Presented at the
WBF
North American Conference
Atlanta, GA
March 5-8, 2006

195 Wekiva Springs Road, Suite 200
Longwood, FL 32779-2552
+1.407.774.5764
Fax: +1.407.774.6751
E-mail: info@wbf.org
www.wbf.org

The road to full MES integration

Practical experience from the pharmaceutical industry

Niels Andersen, M Sc (Physics)
Automation Specialist
NNE A/S
Gladsaxevej 363
DK-2860 Soeborg
Denmark
Tel. +45 30 75 94 88
Fax. +45 44 44 37 77
email: nca@nne.dk

KEY WORDS

S88, MES, Integration, Batch, Engineering

ABSTRACT

This paper describes the successful implementation of a MES system on top of a PCS system in the world's largest insulin API plant, operating more than 300 units with more than 500 recipes and 600.000 parameters.

The functionality of the actual database MES application is described, along with it's evolution from early versions in finished goods production to a modular and flexible large scale MES application within API production.

The solution adheres to S88. The architecture introduces kernel modules encircled by libraries of software extensions. The general database model has proved very robust and the simple principle of recursive operations combined with a simple execution principle of an operations state-machine forms the basis for a very versatile MES integration platform and a powerful batch modeling system. Further it includes effective tools for master recipe and report definition facilitating combined development.

For integration, the importance of a well defined interface to distribute different aspects of control between database MES and real-time PCS systems is described. Finally, the paper describes the impact of the modular design and reuse on execution in fast track projects, which recently contributed to substantially reduced project execution times for both pharmaceutical greenfield and upgrade projects.

PAPER

Project background

Two years ago NNE A/S handed over the Insulin Bulk Plant (IBP) project to the customer Novo Nordisk A/S, a world leader in diabetes care. The plant is the world's largest insulin API production plant with more than 300 units, tanks, centrifuges, columns etc. The insulin factory is a so-called two-lane, full length superhighway, producing two products from fermentation through recovery and purification to crystallization.

During the conceptual design, it was decided to install a PCS system with a MES system on top for combined batch execution, thereby using the traditional definition of MES as abbreviation for Manufacturing Execution System.

The decision was based on experience from similar projects where it had been difficult to engineer the recipe control and unit synchronization at high levels within a PCS system alone. Further, additional requirements to provide data as basis for continuous process optimization and automatic, paperless batch and quality documentation directly during production would be difficult to implement in the PCS system too. Also the system should prevent errors and defects by keeping track of all operations and by automatic identification of materials, equipment and personnel. Finally it should ensure and demonstrate regulatory compliance, including compliance to CFR 21 Part 11.

After screening the market for suitable standard MES solutions, it was decided to build on experiences drawn from an in-house database application implemented in a formulation area a few years earlier.

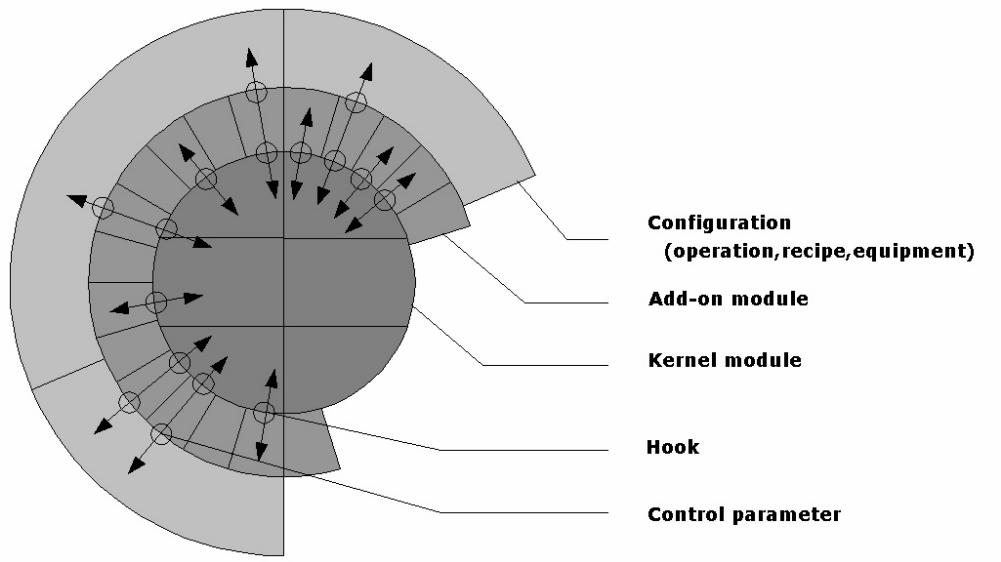
It was further decided to split the execution responsibility between the real-time PCS system and the database application at the S88 standard operational level. Also, the design and implementation of the PCS system took its starting point in a similar application, which was broken up in operations, leaving both procedural and recipe control to the MES system along with handling of manual operations.

System architecture

The building blocks of the MES application consisted of subsystems for handling of master recipes, material definitions, material control, equipment modeling and batch execution. A mapping proved that the existing toolbox covered some 2/3 of the functional requirements, exposing gaps at the reporting, material handling and execution functionality regarding arbitration and queuing. Also a proper OPC based interface between PCS and MES had to be developed.

A subsystem was added to handle report definition and generation. Much of the added functionality was developed as general purpose tools and organized in a number of library modules.

The configuration of the system defines master recipes, reports, units, materials, etc. The entire application may be described as a number of concentric circles:



The inner circle depicts the 6 kernel subsystems for

- Master recipe handling
- Equipment modeling
- Material definition
- Material control
- Batch execution
- Report definition and generation

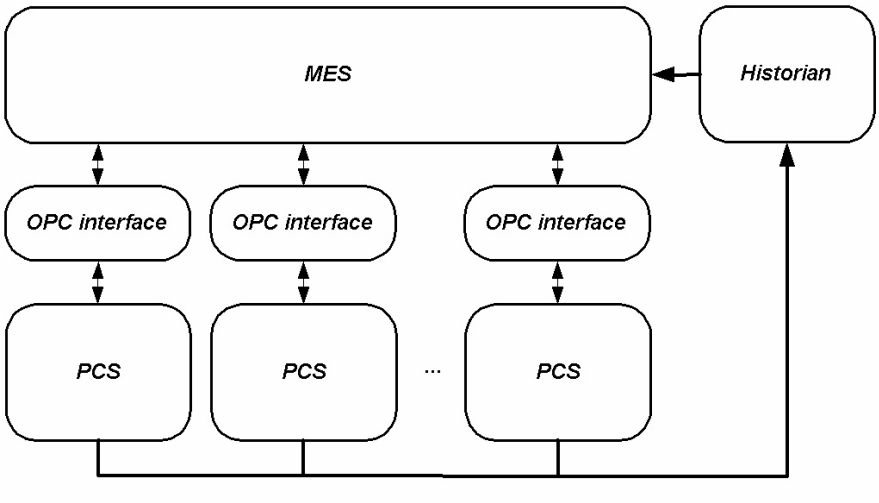
The kernel provides the basis for a MES system within any kind of batch production.

The middle circle illustrates the libraries of add-ons necessary to operate a bulk factory. It provides manual operations and material handling with barcode identification for quality sampling, weighing, charging, filter testing etc. This level, the library of add-ons, defines a bulk factory MES application.

Finally the outermost circle shows the configuration of materials, equipment, locations, master recipes procedures and reports. This layer makes the system a specific insulin API production MES application.

The interactions between the layers are typically controlled by hooks and control parameters. The kernel contains a number of so-called “hooks”, i.e. calls at predefined events to library procedures that have been registered through a standard hook interface table.

As the database application is generally not suitable for collection of time series data, a data historian was added. The complete architecture, which in a coarse overview, ignoring interfaces to barcode readers, scales and any other peripheral devices, may be outlined as:



The figure shows the division of PCS into a number of independent systems, which allowed for individual, parallel implementation, test and validation during project execution and which added to maintainability during operation.

Data model and general execution principle

The data model of the MES solution adheres to S88. The interface between MES and PCS is at the operation level, meaning that MES controls the sequence of operations within procedures and PCS executes these operations one at a time. However, the MES data model relies on the S88 principles of collapsibility and expandability of the procedural model leaving only the recipe level and an infinite recursive operation level. To distinguish between basic automatic or manual operations and their containing operations, the term compound operation is used for the latter. An example is shown in the section on the master recipe handling. This implies that the master recipe programmer freely can model both the traditional recipe and unit procedures into a number of compound operation levels.

Every operation has a status associated with it. The individual states are specified as part of operation definition along with the possible transitions between them. During execution an operation will pass through the relevant states, driven by a state machine. The state machine is the core of the batch execution. In principle it just waits for an external basic operation status change to occur, which can be caused by only events: 1) start of control recipe, 2) by execution of a manual operation or 3) by automatic operation as response from the PCS system. A few mandatory states exist with any optional state in between.

Operations are lined to sequences by status relations. Typically the next operation will ‘start’, when the previous is ‘completed’, but branches can be defined at any intermediate state. Operations are combined into a hierarchical tree-like structure through relations. Relations may have conditions associated and operations may have start conditions.

At execution, the state machine responds to an operation status change in a database transaction which evaluates the entire operation tree of resulting operation status updates through their relations. Having evaluated all transitions to be legal and all conditions to be fulfilled, it commits the new state of the recipe. This in turn results in activation of either a manual operation invoked through a hook or an automated operation invoked through the interface to PCS.

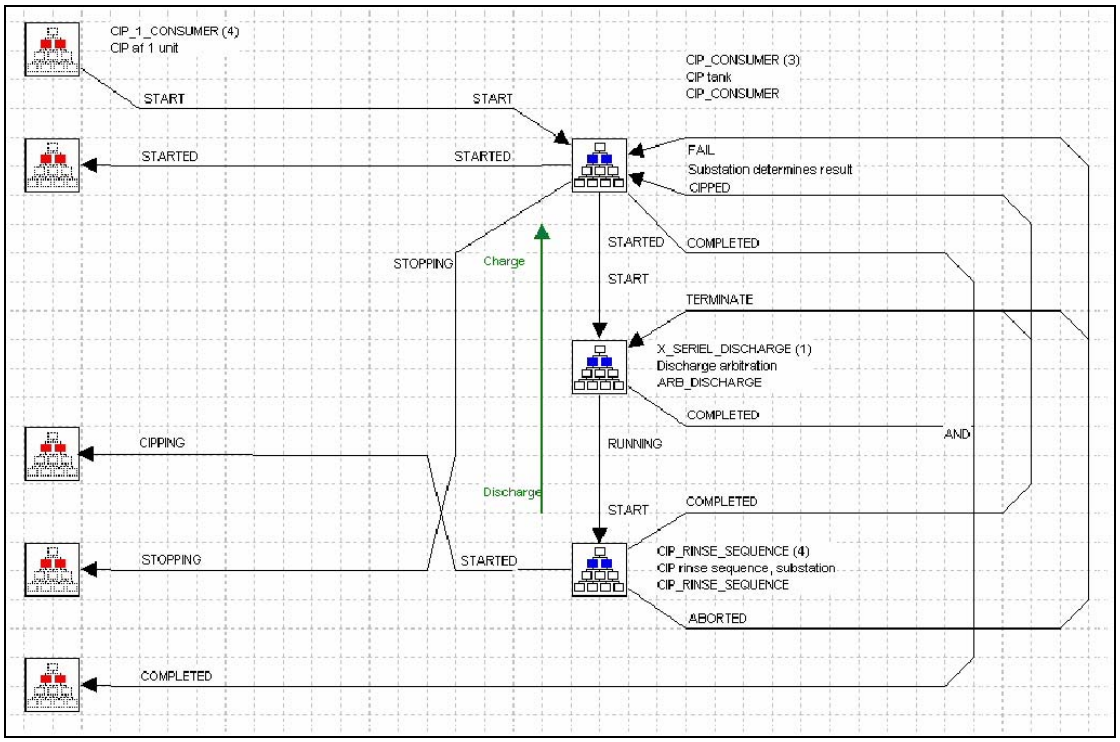
Master recipe handling

The master recipe definition is done in a graphical tool. The following two illustrations show the general CIP recipe procedure and part of the CIP unit procedure displaying only the first three rinses. The CIP concept is that the consumer unit starts its CIP procedure, followed by start of the valve matrix (x-unit) operation and finally the procedure executed at a CIP substation, which contains the sequence of rinse operations with water, base and acid in proper amounts and temperatures.

The CIP procedure (sequence) consists of a number of instances of the same rinse unit operation. The operations are instances of the same basic operation. They have identical alias tags (CIP_RINSE) but different instance tags (RINSE_1,_2,). The rinse operation source has a number of parameters defining media, amount, temperature etc. The value of the media is overwritten in each operation instance, for example first rinse with water, the second with base. Combining rinse operation instances to procedures and procedure instances to recipes, you may choose to perform overwriting at various appropriate levels, i.e. the sequence of medias remain unchanged irrespective of which unit is to be washed, hence you can overwrite the media parameter at the instance level, thus resolving the media values at the unit procedure compound operation level, while the amounts of media depends on the type or characteristics of a unit, so amount parameters are left for overwriting at the recipe level.

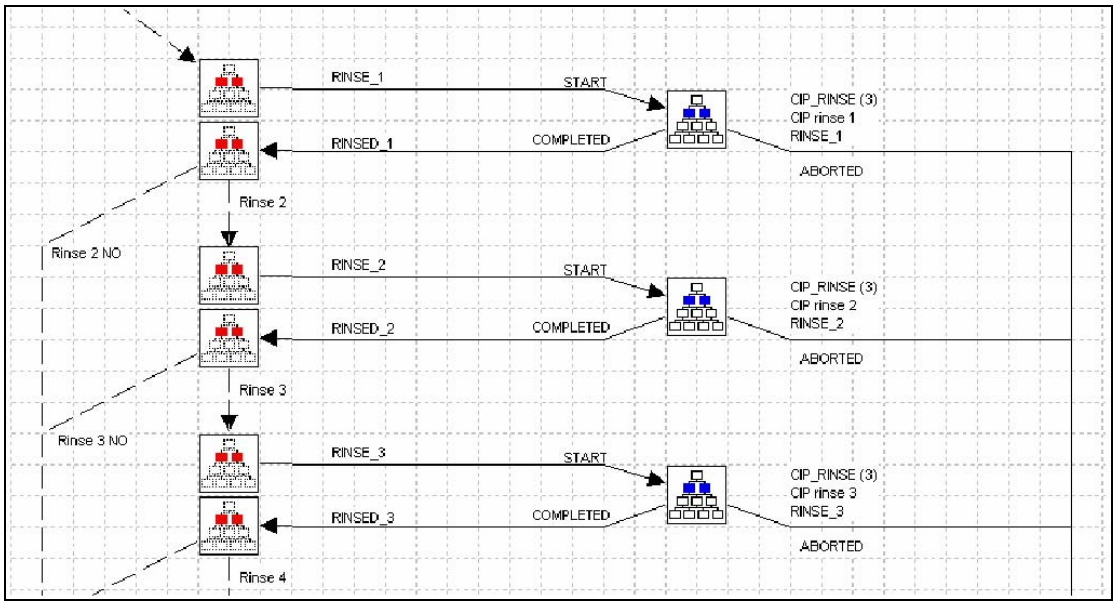
The overwriting principle also applies to equipment types, which are assigned to master recipes and their operations. The principle allows one compound operation as shown in the first figure to control the CIP of all units irrespective of their type, being tanks, centrifuges, x-units etc. This compound operation controls the cleaning of all 300+ units. The compound operation is instantiated in approximately 100 master recipes, since one master recipe can clean any unit of a distinct type. The actual unit is specified at control recipe generation, where equipment is also allocated to the CIP substation and the x-unit.

Typically operation start conditions concern locking and cleaning. The condition names refer to functions stored in the database, which will return true or false depending on the value of the equipment status in question. Cleaning operations will omit the condition on cleaning status, while it is mandatory in production operations. Actually the start conditions apply only to the first unit operation, which, if started, modifies the current status parameters, setting for example the locking status and the current batch number. Subsequent operations may then depend on start conditions specifying same batch to prevent batch mix-up. Finally when releasing the unit, the last operation resets the locking when completed. The equipment status modifications are defined in the master recipe and prevent batch mix-up.



CIP recipe procedure, CIP_1_CONSUMER.

(Icons far left refer to state of parent compound operation)

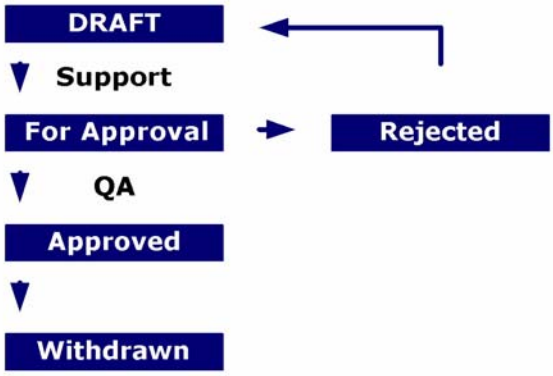


Part of CIP unit procedure, CIP_RINSE_SEQUENCE.

(Icons far left refer to state of parent compound operation)

Operations, their parameters and journal parameters are specified in the master recipe editor using alias tags. The translation of these to actual tags in PCS is done in interface configuration files.

The modular building blocks, the master recipes, compound operations and basic operations are all subject to extensive version control. An operation version is either in ‘draft’, ‘approved’ or ‘withdrawn’, while master recipe versions are running through a life-cycle as shown:



Note, that it takes two electronic signatures to approve a master recipe, one from production support and one from quality assurance. A master recipe cannot be set for approval until all operations within have been approved. Each step is accompanied by a revision note, normally referring the change request that initiated the change. From the master recipe handling tool, it is possible to print these revision notes and a change log listing all differences between any two versions of a master recipe. When approved, an item is locked for further update.

To modify a master recipe, you have to create a new version. If only recipe parameters are to be changed, this will suffice because of the overwriting principles. If any control aspects ruled by relations etc. at operation level have to change, new versions of compound operations must be created and inserted to substitute the approved ones.

For configuration control, you may generate so-called baseline reports telling which master recipes were approved at any point in time or during any time interval.

Equipment modeling

The equipment modeling system represents the equipment hierarchy using both types and instances. Both have status and parameters associated. In the IBP project, however, equipment parameters were kept in the PCS system. Again equipment is a recursive data entity, allowing equipment within equipment as for operations.

Equipment types can be modified by characteristics. The types are used to model equipment at master recipe level. A master recipe may be specified to run in an area. When generating a control recipe, the merge package will translate equipment types to actual instances of equipment belonging to the actual area chosen for the control recipe to execute in.

At the time of the IBP project, minor modifications in the equipment modeling system introduced connections, the model equivalent to pipes, which could be used in arbitration of equipment.

Material definition and control

This subsystem handles definition of article numbers, bills of materials, reservation, picking, goods reception and dispatch. Certain materials will upon reception have their concentration and similar critical parameters recorded together with standard information on expiry dates, quality and amount.

Also, the subsystem has a complete storage control keeping track of the materials at locations in load carriers, again in a recursive manner allowing for bottles in boxes or sacks on pallets. During setup, you may specify articles to locations, articles to load carriers and carriers to locations, thus modeling for example cold storage for some articles.

Materials are received, stored, picked, consumed and produced, stored, released and dispatched using bar code scanners. Materials must have adequate quality status to be released for production.

For production in the bulk environment, no reservation or picking will be performed during weighing and charging in the raw material and solvents dispensing area. These functionalities are automatically performed by the specific add-on operation in order to comply with the general principles of the material control subsystem.

Also, in this bulk environment most materials flow in pipes. The inlet of batches from one tank to another is controlled by the PCS system, which in turn will journalize the batch number and amount as a journal entry on the active unit operation in the MES system. These journals are then registered in a common journal type in the batch tracking system of the MES system allowing for unified tracking of automated and manual consumption of materials. The batch tracking can display both downstream and upstream relations between batches.

Batch execution

Basically the batch engine consists of a state machine. It controls and coordinates the execution of all control recipes, which are generated on the basis of information regarding master recipe, equipment, bill of materials and batch size.

The execution is responsible for comprehensive and reliable recording of process and quality data, allowing for analysis and optimization of the manufacturing processes.

The HMI supports monitoring and operator intervention. In general all recipe and operation status, parameters, journals and equipment status information is available to the operator, generally as lists typical for database applications.

For large recipes, spanning prolonged execution, the immediate allocation of equipment performed at control recipe generation is not optimal, since equipment may not be available at time of execution. Recipes therefore contain operations to bind equipment dynamically during execution based on equipment connections and start conditions regarding availability and cleaning. If no suitable equipment is available, the operation is queued. The queuing mechanism is typically used for queuing alternate execution of production and cleaning batches.

The batch execution subsystem is responsible for invoking any manual operation hooked into the state machine like filter handling, sampling and weighing/dispensing.

The exception handling includes facilities for disabling relations between operations at individual levels and for setting operations in a special error status from which in can freely be set to any appropriate status. A suspend mechanism will allow operation changes from the PCS system despite problems with execution of subsequent operations within the entire transaction.

Report definition and generation

Experiences from earlier projects had shown that batch report definition often fell behind, and reports were commonly defined using standard spreadsheet or database tools. Now, the idea was to combine the recipe definition with report definition to have both recipes and reports defined at approximately the same time during project execution, and most important in a tool suitable for end-user operation. This led to the introduction of a report definition and generation subsystem.

The report structure is built by referencing operations from the master recipe and defining how each operation must be reported relative to each other. For each operation a report component is defined. The contents builds on parameters and journals as defined in the recipe editor. A report component is most easily described as the method to report a compound operation. As compound operations builds treelike recipes, so will components build treelike report definitions, which become chapters, sections and lines during reporting. The same way as a single top level compound operation defines all 300+ CIP recipes, so can its report definition specify all associated batch reports.

Batch reports are subject to the same version control and life-cycle as the master recipes. The system checks for consistency and warns against approving reports associated with unapproved master recipes.

The reporting system has an interface to the historian for curve data, which is extracted at time intervals defined by the started and completed timestamps.

Operator Access Control

User access control was established through a plant-wide shared system to enforce only authorized access to critical operations. It supports electronic signatures as required by FDA CFR 21 Part 11. The system is shared by MES and PCS, so users have their access rights defined in one place only.

Integration

At the interface towards the process control system, OPC was introduced as the communication standard. The MES and PCS systems are connected through an interface program. Actually there are some 20+ instances of the program serving each a cell in the plant.

At start-up, the interface program reads a configuration file, which defines the units, their operations with parameters and journals, each mapped from MES alias tags to PCS OPC names. The interface creates and enrolls OPC groups within a PCS OPC server.

On the MES side, the interface program will regularly poll a database view providing information on the actual operations to start in the PCS system. When matching unit and operation alias tag in its configuration, the interface will activate the corresponding unit operation in the PCS system.

MES covers the non-time-critical aspects in the database. The PCS and historian systems are responsible for the real-time aspects such as continuous data collection, event logging, equipment monitoring, alarm handling, trend curving and phase sequence control.

The protocol for executing an operation is simple. The interface selects the operation from the list of operations to start, changes its status to starting, invokes the PCS system, which in turn reads all the production parameters through the interface, and updates the operation status to started. While running, PCS will journalize a number of entries before finally setting the operation status completed.

Since the recipes mix manual and automatic operations, special operations are inserted at the beginning and end of a unit procedure for allocation and release of the unit. The start operation is kept active and

may serve as repository for journals regarding critical alarms and events during execution of the unit procedure.

The cleaning management is a good example of how to make the two systems cooperate, using their inherited characteristics of database and real-time systems. Both systems have registered a unit cleaning status. It is the responsibility of the PCS system to run timers on clean states in order to degrade these when holding times expire. The MES system must record the actual cleaning status to prevent production on un-cleaned equipment. Consequently PCS informs MES through the interface on changes in equipment cleaning status, even if no batch is running. Note, the actual status is always returned from PCS to MES.

On the other hand PCS is in principle not aware of the resulting cleaning status after a CIP operation on the unit, since it is controlled by operations and parameters running on the CIP substation. Therefore the special start and stop operations have hygienic parameters added to their list of parameters, indicating to PCS which status to set after start or end. For example in a production procedure, the hygienic parameter at start sets the 'production' cleaning status, and at the end it sets the 'used' cleaning status. In both cases, PCS updates the current cleaning equipment status accordingly. Also, at start, a check is performed to ensure the synchronization of the two systems. If not synchronized, PCS will abort the procedure start.

For CIP procedures, the cleaning results are calculated in MES, which looks up the resulting cleaning status in a table depending on the cleaning status at start, the action and its result. The resulting cleaning status is transferred to PCS as a parameter value during stop.

This completes the example of how transfer to MES of responsibility, traditionally placed in PCS reduces the complexity of the PCS code.

Highlights and benefits

The IBP project produced some 600 master recipes and reports covering more than 17,000 operations with a total of about 600,000 associated parameter values. During the passed 2½ years many recipes have migrated to higher versions due mainly to parameter changes. The largest production master recipes now hold version number in the mid-twenties.

A production control recipe in the purification area typically spans some 30 units, while CIP and solvent mixing only spans 3 units. The MES database contains data from approximately 125,000 executed control recipes of which some 6,000 are production recipes. This corresponds to 1.83 million automatic unit operations. The journal data table contains 275 million entries of which some 20 millions are entries directly linked to production data, while the rest are related to tracking of execution. The current size of the database has exceeded 250 GB and it grows at a rate of approximately 80 GB/year.

So what are the benefits? First of all, - it works, - flexible, modular and coordinating the execution and collecting the data in this huge factory, providing insight to the process and generating consistent documentation thereof. The separation of control responsibilities between PCS and MES at the operation level not only made parallel development possible during the project, but it also resulted in a robust architecture which has avoided most of the could-have-been problems on coordination, timing, performance etc., which could have been a major concern handing over a large system for operation. The reason seems to be that the splitting was done at a level that combines the real-time action of PCS and the database transaction scheme of MES in a synergy that exceeds the performance of the individual components.

The MES system includes batch tracking capabilities both upstream and downstream, which helps during batch release. The reporting system has later been extended with a deviation reporting facility, which is able to extract records of journals deviating from predefined acceptance intervals.

The system is robust and users have gained so much confidence with it, that they consider abandoning all CIP reporting, keeping only deviation reporting and otherwise relying fully on the MES system for handling the cleaning of units. Ongoing projects also aim at disposing of log books and replacing them with manual operations during batch execution.

The system has been easily expandable over the time. Additional equipment has been integrated into the system without problems. The system modularity also proved its flexibility in a recent upgrade project regarding one of the purification lines. As the piping changed, so did the recipe procedures, while the compound operations reflecting process modules and units remained unchanged to a large extent. Recipes were easily cut at relations between compound operations, rearranged and reassembled into new production recipes and overwritten with new production values.

An interface to corporate ERP or material control systems, however, has not been developed. Neither has an electronic documentation management system. Instead, a new generation of corporate projects on lean manufacturing is coming up. The database provides data for a number of optimizing projects. Data is simple to extract either for reporting or for further aggregation at higher level IT systems. For all these extensions, the database is ready and available.

History

It should be obvious, that you cannot create a system like this by just looking at requirements. You must have the necessary experience to build on. IBP was the peak of a long haul within the MES domain. Looking back at the evolution of the MES integration toolbox, some decisive milestones can be pointed out.

The first experiences came from an intensive automation project which was aiming at establishing the first fully automated, “paperless” factory. The factory was designed to manufacture insulin disposable devices. The production comprised the full range of finished pharmaceutical processes from formulation through filling, inspection and assembly to packaging.

Originally the database was introduced to fill the gap between a central ERP (SAP) system and distributed SCADA systems. The database contained just the batch engine. It was introduced to assembly and packaging cells in 1998, but it was soon realized that the combination SAP/SCADA was insufficient to model the more complex recipes used in the formulation cell. The logical formulation model at that time included no less than six layers of operations within operations, which led to the construction of the graphical recipe editor.

The “paperless” factory, of course, also included an interface between the original MES system and a document management system for both batch reports and deviation handling. The aseptic area started production in 2000.

The solution included an interface to an independent third party material control system. The experiences resulted in the design of a material control subsystem within the database itself for the next generation of projects over the years 2001-2003. Although the automation projects were not quite as ambitious as the original, the MES toolbox proved to be a good scaleable solution for semiautomatic manufacturing, providing easy to operate interfaces for the operator.

Impact on project execution

The fast and safe execution of this large project led to more contracts. Currently MES solutions are introduced in a number of purification plants, for which the project execution time is continuously decreasing, reducing the time to market. The first successor was completed in just 14 months from contract to handover.

The main reason for this, is the modular engineering concept, to which system re-use of compound operations and report components is a contributing factor. The combined modular development of recipes and reports has proved to be a very productive way of engineering an application.

The master recipe tool has demonstrated its abilities as an intuitive specification and design tool as well as a powerful engineering tool which in combination with the execution engine has the flexibility to meet and execute complex recipe structures. It provides a good starting point for maintenance of the solution after handover to the customer.

Project execution in parallel modules allows for fast track engineering. Fast track means shorter time to market and therefore faster return on investment. Fast track makes it possible to initiate capacity expansion projects later and thereby reducing the risk. A library of validated compound operations to control any modular building block is a reassuring starting point.

Validation principle is bottom-up. Compound operations are validated off-site, progressing upwards towards the recipes. Recipes are commonly validated on site. For families of recipes, using the same top level compound operation, a so-called golden master recipe is selected for test, while the other members of that family are validated automatically against the golden one.

For a pair of identical twin factories, it even succeeded to install a copy of the entire MES database from one to the other, emptying it for production data and otherwise modifying just one single installation parameter: the PCS prefix identifying the plant coordinates. The entire MES system including configuration was consequently validated by an installation qualification only.

With the configurable simulator at hand, even batch reports can be extensively tested off site, using the system abilities to simulate PCS feedback and journalization of process data.

The future

The integration toolbox fits into a void between PCS and ERP systems as they used to be. These days, vendors on both sides squeeze the gap by extending functionalities normally belonging to the MES domain, and vendors of standard MES packages provide more and more features to bridge the gap.

The system integration toolbox may however still find a niche in a world of upgrades and stepwise implementation of MES functionality due to its unique flexibility, especially regarding recursive operation levels.

However, from an engineering consultant’s point of view, solutions based on commercial MES systems are preferable. Our experience within automation integration calls for broad flexibility in the MES domain, which we hope to be able to find in the coming standard systems.