A paper for System Integrators

Managing Design and Documentation on Automation Projects

This document is for companies supplying DCS or PLC or PLC/SCADA systems, and working in the process industries. It covers:

- Providing excellent Control Systems.
- And providing excellent documentation
- On large fast track projects.
- Ensuring teams work cohesively
- The advantages of using ControlDraw, a specification modelling tool.

The reader is presumed to have a good understanding of process control projects, S88[1] and software development activities.

ControlDraw Ltd, 2002
Author Francis Lovering[2]

[1] In general the paper conforms with S88 guidelines, but does not discuss alternative interpretations of them.

[2] Francis Lovering has had a long a diverse career in the process control industries, working at all stages of the life cycle. He now runs ControlDraw Ltd, and concentrates on providing tools to assist suppliers and users in the automation industry, whilst still working as a practitioner of control systems design.

Contents

Rev 71

Summary

The paper discusses the issues involved in providing excellent process control systems on large, fast track projects. It is written from the point of view of a system's integrator charged with the responsibility of supplying the Control System.

Excellent process control systems and projects meet the following objectives:

- Users have the opportunity to influence the design to recognise their operating culture, and their specialist knowledge. And like the end result.
- The systems contain the codified expertise of the equipment suppliers in how their equipment should operate and the process designers in how the process should work.
- Documentation is clear, understandable, accurate and timely
- Testing and validation is done efficiently, using data that is clearly traceable to its origins.
- Project managers have control over the design and implementation process because the process is measurable. Consequently the system is delivered on time and in budget.

The introduction describes the issues involved for systems integrators, especially as projects increase in size and tasks have to be divided up. It looks at conventional methods used by most practitioners and then the paper shows how ControlDraw modelling can be used to help the complex process of succeeding on such projects.

Introduction

The Integrators task

As the system integrator you have to make sure that the entire system achieves the required functionality and works homogeneously.
Systems Integration involves

- Ensuring all the IO, controllers, workstations etc are correctly specified, found a place, ordered, tested etc
- Parameterising the control system with the details of the plant, for example setting all IO address and all the scales and ranges.
- Designing the procedural control – the recipes
- Mapping the recipes to the equipment
- Efficiently deploying the capabilities of modern control systems
- Considering all along the needs and wishes of the end users

You have to:

- Collect all the information that is required from the project that is needed to program the system. All the control modules and their settings, the phase logic steps and transitions, the Tagnames for everything
- Map all that data into the target system
- Define the required states of all objects and modules.
- Provide a system that meets your users expectations of what you can supply.
- Provide an accurate documented history of the development of the system and the origin of the data. This is mandatory for validation on pharma projects, but even on less regulated projects it is also a good idea for other reasons. Even in case you get a problem client.

The difference between small and large projects

One good person in a systems integrator can design a small project. They can understand the requirements, write the functional specifications and the design documentation and maybe even program and install the system. Such projects and engineers are however rare, and possibly wasted on small projects (another paper will address single person ControlDraw projects)
A larger project may require a small team to do the same.
Many projects are too large for even a small team, and may need a team just to define all the functional requirements, and another to design and implement the system, and often another to test it.
Large projects are often also split on an area basis, with a team looking after each process area. This may reflect the overall contract, as when an Engineering Contractor designing a large plant sub contracts the design of one area to one supplier, another area to a second supplier, the utilities to another.
There is a risk of different styles appearing in each process area.
Plant operators do not want to have a different style of operator interface depending on which bit of the plant they are operating. Operational requirements are not the only reason to have a consistency across all the packages. The same applies to software and documentation. You, the systems integrator, have to reconcile all these.

Some of the process package suppliers do not have the required skills to define the phases and so you send your engineers in to help them. This may be a business opportunity for you. But you have to be careful to avoid the conflicts that can arise when the boundaries of responsibilities are not clear and agreed.

Issues

The need for technical management

Large distributed projects like these needs careful technical management in order to co-ordinate the various parties. Especially in fast track projects where it has to be right first time. Standards are required, and these are not yet available for process automation from an ISO numbered document. Those standards, such as S88, that do exist are just guidelines with many opportunities for different interpretation.

This paper discusses the problems of using conventional 'last century' specifications and tools on such projects and explains how ControlDraw can improve the design and documentation process.

Last century documents – the review problem.

Lets assume that your customer, the EPC Company has divided the project the project into the following sections.

Area A	- Process Supplier A
Area B	- Process Supplier B
Utilities	- Process Supplier U
Control System	- You, the System's integrator

Each has to produce detailed specifications covering their part of the overall application.

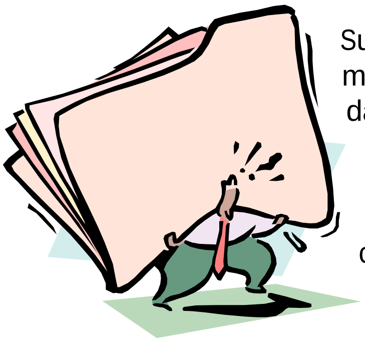
Suppliers A,B and U have to provide a detailed functional description of how the equipment that is in their scope is to be controlled. These are to be agreed with the end users and approved by the Design Managers, and then given to Supplier CS.

You, the system's integrator, have to provide a system that achieves the required functions.

This means that you have to take the process suppliers functional descriptions and incorporate them into the system design documentation – the Functional Spec, Design Specs, Test specs etc. You must also ensure that all the parts fit together.

This involves an intense process of reading all the documents.

And, as this is a large project, you also need to divide the task up across a few teams.



Suppose the various suppliers all work in their 'old' way, providing a mixture of long text documents and disparate spreadsheets and databases, typically based on their last project. It is very hard to read all these documents and there will be technical and style inconsistencies between them making for many problems in reviewing them. The distribution of the ownership of the various parts of a whole control system will increase the problems.

When reading the control requirements text sent by an equipment supplier, is it clear whether it describes something that is really in the scope of the system supplier? (eg, Supplier A says 'If the valve fails to open an alarm is generated and the phase holds'. But the standard system provides valve failure alarms and a command to the standard system provides a hold. And is there a rule that all valve fails in an equipment model cause a hold?)

There is no single search, let alone an intelligent one to help to rapidly review the documents. And there is no means of measuring the contents of the document. You can count how many words are in a document. But how many useful words though?

You probably have to painstakingly go through each paragraph and tick off a checklist.

The system's integrator will then produce a functional specification that attempts to reconcile all the different suppliers requirements. Often the system's integrator will take portions of each of the process supplier's specifications into their own specification. This will be yet another document.

It provides plenty of work. And data accuracy, version control problems, and consequent time delays become a major problem.

Many documents include diagrams, for such aspects as control loops, phase logic, graphics, and state transition diagrams. These greatly help understanding.

A new version arrives from a supplier– what has changed? You hope to find a good change log in the document. And you too have to clearly identify the changes in each revision of each document that you produce. But it takes so much time to document change.

Unfortunately this change cannot be prevented. Suppliers and the systems integrator have to change things, as developments and user needs are added into the design.

This means carefully documenting each change, maintaining version controls.

Document management systems may help to do that and to ensure that the current version is used, but that is only a thin layer of version control if applied to a collection of dissimilar documents. Word processor compare facilities, whilst very useful, provide a very unstructured way of finding changes; even the issue number of a document is mostly manually maintained. And the compare facilities fail to identify changes within diagrams or other documents like databases.

Data Ownership – and databases

The issues here are concerned with who owns (produces, reviews, approves) each aspect of the overall application. And who supplies the data for each part.

Most process engineering companies use one or more project databases, often linked to their CAD. For example a typical 'instrument IO list' contains the IO addresses, and possibly the instrument scales and ranges. Some contain alarm and trip settings, but only usefully for continuous plants were these are fixed.

But there are may other parameters, such as time setting, physical dimensions, event counts and the parameters that are required by complex control loops, equipment modules and procedural logic.

CAD databases just do not handle these at all. S88 recipe software only really addresses the last of these, and even then is often not appropriate because it is not the sort of software that most suppliers are in the business of using.

Typically suppliers will add a collection of spreadsheets to their deliverables to cover the additional data.

These will cover items such as recipe parameters, and matrix tables for showing equipment states.

ControlDraw - This century documents

An integrated collection:

All the information is in one searchable intra-connected file.
It is presented in concise and clear format, as diagrams, lists, matrices and words as appropriate.

Diagrams

Diagrams are used, for each type of object. These are familiar and well established diagrams from the world of process control.

and data

The Data stored in a Control System is massive - from all the Alarm set points though to the Zero for each input, Tagnames of each device to the message text for each event. Much of this has to be defined by the suppliers and then sent to the Systems integrator, and then managed by the SI before finally handing it over to the end user.

Version managed

All FRS's go through a long a detailed development process, version management is essential, and it is built in to ControlDraw.
It also provides a progressive version management regime where designs can develop freely but then when they are agreed be place under a higher level of version control.

And stored in a database

It just makes good sense to keep data in a database.
ControlDraw models are Microsoft Jet databases, created with the Microsoft Jet database engine.
Jet is fine for small teams of people and is in fact a very powerful database engine, and is highly suited to the requirements of a small team developing a functional requirements specification and then turning it into a design.
Access is not required to run ControlDraw of course. But a systems integrator has Access on the standard office desktop and database skills. They can use these to enhance the models, for example to build front-end databases that generate much of the configuration of the target system. Many Control systems companies have such tools already, often in Access. But using ControlDraw may not mean that you have to abandon these - you may be able to link your existing tools to the models.

Easy to read.

The diagrams are use familiar well established styles appropriate for each aspect of the requirements and the design so that users can understand them easily

Ownership clear

Each supplier of information is traceable and responsible for maintaining the information as the project develops. At some point it may be handed over to the users, this is also clear and explicitly done.

Fast review

You need to be able to quickly go through the entire specifcation for many reasons, for counting, for checking, for exporting and for searching. And you need to be able to make comments back to the authors easily.

Model Review system, Search and reporting built-in

The structure of the model can be explored quickly and errors in the requirements identified. The documents can be rapidly searched, including into the details of diagrams. Ctrl-G. Original (Read-only) copies of the model can be reviewed and commented on with automated recording of the comments in a database.
Just try commenting on an FDS using Word Comments and then commenting a ControlDraw model using the Reviewer. For a start Reviewer comments are not stuck into the model, they are separate and can be moved to the next version, accumulated over time and versions and used to build a minutes of meeting.

Metrics available for all content.

ControlDraw counts – database queries, return not just a word count, but the numbers of each of the objects, including generic and instance counts. These are measurements that can be used to estimate the next stage of the life cycle and to monitor progress and change..

Version management and recording is built-in

This includes the automatic incrementing of the overall version and the ability to raise Issues of the model to mark project stages. The Issue functions also include the archiving of older versions and recording a history of the archives.

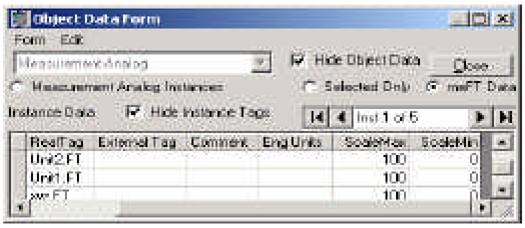
You can also record within the model additional manual notes on, for example, the reasons for changes.

Detailed object based compare functions

Since everything in the specifications is stored in a single model advanced compare facilities are available. So even if the supplier has not recorded all the changes, it is still possible to find them.

A process control database.

The database covers all the data recording requirements for a process control system specification such as IO Lists, Alarms, equipment parameters and many more.



Measurement Analog

☑ Hide Object Data

○ Measurement Analog Instances

○ Selected Only

● mxFT Data

Instance Data

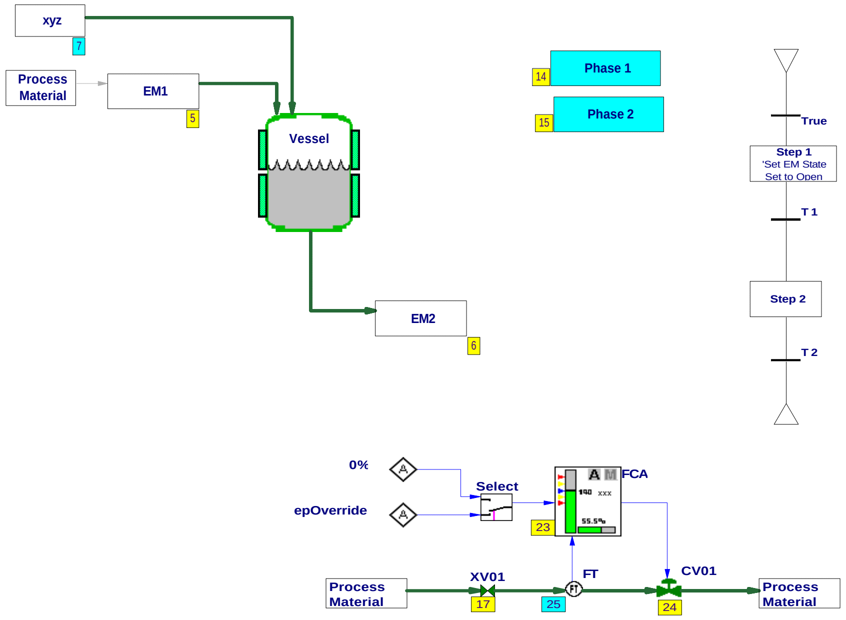
☑ Hide Instance Tags

Inst 1 of 5

RealTag	External Tag	Comment	Eng Units	ScaleMax	ScaleMin
Unit2.FT				100	0
Unit1.FT				100	0
xyz.FT				100	0

Process control diagrams

ControlDraw also keeps the diagrams, and matrices in the model. Diagrams include equipment flow diagram suitable for graphics, Function block type diagrams for control loops, Sequential function charts for procedural logic such as phases and many more.

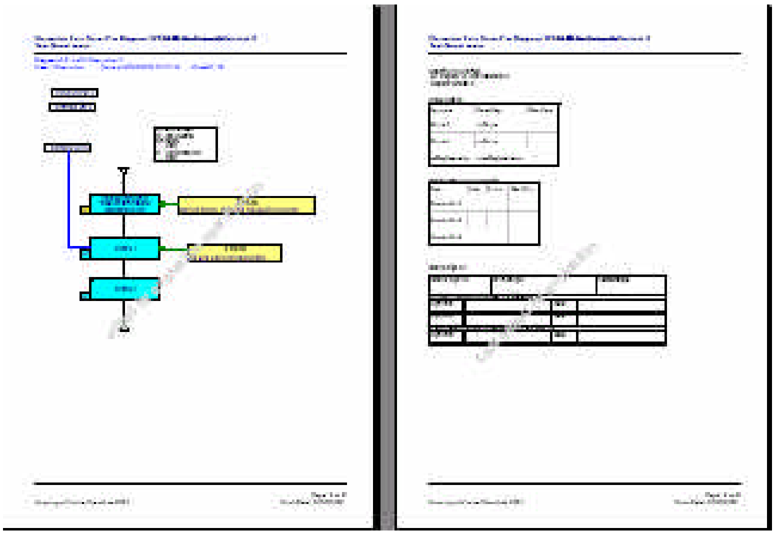


Tools to accelerate review meetings.

The Diagram reviews can be conducted by displaying each diagram on a screen using the ControlDraw Reviewer. If there are any comments to be recorded then these are added as Yellow notes.
This automatically builds an MS Access database of the comments. If preferred reviewers can make comments in the same way with a copy of the reviewer free version and send their comments database back to the modellers.

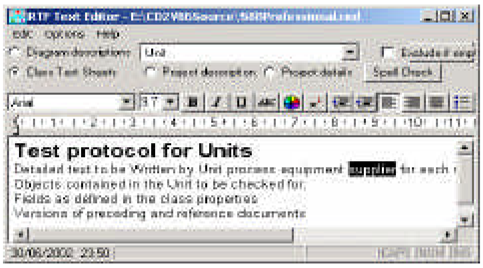
Automatic test sheet generation.

One of the biggest and most boring tasks is the production of the test documentation. ControlDraw can generate test documentation automatically from the functional requirement model.
These are configurable to include tables of object data, signoff sections and all the words.



The Words and the test records

You can write a protocol for each class of object and include it in the test sheets



Test protocol for Units

Detailed test to be Written by Unit process equipment supplier for each

Objects contained in the Unit to be checked for.

Fields as defined in the class properties

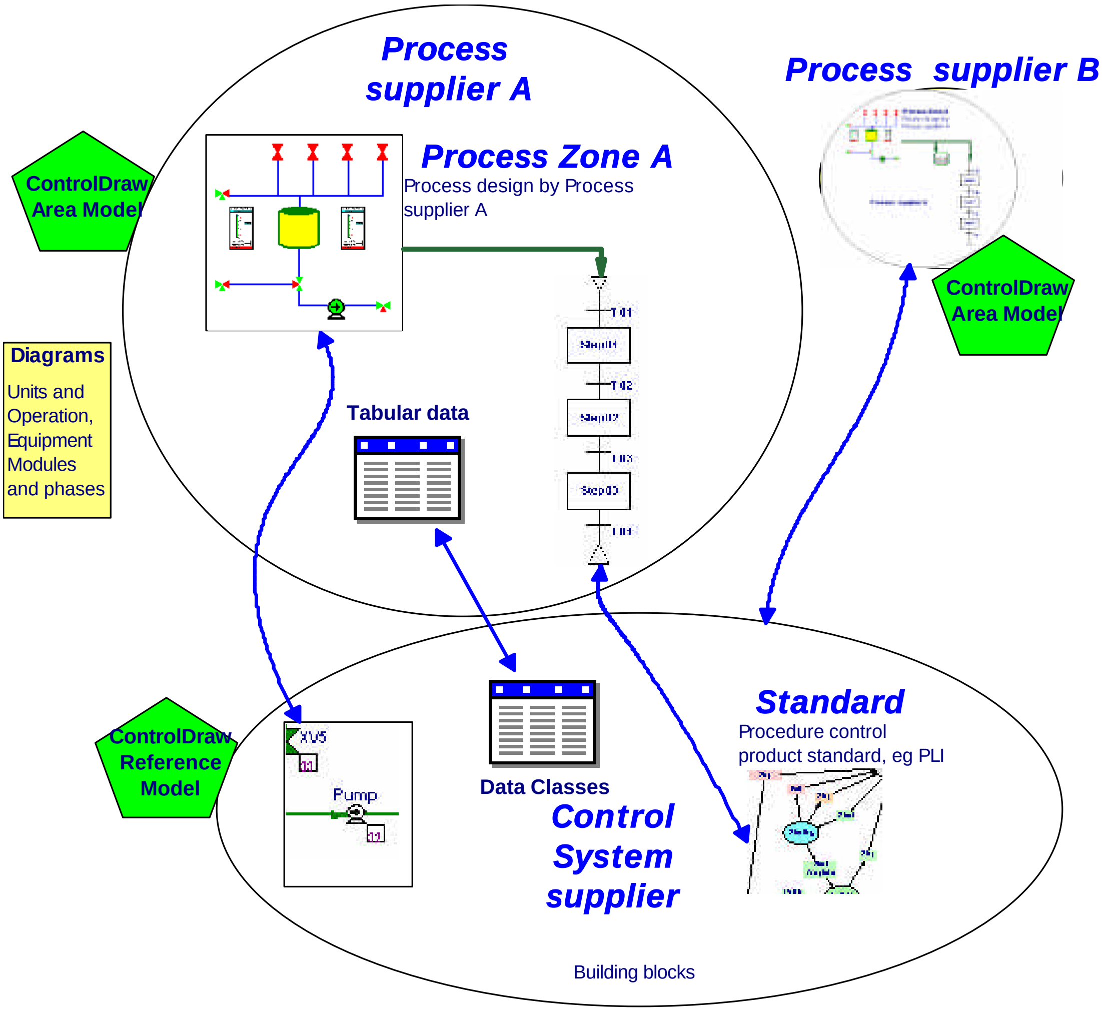
Versions of preceding and reference documents

30/06/2002 23:50

Copyright ControlDraw Ltd, 2002

Zones of Responsibility captured in models

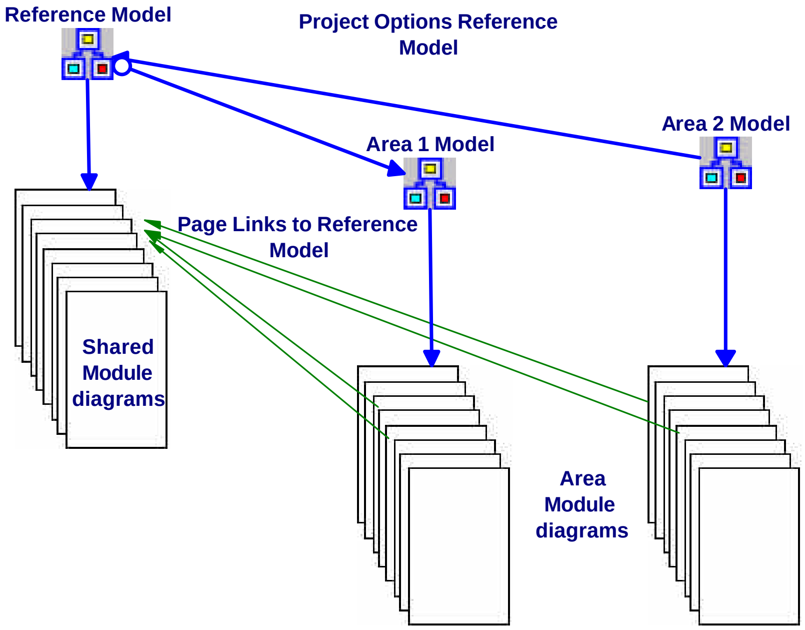
Each supplier produces a ControlDraw model for their part of the project. The diagram shows simply how it works.



Each supplier has a zone of responsibility, and depends on the other suppliers for the supply of items that are not in their zone. There is nothing special about that, it has been done with physical objects. But it has always been a problem with software.
The PC based ControlDraw Approach combines database and graphics technologies to dramatically improve the process of standardising software objects. And it does by retaining the idea of individuals empowered by PC technology.

Diagrams and data

As explained above, the approach uses Diagrams, lookup tables such as state matrices, and data tables as appropriate to the type of information being conveyed. ControlDraw provides the means to produce and these and manage the data for the application in one environment.
In a large distributed project the issue arises as to how the diagrams and data can be coordinated across multiple models. The following diagram shows simply how the Reference models feature of ControlDraw can be used for this purpose.



Use classes to define who owns the data

Some data is likely to be dependent on the recipes that the end user is planning the run. This is true recipe data.
Some depends on the control system, for example I/O addresses.
Some data is likely to be known by the supplier, for example the scale and ranges of the instruments that they are providing with their package. This is physical data, equipment parameters.
Some is neglected during the design and only corrected at huge expense during testing and commissioning.

All ControlDraw diagrams and the objects on them have a Class. For example, Control Module, Phase etc. Class defines the type of data being modelled. With ControlDraw you can use the classes to define who owns (produces, reviews, approves) each diagram and item of data.
To do this you need to define the owner for each class at each design phase.
For example

> Area Designer A produces a model covering area A with the definition diagrams for Units, Equipment modules, operations and phases.
> The Control system owner produces the Reference model. This includes the shared objects such as the control modules.

So you can produce a responsibility matrix like the following

Classes	Area Supplier	System Supplier
Control Module	Special ones	*
Control System		*
Effector Analog		*
Equipment Module	*	
Equipment Parameter	*	
Measurement Analog		*
Measurement Switch		*
Motor		*
Operation	*	
Phase	*	Common phases
PID Control Loop	*	
Process Cell	*	
Recipe	*	
Recipe Formula Value	*	
Recipe Procedure	*	
Site		*
Unit	*	
Unit Procedure	*	
Valve		*

System Independence

There are many aspects of the overall life cycle documentation that should be independent of what system is used. For example the

> "The Reactor uses Cascaded temperature control" Is a Requirement - and does not depend on the system!
>
> "On Step 3 open valve XV1245" is the same.
>
> This can even be applied in a large part to graphics.

To a systems integrator the ability to clearly identify the Independent requirements from the system details is important and useful:

It can be used to identify project stages. For example the generation of a system independent functional requirements specification can be a profitable project in itself. And since a systems integrator can in general only help to produce this, whilst leaving the ownership with the client, problems over functions can be isolated from those over the code.

An integrator often provides systems using different makes of PLC or DCS depending on the clients requirements and commercial factors.

Recognising that much of the documentation does not depend on the system is advantageous in that it improves the opportunity to reuse documentation even when a different system is to be supplied.

System Specific Design

Mapping system independent requirements to the actual system can be done in ControlDraw by adding system specific classes and linking them to the system independent objects. If this is done well then there are opportunities to automatically generate much of the system from the model. This applies in particular to tag database, IO configuration and similar table based aspects of the system. Custom developments can go further. As a systems integrator, you may well have developed some tools of you own.

It may be that you can leverage these by linking to or adding to ControlDraw models.

The Database

One approach that has been developed for process engineering companies is the large-scale database, typically Oracle or SQL Server. Sometimes this is imposed on the suppliers, including the System Integrator. The mainstream CAD companies and the engineering IT departments of the major EPC companies have struggled with this for over a decade and have made some major achievements in respect of the physical design of plants, but have never achieved this with the deeper levels of control system functions and their operation.

ControlDraw does not at present use a large database, it only uses MS Access 97. This is capable of handing up to 10 or so user sessions at the same time, not the hundreds that large EPC companies may have working on a project. However, this is not a problem because ControlDraw is oriented around small teams and has the tools to integrate the work of several teams. Front ends databases can also be used to make the data in the model available in for example Access2000.

How does your existing system compare?

	ControlDraw	Your system
S88 based structure and database	YES	?
Links to source documents	YES	?
User configurable database	YES	?
Equipment module design tools	YES	?
State matrices	YES	?
State animation and simulation	YES	?
Generic and Instance objects	YES	?
Configurable model structure	YES	?
Multiple model synchronisaton	YES	?
Grafcet / SFC diagrams	YES	?
Logic diagrams	YES	?
Procedure function charts	YES	?
Process mimic diagrams	YES	?
SAMA / Function block diagrams	YES	?
State Transition diagrams	YES	?
Configurable audit trail	YES	?
Auto Version numbering	YES	?
Publish and archiving functions	YES	?
Linked Review comments database	YES	?
Print, PDF and RTF Output	YES	?
Test specification generation	YES	?
Start templates	YES	?

What Next?

By now, you should be convinced of the benefits of the ControlDraw approach. But you are an experience systems integrator, with your own ways of doing things. Why change?

Do a Trial?

If you want to explore more then there are some points that we would like you to keep in mind.
There are no silver bullets in the design of plant control systems. ControlDraw will not make badly designed software work better. It can help you design quicker and more accurately.
It is a design tool, aimed at the stage during software development where all the software engineering texts say that the payoff for doing some work is greatest. That stage is Defining Functional Requirements.

Use it to document a past project?

Very few people have the time to do this, but sometimes it is necessary to produce a design document after you did the project. You may have subcontracted to an 'expert' who did produce software that everyone is happy except for the validation people. And now you need to produce a spec. Yes ControlDraw can be used to accelerate the production of reverse engineered specifications. It happens. The rest of this paper is concerned only with design first/

How long will it take to produce the model?

You can reckon on it taking as long as you presently spend producing the design documents the way you do it now. Possibly, on your first project it may even take longer. But it is worth doing.
The first payoff comes in the increased understanding for the users, reducing the probability of dispute.
The next payoff comes in the implementation, where coding time will reduce due to the accuracy of the functional requirements.
Then in testing, where test documentation is produced in a fraction of the time the old way took. On subsequent projects times will reduce as you build a library of re-usable objects in the models.

Find out more

ControlDraw and associates can provide:
the software to establish the system
the training to get you going
consulting on the management issues
skilled staff to help on your project

Contact fl@controldraw.com for quotations