MatrikonOPC

OPC Dynamic Recipe Loading

OPC Best Practices for Batch

Mustafa Al-mosawi

Executive Summary

Batch industries face unique challenges in getting the most out of PLC Automation. One of these challenges is recipe loading and management. Traditional methods of recipe loading and management involve excessive operator input and are extremely vulnerable to error. The result: wasted batches, quality control risks, and missed process improvement opportunities.

A simple open architecture based on OPC and ODBC is a great place to start on the road to dynamic recipe management.

Over the past decade, there has been accelerating pressure on manufacturing industries, such as food and beverage, automotive, pharmaceutical and electronic fabrication, to improve efficiencies and achieve the same optimized operations traditionally enjoyed by other industries. The unique challenges of these industries can be met with a variety of solutions available on the market today.

One of the common demands of recipe and batch systems is the ability to dynamically load recipes from a database and push the recipe values dynamically into the PLC or DCS.

TRADITIONAL APPROACHES

Traditional methods involve an operator sitting at a control station. The operator loads a recipe from a recipe management system then manually transfers the values for the recipe into the process set-points on the HMI. For particularly complex batches where there are a lot of parameters, constant rechecking is required. This approach relies heavily on operator input and is extremely vulnerable to error.

Larger enterprise solutions may be able to solve this problem, but the solution comes with a significant price-tag, system lock-in and management and deployment overhead that not all manufacturers are ready to take on.

The result: wasted batches, poor quality control, and missed process improvement opportunities.

A simple solution consisting of standard off-the-shelf software that uses open standards is possible today.

Where do we start?

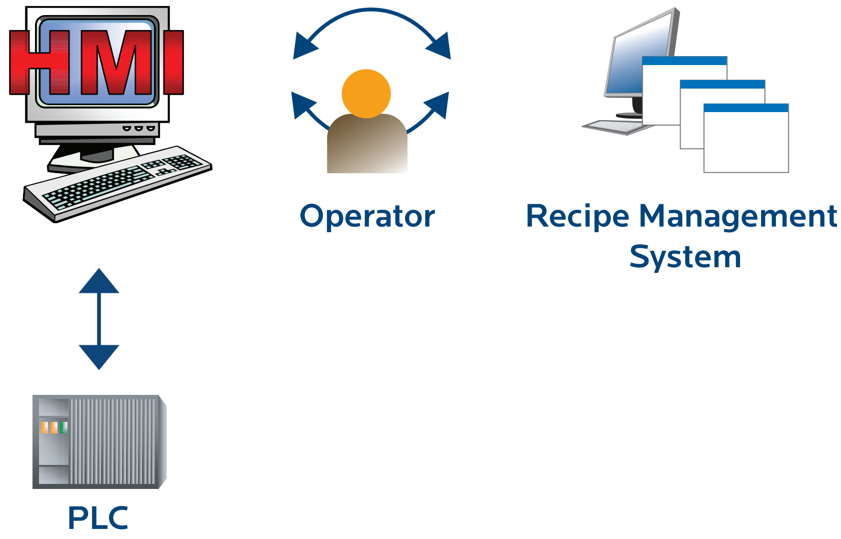


Figure 1 - Traditional Approaches are vulnerable to operator error

One of the common demands of recipe and batch systems is the ability to dynamically load recipes from a database and push the recipe values dynamically into the PLC or DCS.

A FEW WORDS FROM ISA-88

Let's start with a brief examination of the ISA-88 Standard for Batch Control. Although not all batch systems have strict adherence to ISA-88, it does provide a useful framework for understanding batch systems generally.

At its core, whether you are extruding widgets, laying down semi-conductor, or mixing medicines, the set-up of the plant equipment is intimately related to the product being produced.

According to ISA-88, a batch Process can be organized in a hierarchy. The structure defined by ISA-88 is: Process, Process Stage, Process Operation and Process Action.

These processes are carried out by equipment within a control scheme. There are three elements of control typically required in the batch industry: basic, procedural and coordination. Basic control includes regulatory, interlock, discrete control, etc. Procedural combines basic controls in such a way that a particular process can occur. Coordination control is the highest level of control and initiates, directs or modifies procedural control. Like processes, Procedural control can be organized in a hierarchy. The structure defined by ISA-88 is: Procedure, Unit Procedure, Operation and Phase.

The elements that we want to focus on are the Process Actions and Phases. A Phase is best understood as a series of instructions. These instructions control equipment. Through a series of instructions, the equipment performs a Process Action.

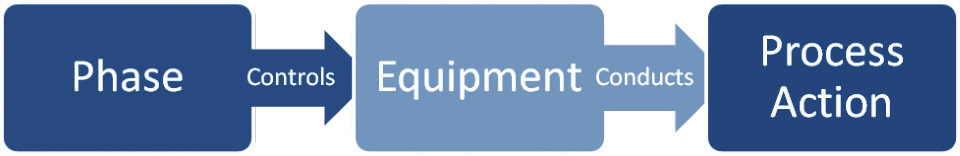


Figure 2 - A phase is the smallest element in a procedure

If these instructions can automatically be loaded into a PLC, the basic building blocks will suffice for a batch of any complexity, since a complete batch procedure can be simplified to a sequence of many phases.

ON OPC AND RELATIONAL DATA

There are a few more basics to cover. One of the basics of any system architecture is the underlying data. Process and equipment data is time based. That means the data can be generalized as a set consisting of Tag, Value, and Time parameters.

Most tools for analyzing time-series data only acknowledge the time-based nature of this data. As far as such systems are concerned, no two data points, such as temperature and pressure, have any relationship to each other. Relationships only exist in the mind of the engineer or technician.

With batch processes, there needs to be relationships between data-points. At its core, whether you are extruding widgets, laying down semi-conductor, or mixing medicines, the plant configuration is intimately related to the product being produced. The particulars for each product are commonly called the recipe, and the time period over which that recipe is valid is called a batch.

The common way to represent recipes digitally is in a recipe management system based on a relational database. The tension here is obvious - how to represent these recipe phase set-points (stored as relational data in a database) in a form that equipment can understand?

This is where OPC comes in. OPC-DA is the open standard for real-time data communication between equipment and other systems. By choosing an OPC server for relational databases, relational data is now available in a standard that equipment can understand.

Now, how can we transform the relational data structure into the Tag, Value, TimeStamp format of OPC? This is done within the OPC Server.

Within the OPC Server for relational databases, we can create a query that extracts the desired data from the recipe database. Not all recipe databases will have the same structure, but for example let us consider a simple database with the following tables: Recipe, joinRecipeParams and Params.

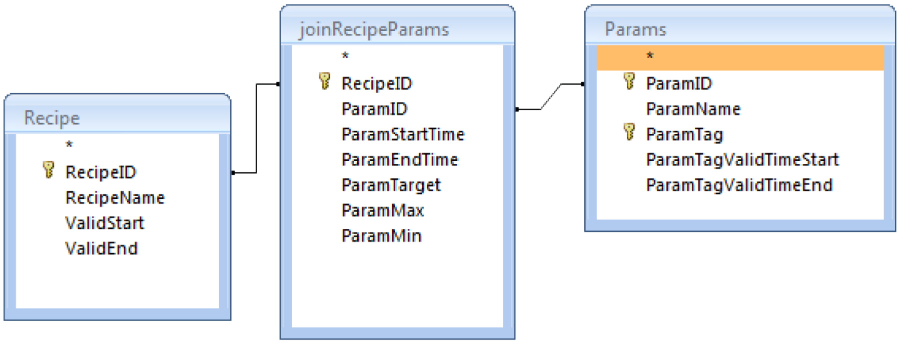


Figure 3 - A simple recipe database structure

The table fields that are of interest are ParamTarget, and ParamTag, but they are in different tables. A simple join with a 'WHERE' clause for the recipe name will suffice to return the desired values.

SELECT Params.ParamTag as Tag, joinRecipeParams.ParamTarget as Value FROM Recipe INNER JOIN (joinRecipeParams INNER JOIN Params ON joinRecipeParams.ParamID = Params.ParamID) ON Recipe.RecipeID = joinRecipeParams.RecipeID WHERE Recipe.RecipeName=@RequestedRecipe;

More complex structures and recipes will require more complex queries that are better expressed as stored procedures. When considering an OPC Server for relational databases, ensure that it can support both simple table expression and stored procedures.

Four pieces:

- An OPC Server for the PLC.
- An OPC Server for Relational Databases
- An OPC Client to transfer between two Servers
- An OPC Capable HMI.

DELIVERING DATA FROM THE DATABASE TO THE PLC

Now the recipe data is available in an OPC Server - how do you deliver that data to the equipment?

Once again, OPC simplifies the task. All the major PLCs in the market have excellent OPC Servers available. All that is required is an OPC Data Management Client capable of transferring data from one client to another based on a trigger. A trigger - such as a change of a value on one OPC Server - causes the values to be transferred from the OPC Server for Relational Databases to the PLC OPC Server.

DYNAMIC RECIPE LOADING

The next step is to close the loop with an OPC capable HMI. Most HMIs have supported OPC for some time.

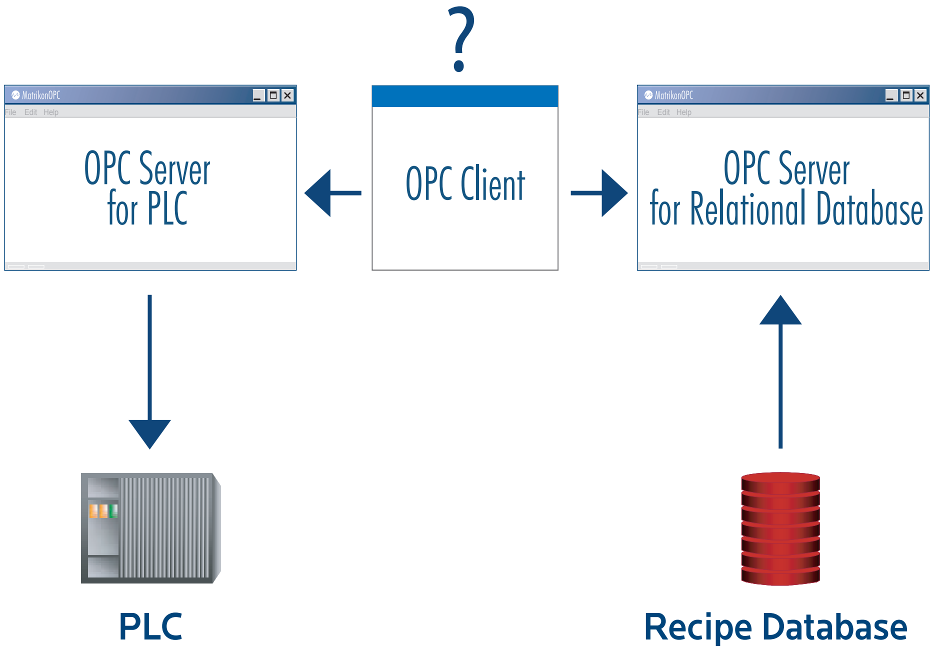


Figure 4 - The Building Blocks of the Architecture

Here we see the power of an architecture that uses OPC and Relational Data. First, build a drop down on the HMI that will load the Recipe IDs and names. Selection of the Recipe ID will write to a tag on the PLC. The data management client detects the change and sends this tag to the OPC Server for Relational Databases.

This executes the table-request or stored procedure, and the results are OPC Tags containing the recipe parameters. These newly updated tags are also read by the data management client and delivered back to the PLC via OPC.

These parameters, now having changed, will display on the HMI. All that's left is a visual confirmation on the HMI, and a single operator interaction; run the batch!

The architecture diagram below shows how one might be able to dynamically load recipe data into a PLC from a relational database using common off the shelf products.

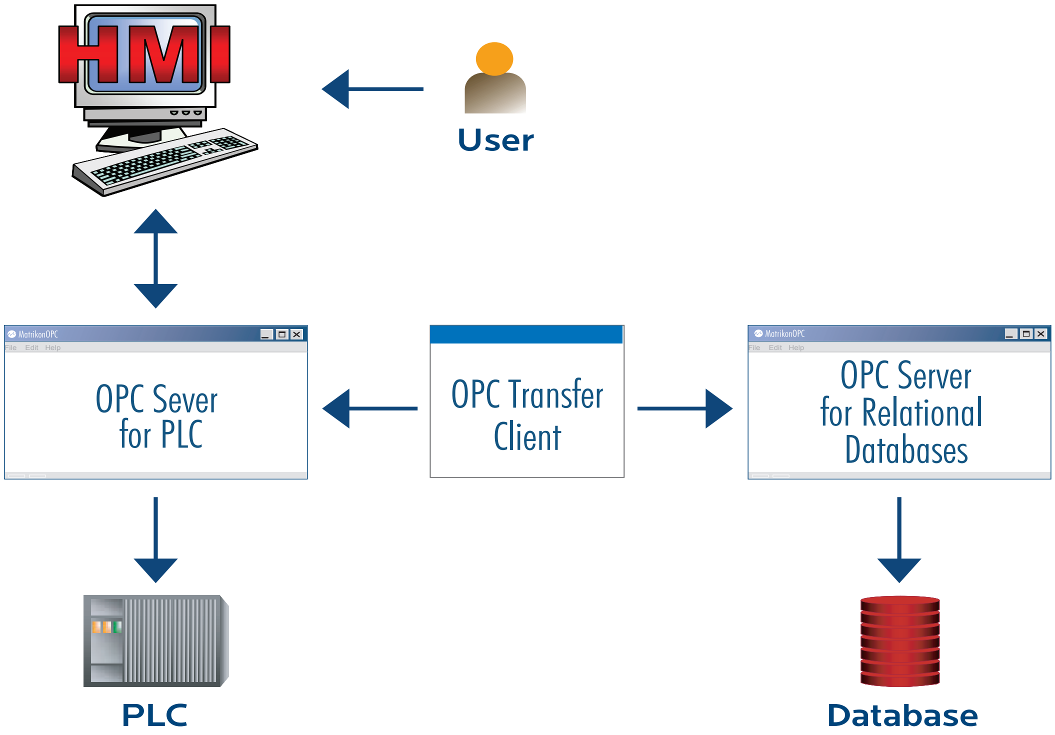


Figure 5 - The Completed Dynamic Recipe Loading Architecture

DYNAMIC RECIPE LOADING - MADE EASY WITH OPC.

Today, Dynamic Recipe Loading is easier than ever to solve. There are simple, low-cost, small footprint, standards-based solutions for dynamic recipe loading. Solutions that enable process engineers, integrators and IT Professionals to build a reliable system that loads recipe data from relational databases on a trigger from an HMI without having to write and maintain code. By standardizing on OPC-DA, the solution stays open and flexible, regardless of the PLC, Database or HMI vendor.

SOLUTION
There Are Three Components To The Solution:

MATRIKONOPC UNIVERSAL PLC SERVER

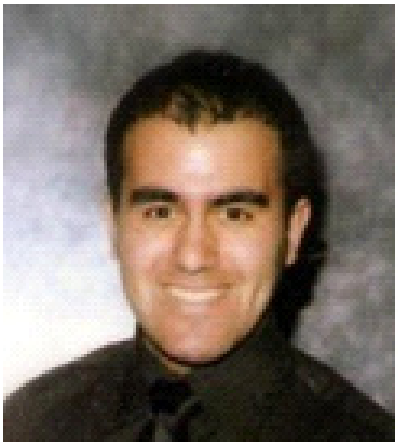
MatrikonOPC Universal PLC Server is a single OPC Server that provides connectivity to multiple devices, protocols and APIs. MatrikonOPC Universal PLC Server offers a wide range of plug-ins to support most popular PLC protocols delivering a cutting edge and new horizon of connectivity between vendors.

OPC DATA MANAGER

OPC Data Manager (ODM) is a software application that transfers data from one OPC server to another. Use ODM when you need to share, map, and bridge OPC data between two OPC Servers, such as a PLC and a Database. With ODM this connectivity can be accomplished with standard, off-the-shelf software.

MATRIKONOPC SERVER FOR RELATIONAL DATABASES

The MatrikonOPC Server for Relational Databases (ODBC) supports both real-time and historical data access to ODBC and Oracle compliant databases. Users may map the OPC point name, value, quality and timestamp by ODBC source table column, custom queries or through pre-configured stored procedures.



ABOUT THE AUTHOR

Mustafa Al-mosawi, BSc, is a product manager at MatrikonOPC. He has 8 years experience in design, development, commissioning and testing of industrial software products. Mustafa previously worked in the nuclear industry as a business and systems analyst with Ontario Power Generation-supporting next generation initiatives. Prior to his current role, he was a senior product designer for the Matrikon Suite of products, including Operational Insight, Alarm Manager, and TaiJi PID.

For More information on OPC Architectures

For more information other OPC topics, be sure to follow up with these associated titles on our website at http://www.MatrikonOPC.com