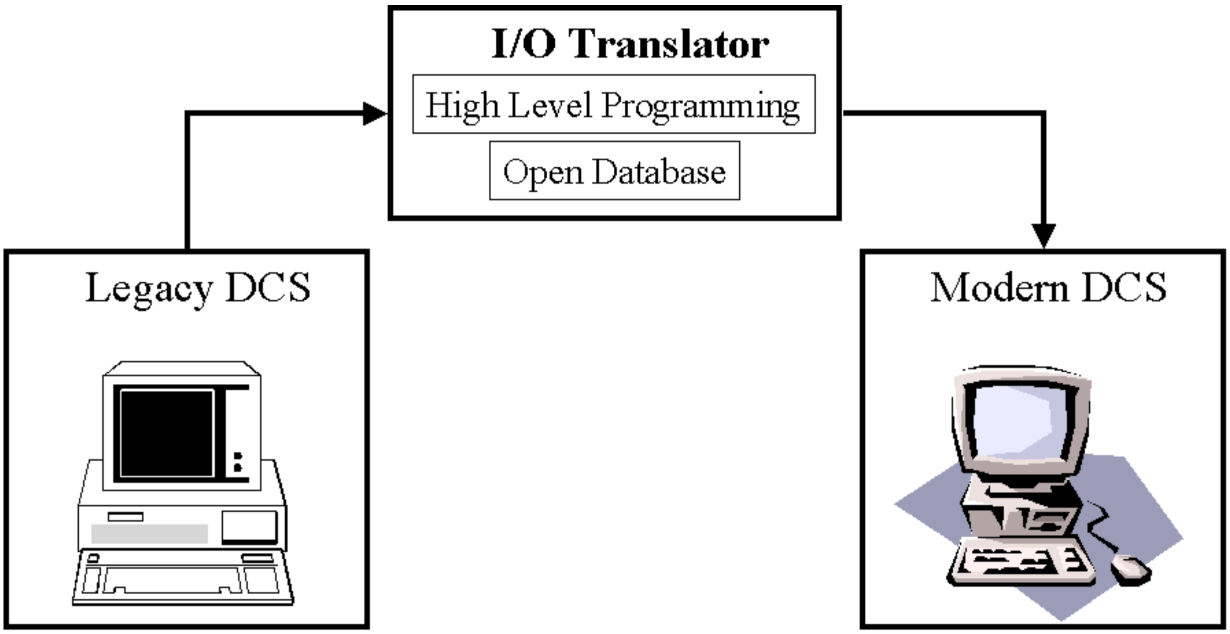
Modern Database Tools Applied to DCS

by Patrick J. Dixon,

Dixon Process Automation Services Inc.



1. Introduction

Control systems have traditionally been distinct from commodity computers. Due to the unique real-time and reliability demands of these systems, they were built from their core by the supplier with these considerations in mind. This meant that the operating system, file format, and application tools were unique to the system.

Vendors began to develop internal tools to transfer files back and forth to make use of more powerful or easy-to-use tools on more commodity machines. These were used to improve the productivity their application engineers. It took some time before such capability was available to the end user.

As commodity systems have improved in reliability and performance, many vendors have moved at least part of their system to a more commodity environment, such as Microsoft NT or Unix. These tend to be the user interface and advanced application server machines, while the process connected devices remain in a unique environment for performing fast regulatory control and logic. Software tools available in this open environment can now be used in place of those that are built-in to a proprietary system. These tools tend to be more powerful and productive than their predecessors. This is due to several reasons:

- A tool which is more commonly used in the marketplace has a lower learning curve. It is much more likely that someone already has experience with it, whereas there are far fewer with experience on a particular vendor's control system.
- There tends to be a lot more functionality in a software tool that has orders of magnitude greater sales volume than a control system.

In this paper, I will present an example of how modern database tools can be used to greatly improve the productivity of tasks on a control system.

2. Problem Statement

The DuPont facility in Richmond, VA produces several products. One of these is Tyvek© , which has end uses such as house-wrap, medical packaging, and apparel. This site has 4 production lines for Tyvek© . The 2 newest of these lines have all the latest Honeywell field devices. The network that these devices reside on is called the Universal Control Network (UCN).

The 2 older production lines are together in a separate building and share a Honeywell system. The field devices on this system were installed in 1988 and they reside on the Honeywell Data Highway. There are 64 field-connected devices on this system with approximately 2800 I/O points. In addition to the Data Highway, this system also has a UCN network with some I/O in service.

In order to take advantage of newer technology and have consistency across all production lines, DuPont will upgrade all Data Highways on these older lines to UCN networks. All of the I/O from the Data Highway will have to be moved to UCN devices. This will require several phases that coincide with scheduled shutdowns over the period of 1.5 years.

It is desired to make this transition as transparent as possible to operations, so that the system behaves the same way as before. There are approximately 2800 I/O points involved in this upgrade. For each I/O point, parameters such as ranges, alarms, and descriptors must be filled in. This makes for a great deal of tedious data entry, and is error prone.

Take as an example the translation of analog input points from the Data Highway to the UCN. A UCN analog input tag has over 41 parameters to configure. Just to get the point descriptors and unit assignments translated for these tags requires tens of thousands of keystrokes. The ranges need to be copied in order to get the proper signal reading. Alarming configuration must be identical to maintain the same functionality. There are parameters that are renamed between the 2 systems. Some parameters that exist on a Data Highway analog input do not exist on the UCN. To further add to this effort, some parameters do not translate directly. An Alarm Deadband is represented by a floating-point number in the Data Highway (0.00347% to 7.11%), but is a string in the UCN ("HALF", "ONE", "TWO", "THREE",…). These analog inputs do not only come from analog input tags in the Data Highway, but also from regulatory tags. Therefore, the person who is translating these points must look at all possible connections for the reading in the Data Highway to know which analog input tags to create in the UCN. Lastly, the engineer has to assign new I/O addresses for each analog input in the UCN, since a tag cannot be loaded without an address. These addresses must not conflict with addresses for other analog inputs or any other type of I/O in the device. It is desirable to have a systematic approach to assign the addresses, so that it is intuitive for later engineers who need to maintain the system. The engineer has to remember not to assign more tags on an I/O module than its maximum, and not to assign too many I/O cards in a device.

The estimated labor effort to rebuild all of the I/O points is over 500 hours. This does not account for any time correcting errors, which is likely to be significant when data is manually entered.

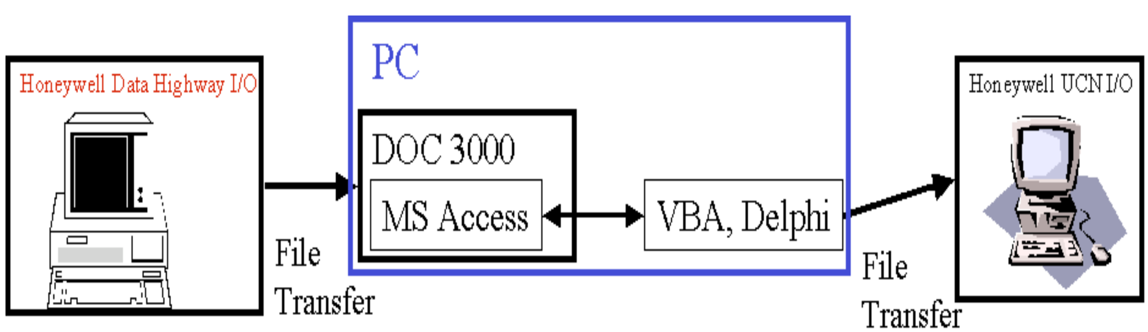
The address assignments mentioned above are also needed in order to determine which I/O cards to order and where they should be placed in the HPMs. The type of I/O processor and FTA will affect the number of I/O points which can be built in each module of an HPM. These card types need to be selected before the I/O is assigned.

Also, any inconsistencies between the existing database and the I/O drawings need to be reconciled. There may be cases where the design shown on a drawing was never implemented, or an I/O tag was built but is not connected to the field.

It must also be considered that changes may occur as the project proceeds, which can effect the I/O database. The thought of having to manually correct this database over the 1.5 year duration of the project is not very appealing, to put it mildly.

In summary, the work involved in moving the I/O from the Data Highway to UCN devices on a system this large is complex, laborious, and error prone.

3. Solution



DuPont uses DOC 3000, from Plant Automation Services Inc., to document the entire control system. It stores the entire control system database into Microsoft Access tables, as well as process schematics, programs, and other system features. This tool has been extremely helpful in troubleshooting.

In order to dramatically reduce the effort and improve the accuracy of building the new I/O database, a feasibility study was initiated to develop software to convert the data highway I/O points from the Access tables into UCN I/O points. If the solution proved to be feasible, then a project would be undertaken to develop a software translator from the Data Highway I/O database to the UCN I/O database.

The translator must also assign the UCN address to the translated points. In order to do this, the software translator must keep track of the spare module and slot addresses in the UCN. When complete, the UCN configuration will include all I/O module assignments necessary to accommodate the I/O.

The deliverables from this development will include:

- Files used to build all I/O points in the UCN
- Table that shows the configuration of all I/O modules in the HPMs
- Listing of all I/O modules that need to be ordered
- Listing of all used and spare slot addresses for UCN points
- Cross references to verify I/O against drawings and field examination
- Software application that can be re-used during each phase of the upgrade as unforeseen changes occur in the database

4. Prototype

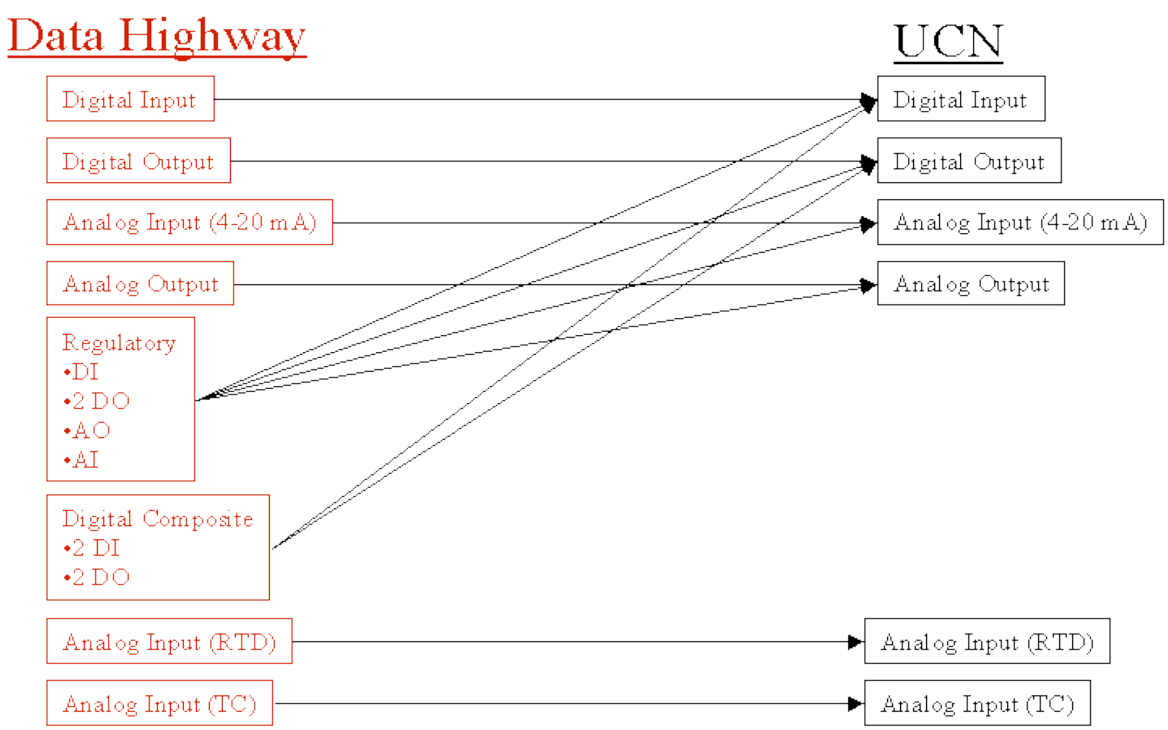
A prototype was developed to translate points from the Data Highway analog input table to the UCN analog input table. The steps involved in building this prototype were:

- Determine the mapping of Data Highway analog input parameters to UCN analog input parameters. There are some parameters that exist on one network, but not the other. Also, the names and data format of these parameters may vary.
- Develop a Visual Basic for Applications macro in Microsoft Access to translate points from the Data Highway table to the UCN table.
- Update the UCN node configuration table in Access for all necessary I/O modules.
- When the prototype is complete, estimate the effort to continue this work for all remaining I/O points. This will be based on a review with the client,

size and complexity of the software code, and amount of time required to complete the prototype.

The prototype was completed after 31 hours of development time. After review with the client, it was demonstrated that the points were properly translated and the UCN configuration showed the proper assignment of I/O modules. The software code was of reasonable complexity, and was designed in such a way to be easily modified. Therefore, the prototype proved that this development is quite feasible. It was agreed to proceed with the development for the remaining I/O points.

5. Development Process



There are multiple data sources in the Data Highway for each type of I/O in the UCN. The data sources are as follows:

UCN Analog Inputs

- Data Highway Analog Inputs
- Data Highway Regulatory points

UCN Analog Outputs

- Data Highway Analog Outputs
- Data Highway Regulatory points

UCN Digital Inputs

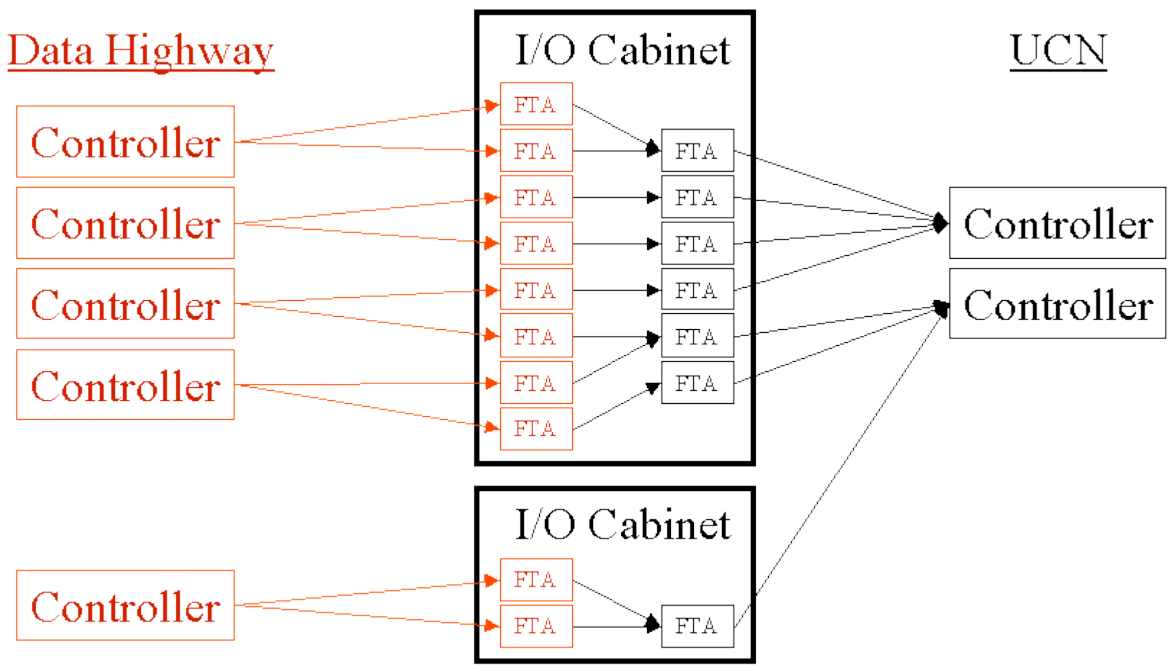
- Data Highway Digital Input points
- Data Highway Digital Composite points

UCN Digital Outputs

- Data Highway Digital Output points
- Data Highway Digital Composite points
- Data Highway Regulatory points with status outputs

For each of these cases, translation logic was developed to build the UCN points.

A table was built to cross reference which Data Highway box would be replaced with which UCN node. This table is used by the program to assign the I/O from the Data Highway to the proper UCN node.



Tagnames for sources which are not regulatory points have their names preserved. Tagnames from the inputs or outputs of regulatory points must not be the same as those of the regulatory points, since the regulatory points are built separately in the UCN. A naming convention was embedded in the code for changing these tagnames. The code also ensures that all tagnames are unique.

After all points have been translated to tables in Microsoft Access, a separate routine checks to make sure that there are no duplicate address assignments. If this occurs, a message is generated. Otherwise, a table is built that shows all used and spare I/O addresses.

Additionally, it was desired to have UCN nodes which are used for similar functions on different production lines to have identical I/O addresses. This would ease documentation and maintenance. However, it was discovered as a result of this work that too many I/O slots were wasted, so this approach was abandoned. Instead, the I/O addresses for each node were assigned in alphanumeric order. This was reasonably easy to change in software, but would have been extremely laborious and frustrating if done manually.

6. Results

Following 160 hours of additional work, the application was complete. This work included the experiment to have identical mapping for UCN nodes in different production lines. All objectives were met.

The translated database was compared to a database built independently from I/O drawings. There were approximately 150 discrepancies found which identified items that were incorrectly configured in original database or were not accounted for in I/O drawings. These discrepancies could not have been found without the translated database. Fixing these discrepancies will make the system more maintainable, clean up unnecessary wiring, and free up spare addresses in the database.

The entire I/O database has been translated into files for all the UCN nodes that will be installed. All parameter settings (ranges, descriptors, alarm settings) have been preserved so that the system will function identically to the current system.

When the results of the automated approach are compared to the manual alternative, the following advantages are shown:

- The automated approach took 200 hours to complete vs. an estimated 500 hours to manually generate the database
- If changes occur during the project, such as address assignments, the program can be simply run again. A manual approach requires a lot of tedious re-work and results in a frustrated engineer.
- By automatically translating all parameter settings to the new points, the probability of errors is drastically reduced. Also, any errors in the translation can be easily fixed in code. Errors from manual entry can be random in nature, take a long time to fix, and are likely to increase as the tedium of the task continues.
- Discrepancies between the existing database and design drawings would have been practically impossible to find on a system this large using a manual approach.
- Different approaches to I/O addressing were examined with changes to software code, which can be modified in a reasonable amount of time. Attempting to do the same with a manual approach will certainly result a high level of frustration.

The results of this work will save an enormous amount of effort on this project and give peace of mind to operations by ensuring that the system will behave the same as before the upgrade.

7. Conclusion

The combination of a data import tool (DOC 3000), a commonly available database (Microsoft Access), and a high-level programming language that easily interfaces with the database makes this solution possible. This is a powerful and flexible combination of tools that eliminates many barriers to productivity. The application described here is just one example. It can be easily imagined how many other solutions to control system maintenance and configuration tasks can be provided with tools on commodity platforms.