

**Presented at the
World Batch Forum
North American Conference
Baltimore, MD
April 1998**



**107 S. Southgate Drive
Chandler, Arizona 85226-3222
480-893-8803
Fax 480-893-7775
E-mail: info@wbf.org
www.wbf.org**

A Methodology for Batch Control Implementation - Real World Lessons -

David A. Christie
Yokogawa Industrial Automation America
125 Huddleston Road
Peachtree City, GA 30269
(770) 486-9893

KEY WORDS

Implementation, Batch Control, ISA S88.01

ABSTRACT

ISA S88.01 provides great benefit to users and vendors by solidifying batch operating models and standardizing batch control terminology. On the other hand, as befits an industry-wide standard, ISA S88.01 leaves great flexibility for individual users to apply their own project practices and implementation methodology. Even when utilizing ISA S88.01 models and terminology, many pitfalls remain for those who implement batch controls and recipe management without thinking ahead. Real-world project experience has demonstrated a number of design and implementation “lessons” which might be overlooked to the eventual grief of the batch engineer. A successful implementation methodology should consider these “lessons” in addition to the formal models and terminology of the ISA S88.01 standard:

- Understand the process before generating the design
- Don’t implement until you have designed
- Get the end user group involved in the design
- Write it down
- Agree on the recipe
- I/O assignments are important
- Don’t underestimate the State Transition Matrix
- Pay particular attention to exception handling—the hard part of batch
- Batch reports are not an add-on

The purpose of this paper is to discuss these implementation concepts that have been learned from real-world experience with batch projects.

INTRODUCTION

I love to hike. By living in Georgia, I have plenty of fascinating outdoor areas for this hobby. Every Sunday, I read the Atlanta newspaper's "Outdoor" section, and it really whets my appetite with neat descriptions and photos. They make it look so inviting. I can just see myself sneaking along a trail through the Okefenokee with a sharp eye out for a bald eagle, or hiking along the Appalachian trail in northern Georgia, surrounded by gorgeous spring dogwood blossoms and listening to the rolling gobble of a wild turkey. In my mind's eye, it is perfect and beautiful, and I want to put down the newspaper and go.

But, it seems like my own trips don't always go so smoothly. The Okefenokee has oceans of water, mud, vines, alligators, and bugs, and bugs, and bugs... Without a good map and a compass, a person can have a pretty miserable time wandering around in the Okefenokee. In fact, someone can wander around for the short remainder of their life. So much for the gloomy swamp, maybe the mountains are the place to be.

The last time I went to the Appalachians, it rained. Then, on top of the mountain, the rain started to turn into sleet. I decided to take a shortcut to get back to the warm, dry jeep, and I eventually ended up down in a steep, slippery canyon full of tangled rhododendrons. After I slid down the hillside a third time, I was muttering pretty strongly, and I certainly did not look like the guy pictured in the Atlanta newspaper. As in the Okefenokee, I needed a map and compass and a well-thought-out travel plan. Even then, the trip might not have been a dream, but with a little forethought and planning it would have been a lot less miserable.

The point is that we often have a vision of where we want to go, and how perfect and beautiful it will be when we get there, but we often bog down between the vision and the reality. Success requires two elements: a goal and a plan for reaching that goal. Without a goal, we do not know where we are going, why we are going there, or when we have arrived. Without a plan, we may never arrive and might even wish we had never started out in the first place.

This concept applies to batch control as well as to hiking.

As a practicing batch control engineer, I use and appreciate the importance of ISA S88.01 on an almost daily basis. First, ISA S88.01 clarifies concepts such as Unit Supervision, Process Management, Recipe Management, Production Information Management, and Production Planning and Scheduling. These are

each levels of the Control Activity Model. In addition, ISA S88.01 provides standardized batch control terminology which gives meaning to specialized batch control terms such as control recipes, units, operations, phases, etc. The control models and terminology facilitate communication and understanding between vendor and customer, between vendor and vendor, and, most importantly, within the vendor/customer project team charged with implementing any specific batch project.

In hiking terms, ISA S88.01 is the equivalent of the Atlanta newspaper's "Outdoor" section. It provides the goal. It allows the batch engineer to visualize what a completed batch control system will look like, what features it will have, and how everything will be perfect and beautiful in the end.

However, ISA S88.01 does not address project practices and methodology. It is not a map and compass. The ISA S88.01 models and terminology discuss where we are going with a project and how it will look when we are done, but (rightfully so) ISA S88.01 does not dictate how we should get there. We are on our own to have a smooth journey or to wallow in the Okefenokee through our misguided efforts. As a project journeys from concept to the plant floor, there are many pitfalls for those who do not plan ahead. On the other hand, there are many rewards for the engineering team that combines knowledge of ISA S88.01 with a proven design and implementation methodology.

Within my own company, the implementation concepts that have been learned from real world experience have been formalized in our ISO9000 procedures, peer review process, and Batch Engineering manual. Our ISO procedures dictate what activities should occur, in what order, and with what quality barriers. These project activities are generally broken down into the areas of Functional Design, Detailed Design, and Implementation. Each general area includes its own peer review and internal testing requirements. These internal practices are flexible to accommodate a wide scope of projects (such as validated or non-validated) but are specific enough to be our "map and compass" to keep us on the happy trail toward a successful project. I would like to share some of this experience.

UNDERSTANDING THE PROCESS

My first and foremost rule is, "Don't do anything until you thoroughly understand the process." This is easy to say, and even easier to overlook in the rush to get a project off the ground. Difficulties lie ahead for the project team that does not stop to consider the process. The uses and interconnections of process vessels seem obvious at first. Then, upon further analysis, there always seem to be those non-standard recipes which require unusual operations and make the life of a batch engineer so interesting.

Sometimes, one vessel may have multiple roles. For example, the catalyst feed tank might occasionally be used to mix reactor cleaning solution. In other cases, a vessel may have a single role, but its individual devices may perform different functions. It may seem obvious that a certain valve is used to

transfer material out of Vessel A into Vessel B. But that same valve may also be used for transferring another material from Vessel C to Vessel D, or for transferring chemical cleaning solution into vessel A. All of these roles must be understood before deciding which unit the valve should be assigned to, or whether the valve should be designated as a shared resource.

Make a list of the activities (which may eventually equate to ISA S88.01 unit procedures or operations) performed by each equipment module, and flow chart any activities which are unusual or confusing. Then, using the P&ID's, construct a block diagram of the overall process and mark each individual transfer path in a different color. Using these visual aids, consider the multiple roles of each individual control device, such as valves, pumps and meters. Analyze the devices' non-standard roles (such as during vessel cleaning) and keep these exceptions in mind during the design phase of the project. Consider safety, hazardous materials, and potential environmental impacts. If the process line being automated already exists, the operators are a great source of information about situations which require exception handling. Ask questions and listen to their answers. Finally, look to the future and consider any upcoming changes in the process equipment, recipes, or methods of operation. Only when this basic process knowledge is in place is the project team ready to move on to the design phase.

BATCH CONTROL DESIGN

Rule number two requires good self discipline (or a strong ISO auditor): "Don't implement until you have designed." How tempting it is to think, "I know where I'm going. I'll get ahead on the project," and then rush to start punching keys. Then, how confusing the design changes and software patches will be before the project gets finished. Successful batch control is very structured, consisting of ordered and interrelated pieces such as formula data, unit procedures, operations and phases. These are difficult to implement and maintain if the underlying structure does not fit the process. So, after understanding the process, it is important to have a complete and well thought out design which fits the plant and its objectives.

ISA S88.01 gives us the building blocks such as units, unit procedures, operations and phases, but these building blocks can be combined in many, many ways. Some of these combinations will be disastrous for any specific project. Batch control can be viewed as the pinnacle of the lower control levels, such as process I/O, regulatory control, and discrete control. The batch control operates successfully if these lower levels are consistent with the ultimate batch structure. This is why it is dangerous (meaning "expensive") to say, "We will just get the regulatory control in place for now and consider the batch control after the plant is up and running."

As the design is developed, it should be written down. It is amazing how gaps in the thought process are revealed by putting ideas on paper and letting others review them. A written design document helps separate facts from assumptions, lets others see the overall automation picture, and secures end-user "buy-in" by requiring formal signoff. For maximum effectiveness and comprehension, the design standards should use documentation conventions which the end-user is most comfortable with. This might be flow charts, structured text, Sequential Function Charts, or prototype program code.

One secret is to get the end user's Operations department involved early in the design. If the plant union situation permits, it is suggested that the plant designate an "Operator Advocate" to represent the operators in design meetings. Let the operators design, or at least mark-up, the preliminary graphic displays. Provide them with advance knowledge about what is coming and give them ample hands-on training before production. All of these activities foster buy-in and cooperation which will eventually make start-up a more pleasant experience and the control system a greater long-term success.

When creating the batch design, one commonly thinks of the recipe procedure, but that is not the only thing to consider. A whole set of factors should be considered, any of which can cause difficulty later if not properly integrated with the overall design. These include the following:

- The overall requirements of the plant's automation master plan
- Types of products manufactured
- Process lines and paths
- Equipment and device functions and dependencies
- Pre-conditions for starting each recipe
- Operation mode and status changes (state transition)
- Operator and Engineer interface locations and conventions
- Operator console segregation between lines or equipment areas
- Operator messaging and manual actions
- Exception handling
- Alarm management
- Emergency Stop procedures
- Batch reporting and data archiving
- Interfacing to higher level computer systems and to lower level subsystems
- Consistency with legacy systems
- Required system response times
- System security and reliability

- Recovery from power failure and other disasters
- Environmental and safety codes, standards, and regulations
- Ability to implement future control modifications without impacting plant production

A good, solid batch design documents all of the aforementioned factors and subjects them to peer review by other engineers who are knowledgeable in the art and science of batch automation. Thus, a document is available which captures the thought process for customer approval and which freezes the design prior to the start of implementation. Failure to generate a firm, frozen design usually results in a project which is in constant flux, and a system which may not operate the plant properly after startup. This situation is not desirable to either the end user or the control vendor.

RECIPE PROCEDURES

Regarding the recipe procedure itself, it is amazing how many different people will say, "Don't worry, we know exactly how to make that product. We've been doing it for years.", then proceed to disagree among themselves exactly how the batch should be run. Automation of a batch process often provides an opportunity to re-think the "best" recipe procedure, and different internal groups may differ in perspective. For example, Operations may discover real-world deficiencies in a chemist's proposed procedure which makes them deviate from that procedure during production. If these situations are not managed effectively, disputes between internal groups can result in unanticipated project delays. In such situations, the vendor can act as a third party consultant regarding ISA S88.01. The vendor can also convey the specific control system's capabilities and limitations and provide broad based experience from previous jobs. However, the end user is ultimately responsible for providing recipe procedures and must reach consensus on what those procedures are. It is important for everyone to keep the project schedule in mind as these differences are worked out.

Often, a plant's existing recipes have been written by different people or evolved over many years, and an automation project is a good opportunity to standardize the recipe procedures as much as possible. If every different recipe is a random collection of activities, then it becomes very difficult to automate the process. When a standard structure is possible, it aids operator understanding of the recipe and often improves product consistency. A consistent recipe structure also increases engineering efficiency by providing less opportunity for implementation error, allowing the possible use of generic program code, and requiring less repetitive testing.

COORDINATION OF APPLICATION AND HARDWARE DESIGNS

In many organizations, the control hardware and the batch application are engineered by different groups. The Project Engineer may lay out cabinet designs, assign networking addresses, specify cabling,

and make process I/O channel and point assignments. Concurrently, the Batch Engineering team may be working on the definition of unit instruments, regulatory control, interlocks, and phase logic. It can be a challenge to get these groups to talk to each other, and this can be a source of problems.

The I/O assignments are relatively easy to coordinate up-front, but difficult to work around later. Some systems handle node-to-node communications better than others, but splitting phase logic from its I/O can result in unnecessary programming complexity and the potential for timing errors. In addition, undesirable delays in communications and control feedback can occur. The batch control logic may even fail unpredictably if there is a communication error between two nodes that each contain half of the logic. Finally, the system's documentation and ease of understanding are improved if it is possible to avoid external link blocks or programs. There may be hardware-specific reasons for assigning a particular point to a certain channel, and there may be equally good application reasons for assigning it elsewhere. Consequently, each design sub-team needs to understand the other's perspective. In many cases, all that is required is a little up-front communications between the engineering disciplines.

STATE TRANSITION MATRIX FUNDAMENTAL TO SYSTEM BEHAVIOR

The State Transition Matrix describes the "rules" for switching between various recipe modes and statuses. For example, it dictates whether the operator can "Abort" directly from a running batch with a single action or must first go to some non-running state such as "Pause." Also, the State Transition Matrix may protect the process by prohibiting a new recipe "Download" while the current recipe is "Running." The Transition Matrix may force the recipe to run a rigid sequence or permit the operator to jump forward or backward in response to certain situations. In some cases, it may make sense for some batch units to have one State Transition Matrix while others in the same train have a different Matrix.

The design team should not underestimate the importance of the State Transition Matrix. It is one of the most fundamental aspects of the batch control system; consequently, it may be one of the hardest things to change after the system is in operation. The State Transition Matrix is one of the first items which should be designed, but is one of the easiest items to overlook in the rush to define units, flow chart unit procedures, and lay out recipe data. Adequate time should be allowed to lay out the State Transition Matrix, and it should be reviewed and understood by everyone, especially the end user Operations group.

It is a worthwhile practice to prepare some visual representation of the Transition Matrix, but this activity is often harder than it seems. Within a single mode (such as "Auto"), it is relatively easy to illustrate the changes between statuses (such as "Running" and "Paused"). However, it is possible for the status transitions to be different within each mode, which necessitates additional dimensions to the documentation. Then, there are the permitted and prohibited changes between modes within each status.

Finally, one must consider the methodology of transitioning between recipe steps, and of proceeding from the “End” of one recipe back to the “Download” and “Start” activities of the next batch. The State Transition Matrix can be considered complete only when all of the above are considered, coordinated, and documented.

Special care should be exercised to explain the State Transition Matrix to the “Operator Advocate.” Plant operating personnel are usually inherently aware of how they want the system to behave, but they typically do not associate with the term “State Transition” and are not conversant with the bubble diagrams or matrices often used to communicate this concept. The batch control engineer is advised to go out of the way to ensure proper communication, understanding, and agreement on the proposed State Transition Matrix. The transition concepts may be difficult to modify if they are not discussed until system start-up.

EXCEPTION HANDLING IS CRITICAL ELEMENT

Exception handling is easily overlooked but is a critical element for achieving long-term control system success. Based upon my company’s real-world project experience, exception handling constitutes 40-60 percent of the batch control design and implementation effort. It is easy to get caught up in the excitement of automating a recipe, but it is more tedious to go back and figure out how to handle all of the unpleasant complications that might occur. However, the handling of these complications is a key element in process safety, operator satisfaction, consistent product quality, and minimization of batch cycle times.

Some of the situations that should be considered include:

- A raw material that is charged to a unit and then found to be out of specification
- A batch unit that is not "ready" for charging when the preceding vessel expects to transfer material
- A valve that fails to respond to an “Open” or “Close” command
- A chemical reaction that does not begin as expected after ingredients have been charged
- A pump that shuts off unexpectedly during mid-transfer
- A recipe that requires a shared resource which is already in use

- A transfer that is delayed by an equipment problem, necessitating a mid-transfer recalculation of ingredient flow rates
- A need to manually skip certain steps in a batch procedure, or to re-execute certain steps due to off-specification product
- A batch that must be “Aborted” with partially processed material in the vessels
- An Emergency Stop that must be performed due to potentially hazardous conditions
- A final product that is off specification, requiring a transfer to a re-run line instead of the normal storage tank

The above list is not all inclusive, and all of the above situations do not apply to every plant. It is excellent practice to seek the operators’ input regarding exceptions. Ask a lot of “What if” questions, listen carefully to their answers, and try to meet with more than one operating team. It is amazing what happens at night that is not on the official batch ticket.

As each applicable "exception" is identified, it is necessary to determine how the batch logic will respond. Often, changes in the control design should be made to provide more efficient handling of abnormal situations. These changes are much easier to make on paper than by re-coding the system during startup. For this reason, exception handling is a major area of grief for those who cannot wait to start implementation instead of thinking through an up-front design on paper first.

BATCH REPORTING

The batch report is mentioned last to emphasize that it should not be considered last in the batch design. A successful batch report is not an add-on and requires different considerations than the process report for a continuous plant.

In a batch plant, a batch of product usually travels from vessel to vessel along a process train. The first set of vessels often receive material for a subsequent batch while the original batch is in the train's final set of vessels. As a result, batch data must be collected from each piece of process equipment while the batch's material is in that equipment. The data from various vessels are not gathered concurrently, but the batch report is typically printed at a single moment. This requires that each unit's data be saved until the batch’s material leaves the final vessel. In fact, the system often needs to store data from multiple batches which have passed through a given unit. Thus, it is easy to inadvertently mix or overwrite one batch's data with that of a subsequent batch--especially if multiple reports for multiple batches are to be

archived. All of this data storage must be considered and managed. Some systems do this automatically, while others may require extensive manipulations by the batch engineer.

In addition, the batch report may involve computations such as averaging, summing, or calculation of the deviation of actual ingredients from their control recipe amounts. Some of these calculations, such as summing, may be handled at the end of the batch within the reporting package itself. However, more difficult calculations, or the capture of time-critical data, may have to be performed on-line by the batch control logic, and the calculated data may need to be stored for later use by the report package. The capture and storage of batch data varies from trivial to difficult depending on the system, the amount of data, and its format (for example, character string data).

Finally, the batch report may require the collection, storage, and appending of batch event data, batch journal information, and batch trend data. These types of batch related data cannot be easily extracted from a continuous control system's time-series trending functions.

For all of these reasons, the structure of the batch reports should be considered early during the design. These reports should receive the same deliberation that is given to the handling of recipe formula data. Thus, there will not be any unpleasant surprises and hurried re-engineering when the report is formatted near the end of implementation.

CONCLUSION

This paper has discussed a batch control design methodology which is equivalent to a map and compass which are used to find one's way during a hiking expedition. So far, we have not laid a finger on the keyboard to begin control implementation, and that is exactly my point.

I leave you with three overall rules of thumb:

1. Do not design the controls until you fully understand the process
2. Do not implement the system until you have created a written design
3. Resist attempts by others to ignore Rules #1 and #2