MES-DCS INTEGRATION CASE STUDY

Presented at the
World Batch Forum –2001 North American Conference

Richard E. Parapar
Principal Systems Architect, Operations Systems
Genentech, Inc.
1 DNA Way, S. San Francisco, California 94080
650/225-5044
parapar@gene.com

KEY WORDS
Manufacturing Execution System, Distributed Control System, Application Integration

ABSTRACT
This paper addresses the design and implementation of a real time, batch scheduling and material reporting interface between MES and DCS applications in a good manufacturing practice (GMP) environment. Included is a discussion of the process functionality and data model applied to successfully implement such an interface. The design approach applies standardized technologies where possible to help ensure inter-application compatibility, data integrity, and continued expandability.

INTRODUCTION
Genentech’s new biopharmaceutical production facility in Vacaville, California, is a state of the art bulk-manufacturing plant capable of high volume multi-product, multi-campaign production. The plant is highly automated and employs a centralized distributed control system (DCS). The control application design is based on the ISA S88.01 Standard for Batch Control, Part 1: Models and Terminology.

To increase production efficiency and ensure complete, accurate material management, the DCS was integrated into Genentech’s production operations and material management environment, called SYNOPSIS (Synergistic Operations Information Systems). This facilitated automatic scheduling of batch recipe execution in the DCS based on the release of manufacturing Shop Orders from the
enterprise resource planning (ERP) system. Material consumption and production information collected during subsequent recipe execution is reported by the DCS to the manufacturing execution system (MES). The MES is responsible for maintaining the comprehensive material lot genealogy of production. This information is required by Quality Assurance to ensure only properly manufactured lots are released to market. The DCS also reports operational data for lot status updating to ERP, via the MES, in support of production planning and finite scheduling applications.

**PROJECT REQUIREMENTS**

The significant business objectives of the MESDCS Interface Project are three fold:

- **Provide real time lot status from the plant floor to the ERP System.** Visibility into batch schedule execution supports phase material delivery, resulting in more efficient plant operations and cost-effective material management.

- **Update material inventory in ERP.** DCS material consumption and production reports applied directly against inventory levels in ERP drive the demand for future raw material and work in progress (WIP) orders and satisfy the demand for production orders driven by MRP.

- **Integrate DCS Batch History into MES Genealogy.** A single, comprehensive material genealogy, supporting both forward and backward lot trace capability, facilitates more efficient material lot review and release by Quality Assurance.

**FUNCTIONAL SPECIFICATION**

To accomplish its objectives, the MESDCS Interface must satisfy the following functional requirements:

1) **Integrated Recipe Initiation.** Control Recipes in the DCS are automatically initiated from the MES and ERP applications. Recipe header information provided by MES serves to identify and classify material and operations data subsequently reported by the DCS during batch execution.

2) **Automated Material Consumption and Production Reporting.** Raw material, utility, and intermediate consumptions are reported by DCS, logged in lot genealogy and automatically decrement inventory levels in ERP. Intermediate and Final Product production reports increment inventory and are applied against production Shop Orders in the ERP.

3) **Automated Operations Status Reporting.** As Shop Order-initiated batch recipes execute in the DCS, manufacturing operation, work center and labor parameter values are collected and reported to ERP.

**Integrated Recipe Initiation**

In the DCS, automated recipe driven procedures are defined and executed by the Direktor Batch Management application. These procedures are called “Master Recipes”. They direct the DCS in the production of finished goods and intermediate materials, as well as preparation of manufacturing equipment (e.g., CIP, SIP, sanitization.)
Master Recipes are categorized into three groups of automated procedures: 1) Product and Multi-use Intermediate Production, 2) Single-use Intermediate Production, and 3) Equipment Support. Product and Multi-use Intermediate Recipes are executed on behalf of Shop Orders generated by ERP. Single-use Intermediate and Equipment Support recipes are initiated at the discretion of plant personnel, independent of ERP.

To maintain lot trace capability between MES Lot Genealogy and the DCS Batch History Record, DCS recipes are initiated using a unique Lot Number identifier. As Shop Orders are issued by ERP, the MES identifies and initiates the appropriate DCS Master Recipe based on the Item Number and Lot Number specified in the Shop Order. For recipes initiated manually by the Production Scheduler, the MES assigns a new Lot Number for the batch.

Material Consumption and Production Reporting

During the execution of a recipe, the DCS prompts operators for material to be consumed into the batch. Consumption of intermediate materials (e.g., buffers and media) are reported automatically by the DCS as equipment units with the appropriate status and contents are acquired by a production recipe.

Material consumption information is collected in the DCS Batch History Record and automatically forwarded to MES. The MES updates material status in its Lot Genealogy database and forwards the material consumption to ERP for update of inventory. When DCS consumptions are reported on behalf of Shop Order-initiated recipes, the MES also determines the corresponding Bill of Material (BOM) for that item and charges the consumption against the Shop Order in ERP.

The DCS reports Shop Order-initiated recipe material production to ERP, via MES. As control recipes execute, the DCS logs the Material ID, Lot ID, Quantity Produced, Unit of Measure and Completion Date & Time in the DCS Batch History Record. This information is collected and reported to MES. The MES then issues the appropriate ‘S/O Receipt’ transaction to ERP to satisfy the Shop Order and update inventory. Also, bar code labels identifying material, lot, quantity, unit of measure, and expiration date are automatically printed as per recipe specification.

Operations Reporting

For recipes initiated on behalf of ERP Shop Orders, the DCS reports batch execution status at significant checkpoints in the recipe’s procedure. These checkpoints are configured using “MES Report” Phases that contain appropriate operations parameters (e.g., Item #, Lot #, Operation ID, Work Center, Labor Code, Date/Time). When the phase executes as part of a recipe procedure, its parameter values are automatically stored in the DCS Batch History Record. This data is collected and forwarded to MES and ERP.
**SYSTEM SPECIFICATION**

**Application Overview**

Moore Products, Co.’s *Advanced Process Automation and Control System* (APACS) is the DCS platform. It is responsible for automated process control and maintenance of its own Batch History Record. Batch management is performed by Wonderware’s *Direktor* application. Honeywell-POMS Corp.’s *Production Operation Management System* (POMS) is the MES application. It is responsible for production control, inventory management, and maintaining lot genealogy. SSA’s *Business Planning and Control System* (BPCS) is the ERP application responsible for production planning and inventory control.

The following figure summarizes the major functional interactions between the integrated applications.

![Diagram](image)

**System Architecture**

The MESDCS Interface’s system architecture spans both the DCS and MES platforms. ERP software components are not impacted. Common SQL*Net functionality provides secure, reliable communications between DCS-resident and MES-resident Interface components. Message passing is accomplished by inserting and retrieving transactions from MESDCS Interface tables defined in the MES database.

In addition to providing new Recipe Selection & Initiation functionality, MESDCS Interface components residing in MES are responsible for integrating with existing MES functionality for Shop Order Processing, Lot Genealogy and ERP Reporting. MES logic components are written in the POMS proprietary language and operate against an “MES” data schema implemented using an Oracle database.

MESDCS Interface components residing on the DCS are responsible for automated recipe initiation in the Direktor application and collecting/forwarding specific reporting information to the MES. They are written in the C Language and execute in a UNIX environment, employing Pro*C library functions for interacting with tables in the POMS Oracle database. These components also utilize a vendor supplied Application Program Interface (API) for the Direktor Batch Management application. “Tool Kit” API functions allow custom programming to perform remote recipe initiation and retrieve information from the DCS Batch History Record.
DCS Recipe Initiation

The MESDCS Interface supports both automatic Shop Order-based and manual MES-based recipe initiation in the DCS via a common interface to the DCS Direktor application. The following diagram illustrates the various MESDCS Interface components responsible for performing DCS recipe initiation:

The DCS-resident **GNE_MesDcs** module processes ‘Recipe Initiation Requests’ from MES. These take the form of entries into the **Recipe Initiation Table**. The **GNE_MesDcs** module periodically awakes and queries the table for new recipe initiation requests. For each request found, it invokes the Batch Talk API **btAddScheduleEd** function to initiate the indicated DCS master recipe. When successful, the request is deleted from the table. If unable to initiate the recipe, the table entry is updated with an error indication and additional diagnostic information. The module processes all pending requests before again putting itself to sleep for a configurable period.

Entries into the **Recipe Initiation Table** can be made by either of two MES-based components. The first, called the **Shop Order Processor**, is responsible for initiating recipes in the DCS on behalf of Shop Orders issued by ERP. The module determines if the Shop Order warrants initiation of an associated...
recipe in the DCS. It compares the Item Number specified in the Shop Order against a configured list of DCS Master Recipes maintained in the **Master Recipe Table**. If a match is found, a new entry is created in the **Recipe Initiation Table** so that the recipe will be initiated in Direktor by the **GNE_MesDcs** module.

The second MES-based component creating entries in the **Recipe Initiation Table** is called the **Recipe Selector** module. It provides an HMI interface through which MES Production Schedulers can select specific recipes to be initiated in the DCS. MES workstation displays provide a view to the categorized and sorted contents of the **Master Recipe Table**. After the user makes his selection, the **Recipe Selector** module obtains a new Lot ID and inserts a recipe initiation request into the **Recipe Initiation Table**.

Both the **Shop Order Processor** and **Recipe Selector** modules keep a record of all recipe initiations in the **Lot Log Table**. An entry is created for every recipe initiation performed. This information is subsequently used in tracking recipe execution, as well as to process resulting consumption, operations, and production reports from the DCS.

**DCS to MES Reporting**

Three different functions of the Batch Talk API are employed by the MESDCS Interface to collect batch data from the DCS. The **dfHistHook** function is used to intercept and examine Direktor Material Input and Material Output records on their way to the DCS Batch History Record. These transactions contain the information forwarded to MES as DCS Consumption and Production Reports.

The **dfPhaseComplete** function allows the MESDCS Interface to detect the execution of specific recipe phases (e.g., “MES_REPORT”) as they are processed in real time by the DCS, and collect relevant phase parameter values. These parameters are configured to contain work center and labor code values specific to that recipe. This information is collected by the MESDCS Interface and forwarded to MES as DCS Operation Reports.

The third API function employed by the MESDCS Interface is the **btStartScheduleEd** function. This function allows a custom application to subscribe with the Direktor Batch Manager to automatically receive schedule (i.e., batch) status changes. Via this mechanism, the Direktor Batch Manager informs the MESDCS Interface in real time of status changes during recipe batch execution. This information is forwarded to MES as DCS Recipe Status Reports.

It should be noted that the Direktor Batch Manager executes as a single threaded application and that utilization of these API functions implies that the user’s custom modules will be executed in line with the Direktor application’s process. Both the quantity and complexity of activity performed by a custom module must be minimized so as not to impact Direktor performance and reliability. The MESDCS Interface modules which directly interact with the API functions are identified below:

- **Batch Talk API History Hook**: GNE_HistHk
- **Batch Talk API Phase Complete**: GNE_PhsCmp
- **Batch Talk API Schedule Update**: GNE_BTMessenger

Each of the MESDCS Interface API modules is responsible for identifying target data as it is processed by the Direktor Batch Manager, and forwarding the information to the **GNE_MesDcs** module.

The **GNE_MesDcs** module is responsible for the transfer of all information to the MES. It assembles transaction data collected by the API modules and inserts “DCS Report” messages into the **DCS Report**...
Table in the MES database. The **DCS Report Table** provides a standardized mechanism for conveying all batch status and material data required of the DCS. Each entry in the table represents a specific type of DCS Report: Consumption, Operation, Production, and Recipe Status.

The MES-based **MES Reporter** module is responsible for retrieving and processing the various report messages found in the **DCS Report Table**. The **MES Reporter** first qualifies each message by matching it’s header identification with a corresponding entry in the **Lot Log Table** (created at recipe initiation). If no match is found, the invalid DCS Report is deleted without further processing. Only after successful qualification does the **MES Reporter** continue to process the DCS Report according to its type. Thus, the **Lot Log Table** acts as a filter, allowing only qualified DCS Reports to impact the MES and ERP databases.

Information from the **Master Recipe** and **Lot Log Tables**, as well as in other tables in the MES database, is used in preparing subsequent inventory adjustment and lot status update transactions to ERP. The following diagram illustrates the MESDCS Interface components responsible collecting and forwarding all consumption, operations, and production information between the DCS and MES.
Master Recipe Configuration

The final component of the MESDCS Interface is the MES-based Recipe Configuration module. This module provides an HMI interface which allows a user to configure DCS recipe information in the Master Recipe Table. This is the only module permitted to update the Master Recipe Table. All other modules have read-only access.

A recipe cannot be initiated by either the Shop Order Processor or the Recipe Selector modules while its entry in the Master Recipe Table is being edited by a user. The MES-based Recipe Selector HMI display prohibits the user from selecting the recipe while its Master Recipe Table entry is locked for editing. In the case of automatic Shop Order initiation, a Lot Log Table entry is created with a status of “EDITLOCK”. No further processing of the recipe initiation request will occur until its status is reset.

PROJECT SUMMARY

The MESDCS Interface has operated successfully in production since May, 1998. To date, it maintains a 99.999 % availability rate and processes an average of 1100 recipe initiation and report transactions every day. The GNE_MesDcs component has been integrated into the Direktor Environment Management application so that it is started and stopped in tandem with the Direktor Batch Manager. It automatically generates detailed log files describing every transaction processed. This maintains the accurate data trail required in a GMP environment.

Implementation of the MESDCS Interface components using basic, readily available and established technology results in a system that is robust and reliable. It also helped to make the testing and validation phases of system implementation straightforward and more efficient to execute. The resulting application has the flexibility required to handle the complex data mapping and synchronization requirements often encountered when integrating dissimilar applications.