally includes:

Current state of the device
Current state of major subsystems
Configuration of device and subsystems
Current workload in amount complete, amount to process
Current operator
Supply status
Hardware error/failure statistics
Device, operator, and shift performance

This information, for every device in the operation, can be displayed on the LMO GUI and saved in a relational database for future analysis. This provides valuable insight into the overall operation and gives both operators and administrators the information they need to improve job scheduling and device utilization, identify recurring problems in devices, identify operator training or performance issues, and give operators the ability to determine when supplies will need replacing.

Job Tracking Functions

In addition to device monitoring functions, the LMO provides job tracking capabilities. Although these job tracking capabilities are tailored to large mailing operations, the system is designed to provide general job tracking functions, and it can be tailored to other environments as needed.

The job tracking process starts with the host data processing components. These components use the LMO network interface to initialize product, job, and mailpiece information on the LMO server. The products, jobs, and mailpieces each have a unique identifier associated with them. These identifiers are the basis for job tracking throughout the system. As each device in the system processes a mail object (product, job or mailpiece), it updates the status of the object on the server.

As is the case with device monitoring, the system is table driven and can associate any information with a product, job, or mailpiece that is valuable to the operational and administrative staff. Product level information typically includes the following:

Product Name
Instance Qualifier
Owner
Date Created
Input Origin
Job Size
Number of Jobs
Accounting Data
Description

A product consists of one or more jobs. Job level information usually includes the following:

Job Name
Job Ticket Name (Processing Instructions)
Print File Name
Print File Format
Date Created
Job Size
Current Device
Current Status
Description

A job consists of one or more mailpieces. The LMO mailpiece information consists of a cross reference table that identifies the product and job, as well as its current status.

In addition to retaining information on products, jobs, and mailpieces, the system provides the capability to define process steps, record process history, and define bill of materials and job ticket information.

The LMO provides agents that facilitate the tracking process. These agents use the attributes described above. They give operators and administrators access to the following types of information:

What supplies were used by a job?
How much work is complete on a product?
How much work is remaining?
What devices are ready or will be ready?
What jobs are ready to be processed?
Where is a particular mailpiece in the system?

What mailpieces where mutilated?
What mailpieces need to be regenerated?
Can a job or product be reconciled?

This information provides valuable insight into the printing and finishing operation and is valuable to a number of departments, including operations, accounting, quality assurance and customer service.

Integration

The open architecture of the LMO provides interfaces for integration with third party applications. Users and vendors have shown considerable interest in integrating existing supplies/inventory accounting systems, scheduling systems, and archival viewing/reprint systems.

Future Directions

The architecture and system described herein were originally designed to solve problems in the large mailing operations environment. As the design and implementation progressed, it became clear that the system was generic and could be applied to other problems. For example, the print on demand environment is similar to the printing and finishing process for large mailing operations. The developers of the LMO system believe the LMO technology can provide device monitoring and job tracking functions for this developing industry.

Conclusion

The LMO has demonstrated itself to be a valuable mechanism for both users and manufacturers to apply advances in computer and network technology to the problem of managing the printing and finishing operation in a large mailing operations environment. Factors that have contributed to the success of the project include:

Demonstrated customer need
Widespread manufacturer support
Standardization of common attributes
Support for vendor extensions
Support for product differentiation
Open systems technology that is open to all vendors
Clearly defined interfaces and responsibilities
Easy to use network interfaces
Easy to use object model

The LMO system's open architecture will contribute to its longevity by allowing it to adapt to the rapidly evolving needs of the printing and finishing industry.

1994, "Desktop Management Interface Reference Manual" (Desktop Management Task Force).

1994, "Desktop Management Interface Specification" (Desktop Management Task Force).

1994, "Large Mailing Operations Standards Specification" (IBM Corporation).

1995, "LMO System, Guide to the GUI" (IBM Corporation).

1995, "LMO System, Migration Planning Guide" (IBM Corporation).

1995, "LMO System, System Administrator's Guide" (IBM Corporation)

1995, "LMO System, User's Guide" (IBM Corporation).

Appendix A

The LMO Advisory Council was convened by the IBM Printing Systems Company in September, 1992. It consisted of the following customers and vendors:

Customers

Advantis
Aetna
Bell South
EDS
Merrill Lynch
Pacific Bell
Tritech

Vendors

Bell + Howell
BESTE Bunch
C.O.P.E.
Champion Paper
IBM Printing Systems Company
Johnson & Quin
Pitney Bowes
Roll Systems
Wallace

In December, 1993, the vendors on the LMO Advisory Council formed the LMO Vendor Council to develop the Large Mailing Operations Standards Specification. Today, the LMO Vendor Council consists of the following companies:

Bell + Howell
BESTE Bunch
Bowe Systec
C.O.P.E.
Gunther International
Hunkeler
IBM Printing Systems Company
J.Q. Technologies (Johnson & Quin)
Kern
LASERMAX (Wallace/Stralfors)
Moore Business Forms
Pitney Bowes
Roll Systems
Standard Register

The LMO Systems Manager uses the Desktop Management Interface (DMI). The DMI Specification is a product of the Desktop Management Task Force:

Digital Equipment Corp.
Hewlett-Packard
IBM Corp.
Intel Corp.
Microsoft Corp.
Novell Inc.
SunSoft Inc.
SynOptics Communications, Inc.