

**Presented at the
World Batch Forum
North American Conference
Orlando, Florida
April 2001**



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

Automatic Adaptation of Batch Logic to Changing Levels of Field Automation

David A. Christie
Consulting Batch Applications Engineer
Yokogawa Corporation of America
2 Dart Road
Newnan, GA 30265
U.S.A.
Telephone(770) 254-0400, ext. 2154
Fax (770) 254-4914
dave.christie@yca.com

KEY WORDS

Implementation, Batch Control, ISA S88.01, Unit Supervision, Field Devices, Manual Operation

ABSTRACT

As batch processes are automated, it is common to leave certain less essential field devices without automatic actuators. Thus, the initial control design must accommodate both automated and manual activities. Later, the manual field devices may be automated, either one-at-a-time or in related groups as equipment modules. These field changes, often occurring over a period of years, each require rework of the batch control logic, which can easily exceed the cost of the actuator. In response to this undesirable situation, a technique has been developed which permits the batch design to automatically modify itself, or evolve, to accommodate changes in field automation. This approach yields significant benefits:

- Initial design treats all field devices as if they are automated,
- Recipe includes logic for both automatic and manual devices,
- Batch Journal logs manual device actions,
- Field devices may be automated over any time period,
- Virtually zero modification time for batch logic,
- No redesign, patches, or work-around,
- No dead code or wasted engineering.

This paper describes the problem, illustrates a practical solution, and explores the resulting benefits.

AN IDEAL BATCH PROJECT

In an ideal automation project, time is unlimited, and project budgets are generous. The plant implementing such a project has the luxury of automating all of their field actuators, great and small. These might run the gamut from major charging valves, to agitators, to small air operated mixers, to transfer valves, to wash valves, and more. The appropriate actuating solenoids and answerback switches are installed regardless of whether the device is used once an hour or once a quarter. There are no half steps in this ideal world. If the reactor is automated, so are all its supporting ingredient headers and premix tanks. If Train A is automated, so are Trains B and C.

When the plant can ignore the up-front costs, there are considerable benefits to such an all-out approach. If the system is engineered carefully, it should lead to improved process repeatability. With everything under control of the system, the order of operation should not vary from batch-to-batch. Nor should the plant fall prey to unexpected variations in recipe formula values such as ingredient quantities and operating setpoints. Unlike humans, the system does not get distracted or need a break and hence can perform in a repeatable manner. In terms of plant profitability, all of this consistency and timeliness should result in more repeatable batch quality and cycle times.

In addition, the fully automated system should generate an operation record, or Batch Journal, which steadfastly time tags and logs all of the important batch activities. This Journal, easily accessible via a variety of plant and corporate users, reveals what happened, when it happened, and how long each activity required. Besides the obvious operational uses, such as tracking production schedules and cycle times, the Batch Journal is also essential for strategic engineering activities such as troubleshooting bad batches and optimizing master recipes.

Finally, since the system covers the entire scope of plant operating equipment, the engineering team can carefully design the batch operations to accommodate any reasonable usage of that equipment. Therefore, when this highly competitive, ideal company needs to produce a new product, it becomes an almost trivial matter to string together the existing operations as building blocks for the new master recipe. With such repeatability, operating information, and flexibility, the company is sure to blow their competition away. That is, unless they hadn't spent so much money on their automation system.

REAL-WORLD COMPLICATIONS

In reality, the ideal automation project rarely occurs. Instead of complete field automation, the partially automated plant is a more common reality. The following are a few typical scenarios:

- A. A small, single-reactor chemical plant has been operating manually and has been trying for years to get an automation system approved by corporate. When approval finally comes, money is tight. But they are delighted just to get the reactor onto recipe control, never mind the additive tanks and lesser mix vessels. The plan is to actuate the key reactor valves and agitator up-front. Perhaps actuator kits can be added later to additional purge, drain, and wash valves one-at-a-time under a future maintenance budget. Or maybe in a few years, they will lobby for a follow-up project to retrofit all of the ingredient header system. At any rate, they are glad just to get the initial reactor automation system, and nobody is complaining about the longer picture.

B. Perhaps another company has a bit more money and decides to automate an entire production train. They will “do it right” on this portion of the plant, which may manufacture their most critical or profitable products. The other two trains in the plant will remain mostly manual for now. This will be a demonstration project, and if all goes well, automatic actuators will be added to the other trains during future project phases.

These types of real-world scenarios are the norm, not the exception. And the resulting batch control systems, which operate partly automatically and partly manually, are often technical and monetary successes. However, the partial approach, which is dictated by economic reality, also causes these plants to give up a number of advantages.

First, their batch procedures are not centralized in a single, computerized recipe management system. Some of the actions, which may be critical to batch quality, are still performed manually. The operators may refer to paper instructions, or, more commonly, use their own judgement about what to do and when to do it. It is very common to find that each operator has a favorite method or a preferred way to perform a procedure, which introduces batch-to-batch variations. Sometimes, the batch procedures may even be performed out of order due to operator convenience, habit, or lack of understanding.

Likewise, there may not be a single, centralized database of formula data values, such as ingredient names, charging amounts, mixing times, washing volumes, and pressure setpoints. Again, the operator commonly refers to paper instructions, while the DCS operates its part of the batch based on electronic formula data. The formula data are stored in two different sources, which is undesirable.

The above situations can easily impact batch repeatability and quality. It is much more desirable for the entire recipe to be nice and tidy in a single, secure database.

Eventually, the plant will produce an off-specification batch. Engineers will scurry around with clipboards, attempting to deduce what went wrong. Since engineers operate from data, they will print out trends, recipes, and operating records for the batch. The Batch Journal, which has been quietly collecting data batch-by-batch, will suddenly receive high visibility. But, where are the records for the manual ingredient valves? The ingredients were charged in the middle of the night. How long did it take? Are we sure that HV1007 was actually opened before the catalyst charge began, or after? There is no way to know; consequently, there may be no way to be sure what caused the off-spec batch.

Now, suppose that despite its limited budget for field devices, the plant anticipates all these problems, and they are clever enough to devise a solution. Although many of the valves remain manual, they include these valves in the recipe procedure. When it comes time for the valve to be manipulated, the system is programmed to issue a command to the operator and wait patiently for a typed-in response. Thus, the entire recipe is centralized, paper instructions are eliminated, and the system captures a complete operating record of the operators’ CRT responses. Since there are dozens of manual devices, the plant is a bit dismayed by the contractor’s cost to configure the logic and text for all those operator messages, but it seems like the best tradeoff. And it works well enough when the system is first installed. They’ve dodged the bullet without incurring the cost of automating every field device!

The rub comes next summer when a few dollars become available to automate the first additional field device. Perhaps it is an occasionally used crossover valve between two ingredient headers. What happens now? First, the plant engineers, who perhaps were not deeply involved in the initial batch

design, must locate and remove the operator message for that valve. This is probably not too hard; just find it and comment it out. Then comes the more difficult aspect of modifying the logic of existing batch operations that depend upon the valve. Now, they are messing in the program code itself, which may be quite complex and finely tuned to start with. The person making the modification may not be aware of the design criteria, standards, or nuances of the original design. But somehow, they get it done, and after a few glitches are worked out, the new automated valve performs as desired.

Three months later, a bit more money comes available to automate the twelve valves on the water header. What now? They go through the same drill all over again. Then, next year, the Phase II budget is approved to automate that second train, which was manual until now. A bigger, more daunting, and more time consuming task has now fallen to the poor plant control engineer, who perhaps wasn't even working for the company back when the original system was designed. Over the course of time, these piecemeal, evolutionary changes will probably deviate from the original, elegant design of the operation logic. Patches are made, code is commented out, and work-arounds are required. In the end, the system loses some of its flexibility and becomes increasingly difficult to modify. To eliminate their own pain, the plant people stop asking for more automation budget, and the system comes to an sub-optimal, steady-state condition. Their feeling is, "Pleeeeeease don't give us money to automate Train 3."

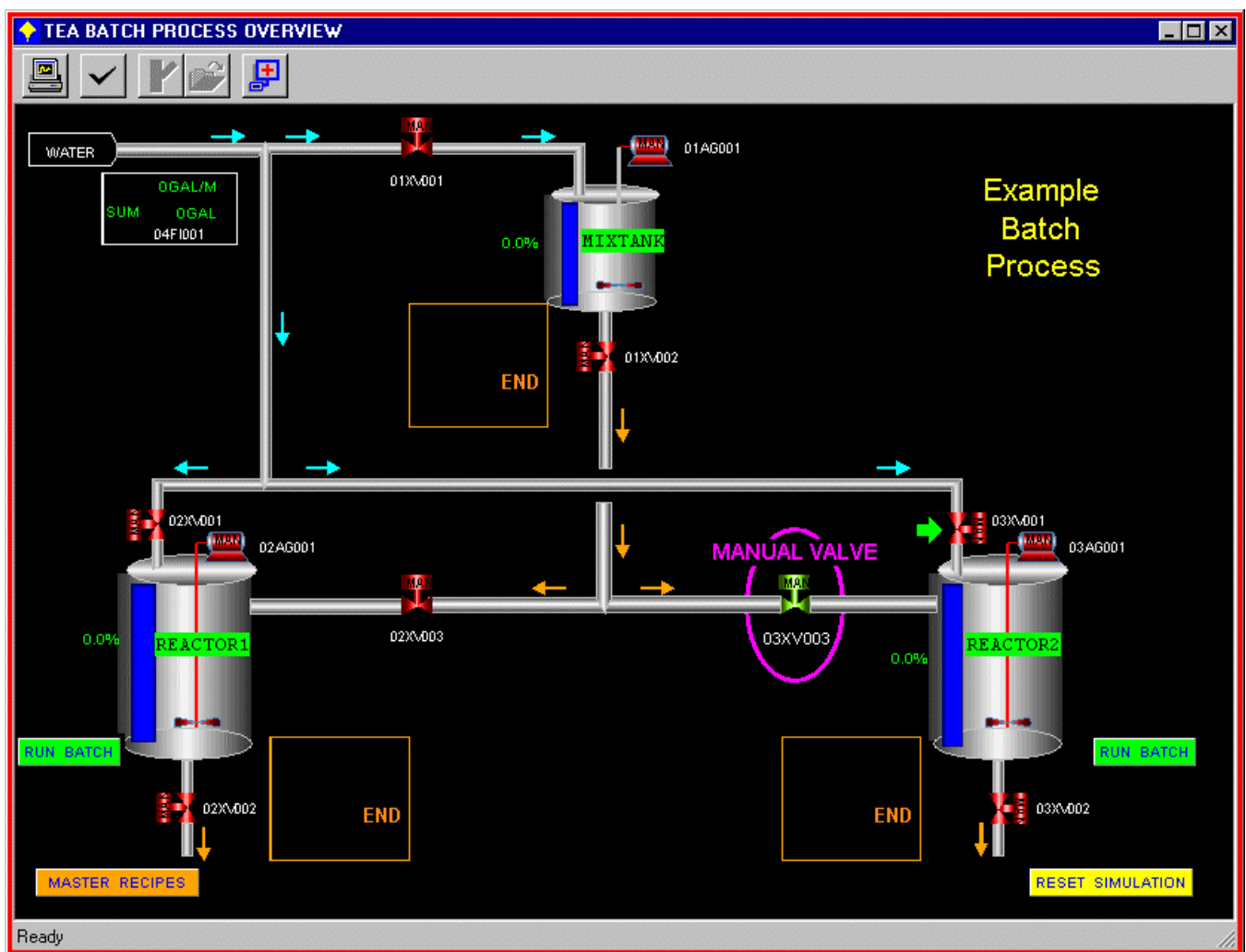
A SOLUTION

So, even with the best, modern, \$88.01 aware batch control system, many real-world plants may eventually reach a plateau of automation. As a batch control engineer, this realization prompted me to try to devise a solution. I realized that the best, in-depth thinking about the control functionality usually occurs by the team performing the original design and implementation. That is when there should be time and money for someone to sit down and think quietly about the present plant operation, and hopefully also about the future recipe requirements.

What if, at that time, the application design team could treat all field actuating devices as if they were automated, regardless of whether they were to be electrically actuated, manual, or even non-existent but planned for future construction? A completely integrated, well-thought-out design could then be developed which would serve both present and future needs. Upon initial startup, what if the system could automatically distinguish the non-automated actuators and issue operator dialogues to manipulate them? Then, what if there was a way for the system's treatment of the manual devices to evolve as field actuators were added in the future? What if all this happened instantly, without having to dig into the batch code at all? I soon came up with an approach that addresses these issues. I would first like to describe its features, and then I will explain its underlying concept and benefits.

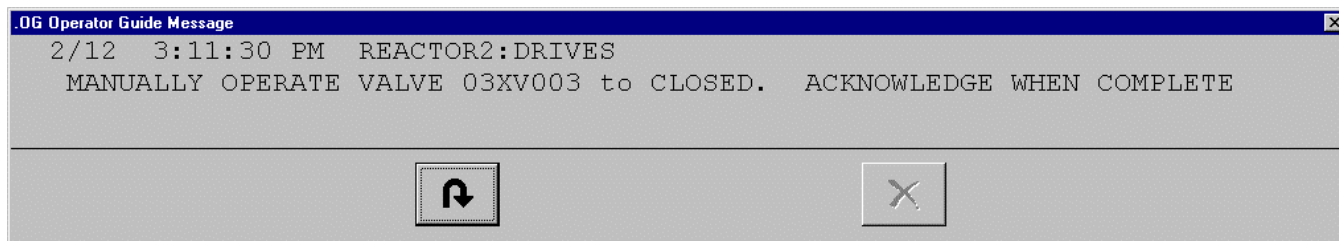
This screen copy is a graphic that represents a very simple batch process with one mix tank and two reactors. Imagine that the operator starts a recipe running on the left path, using the MIXTANK and REACTOR1. This imaginary recipe charges ingredient to both vessels, starts their agitators, transfers material from the mix tank to the reactor, holds for a given time, and then dumps the reactor. The imaginary recipe merely serves to illustrate activities that would be common for most batch plants. In this case, the recipe proceeds automatically, with the system manipulating the automated control modules whenever required.

Next, imagine that the operator starts the same recipe on the rightmost path of this process. On this path, all of the control modules are automated except the REACTOR2 transfer valve (03XV003), which is circled on the graphic merely to highlight it for illustration purposes.



The first thing the recipe does is initialize MIXTANK and REACTOR2 by closing all their valves. But the control system cannot close manual valve 03XV003. So, in this case the system generates an operator dialogue message requesting operator assistance. This message appears on the operator's console and contains the following information:

- The date and time,
- The name of the unit issuing the command, in this case REACTOR2,
- A text instruction to manually close a specific valve, in this case 03XV003.

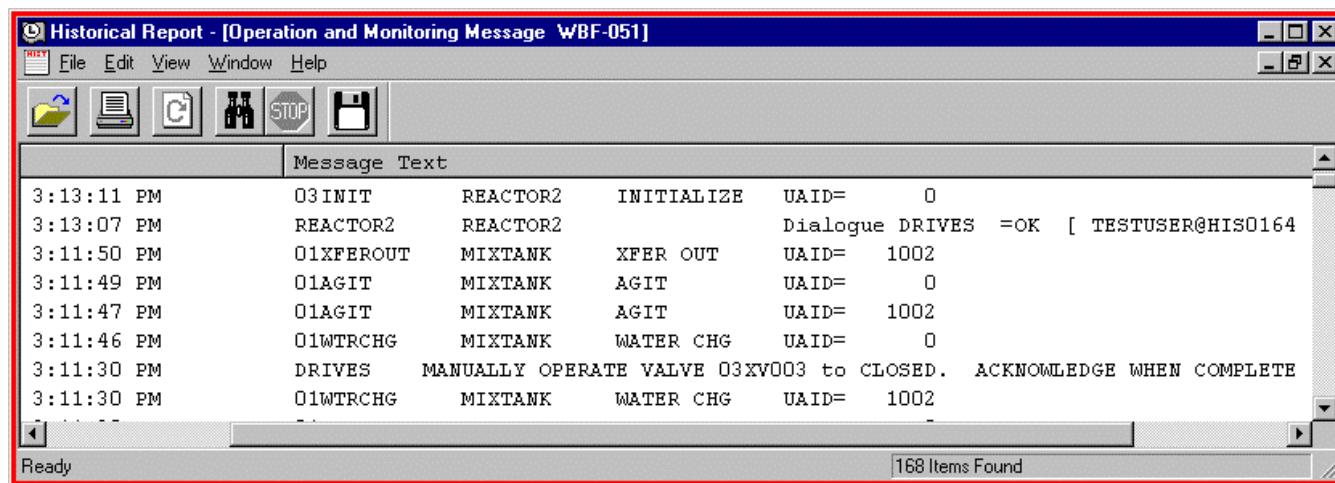


The reactor's recipe procedure waits and will not proceed until the operator confirms the message.

Now, pretend that the operator has walked out to REACTOR2 and verified that this manual valve is indeed closed. Acknowledging the message from any CRT confirms to the system that the condition has been met. The recipe then proceeds to its next step. Before doing so, the logic updates the graphic to reflect that 03XV003 is now closed, indicated by changing the valve's color from green to red.

Later in the batch, when it is time to operate the same valve to transfer from the mix tank, the operator will again receive dialogue messages to OPEN or CLOSE the valve. In this example, this occurs for only one valve, which is presumed to be manual, but the same principle applies to any number of manual devices, be they valves, agitators, etc.

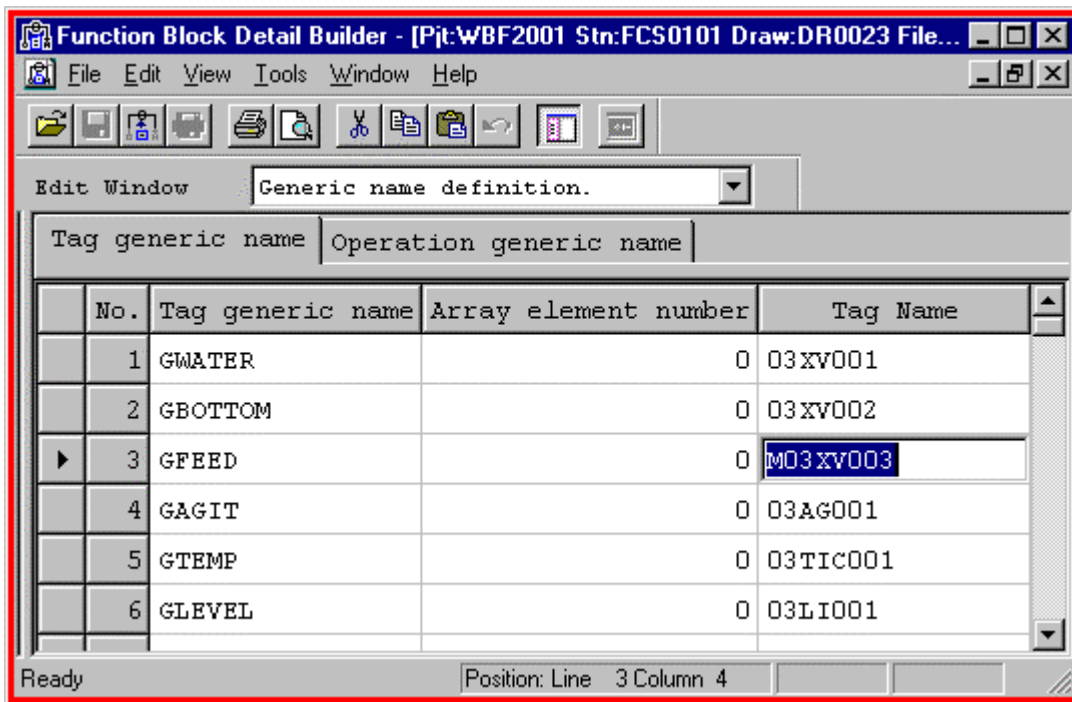
The next view is of the Batch Journal screen, which has been automatically generated by the system as the batch proceeded. The Journal includes two notable entries. The first is the time-stamped message instructing the operator to manually close 03XV003 at 3:11:30 PM. The second is a record of the operator's confirmation at 3:13:07 PM, which is not only time stamped, but also indicates the operator's login name, TESTUSER, and the name of the operator station, HIS0164.



Thus, the control system has overcome some of the problems described earlier. The computerized recipe procedure addresses not only the automatic field devices, but the manual valve as well. And the system has captured a permanent record of the manual device's operation.

Now, move forward in time and imagine that the plant has found enough money to automate 03XV003 and wire it to the I/O subsystem. How difficult is it to update the valve's recipe logic?

The first step is to open the engineering builder for the REACTOR2 unit. This builder contains the reactor's tagname list. Notice that the tagname for the feed valve, 03XV003, is preceded by a special character, "M," signifying Manual. All that is required is to remove the "M", then recompile and download the unit. No other programming changes are needed. It only requires about one minute.



Now, imagine that the operator starts the same recipe again using the REACTOR2 path. As the recipe proceeds, the system now knows that the feed valve is automated. This time, it operates the valve via the I/O subsystem without generating any operator dialogue messages. Thus, with a single, quick, online change, the batch logic has evolved to reflect the new field situation.

EXPLANATION

So, how does all this work? The control system used for this demonstration depicts unit procedures and operations as SFC's (Sequential Function Charts), with the phase logic being constructed in a batch programming language. The concept utilizes two characteristics of this language: 1) the code can be written in terms of generic variables, which are cross-referenced to actual tagnames in the batch unit's tagname table, and 2) the language supports user-written program functions. Thus, the batch phase might close the reactor feed valve by calling a user-written function called OPERATE and passing to it the reactor feed valve's generic name (GFEED) and desired state (CLOSED), as follows:

OPERATE (GFEED, CLOSED)

The user-written function would then perform the following general logic. Specific programming code is not shown because it would vary from platform-to-platform.

1. Fetch the tagname (M03XV003) corresponding to the generic name (GFEED)
2. Parse out the leftmost character of the tagname
3. Test to see if the leftmost character is an "M"
4. If the character is "M"
 - Generate an operator dialogue requesting the operator to manipulate the manual valve
 - Imbed the real tagname (03XV003) and desired state (CLOSED) into the dialogue
 - Wait until the operator confirms that the field device has been operated
 - Update the software status of the device (thus causing the valve to show red on the graphic)
 - Return to main phase logic
5. If the character is not "M"
 - Operate the valve automatically via the I/O subsystem
 - Test for answerback and generate an alarm if valve fails to respond
 - The valve's actual answerback switches cause the graphic color to update to red
 - Return to main phase logic

That's all there is to it. For my prototype, the function only requires 29 lines of code. Since the actual tagname is passed to the user function as an argument, the code only needs to be written once for all field devices. In other words, the same code will work for all the manual valves in the project.

NUMEROUS BENEFITS

This approach has many practical advantages. First, it permits the original design team to create a "clean" design for the entire process, treating every field device as if it is automated, even if it will initially be manual. Thus, a well-thought-out, integrated design can be created that adheres to the designers' concepts and standards, some of which might not be obvious to those making subsequent modifications. The initial designers' subtle intentions regarding exception handling, sharing of resources, etc., are thus not subject to future misinterpretation. This totally integrated design, in turn, supports the creation of a complete, electronic recipe for each product. The combination of computer and paper procedures is eliminated. The operators' lives are less confusing, and the recipe auditor is happier.

With this approach, the system provides guidance for all manual actions, be they opening and closing a valve, starting an air mixer, rinsing a vessel, or whatever. The required activities are dictated to occur in the prescribed order. Nothing is left to operator judgement or whim. At the same time, the Batch Journal captures a record of every manual action, including both the time of the request and the time of response. These Journal messages can be printed in the batch report and/or retrieved by a Plant Information Management System. They provide invaluable clues for engineers tasked with the questions: “Why was this batch off specification?” or “Why are our cycle times so much longer on the weekends?”

Obviously, this approach allows field devices to be automated at any time and in any order. Actuators can be fitted to valves one-at-a-time, or in a larger automation phase, such as an entire header. Significantly, there is no incentive to wait and bundle the activity into a larger project that might face a budget hurdle. Each single device can easily be automated whenever the plant generates a few dollars for the actuator pack.

The most important benefit of this method is that it does not require the plant to re-engineer the batch logic when a device is automated. There is no “throw away” of engineering effort from the original design. Nobody is required to locate and eliminate the original operation messages and their supporting logic because the logic automatically “shifts gears” to accommodate the new situation. As in the REACTOR2 illustration, only a few moments are required for the on-line change. Thus, virtually no plant downtime is necessary. Likewise, there are no engineering labor costs for re-design of the batch logic and for testing the changes. Since the system’s original designers were able to create a totally integrated design, anticipating the future automation of all field devices, there is no need to patch or work around their code by someone who might not understand the subtleties of their original design. And if unanticipated problems do occur, it is very easy to go back to the previous method by simply reinserting the “M” in the unit’s tagname table.

There are also several technical advantages. The user function becomes part of the actual batch logic code, residing in the controller but occupying very little space. In the example case, only 29 lines of code were required, including error handling if the automatic valve failed to operate. The operator message is parsed together from the function’s arguments, consisting of the tagname and the desired state. Thus, a separate message is not required for each individual valve. The same 29 lines of code could serve one valve or one thousand. Thus, the message capacity of the controller is not a factor, nor is it necessary to rely on non-redundant external message storage, such as the hard disk of the operator console. Another technical advantage is that no dead code remains behind after each device is automated.

Finally, all of this is reusable engineering. It is not project specific. Once the concept has been developed for any given platform, it is immediately transportable from project-to-project without modification. Thus, the efficiency of the engineering team is increased, and project costs are reduced.

PRECAUTIONS

Admittedly, there are also potential drawbacks or precautions which should be considered. First, although it is desirable to the plant’s engineering and management groups to log all manual device activities, this can become burdensome to the process operators. It is easier for them to go to the field and open a valve (or perhaps a number of valves) than to receive a dialogue, go manipulate the valve, and return to the CRT to acknowledge each individual action. Depending on how conveniently the

operator stations are located, the operator acknowledgements can take more time, even to the point of impacting total batch cycle time. Also, the operators may grow to resent the system dictating their every move, regardless of whether management perceives benefit from documenting those moves and enforcing consistency. This is especially true if the control system is introduced into an existing facility. These competing factors need to be weighed against one another. Very careful consideration should be given if the number of manual devices begins to creep above one-third of the total control modules.

Another precaution involves the up-front design. This technique makes it very easy and cost effective to add future automatic control modules, with virtually no overhead cost to modify the original logic. However, its effectiveness is highly dependent on the quality of the initial design. If the original team does not exercise proper discipline, poorly modularized equipment or faulty treatment of shared resources can create havoc. The team must carefully avoid the detrimental tendency to focus on the automated devices and to pay less attention to those which are initially manual. Such a mistake will be revealed when the manual devices are eventually automated. The up-front design must be viewed as more than just an immediate solution. It is a framework to ensure time savings and consistency throughout the entire life of the control system. If the original team does not have this vision, or the time and budget to consider the future, then the desired benefits may not be realized.

CONCLUSIONS

In conclusion, this paper has discussed a common, real-world batch automation problem, illustrated an efficient, easy-to-implement solution, and described numerous tangible benefits as well as a few precautions. Because it can significantly reduce future expansion costs, the concept should be advantageous to many processing plants that are trying to automate over time on limited budgets.