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

A Case Study for Batch Integration in A Specialty Chemical Facility

Andy Robinson
Project Engineer
Avid Solutions
102 Towerview Court
Cary, NC 27523
United States
919 468 4334
919 468 4335
arobinson@avidsolutionsinc.com

Shawn Hull
Project Engineer
Avid Solutions
102 Towerview Court
Cary, NC 27523
United States
919 468 4334
919 468 4335
shull@avidsolutionsinc.com

KEY WORDS

S88, Specialty Chemicals, Simulation, object based, batch, integration, centralized control, migration, DCS

ABSTRACT

When multiple plant areas, units and products are all controlled by a single control system in a batch environment, proper control system planning, design, and implementation are crucial to the success of a project. This presentation will examine a recent project completed at a specialty chemicals plant that involved upgrading and consolidating several plant areas, each making a unique product in its own island of automation, onto a single control system. This presentation will explore all stages of the project, emphasizing the use of the S88 standard during system design, the use of object based tools to speed development and ensure consistency during development, the use of medium fidelity simulation tools to ensure software and batch functionality before it reaches the plant floor and aid in operator training, and the use of a single batch manager in a multi-product facility. Additionally, the presentation will highlight successes and lessons learned throughout the project.

PAPER

Overview

Avid Solutions recently completed a large control system integration project at a multi-product specialty chemical facility in North America. The 2400 I/O project consolidated six distinct processes currently controlled by antiquated standalone PLC systems or local panel-boards into a modern integrated control platform. The primary goals of the project included reducing batch cycle times, automation of new processing equipment, and providing a single modern control system platform across the entire plant. Such a platform was expected to reduce control system engineering and maintenance costs and reduce plant operational costs since operators could easily cross-train across the entire facility.

As is typical for many older plants, this plant had not established a solid standard for control system platforms. Thus different platforms had been installed in various islands of automation across the facility. Two of the plant areas within the scope lacked any automation and were controlled solely by hardwired panel-board style controls and single loop controllers. The remaining plant areas were controlled by older PLCs with several different Human Machine Interface (HMI) packages scattered throughout the plant areas. Each of these distinct systems required distinct operators and system specific training. Thus cross-training was cost prohibitive and the ability to monitor and control the plant from a single location was not possible.

Design Approach

In conjunction with the customer, a design approach was developed to ensure the islands of automation were seamlessly integrated into one system using modern standards for batch system architecture, design and implementation as well as applying the latest technology to field devices and bus systems.

S88 Approach

As part of the project conceptual design phase, the customer had adopted the S88 framework to provide a consistent methodology for batch system design and implementation. The S88 physical model was used to organize all aspects of the project including process cells, equipment areas, equipment modules, and control modules. The S88 Enterprise physical model was not utilized since the plant's product lines do not extend beyond the local facility. Since the physical model was separated from the procedural model, hardware and device connectivity design could begin without intimate knowledge of the recipes or procedures that each piece of equipment would be performing.

Procedural development was accomplished by defining activities that equipment modules, operations, and recipes would perform and documenting these activities in a detailed design description that would later be used to implement the software code. In most cases these activities, or procedures, were extracted from existing control code or from documented sequences-of-operation and then refined to ensure that they fit into the new S88 architecture. To facilitate reviews, detailed design specifications were written in "pseudo-code" allowing for all audiences to review the documents without specific knowledge of the control system code. The use of pseudo-code also made the documents relevant for any brand of control system and provided reference documentation for the facility personnel once the system was installed.

Procedure Development

Recipe procedures were based on activities carried out at the process cell level which typically consisted of multiple operations and represented steps needed to produce a single product in addition to any co-products. Operation procedures were developed based on process Units which generally consisted of major pieces of process equipment such as reactor vessels, crystallizers, or centrifuges. Operations consisted of multiple phases. In addition to phases, additional procedures required by equipment modules were developed. The general design philosophy during procedure development used an “Operation-centric” approach which assigned development teams to individual process areas. This allowed multiple design teams to work in parallel thus reducing overall development time and allowing for teams to proceed without the need for a completely designed system.

Phase Development

A key guideline during phase development was to ensure that phase design was as simple as possible to ensure that phases could be re-used, would be easy to trouble-shoot, and would allow for robust hold recovery without complex error routines. Previous batch projects in the plant had implemented larger, more complex phases which had proven troublesome and difficult to maintain. As a general rule, phases were defined based on one process verb associated with the activity to be performed such as “charge”, “heat”, or “transfer”. A second requirement for phase design was that each phase should be able to run stand-alone. In most scenarios phases would be run as part of a larger batch operation, which coordinated the phases, however it was sometimes necessary to run a phase manually outside of the operation. With this idea in mind each phase was designed to handle its own equipment preparation, or phase initialization logic, as opposed to assuming that equipment would be in the proper state based on prior phase execution. The phase design approach simplified phase logic implementation as well as provided the ability to use the smaller, simpler phases to build flexible operations and in turn flexible yet robust recipes.

Equipment Module Development

Another area where S88 principles were applied was the definition of equipment modules. In conjunction with the customer, each equipment module was defined as a group of equipment that carries out a specific set of tasks independently and may be shared by multiple units. In cases where equipment modules could service multiple units, arbitration was used to determine ownership of the equipment modules thereby requiring a unit to “acquire” the equipment module before it could issue a command. Typical equipment modules included tank and pump systems, jacket heating/cooling, and process headers serving multiple units. Equipment modules were also designed such that they could be directed either by phases or by operator action but not at the same time. This design allowed the operator to perform complex tasks on auxiliary systems with a single command to the equipment module while outside the batch environment.

Modern Bus Systems and Specialized Interface Modules

When new automation was added to the existing system as part of the project, modern bus systems and devices were selected for installation. Foundation Fieldbus was the protocol of choice for new analog instrumentation. ASi bus was implemented for control of discrete valves and new process sensing switches. Profibus was implemented for interfacing with motors and variable frequency drives. The use of the newer bus systems decreased installation cost and provided the operating areas with additional process and diagnostic information not previously available.

In cases where the existing input/output systems were still reliable and supported, interface devices were used to connect these legacy systems to the new system. This approach allowed the customer to retain their capital investment in the original input/output system while migrating control features to the more modern and more reliable process controllers associated with the new system. The use of I/O interface devices also prevented the need for rewiring and termination of hundreds of field devices that would have required additional testing had the interface modules not been implemented. This decision avoided additional capital costs, greatly reduced start up check out costs, and shortened the entire project execution time.

Implementation

Phased Implementation Strategy

Due to the large size of the project, we used a proven phased implementation strategy to successfully complete the implementation stage of the project. The team first categorized all of the existing instrumentation and field devices to provide a logical grouping of similar devices. Control modules, the base element in the control system, were created for each major category such as valves, motors, analog controllers, etc. The control modules were developed to be robust enough to handle minor differences between types within a given category thus resulting in a limited number of control module types to maintain and program. The same approach was used to develop equipment modules and then phase modules resulting in a full software library of tested objects that was in place before any logical and batch programming occurred. For large automation projects, this building-block approach is crucial to efficient and consistent implementation.

Object Based Development

The use of object based development tools played a key part in maintaining efficiencies on this large project. Object based tools allow a master object or template to be created and then instances created from the master object. It also allows the programmer to make changes to a large number of objects once they are instanced by simply changing the master object. Object based development was used to develop control module templates, equipment module templates, phase templates, and unit templates based on the software building-blocks discussed in the previous section. As a result, software development time was reduced since a master object could be instanced easily while changing only a select number of parameters on the instanced object. Object based development also resulted in a more consistent product since multiple developers pulled from one master object library during software development.

Simulation Tools

A process simulator package was employed to allow for simulation of both I/O (traditional tie-back simulation) and the physical processes as well as a tool for operator training. The simulation package provided the ability to fully configure the offline system in the same manner as it would be installed in the field. This approach allowed for non-intrusive, full functional testing and debug of the production software and configuration prior to field installation. It was also used as a training environment for control system operators and engineers. By having a simulated process, the operators could execute entire batches just as they would on the live system once installed. Using a simulation package not only dramatically reduced errors found during startup, but permitted the plant operators to comfortably take control of the system once it was turned over to them due the extensive real-life training acquired during the project phase.

Single Batch Manager

One major challenge in this project involved the use of a single batch manager for the entire facility, which already had other units actively running. One major drawback of the system was that adding a new phase to one recipe or new parameter to a phase required the batch manager to be stopped and restarted which impacted the whole plant as all running procedures had to be put into a hold state. This scenario illustrates the importance of properly designed phases and robust hold recovery logic to enable phases and batches to be held and restarted without losing a batch or causing severe operational problems. Another problem with having a single instance of a batch manager is that one problem on one computer could potentially impact every batch in the system and possibly bring the whole facility to a state where no batches can be executed. Unfortunately the system architecture did not provide for redundant batch managers. Although these problems do exist, the customer felt that through proper system design, back-up procedures, change management scheduling, and system maintenance the issues could be mitigated. Thus the client felt that the advantages of a single batch manager, including reduced cost and reduced system maintenance, outweighed the disadvantages.

Project Successes

Several successes were realized for the project resulting from the design approach and implementation techniques highlighted in this paper. Most notable was that the primary goals of the project were reached. All of the plant areas were started up safely and per the project schedule with only minor errors which allowed the customer to start making usable product after only a few dry runs of the new system. In addition to the successful start-ups, in almost all cases batch cycle times were decreased due to the increased automation. A single operator could now comfortably monitor multiple processes from a single location which allowed the plant to free up multiple operators to perform other essential duties. Additionally, due to a well conceived implementation structure, the system was delivered with a configuration that was easily adaptable to future recipe or process changes making it easier to maintain and implement changes.