Requirement Principles Template

This document explains a method for ControlDraw modelling for batch plants
The method is used in the sample models provided with ControDraw software and as a scope basis for consulting contracts where ControlDraw practitioners provide models as a service to ControDraw clients. Subject to agreement it may be used as a template for ControlDraw users to develop for their specific requirements models.

Some parts of this document are not complete

Table of Contents

Requirement Principles Model

This document describes how the project Functional Requirements document is constructed in ControlDraw models for the project.
It also describes where necessary, the assumed generic functionality that underlies the detailed requirements

The document is closely aligned with the S88.01/IEC 61512-1 Batch Control standard and provides a cross reference between the definitions in that standard and their corresponding entities in the models. Small portions of the standard are quoted in this document, to obtain the document (recommended) go to

http://www.isa.org/Template.cfm?Section=Standards&Template=/Ecommerce/ProductDisplay.cfm&ProductID=2887

Several ControlDraw models are produced.
Area models define the functional requirements for Process Cells (or other scope - if so to be defined)
The reference model provides re-usable objects that are used in the Area models.
This Document, the FRP is also produced as a ControlDraw model.
The Reference model could possibly be combined with this FRP, at present the difference is that the FRP covers general principles whereas the Reference model

S88.01 provides a general framework. There are within this many different ways of doing things. Some sections discussed some of the choices that can be made. These should always be made in the context of the capabilities of the users of the system, the type of plant and processes and so on.

Whilst aimed at the Requirements, the models may also have a role in the Software Design Specification, since this generic functionality is also a top level design for the software modules that will be used in the target system. Subsequent models, and/or issues of this document may be produced as the project proceeds.

References:
S88.01/IEC 61512-1 Batch Control Part 1: Models and Terminology
IEC 61131-3 Programmable Controllers - Part 3: Programming Languages
GAMP 4 Good Automated Manufacturing Practice Guide for Validation of Automated Systems in Pharmaceutical Manufacture

Project Information

Item	Value
Project Name:	Requirement Principles Template
File:	J:\F_CDReference\Samples\FRP.cnd
Client:	ControlDraw Users
Last Author:	Francis Lovering
Date Edited:	05/10/2004 17:21:30
Date Printed:	05/10/2004
Version:	143
Reviewer View Name:	FRP in J:\F_CDReference\Samples\comments.mdb

Overview

This document provides a basis for the production of a Functional Requirement Specification made using ControlDraw software.
The objective of the Functional Requirement Specification is to provide an object-oriented description of the application that defines the requirements, as agreed by the users and that is independent of the target control system.

Requirements Analysis independent of implementation

What do you want your plant to do is one question.
What is programmed to do is another.
Ideally the two are similar, however there should be some distinction in order to know if the system is doing what is required and to be sure that limitations are not introduced without a conscious decision.

The ideal is to make the requirements analysis completely independent of the technological solution. In reality this is an exaggeration - any requirements analysis should recognise what is possible. However there should be no interference of specific system attributes with the analysis.

Functional Requirement Detail

The specification comprises a hierarchical diagram based functional model of the process. This model shows all of the required equipment and functions that the system is intended to perform.

The Model is produced using ControlDraw software. A Control Draw model is collection of diagrams with meaningful relationships between the various objects on the diagrams. A diagram contains symbols and connections. Symbols may then be parents of another diagram. Diagrams that are linked underneath more than one symbol define repeated functions. For example a Valve driver diagram would be linked underneath every valve of the same type.
In this way there may be many instances of the repeated function, but only one definition.
Each symbol - or object - on a page can link down to a 'child page' - the lines represent these links. In effect a page is a collection of objects, each of which can be described in further detail on another page. The object hierarchy in the drawings matches the S88.01 physical and procedural models.
Objects that appear frequently can all share the same description - by linking to the same page. Most commonly in ControlDraw models this is applied to items such as Control Module (Motors, Valves) and so on, but can also be used with procedural elements , such as phases.

A modelling rule is that there should be only one path (ie via Objects and their linked Child pages) from diagram one to each physical entity (Valve, motor etc and also Equipment module, Units and common resources). Provided this guideline is followed then the model will generate a complete list of the physical entities in the plant via the Class Instance tables. For example, IO Lists, instrument lists etc.

A similar concept can also apply to the procedural model so that the master recipes and the batch journal requirements are defined by the model.

Description for Diagram 1 - S88 Overview

This diagram shows the S88 physical and procedural model and provides links to examples of the diagrams that are produced in order to define detailed the functional requirements.

Diagrams are produced for the physical model covering each class of :

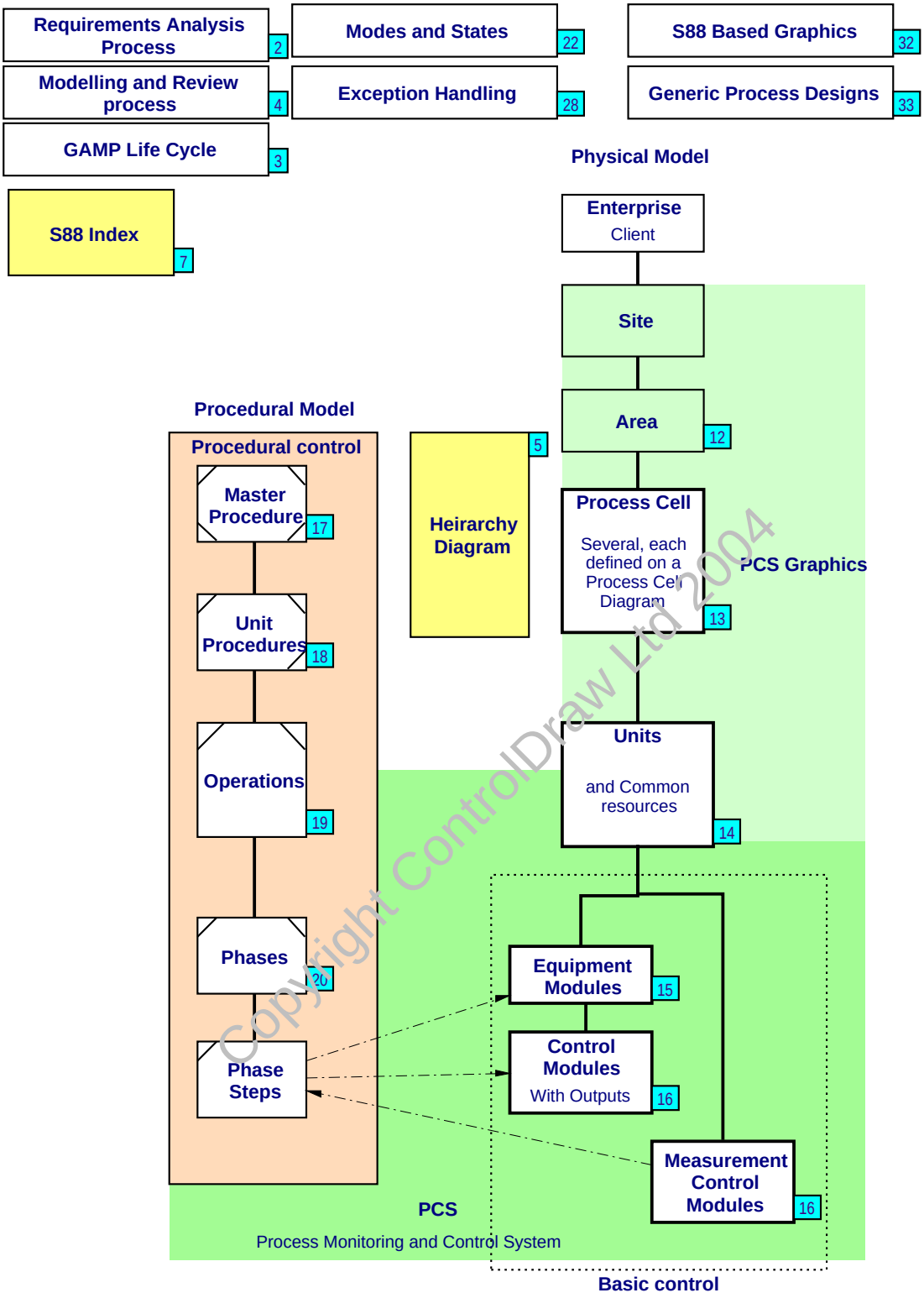
- Process Cell, of which there are several. the process cell These split at the batch boundaries.
- Unit
- Equipment Module
- Control Module
- Analog Measurement Control Modules are contained in the unit so that all measurements are available to all equipment modules in the unit
- For Control Modules With Outputs these are contained in equipment modules

Diagrams are also produced for the procedural model

- Recipe procedure
- Unit procedure (may be part of Recipe procedure diagram)
- Operations
- Phases (may be part of operation diagram)

Diagram 1 - S88 Overview

Diagram Version: 125 **Diagram 1 of 43**



Description for Diagram 2 - Requirements Analysis Process

The diagram shows the main activities for developing the detailed functional requirements. It is oriented around the physical and procedural models.
S88.01 defines these models generically - that is to say that the models are universal and apply to any plant. The analysis process must take these generic models and then apply them to the particular plant. For example the physical model in S88 shows Control Modules. The requirements analysis should identify what are the actual modules required by the particular plant.
The diagram above shows that there are pre-requisites before the detailed analysis is done.
These starting points include:

Piping & Instrument Diagrams:

These contain the details of the equipment in the plant
Usually available before the analysis is done
If not available, the requirements analysis has the opportunity to document or even influence the basic design process

Control & Operability Philosophy:

This should contain the general principles that the end user will want to apply to the project. It should define the level of automation that is required, the information handing needs and the operational requirements.
Typical issues are:

- What level of flexibility is required? The plant may be dedicated to a single product and always use the same recipe. Or the plant may be intended to make a large range of products. These issues can influence the requirements' analysis considerably.
- How far up the procedural model should the automation extend? For example, some operators want phases automated but operations and procedures to be manual.
- What level of batch reporting and management information will be required?
- Who operates the plant?
- Where do the operators work?
- What skill levels they will have?

Process Description

This is typically written by a process engineer often without reference to S88
Useful starting point for the analysis
May be superseded by the document that is generated as a result of the requirements analysis
Should provides a basis for Process Recipes

Equipment Protection Requirements

This is typically provided by mechanical equipment suppliers needs.

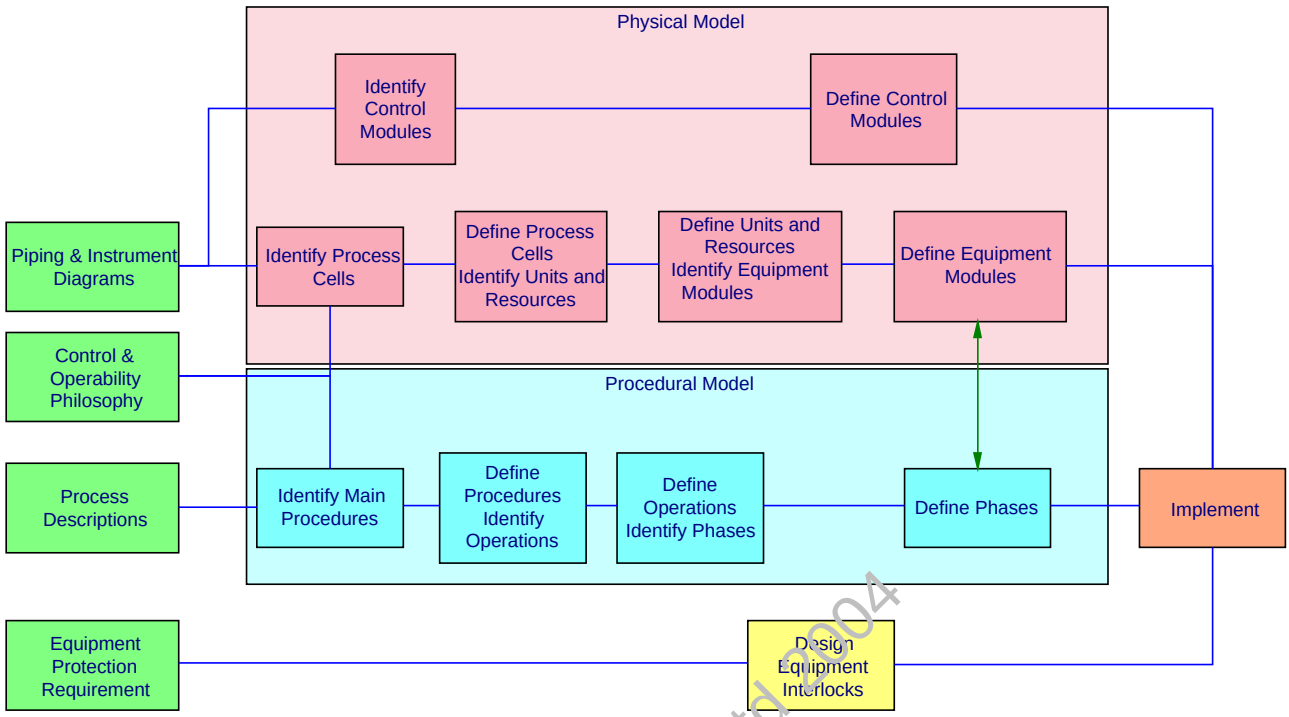
Describe the Physical Model

This involves finding and describing the entities in the physical model, specifically the Units, Common Resources, Equipment modules and control modules

Describe the Procedural Model

This involves finding and describing the entities in the Procedural model, specifically the Recipe, Operations, and Phases

Diagram 2 - Requirements Analysis Process
Diagram Version: 133 Diagram 2 of 43



Description for Diagram 3 - GAMP Life Cycle

The diagram shows a typical life cycle. Note that the ControlDraw modules are relevant throughout the life cycle.

Note also that the waterfall model does not means that All aspect of one stage have to be completed before the next stage can be done. For example the project plan may work through the Units one by one, completing an FRS for one unit even before a P&ID is available for the next. Also, often the Control module level can be completed even before equipment modules and units are completed

Note

For User Requirements Specification Use ControlDraw to define the detailed functions needed for the plant

A Functional Spec can be developed in ControlDraw, and made system specific by developing the classes and detail of the model.

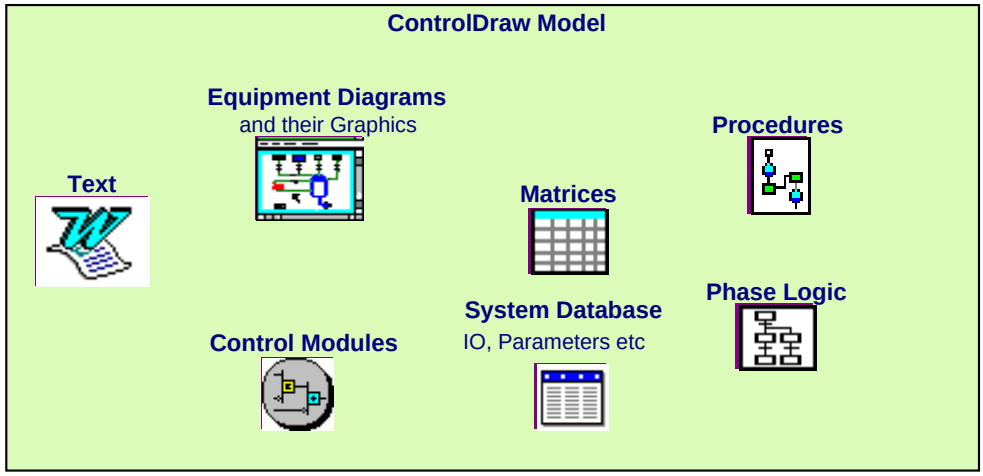
A Software Design Specification can also be developed in ControlDraw

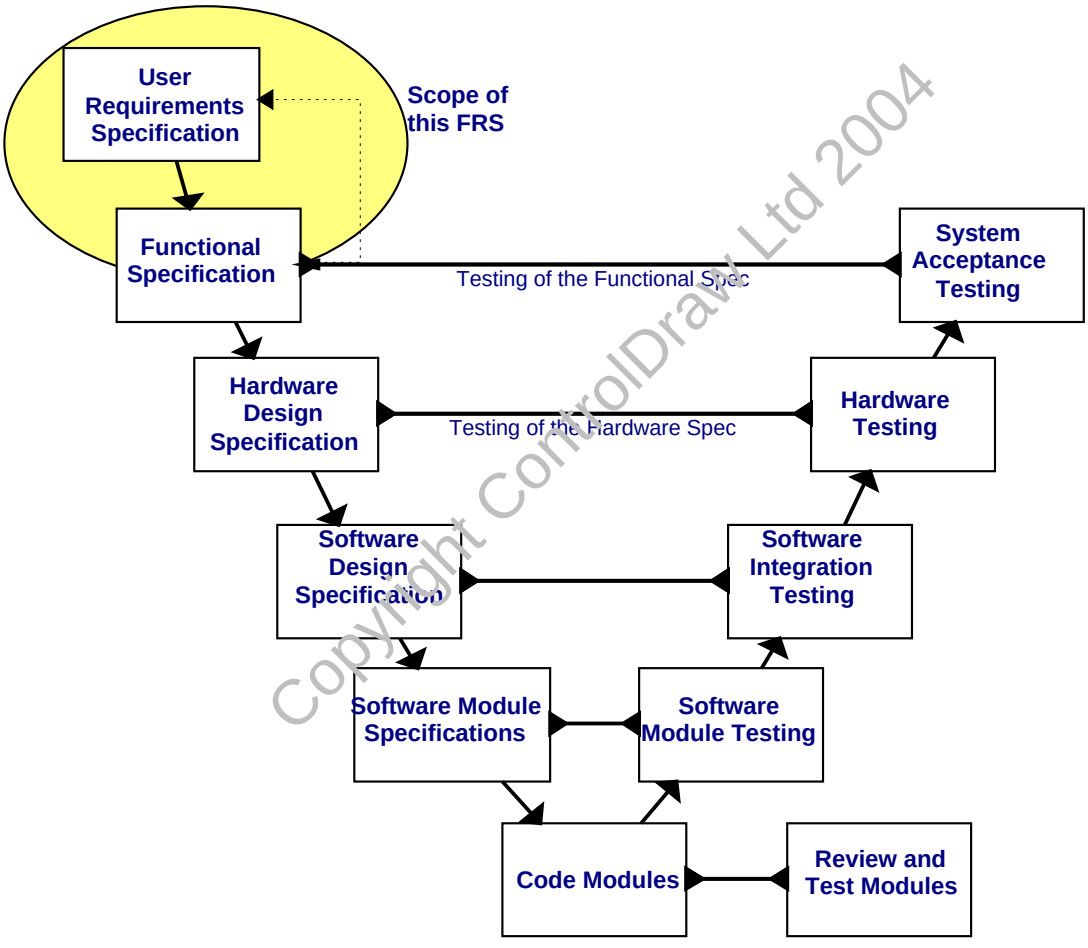
Software Module Specifications are subsets of a ControlDraw Software Design Spec

For Software and Hardware Testing, Test documents can be developed from the ControlDraw models

When Coding, ControlDraw does not write code, but can be used to provide all the data required by the code.

Diagram 3 - GAMP Life Cycle
Diagram Version: 133 Diagram 3 of 43





Description for Diagram 4 - Modelling and Review process

The diagram shows an overview of the process by which the functional requirements are developed.
Typically this process is organised to work through the process cells and the units and into the operations and phases.
Generic Control Modules such as motors and valves are reviewed first. Since these provide much of the manual operation facilities in the plant it is important that the functionality available through
these modules is understood before going into the details of the operations and phases.
Then the reviews move into the actual detail of the various process cells, unit's etc. This can include simple simulations where the valves are opened and closed, motors started etc to show dynamically
the states of the equipment.

The diagram also shows how the model ican be semi-automatically checked against the P&ID's by comparing the IO in the model with that in the instrument indexes derived from the P&ID's

Diagram 4 - Modelling and Review process

Diagram Version: 90 Diagram 4 of 43

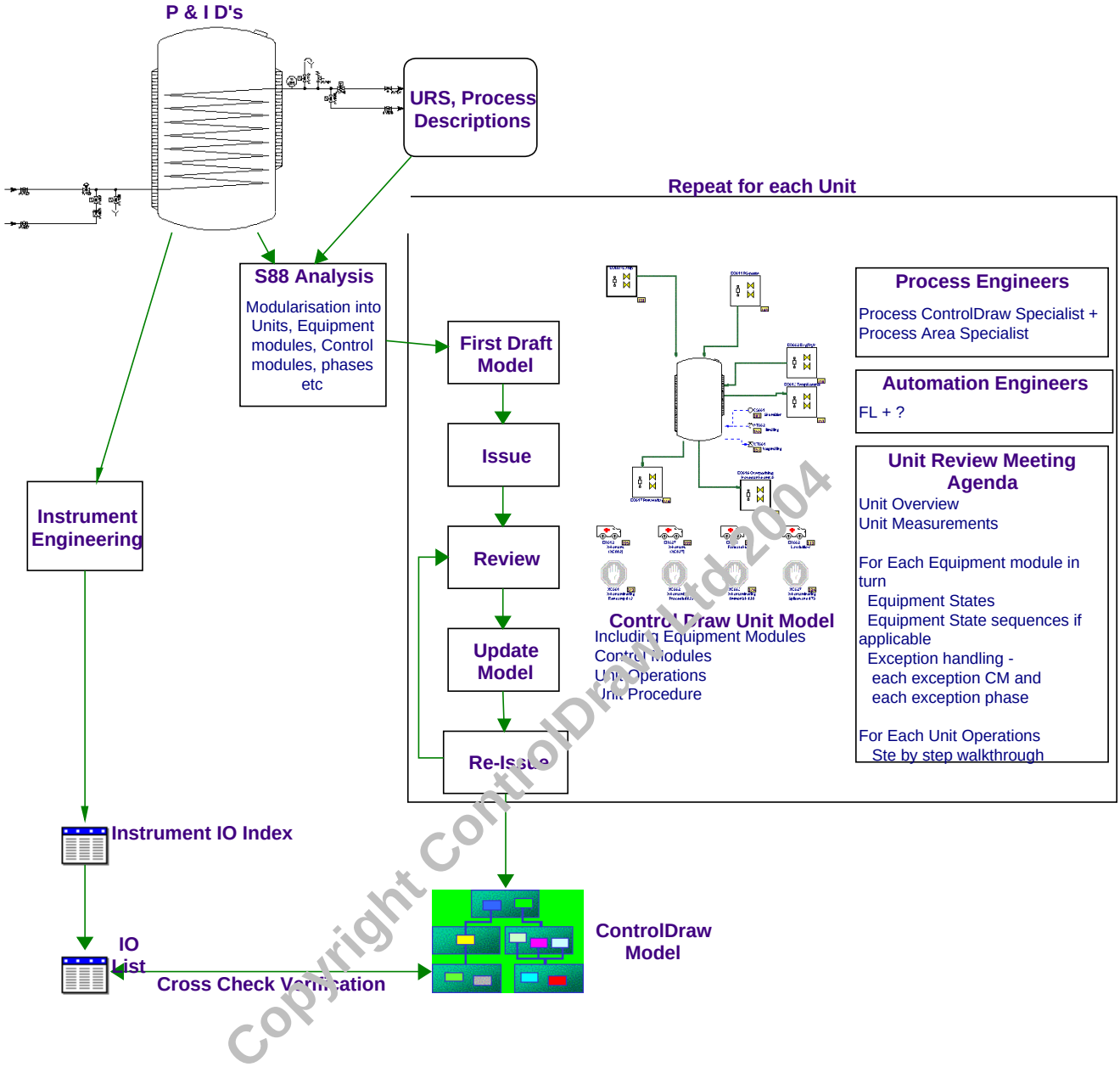
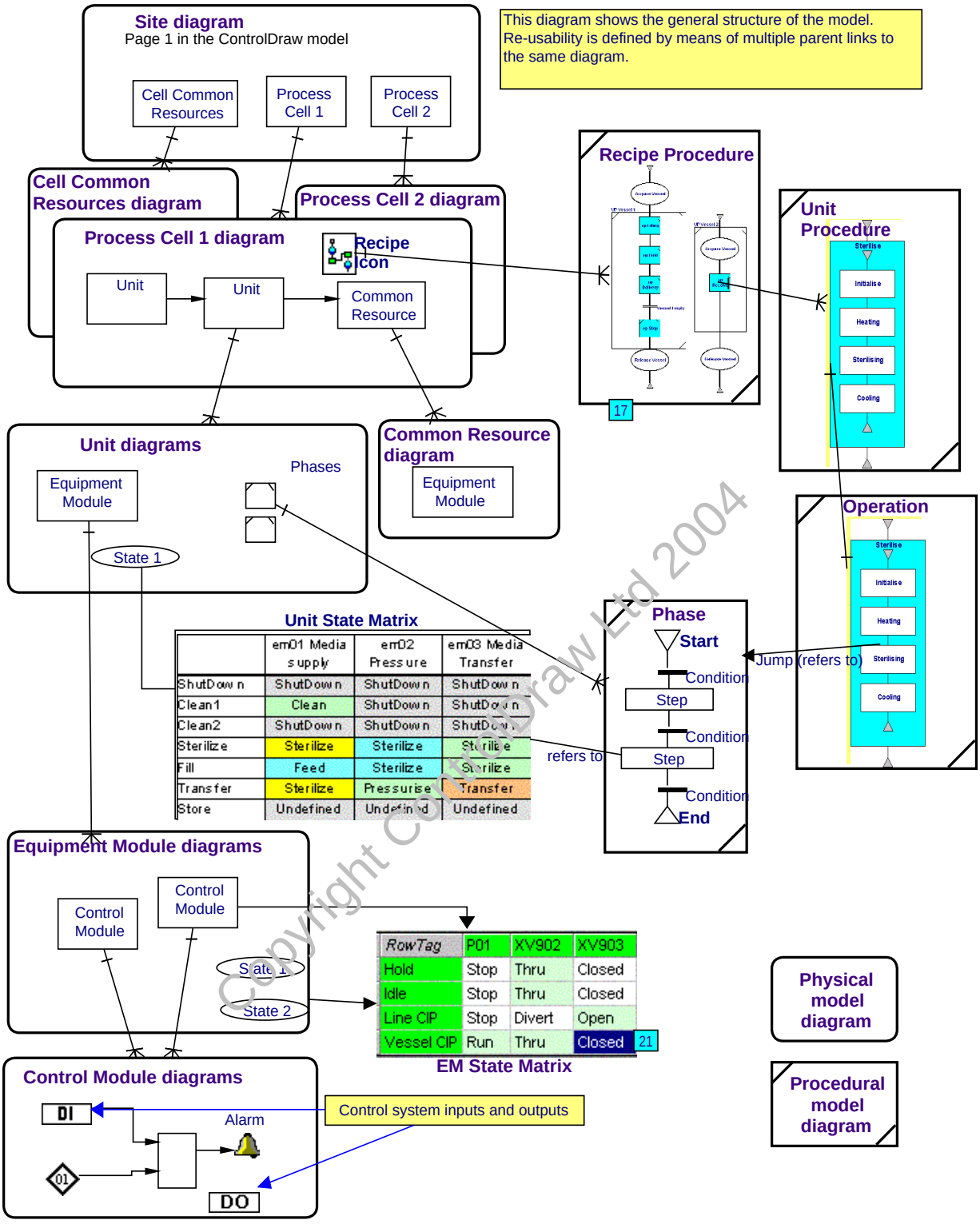


Diagram 5 - Heirarchy Diagram
Diagram Version: 138 Diagram 5 of 43



Description for Diagram 6 - Diagram and Object Classes

All ControlDraw diagrams and symbols are given a class. It is the most important setting.
This identifies the type of information a page or symbol represents and is significant in models, and the database.
The classes are heavily influenced by S88.01

Class is used for:

Determining what data should be stored for the object and instances associated with a symbol
Establishing validation rules for the model
Setting hierarchical tag numbering rules

The Class of an object or page is an important aspect of a model, and so ControlDraw supports many additional features to help to manage the classes.

A toolbar button toggles the display of the classes of each object on the diagrams.
Model rules that check the consistency of diagrams and links are based on the classes.
Sequential Function chart numbering and dynamics are based on the classes

Diagram 6 - Diagram and Object Classes
Diagram Version: 135 Diagram 6 of 43

Summary list of classes

S88	Model Class	Tag	Include
Alarm	Alarm	al	True
Allocation	PFC Acquire / Release	acq	True
Area	Plant Area	pa	True
Batch	Container	ct	False
Batch Journal	Batch Journal Entry	bj	True
	Batch Report	br	True
Common Resource	Transfer Class	tc	True
Control Module	Control Module	cm	True
	Effector Analog	ee	True
	EffectorDiscrete	ed	True
	Interlock Control Module	il	True
	Logic Function	lg	False
	Loop Function	lf	False
	Measurement Analog	me	True
	Measurement Switch	ms	True
	Motor	mtr	True
	PID Control Loop	pid	True
	Software Driver Module	sd	True
	Software Object	so	True
	Valve	vv	True
Control System	Control System	cs	True
	Control System Node	cns	True
Equipment Module	Equipment Module	em	True
Equipment Procedure	Equipment Procedure	er	False
Header	Document Reference	dr	False
None	Action	AC	False
	Datastore	ds	False
	Entity	en	False
	Equipment	eq	False
	Interface	if	True
	MemoryDiscrete	bln	False
	MemoryInteger	int	False
	MemoryReal	sng	False
	MemoryString	str	False
	None	none	False
	Operator Command	co	False
	Operator Display	od	False
	Operator Entry	oe	False
	Person	pe	False
	Transfer Segment	ts	True
	Transition	tr	False
Operation	Operation	op	True
Phase	EM State change phase	emst	False
	EMSubState	emsb	False
	Phase	ph	True
Process Action	Phase/Op Step	ps	False
Process Cell	Cell Common Resource	cc	True
	Process Cell	pc	True
Process In-Out	Process Material	pm	False
Process Parameter	Equipment Parameter	ep	True
Process parameter	Recipe Formula Value	rf	True
Recipe	Recipe	re	True
Recipe Procedure	Recipe Procedure	rp	True
Site	Site	si	True
State	State	st	False
Unit	Common Resource	cr	True
	Unit	un	True
Unit Procedure	Unit Procedure	up	True
Wiring	Wiring	wr	True

Diagram 7 - S88 Cross Reference

Diagram Version: 125 **Diagram 7 of 43**

Note

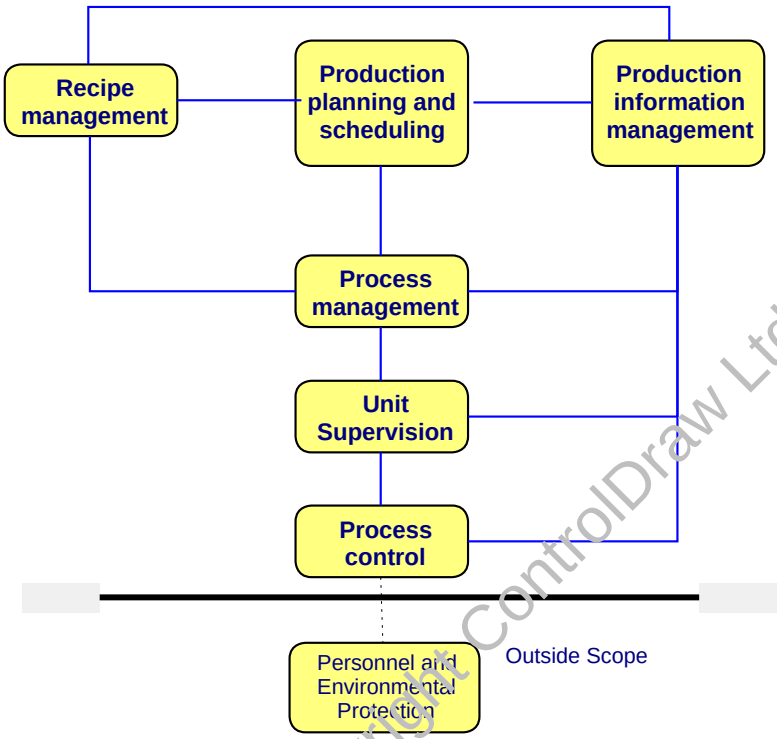
Only covers those definitions that are not specifically covered elsewhere

	Allocation	The models show the allocation requirements by means of the Aquire objects in PFC diagrams
	Arbitration	Sometimes you may just assume standard functions in a batch manager, other times you may want to design some Arbitration logic. Such logic could appear as a logic control function in a common resource
39	**Batch**	Requirements defined by the models
	Batch control	Requirements defined by the models
	Batch process	Requirements defined by the models
	Batch schedule	Not addressed in the version
8	**Control Activity model**	See diagram below
	Coordination control	Requirements defined by the models
	Equipment entity	All the physical model diagrams
	Exclusive-use resource	Requirements defined by common resourcee diagrams
	Formula	Objects on phase diagrams
	Header	Possible to define in the recipe diagrams
5	**Line; train**	Plant Area and Process Cell Diagrams
39	**Lot**	Requirements defined by the models
22	**Mode**	See Modes and States Overview
5	**Path; stream**	Plant Area and Process Cell Diagrams
	Person & environment protection	Not defined by the models, although low level equipment protection may be
5	**Procedural element**	Steps in phase diagrams, phases in operation diagrams etc
5	**Procedure**	Recipe and Unit procedure diagrams
	Process input	Process Material objects on diagrams
	Process output	Process Material objects on diagrams
10	**Process model**	See Process model Object Mapping
17	**Recipe**	Recipe objects and child pages
11	**Recipe - Equipment Control links**	Shown by means of the links between objects in the procedural and in the physical models.
9	**Recipe Types**	Master recipes and Site recipes are defined by the Recipes in the Procedural model.
	Shared-use resource	Requirements defined by common resourcee diagrams
5	**Stream; path**	Plant Area and Process Cell Diagrams
5	**Train; line**	Plant Area and Process Cell Diagrams
	Unit recipe	Derived by filtering the site recipes

Description for Diagram 8 - Control Activity model

This shows the S88 Control Activity model.
As far as the ControlDraw models go, there are not in general specific objects the relate directly to this.
Defining Process control requirements is the purpose of the ControlDraw model.
Process management is an area that is a function of the Control System in real time, however the ControlDraw model implies much of this.
Recipe management is similar
General requirements for Unit Supervision are implied in the Unit State control, and specific requirements may be identified in some models
Production planning and scheduling is out of scope
Production information management is implicit in the batch logging data in a model.

Diagram 8 - Control Activity model
Diagram Version: 123 Diagram 8 of 43



Description for Diagram 9 - Recipe Types

This shows the S88 Recipe Types
Control recipes are a function of the Control System in real time,
General recipes out of scope
Master recipes and Site recipes are defined by the Recipes in the Procedural model.

Diagram 9 - Recipe Types
Diagram Version: 123 **Diagram 9 of 43**

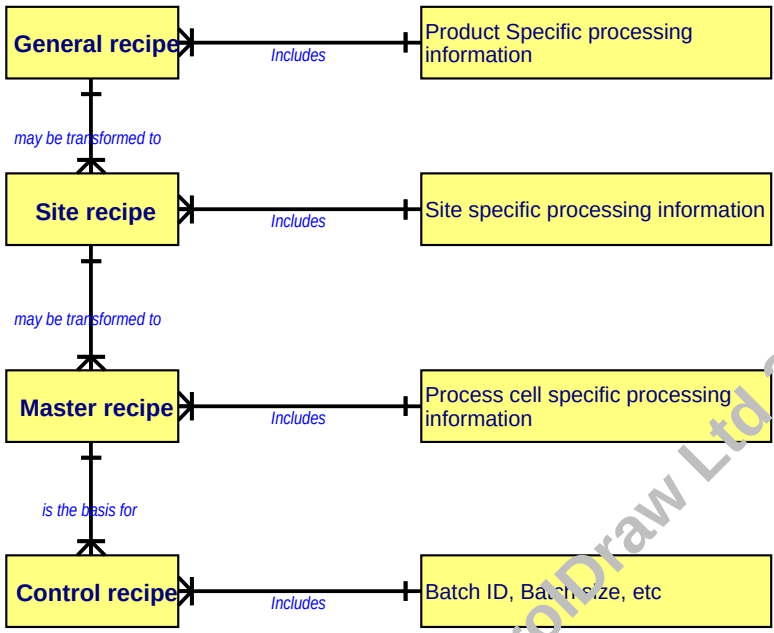
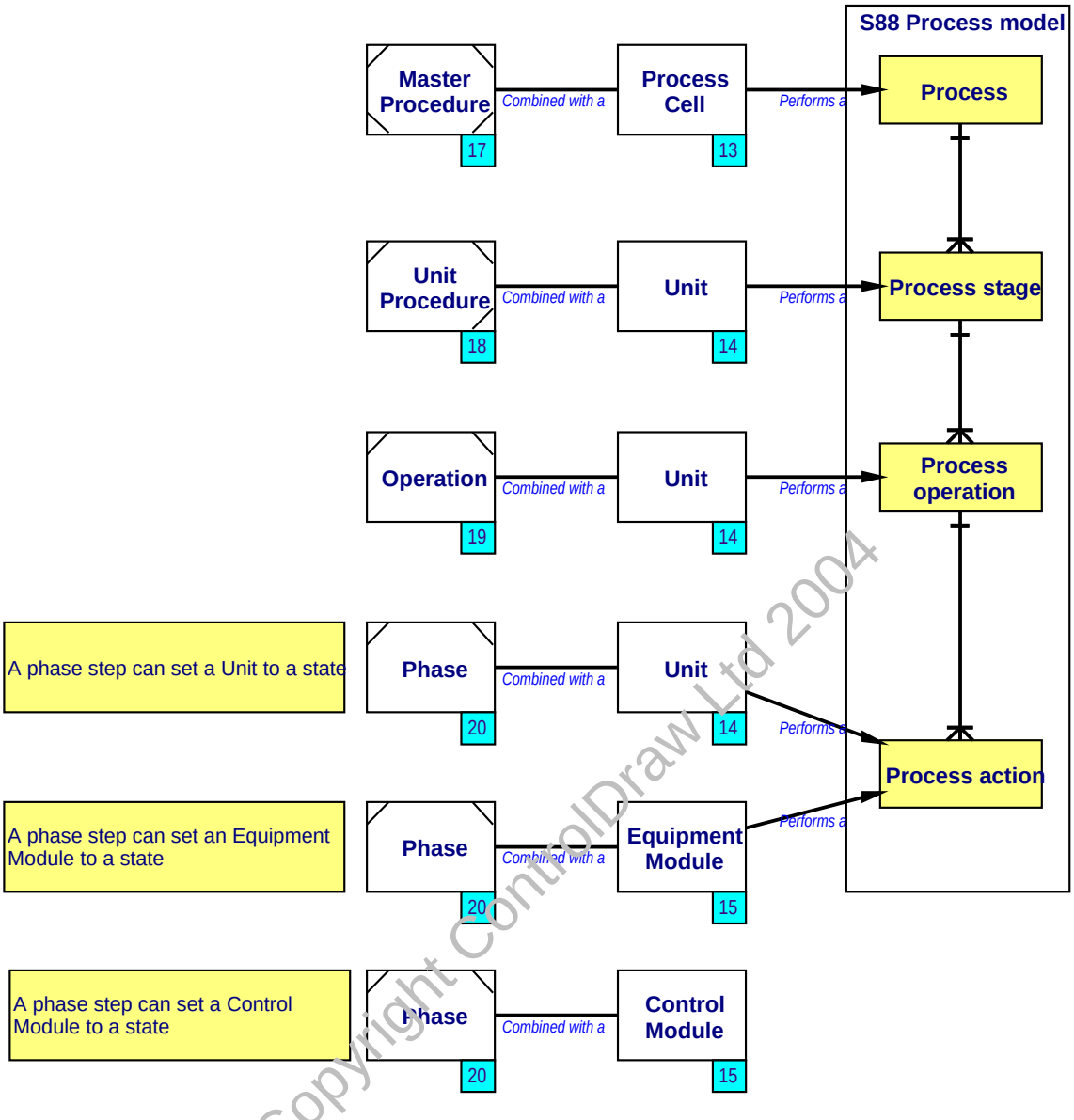


Diagram 10 - Process model Object Mapping
Diagram Version: 133 Diagram 10 of 43



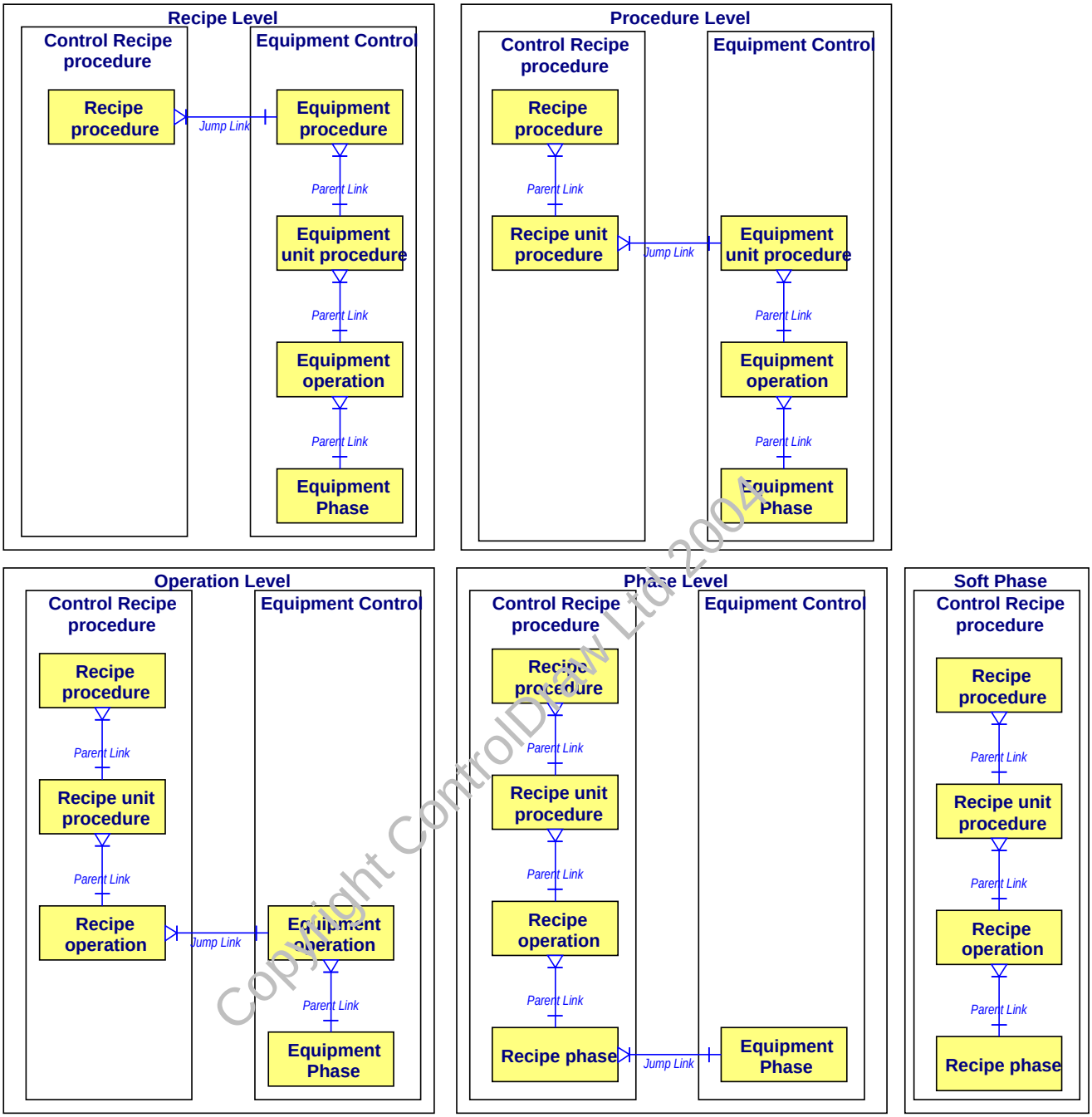
Description for Diagram 11 - Recipe - Equipment Control links

This diagram, derived from the S88.01 standard, shows each possible configuration of the relationship between Procedural and Physical control.

ControlDraw models can show this mapping by means of the links between objects in the procedural and in the physical models.

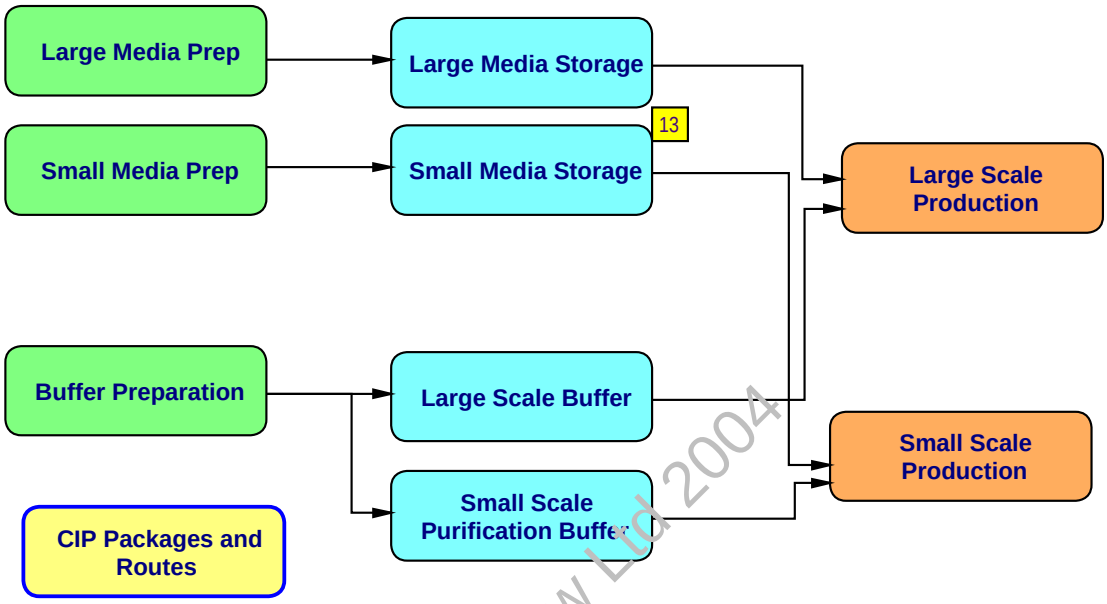
Note - it is entirely possible to have more than one of the levels operating at the same time in the same control system, for example CIP may work at the Recipe unit procedure - Equipment unit procedure level whilst production phases work at the Equipment phase - Recipe phase level.

Diagram 11 - Recipe - Equipment Control links
Diagram Version: 122 Diagram 11 of 43

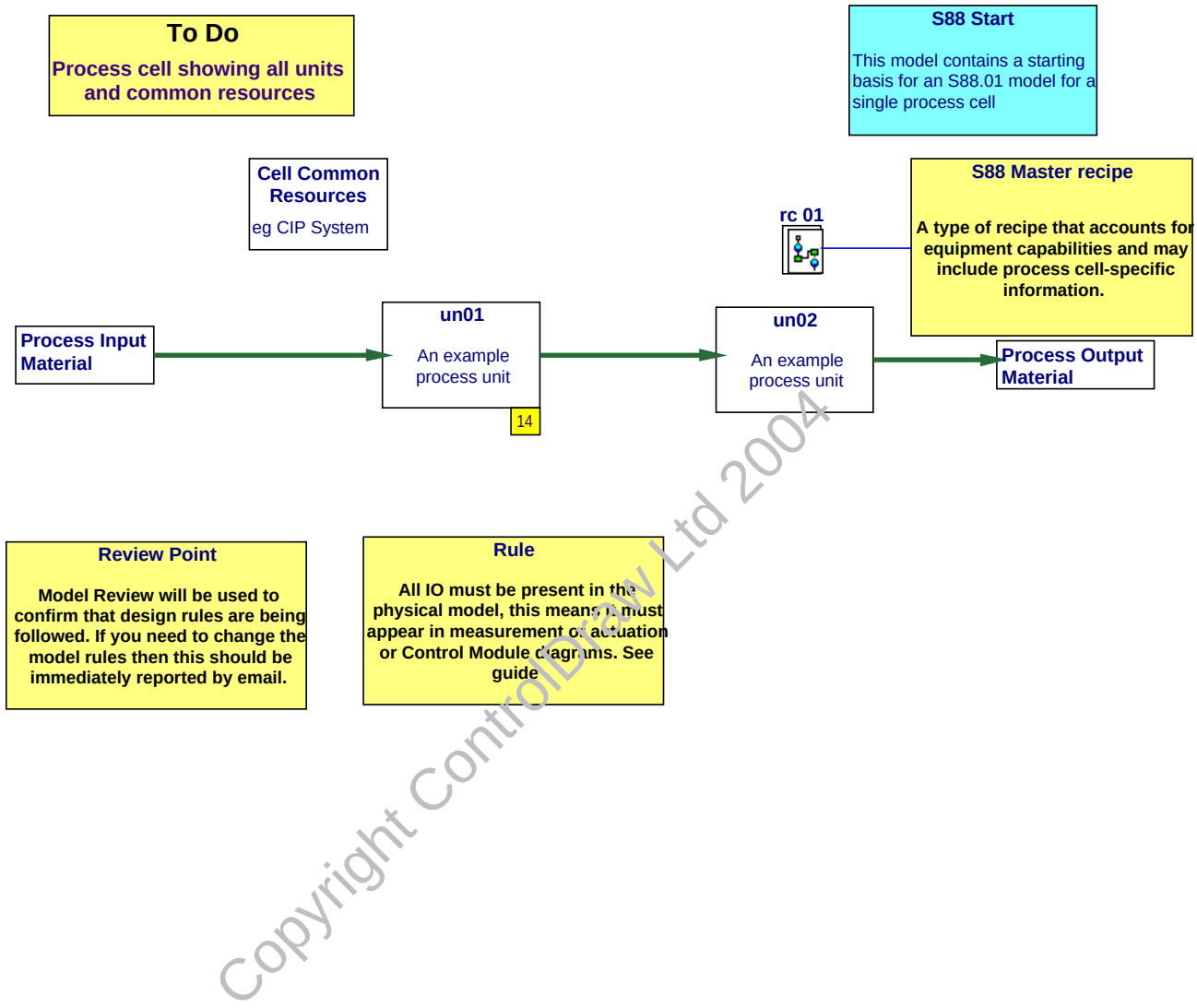


Description for Diagram 12 - Area
This diagram is an example of an Area diagram, and contains all the process cells in the area.

Diagram 12 - Area
Diagram Version: 125 **Diagram 12 of 43**



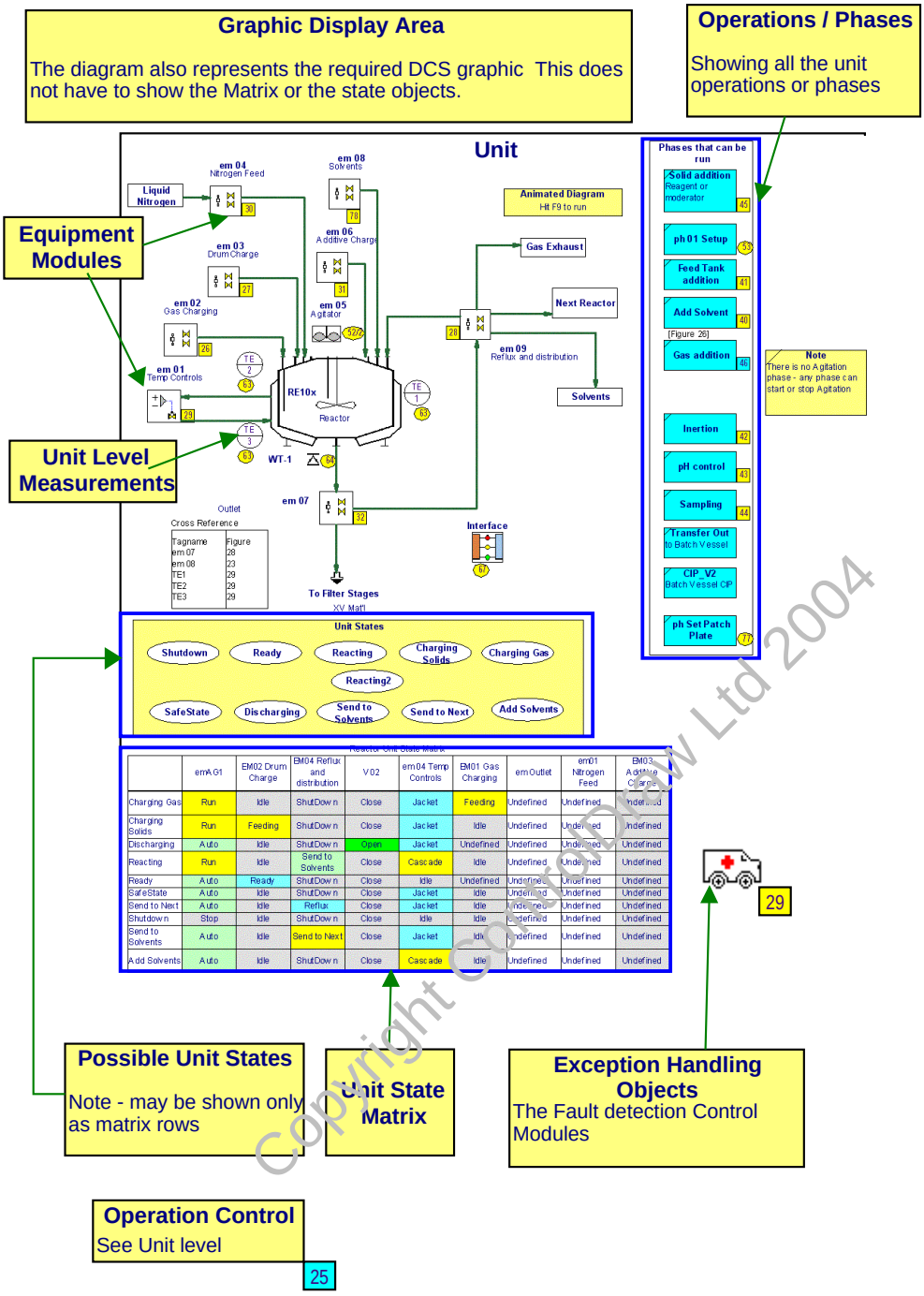
Description for Diagram 13 - Example Process Cell
An example of a Process Cell Diagram
Diagram 13 - Example Process Cell
Diagram Version: 133 **Diagram 13 of 43**



Description for Diagram 14 - Storage vessel Unit Diagram

Example of the make up of a Unit Diagram inthe Model
The Unit diagram includes all the analog measurement Control Modules and the Equipment modules.
Those Control Modules that have physical outputs such as valves and motors are shown in the EM diagrams.
Some very simple units or cr's may only contain one equipment module, in which case it is called EM01 Main.

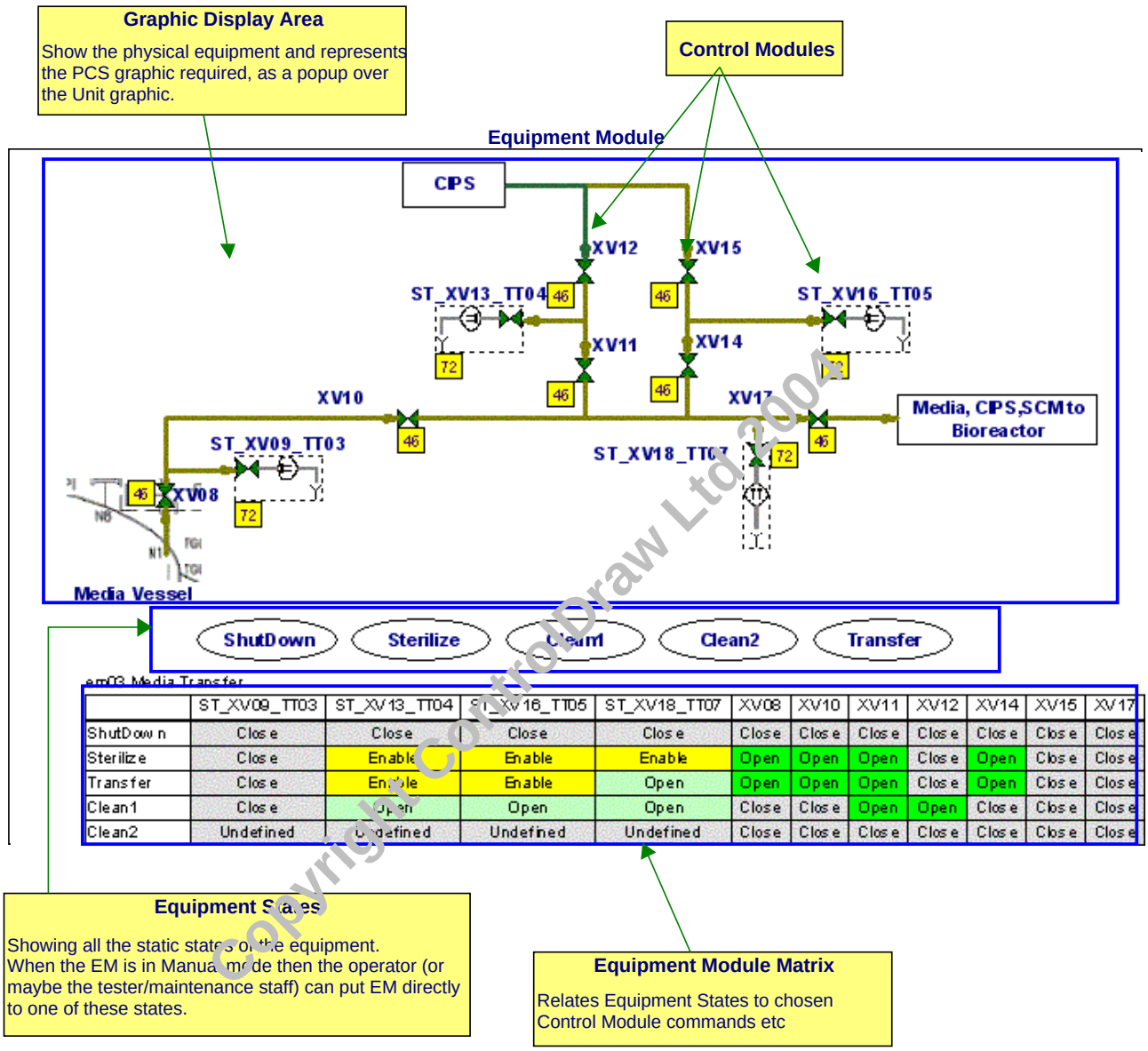
Diagram 14 - Storage vessel Unit Diagram
Diagram Version: 112 Diagram 14 of 43



Description for Diagram 15 - Equipment Module Diagram

Example of an equipment module diagram.
This diagram includes all the Control Modules that have physical outputs such as valves and motors.
Analog measurements that are shown in the unit but required in the EM can be repeated on the diagrams, but with a class of none.

Diagram 15 - Equipment Module Diagram
Diagram Version: 112 Diagram 15 of 43



	ST_XV09_TT03	ST_XV13_TT04	ST_XV16_TT05	ST_XV18_TT07	XV08	XV10	XV11	XV12	XV14	XV15	XV17
ShutDown	Close	Close	Close	Close	Close	Close	Close	Close	Close	Close	Close
Sterilize	Close	Enable	Enable	Enable	Open	Open	Open	Close	Open	Close	Close
Transfer	Close	Enable	Enable	Open	Open	Open	Open	Close	Open	Close	Close
Clean1	Close	Open	Open	Open	Close	Close	Open	Open	Close	Close	Close
Clean2	Undefined	Undefined	Undefined	Undefined	Close	Close	Close	Close	Close	Close	Close

Diagram 16 - Standard Control Modules
Diagram Version: 139 **Diagram 16 of 43**

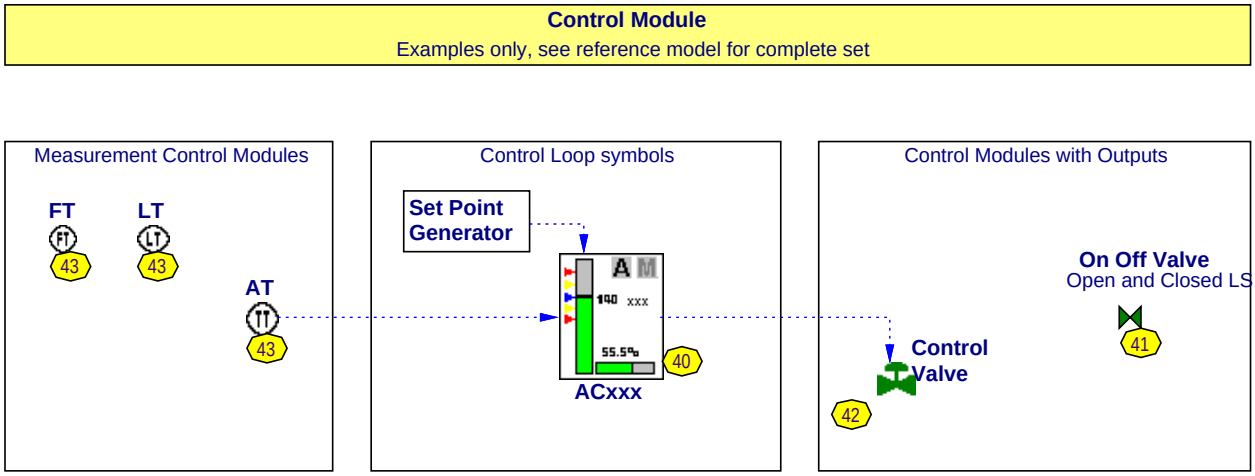
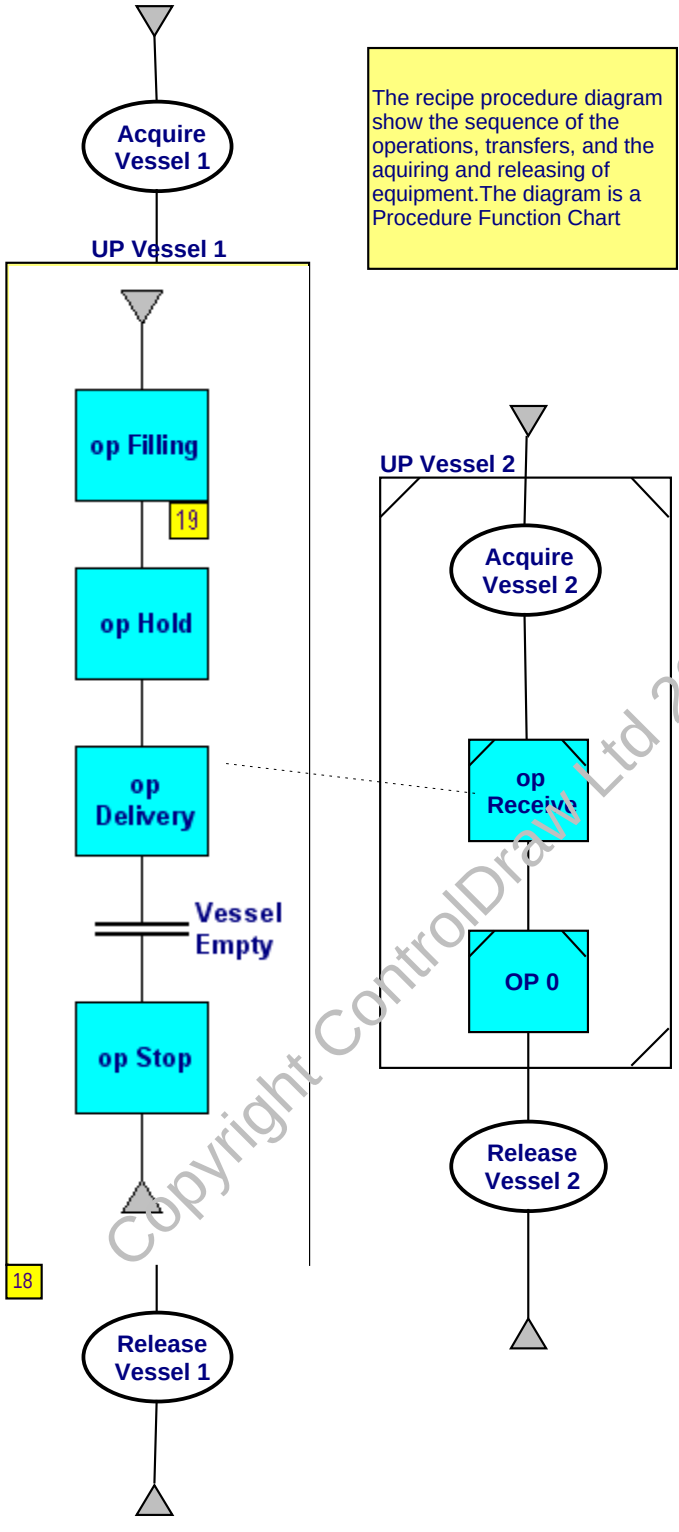


Diagram 17 - Recipe Structure
Diagram Version: 131 **Diagram 17 of 43**

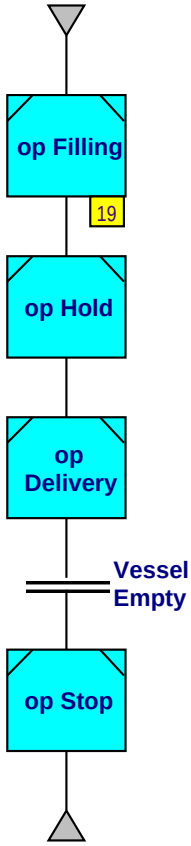


Description for Diagram 18 - UP

A unit procedure works by sequencing the execution of phases, the unit procedure diagram is a Procedure Function Chart.

Diagram 18 - UP

Diagram Version: 129 **Diagram 18 of 43**



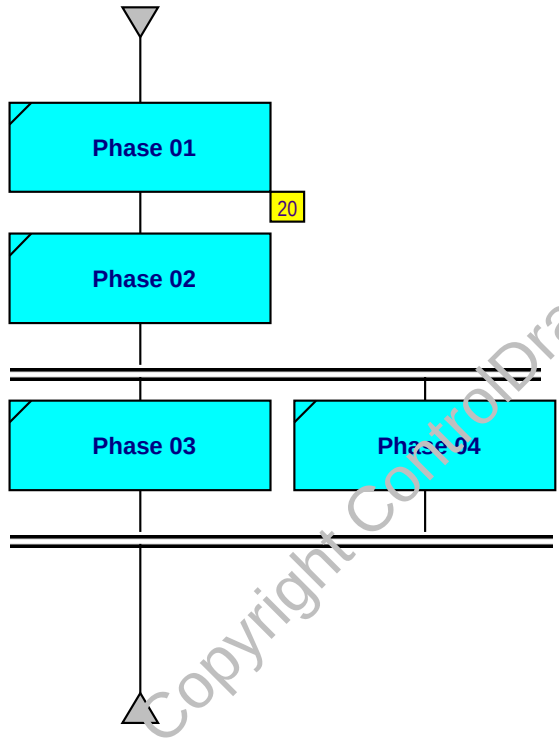
Description for Diagram 19 - Operation
A typical operation works by sequencing the execution of phases, the operation diagram being a Procedure Function Chart.

Generic Operation Timer

Each operation will have a time setting for each step or phase (in the S88 sense) and an alarm will be generated in the event that a step takes longer than it's allowed time.

Normally this will be the only action taken. Exceptionally (perhaps for example when a particular filling step is taking too long) it may be defined that the Timer will also put the Unit to a hold state when it times out. These exceptions will be explicitly identified in the detailed design.

Diagram 19 - Operation
Diagram Version: 130 **Diagram 19 of 43**



Description for Diagram 20 - ph01 Fill

A typical phase works by setting the Unit to states and monitoring process values or equipment states to determine when to move to the next step.
Recipe Formula values are shown on the operation/Phase diagrams.
The model also generates a database table of all these so that the values can be managed.

Steps in the SFC can include:

- Equipment State commands, in order to put the Equipment Module to a particular equipment state
- Messages and prompts to the operator
- Starting a timer
- Control Module commands to put a Control Module to a particular state

. .

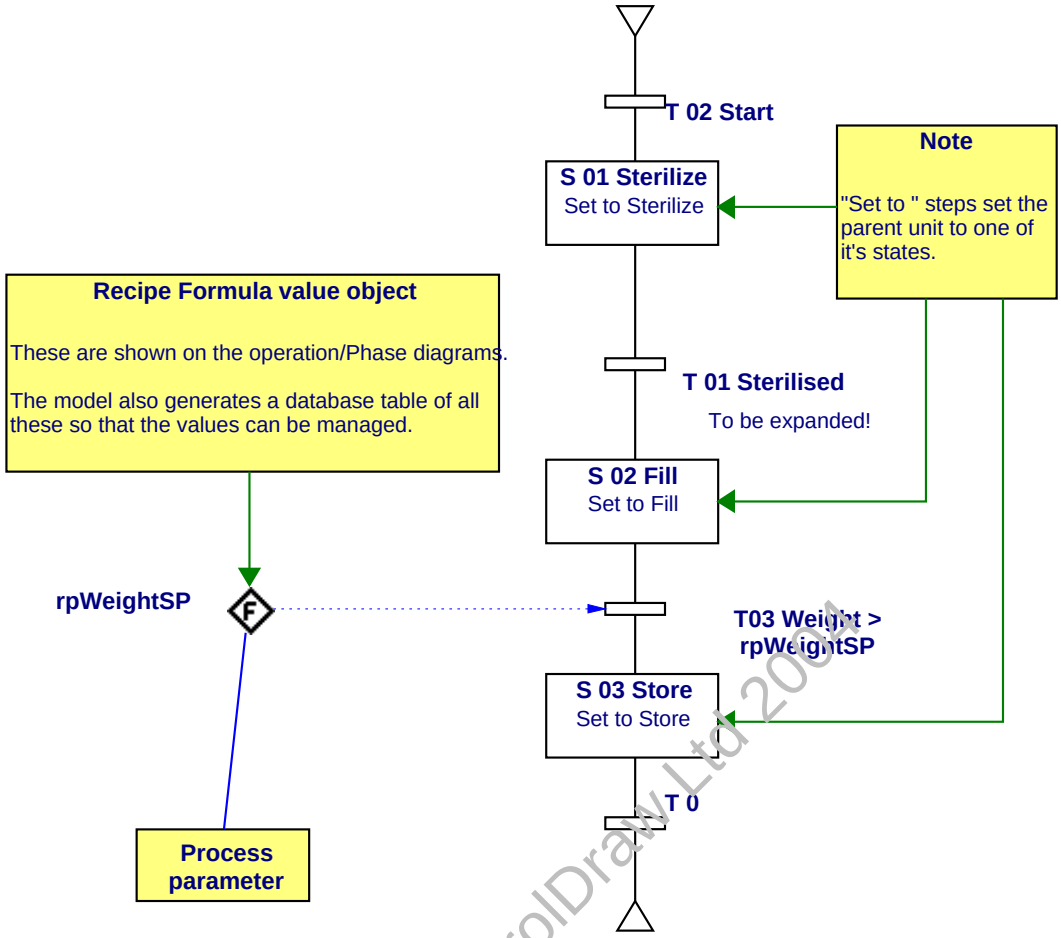
The transitions that end each step and start the next include:

- Operator Confirmation or selection
- Step Timer timed out
- Achieving a process Condition.

. .

Note also that parallel Steps are not generally used except where one branch is concerned with a dialog with the operator and the other is controlling equipment.

Diagram 20 - ph01 Fill
Diagram Version: 80 **Diagram 20 of 43**



Description for Diagram 21 - Matrix Functions

The modelling approach deploys State Matrices, for example for Units and Equipment modules.
Typically the Columns are Control Modules on an Equipment module page and the Rows are the states on the page, the diagram below illustrates this.
Typically, Matrices are used to map the procedural elements in a model into the physical elements.
For example, a step in a phase can set an equipment module to a specific state, so that step then causes all the devices in the em to go to the setting defined in the EM State matrix.

Diagram 21 - Matrix Functions

Diagram Version: 112 **Diagram 21 of 43**

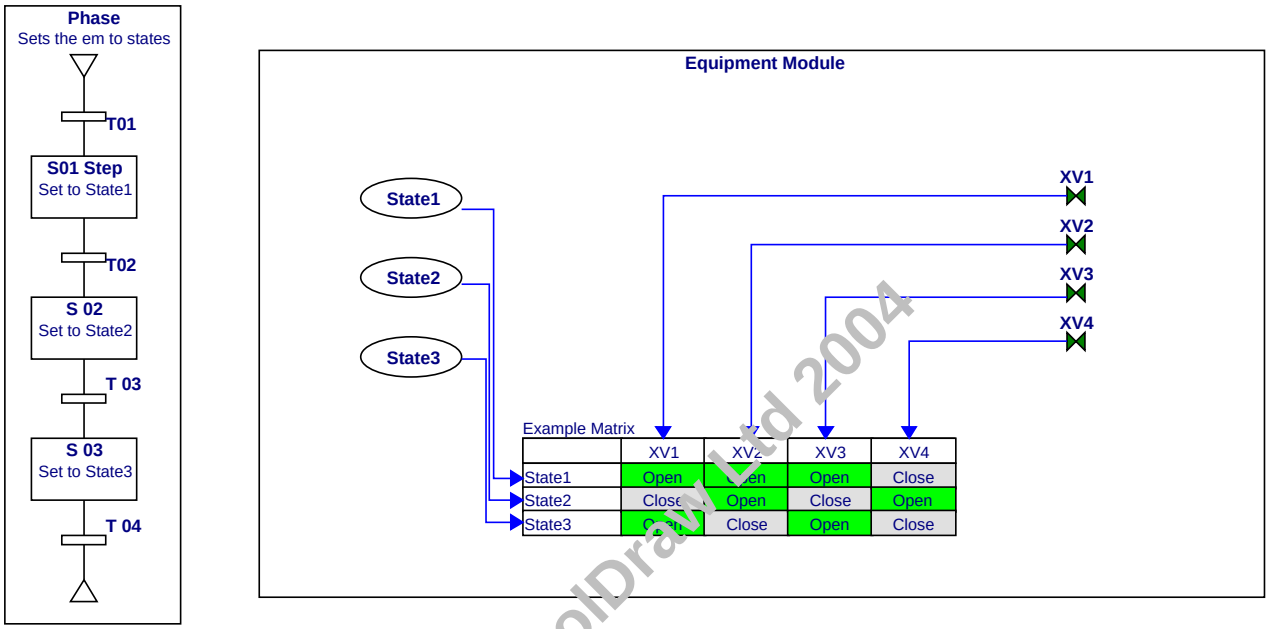
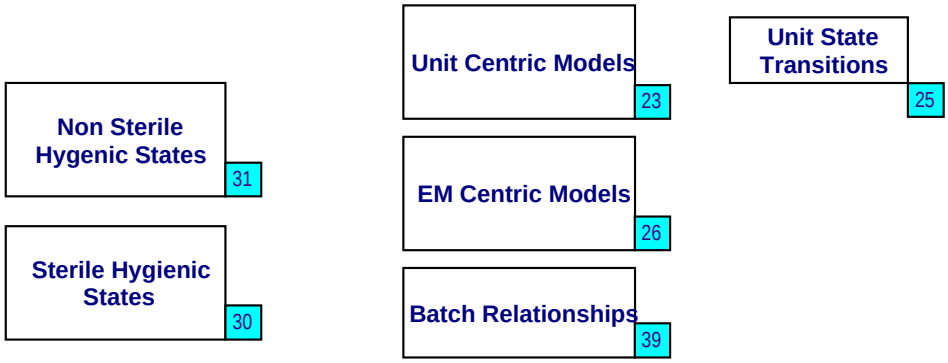
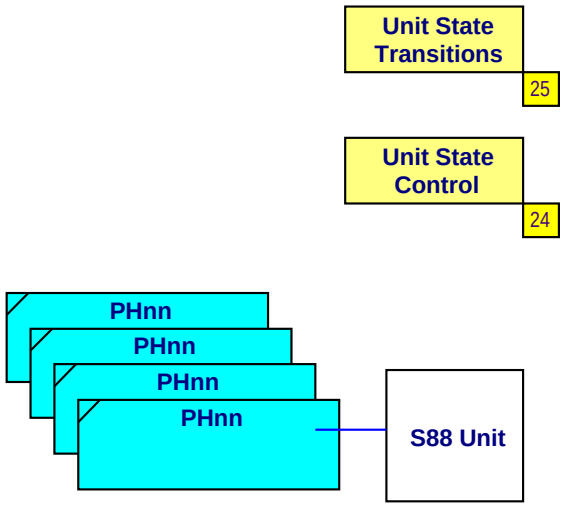


Diagram 22 - Modes and States Overview
Diagram Version: 124 **Diagram 22 of 43**



Description for Diagram 23 - Unit Centric Models
In a unit centric model
Only one phase can run at one time in the unit.
The unit is completely in charge of the and controls what is happening in each equipment module
Diagram 23 - Unit Centric Models
Diagram Version: 80 **Diagram 23 of 43**



Description for Diagram 24 - Unit State Control

This diagram indicates the relationship between Unit Operations or Phases and Sequences and the equipment module. Only one operation can act on the unit at one time. As the operation executes, each step (ie a step within a Phase within the operation) can command the unit to a state. The unit then sets it's subordinate equipment modules into appropriate equipment states as shown in the Unit State matrix.
Note - Units could directly set Control Modules, in cases where the CM's do not need to be in Equipment modules, however in order to provide consistency the model will always include equipment modules between the Unit and the Control Module levels.
Note also that the Unit state matrix may include a Hold State column. This is optional, and is only required when a unit has different hold states for different conditions.

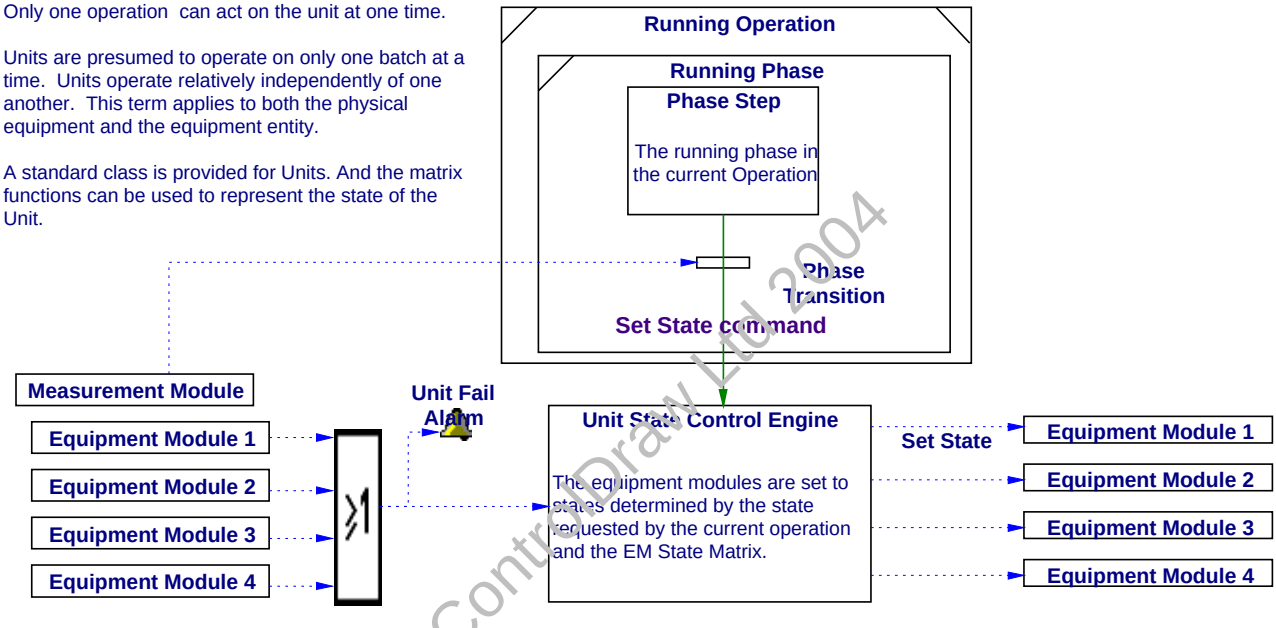
Diagram 24 - Unit State Control

Diagram Version: 80 **Diagram 24 of 43**

This diagram indicates the relationship between Unit Operations and Sequences and the Unit.
Only one operation can act on the unit at one time.

Units are presumed to operate on only one batch at a time. Units operate relatively independently of one another. This term applies to both the physical equipment and the equipment entity.

A standard class is provided for Units. And the matrix functions can be used to represent the state of the Unit.



Unit State Matrix

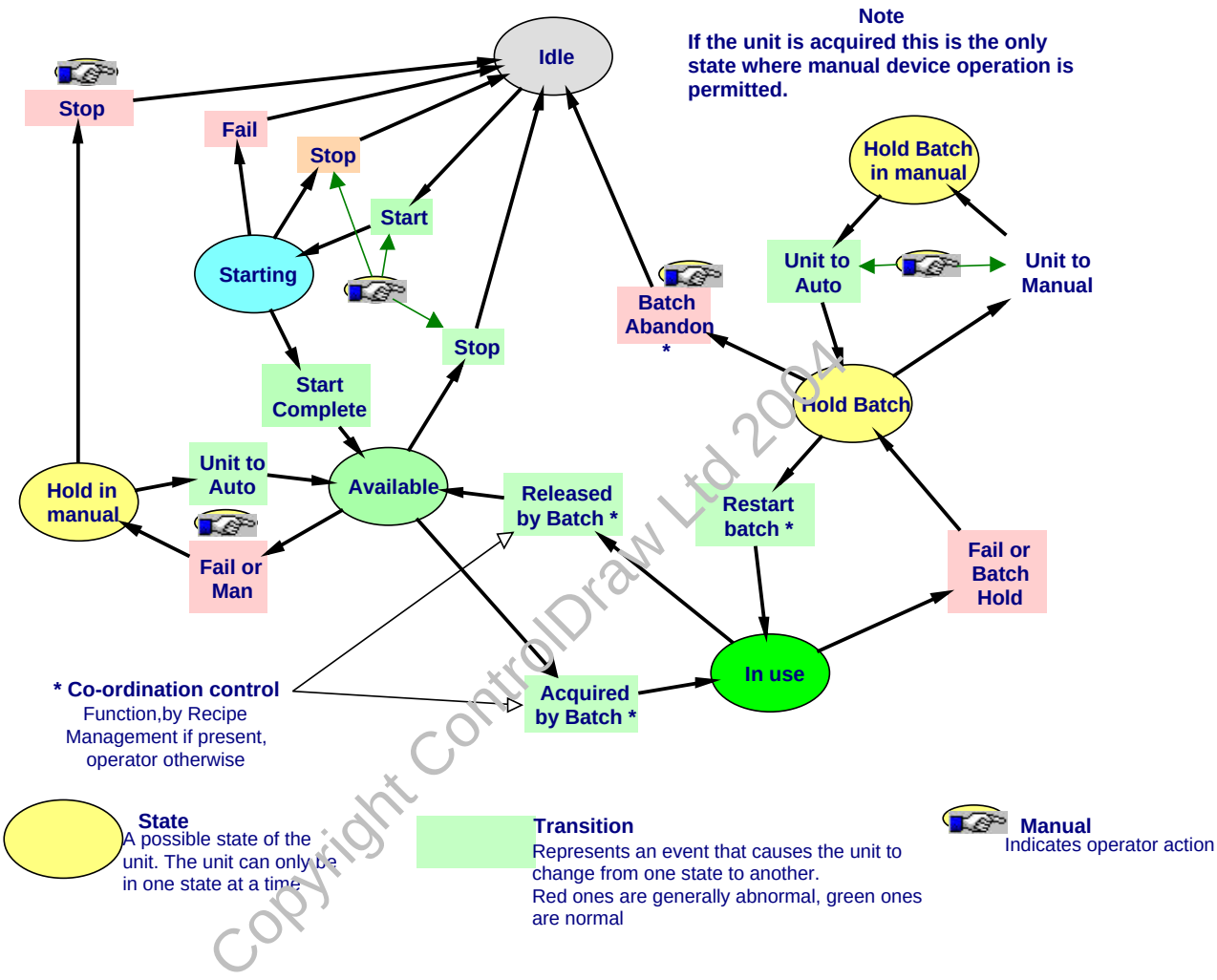
	em01 Media supply	em02 Pressure	em03 Media Transfer
ShutDown	ShutDown	ShutDown	ShutDown
Clean1	Clean	ShutDown	ShutDown
Clean2	ShutDown	ShutDown	ShutDown
Sterilize	Sterilize	Sterilize	Sterilize
Fill	Feed	Sterilize	Sterilize
Transfer	Sterilize	Pressurise	Transfer
Store	Undefined	Undefined	Undefined

Description for Diagram 25 - Unit State Transitions

This diagram shows the assumed state behaviour of units and batches.
This is not intended as a system design, it just shows the assumed functionality that has been used in the requirements analysis.

It may be that the supplier has a different model, for example with phases only acting within an equipment module.
It is entirely accepted that the SI can transform this model to other viewpoints, for example, the phases in em's structure, or any other, provided this retains all the information contained in the ControlDraw model.

Diagram 25 - Unit State Transitions
Diagram Version: 49 **Diagram 25 of 43**

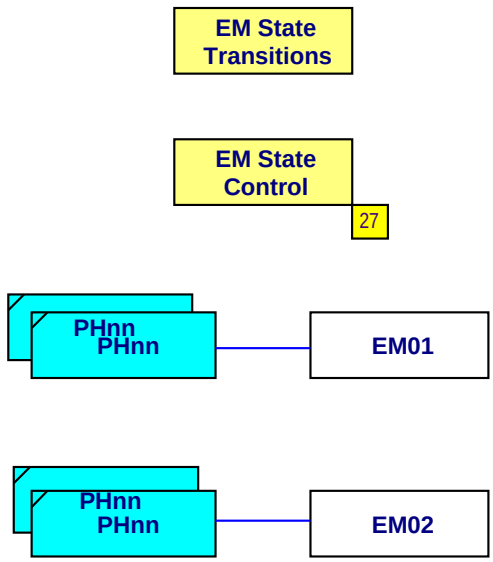


Note
You can only go from a state to the next state thru a transition if the starting state is active and the transition True.

Description for Diagram 26 - EM Centric Models

In an EM centric model
One phase can run at one time in the each EM.
The unit may not know what is happening in each equipment module.

Diagram 26 - EM Centric Models
Diagram Version: 0 **Diagram 26 of 43**

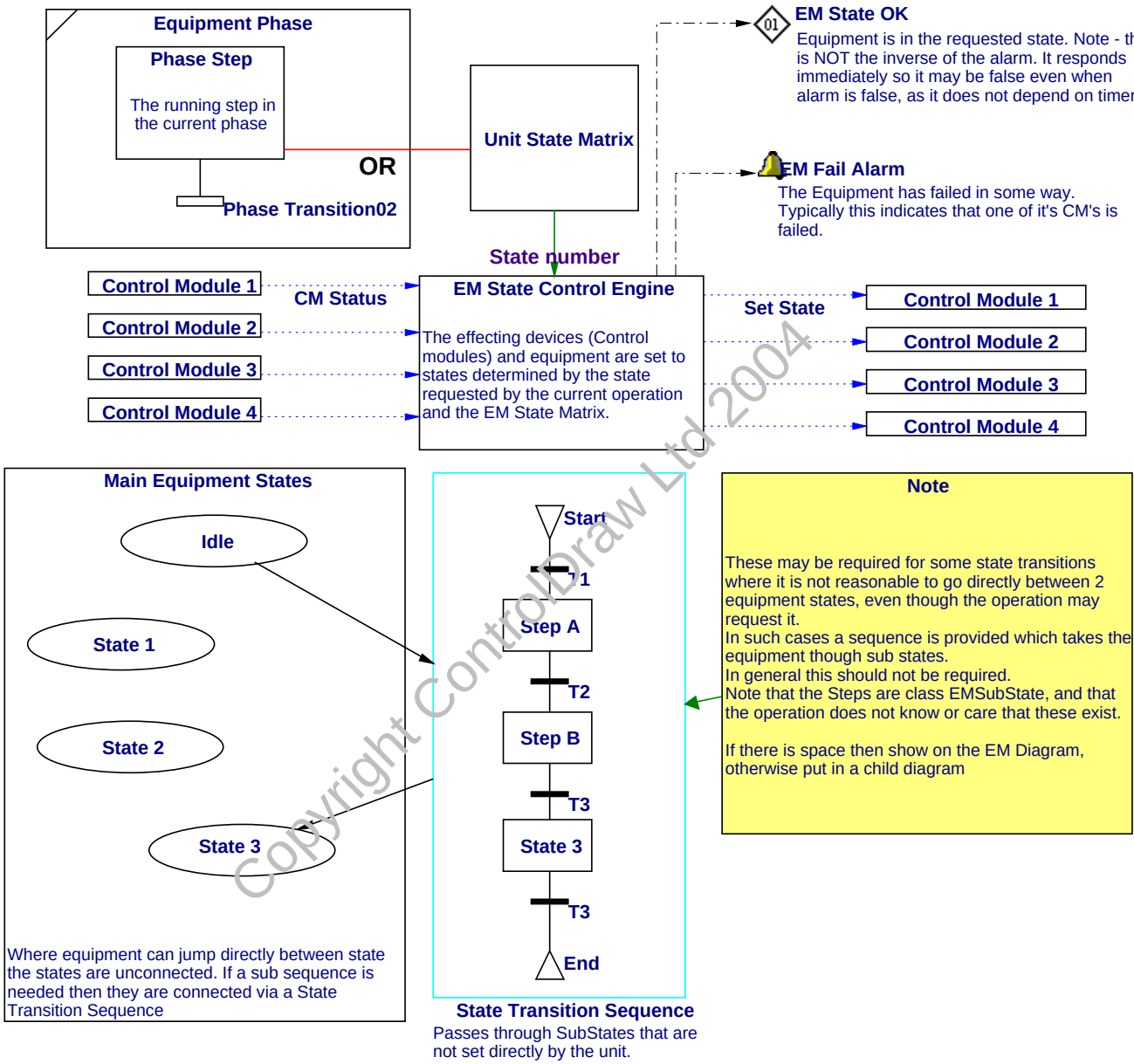


Description for Diagram 27 - EM State Control

This diagram indicates how the equipment module sets it's subordinate control modules into appropriate equipment states as shown in the Equipment State matrix.
Note - Units could directly set Control Modules, in cases where the CM's do not need to be in Equipment modules, however in order to provide consistency the model will always include equipment modules between the Unit and the Control Module levels.

Diagram 27 - EM State Control

Diagram Version: 79 **Diagram 27 of 43**



Description for Diagram 28 - Exception Handling

Exception handling

Exception handling per S88

Those functions that deal with plant or process contingencies and other events which occur outside the normal or desired behavior of batch control.
S88 also says
"An event which occurs outside the normal or desired behavior of batch control is commonly called an exception. Handling of these exceptions can occur at all of the levels in the control activity model and may be part of procedural, basic, and coordination control."

Those functions that deal with plant or process contingencies and other events which occur outside the normal or desired behavior of batch control.
S88 also says
"An event which occurs outside the normal or desired behavior of batch control is commonly called an exception. Handling of these exceptions can occur at all of the levels in the control activity model and may be part of procedural, basic, and coordination control."

Credit to Tom Fisher and the WBF for portions of this text.

An exception is an event that occurs outside the normal or desired behavior.
Can occur at all of the levels in the control activity model. May be part of procedural, basic, and coordination control
Implement Exception Handling in Equipment Control
Implement all exception handling at the unit level
Implement directly in phase logic
Implement in phase logic using states
Implement in control actions

Exception Handling - Location
Implement exception handling that changes infrequently in equipment control
Implement exception handling that changes frequently in the recipe
Many product-related exceptions can be handled in equipment control using formula parameters
Exception Handling In Recipe
Keep as simple as possible
Implement only at recipe phase level if possible

Exception logic activated and managed by the Unit Supervision control activity
States for unit must be clearly defined
Unit supervision executes the exception routines
Exception handling is probably best done with states
User must define what states are needed
User must define conditions that cause state transitions

Functional Requirements for Exceptions

The Approach provided in the Functional Requirements is described below

Basis

Keep it simple
Define a standard Unit/EM Hold mechanism, based on the type of fault in each type of Unit or EM.
By default rely on operators to decide what action to take when exceptions occur.
For specific exceptions take automatic actions such as putting a Unit to hold.
Place the control logic for exception handling within the Physical model, not the procedural as far as possible

Exceptions are detected and acted on within the boundaries of a Unit. An exception within one unit will not cause an exception handling response within another unit. Note that another unit may well react to an upstream or downstream unit's exception, but by normal control. Typically this would be done via the 'Active' signals in the transfer panels that connect the units.
Some exceptions may be handled purely within a Control Module, some in the Equipment Module and some at a Unit level.

In general if an exception causes a hold of any sort (such as within a specific equipment model or at the Unit level) and this causes the processing to suspend, then the phase that was running holds but can be resumed from where it was. If this rule does not apply to a specific case then that is shown in the ControlDraw model.

The components of exception handling are Event Detection, which is concerned with determining that an exception has occurred, and Actions to take in that event.
Specific exception handling actions may be defined that overrule the general rules given in the table. In this case they are defined in the ControlDraw model.

ControlDraw model Exception Handling objects
Where the exception handling requirements are not completely covered by the generic rules then details are defined as follows:
An Exception Handling Control Module is placed on the Unit diagram
The Logic to handle the exceptions is then shown on the Child Diagram for the Exception Handling

If required an Exception handling the diagram may be drawn specifically. This would comprise a diagram of similar appearance and basic principles as the procedural phase diagrams but which is not part of procedural control. This might for example take a Unit to it's Hold State via a specific series of steps. However this is only done when the problem requires a specific action other than the standard Unit/EM Hold mechanism.

Control during exceptions

This section describes control and operation during exceptions - ie when there are failures.

It is important to understand the extent to which standard mechanisms within Basic Control can provide much of the exception handling.

In a large part, control during exceptions is carried out by Basic Control (as per the S88 definition) rather than procedural control.

This includes:

- Driving equipment to safe states which will hold the product in a sustainable state
- Annunciating failures via the PCS alarm handling system
- Facilitating manual control whereby operators and maintenance staff can carry out actions necessary to restore normal operation

Procedure Failure

No specific failure handling is defined at the Procedure level.

In the event that a failure occurs in a batch it will be detected by the standard exception handling for unit control.

Generally, any failure of a batch must cause upstream batches to hold. This will happen automatically as a result of the acquire and release mechanism which will prevent a failed unit from being acquired and thus inherently hold the Procedure.

Normally it should be possible, after the fault has been cleared, to resume the procedure and complete the batch.

Potentially however, a failure could leave a batch in a non recoverable state.

In this case it will be necessary to use manual control to dispose of the batch. Detection of such a condition is assumed to be a manual operational issue, with no control system logic involved.

Operation Failure (also Phases)

No specific failure handling is defined at the operation level.

In the event that a failure occurs in a batch it will be detected by the standard failure handling for equipment

module control.

When an equipment module or unit is failed any operation that is running on the module will suspend due to the module being held. Each operation will have a "Too Slow" alarm. Generally these alarms will NOT cause the operation to go to hold. The basic principle is that when the equipment is in a hold state then the running operation is inherently suspended. This happens because the equipment is not doing any processing. Since operations generally wait for some process event before moving to the next step, the hold state will prevent the process event from happening.

Unit and Equipment Module Failure

When a unit or equipment module fails, a standard exception handling mechanism is invoked.

The lower level objects (ie Control Modules) which the unit or equipment module controls are set to predetermined states. These will generally be equipment states that can sustain the current batch in a stable condition whilst the problem is resolved.

In the Fail condition, valves adopt positions and motors run or stop as defined in the Equipment State matrices.

Failure of an Equipment Module is detected by the failure of any of the subsidiary objects in the module, or by specific sensors such as level switches.

It may occasionally be that some operations require some control over the failure detection. This can if necessary be achieved either by:

Including a Failure Response value that is set by the phase that determines what basic control should do when there is a failure.

Providing alternative Fail/Hold equipment states depending on the current equipment state.

Any alarm should enter the batch log, not all may cause an equipment exception.

(Suggest that the Basic Control includes the batch number, downloaded by recipe management when the equipment is acquired. The Batch number should then be included in a field in the alarm log so that they can be queried into the batch log. Details to be reviewed with RHS input from NNE.

If any part of a module fails the entire module will be deemed to have failed, unless the detailed documents such as EMS or lower level design Documents exclude a particular failure.

One alarm will be generated per unit in the event of a failure. This is shown on the Equipment State Transition diagram.

Operators must correct the problem (eg: by overriding a limit switch) before resuming the recipe procedure for the batch.

Control Module Failure

PCS Control logic will detect the failure of control modules. Generally this involves comparing the commanded state with the inputs for the control module - for example if a control loop is deviating or a valve has failed to move.

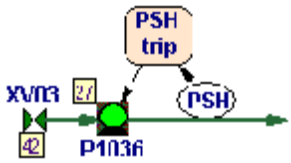
When this occurs an alarm will be generated (one alarm per Control Module) and equipment modules that use the control module will also fail.

Interlocks

- Interlocks exist only to prevent undesirable actions such as dead heading a pump.
- Interlocks only act in exception, typically by closing a valve or stopping a pump.
- Most interlocks act within a control module or are themselves control modules and apply in both manual (operator control) and auto (phase control) modes.
- Interlocks do not carry out normal control, so for example a pump that just starts and stops on the basis of level is not controlled by an interlock.
- Interlocks must generate an alarm when they act - ie when they actually stop a pump or close a valve.
- The Interlock status must be visible to operators, so that when a manual operator action such as starting a pump is prevented then the operator can see the reason.
- Interlocks should be simple.
- High security level Interlock overrides are assumed to be provided for maintenance and commissioning purposes.

· · There are alternative methods of representing these:

Graphically in the Model, by using an Interlock symbol on , for example, an equipment module diagram.



This works well for simple interlocks. If the logic is sufficiently complex, a Logic diagram will be provided underneath it - of course, similar interlocks would be represented on the same interlock diagram.

By rules written into control module diagrams - where the interlock is easy to predict by a rule such as "In the case of valves that feed into flowplates the interlock will include confirmation that the flowplate port is connected". Typically these are shown as notes in the diagram.

Note: The above relates to software interlocks only, hardwired interlocks are shown on the P&ID's only, unless linked to the control system.

PCS Failure

NOTE - Details to be reviewed against PCS functionality.

I/O Failure

PCS Control logic will detect the failure of I/O cards.

When this occurs an alarm will be generated (one alarm per I/O channel) and equipment modules or control modules that use the I/O will also fail.

Processor Failure

PCS Control logic will detect system failures.

These are handled in the same way as I/O failures.

[Types of failures detectable will be system specific - to be defined as part of detailed design]

Power Failure and Recovery

On loss of power to a controller all of it's outputs will be de-energised.

When the power returns, the outputs will be held off and all controlled equipment will be deemed to have failed.

Operations will suspend. It will be possible to re-start the Equipment module in the same way as when recovering from a Unit failure

Communications Failures

Communication Failures can be divided into the following types:

1. Loss of communications from controllers to operator stations
2. Loss of communications from controller to controller.
3. Loss of communications from controller to I/O rack/remote I/O.
4. Loss of communications from server to Operator Terminal.
5. Loss of communications from PCS to RHS.
6. Loss of communications from PCS to the PI database
7. In the event of a failure, Control will continue to the point at which an Operator signal is required. ***** OR should there be an immediate hold??

8. 8. Typically this means: -

Operations will continue to either completion or else the need for an operator confirmation.

Units and Equipment Modules will continue normal running.

Recovery from Failure

In the event of a failure it is necessary for there to be a method for recovery. Ideally this will permit re-establishing control and continuing. A method whereby this can be done is defined below. Exceptionally this will involve abandoning the batch.

Note: Elements of restarting equipment and procedures may be RHS dependent.

Continuing

The method proposed is: -

Whilst the Unit or Equipment Module is still in hold

- Manually Correct the problem
- Restart the Unit/Equipment Module
- With the Unit/Equipment Module re-started and all failures cleared
- Restart the Batch

· · **Correcting the problem**

Typically this is a manual activity carried out in the plant and at PCS screens (eg by overriding a limit switch).

Restart the Equipment

This requires that the Equipment Modules and Unit be put back into their running modes.

Restart the Batch

In order to restart production a Restart Batch command must be sent.

Alarm Handling

Alarms are generated in several parts of the application. To a large degree the alarms that are produced are 'standard', originate in the control module level and are independent of the operation that is being run.

For example, Discrepancy alarms such as Valve Fail-to-Open will occur regardless of the state of the plant if a valve is not at its commanded position as described in the valve driver section. Similarly an Equipment module Fail alarm indicates when a module within the Equipment module fails, whatever the unit was doing at the time.

Occasionally however there may be devices that might indicate an alarm under some process conditions whilst but not under other conditions. The ControlDraw model may indicate this by means of additional columns within the Equipment State Matrix, or by specific steps (such as enable alarm) in the Phase logic or by notes.

Some alarms are generated as standard

- Unit Fail

  Generated when a control module or equipment module in a unit fails. Generally (not always) results in a Unit hold.

- Equipment Module Fail.

  Generated when an Equipment Module goes into hold after detecting a failure, usually when a control module in the Equipment Module fails

- Control Module Fail.

  Generated when a control module detects a discrepancy between sensors and effects.

- Process Deviations?
- Operation too Slow.

  · · An alarm generated whenever an operation exceeds the maximum time allowed.

Sequence of events when there is a failure:

(a) Control Module in a Equipment module Fails

Eg: Motor fails to start or trips when running, or a valve fails to move, or moves when it should have stayed put.

Sequence:

Control Module fails generating the relevant Alarm

and immediately after:

 Equipment Module fails-to-hold generating an Equipment Module Fail alarm

So then the operator sees two alarms, Valve fail and Equipment module fail.

Operator then fixes the problem and then restarts the batch by Selecting Restart.

Alarms Clear

(b) Control Module in a Common Resource Fails

Eg: Motor fails to start or trips when running, or a valve fails to move, or moves when it should have stayed put.

Note - A Common Resource is made up of one (usually) or more equipment modules.

Sequence:

 Control Module fails generating the relevant Alarm

and immediately after

 Common Resource fails to hold generating a Resource Fail alarm

 Equipment module or units using the resource may respond to the Resource Fail alarm by going to their hold state, although this is not always essential

So then the operator sees two alarms plus the number of affected Unit alarms, Valve fail and Resource fail and an equipment module Fail for each equipment module that was affected.

Operator then fixes the problem and then restarts the batch by Selecting Restart.

(c) Phase Failure Without a Control Module Failure

This might occur when some process fails to perform as expected or when a control module fails without it causing an alarm. For example a heating phase might fail due to the heating media being cold, or a pump might fail to run due to a failure of a mechanical coupling (so that the motor is running and hence the motor driver sees no failure.)

In such a case, the operation or phase that is currently running will typically take too long to reach the required process condition (eg Temperature Set Point reached) and so ultimately the "Phase Too Slow" alarm will be generated.

Sequence:

Process failure during a phase

Phase continues to run but does not reach the target end point

 Phase too long alarm.

Operator fixes the problem and accepts the alarms

Alarm Latching and Acknowledgement

There are several issues in respect of the alarm handling system. This includes

- Need to 'Latch' transient alarms
- Alarm Acknowledgement
- Alarm Disabling

· · ***Consider the following:***

A valve in an Equipment module has failed closed, and will not open. The position of this valve in the Hold state of the Equipment module should be closed. At some step in an operation, the valve is asked to open and fails to do so.

The first thing to happen will be that the valve control module generates a Fail-to-Open alarm, which in turns causes the Equipment module to into hold.

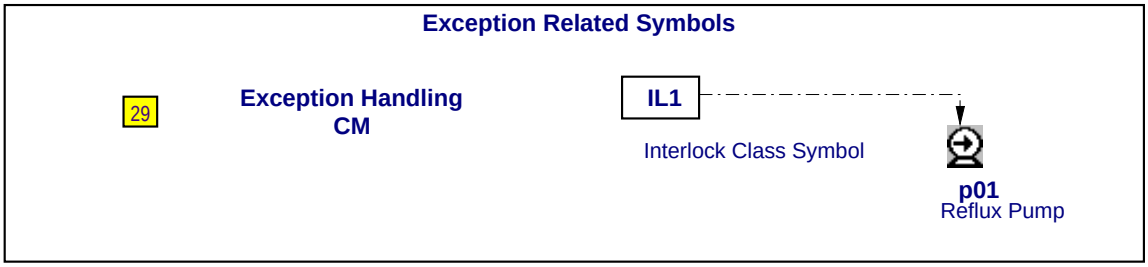
The effect of this will be that the command to the valve (which was 'open') changes back to 'close' and consequently the valve is no longer in the Failed to Open state.

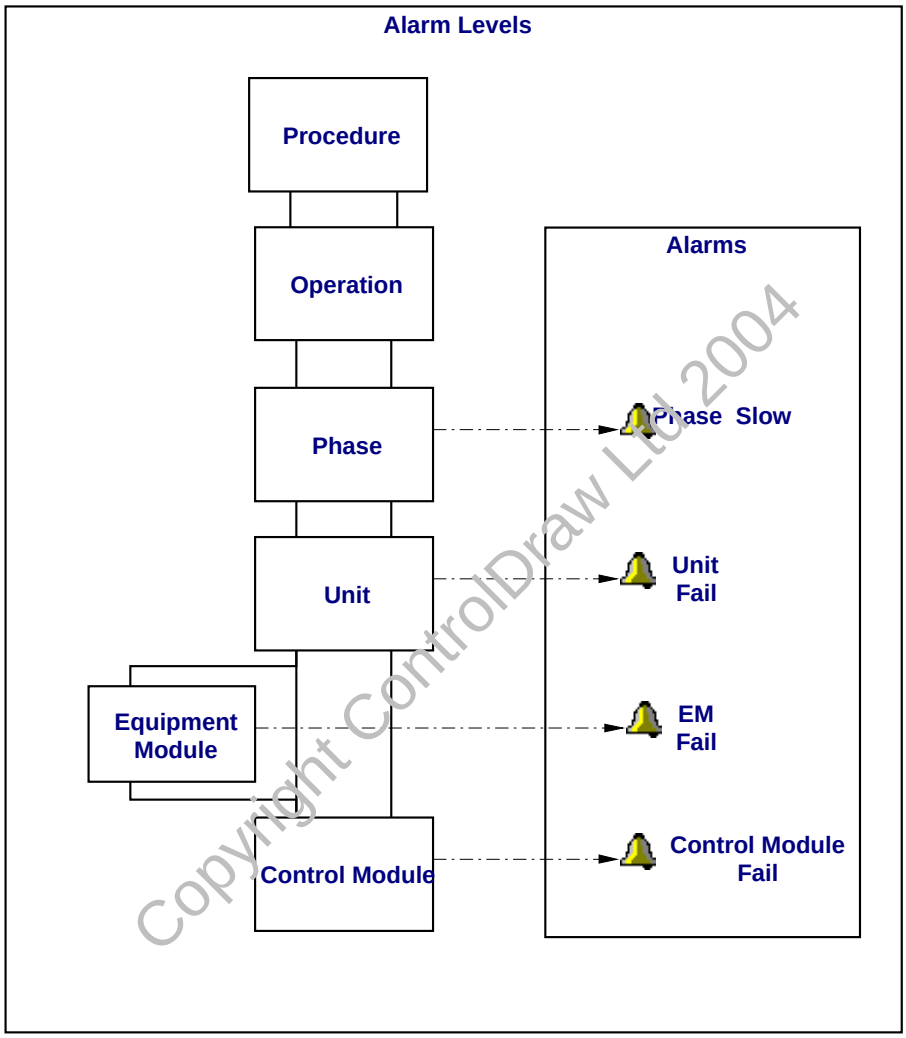
Note - it is even possible that the control system logic could go through this entire process without ever actually generating the Fail to Open alarm on the PCS screens. This can happen in cases where the Alarm scanning in the PCS is not synchronised with the Control logic in the PCS controllers. The condition exists transiently in the PCS controller, but is never seen as an alarm by the PCS workstations.

To avoid this possibility the control system must be programmed to 'latch' such alarms on and not to unlatch them until the operator accepts them.

Diagram 28 - Exception Handling

Diagram Version: 136 **Diagram 28 of 43**



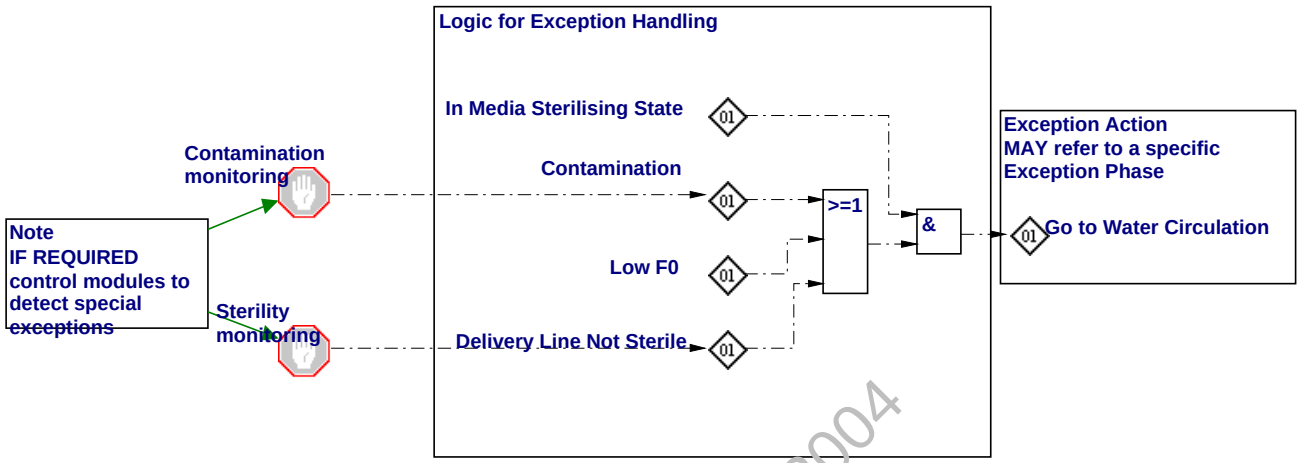


Description for Diagram 29 - Exception Logic

This shows an example of an exception handling object. It generates a flag that is used to drive a unit to a safe staee.

Diagram 29 - Exception Logic

Diagram Version: 0 **Diagram 29 of 43**



Description for Diagram 30 - Sterile Hygienic States

This shows how a piece of physical equipment can moves through the Sterile and Hygienic states in response to process events.

The normal cycle is the green route.

Diagram 30 - Sterile Hygienic States

Diagram Version: 5 **Diagram 30 of 43**

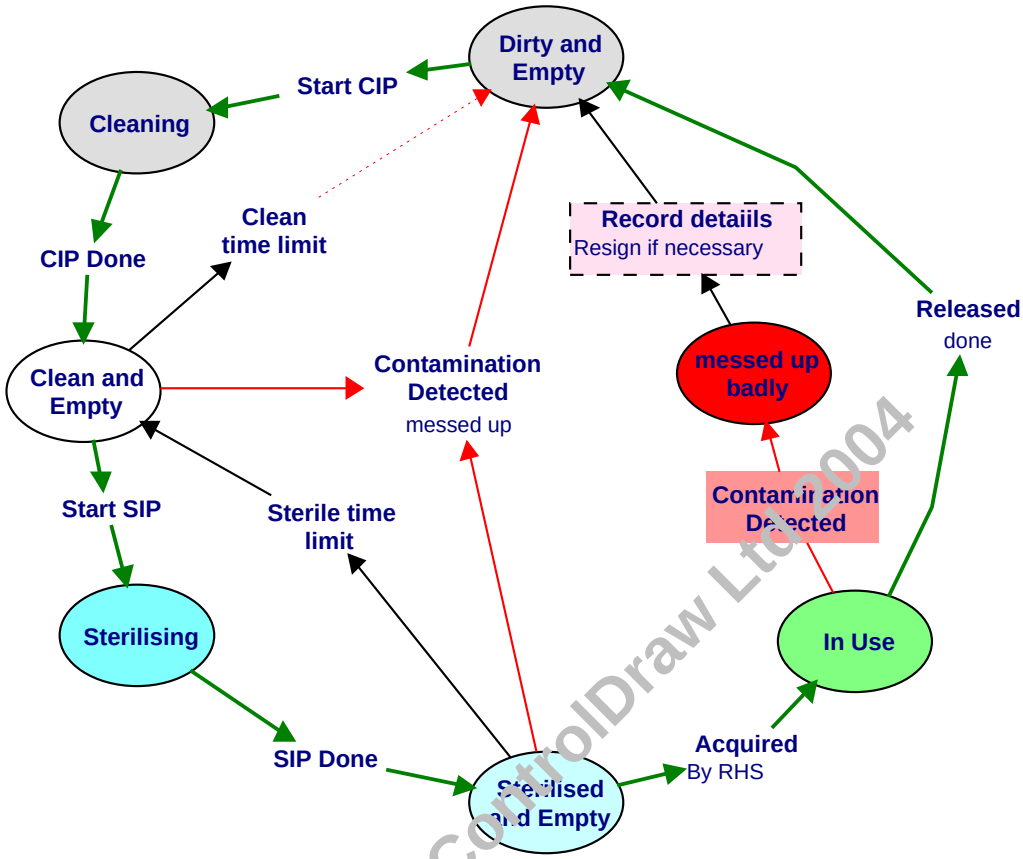


Diagram 31 - Non Sterile Hygenic States
Diagram Version: 134 Diagram 31 of 43

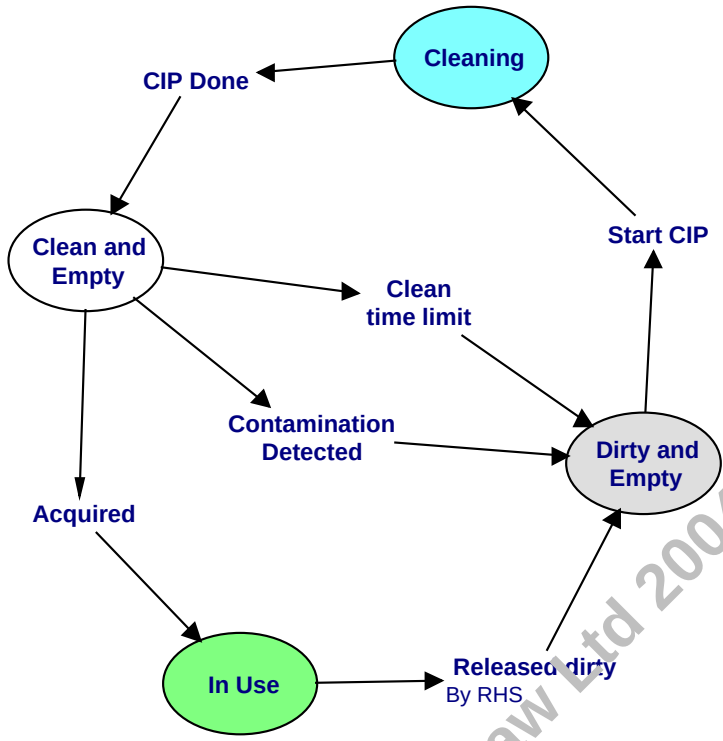


Diagram 32 - Feed Vessel Unit Graphic
Diagram Version: 134 Diagram 32 of 43

This diagam indicates a Unit with a number of Equipment modules as represented on a Process Graphic with Pop up Equipment module graphic

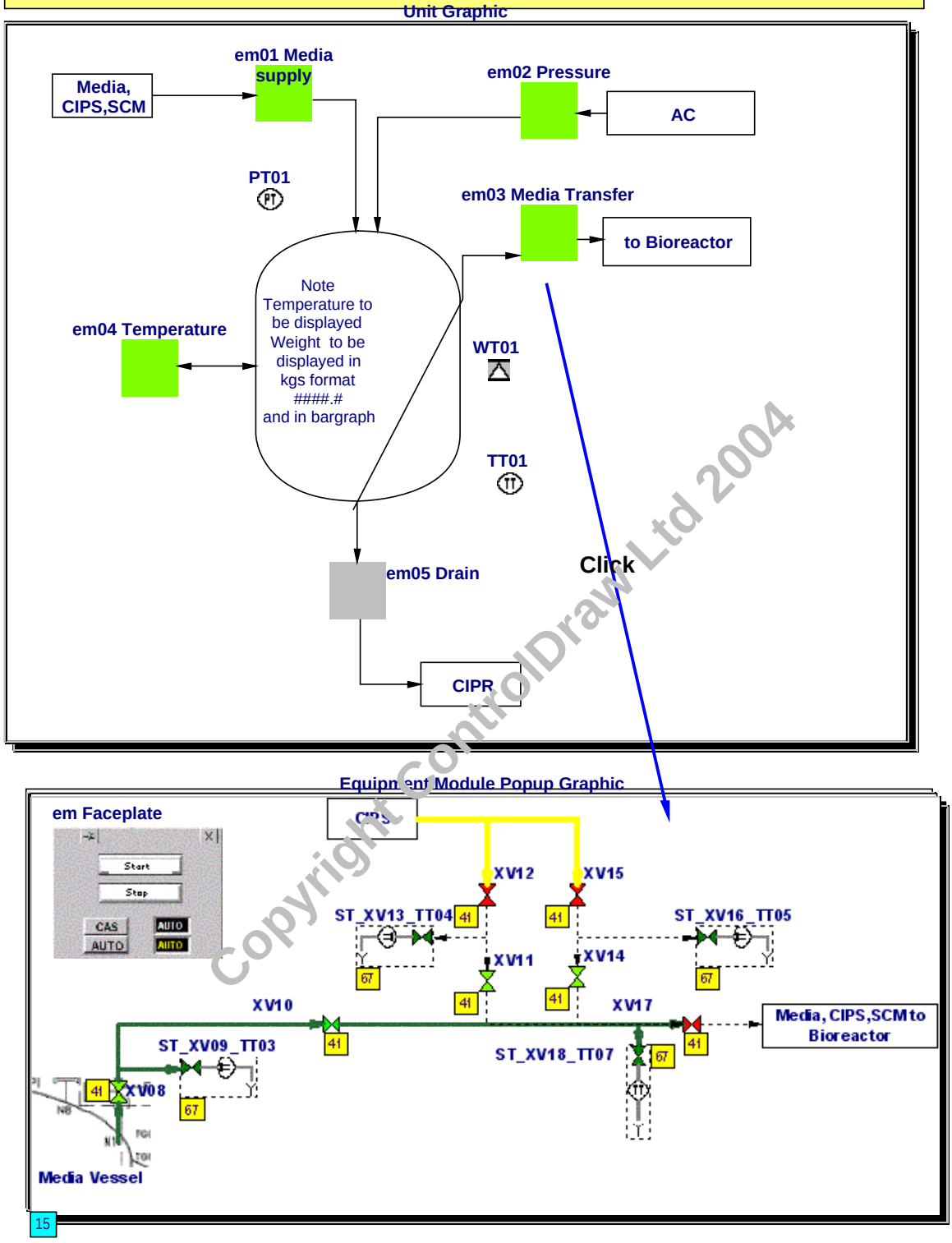
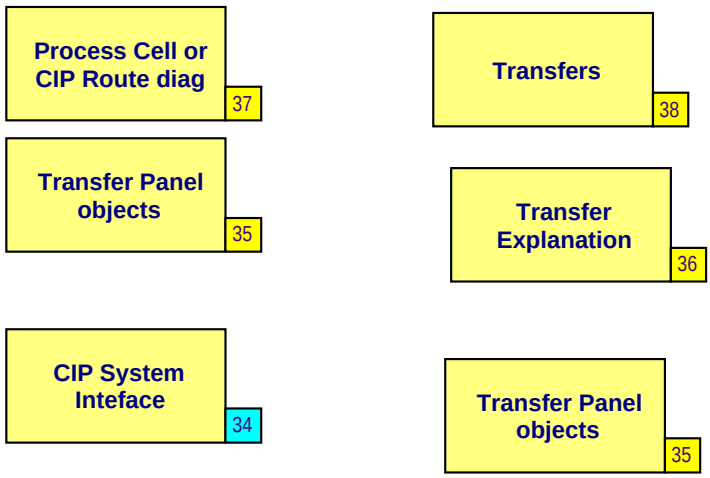


Diagram 33 - Generic Process Designs
Diagram Version: 127 **Diagram 33 of 43**

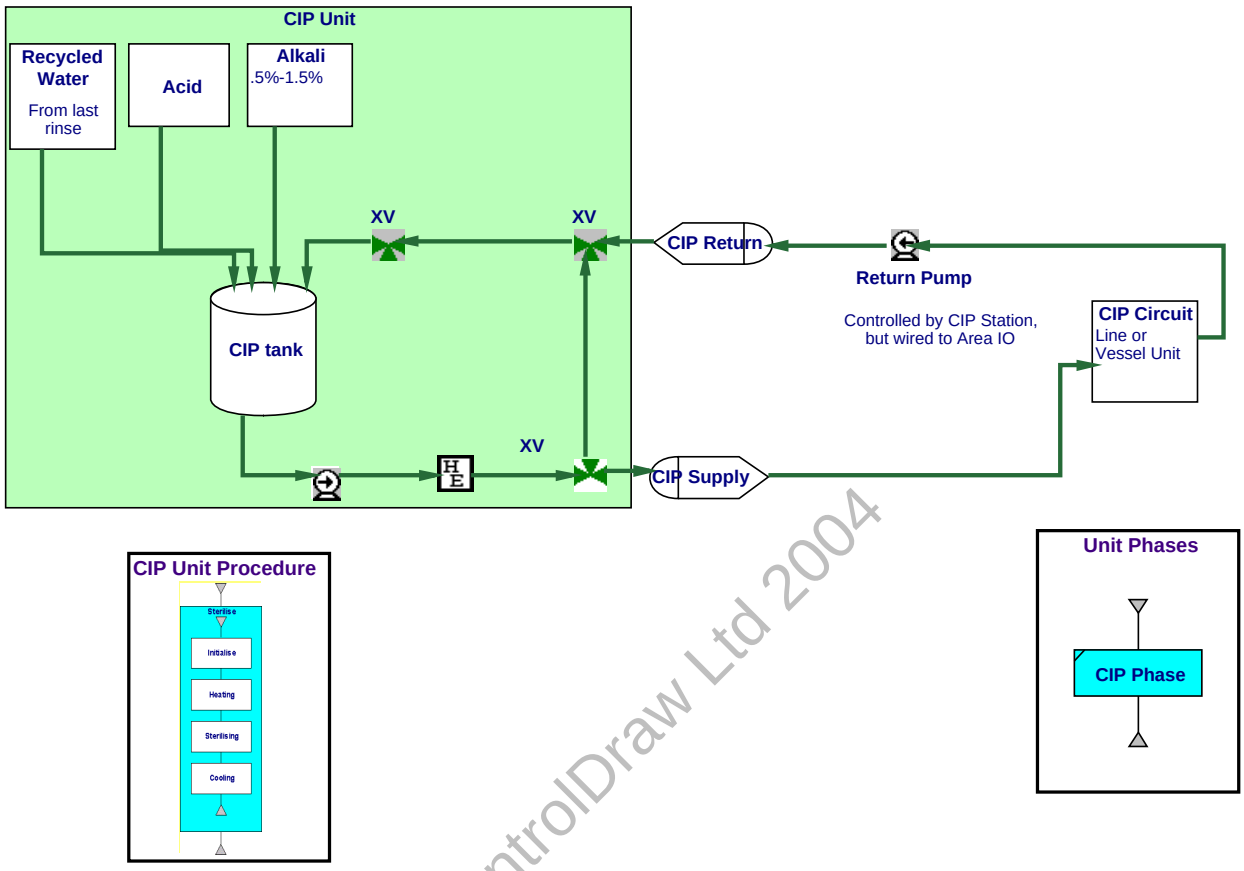


Description for Diagram 34 - CIP System Inteface

This shows how a CIP unit runs the CIP procedure whilst the equipment bneing cleaned runs CIP phases

Diagram 34 - CIP System Inteface

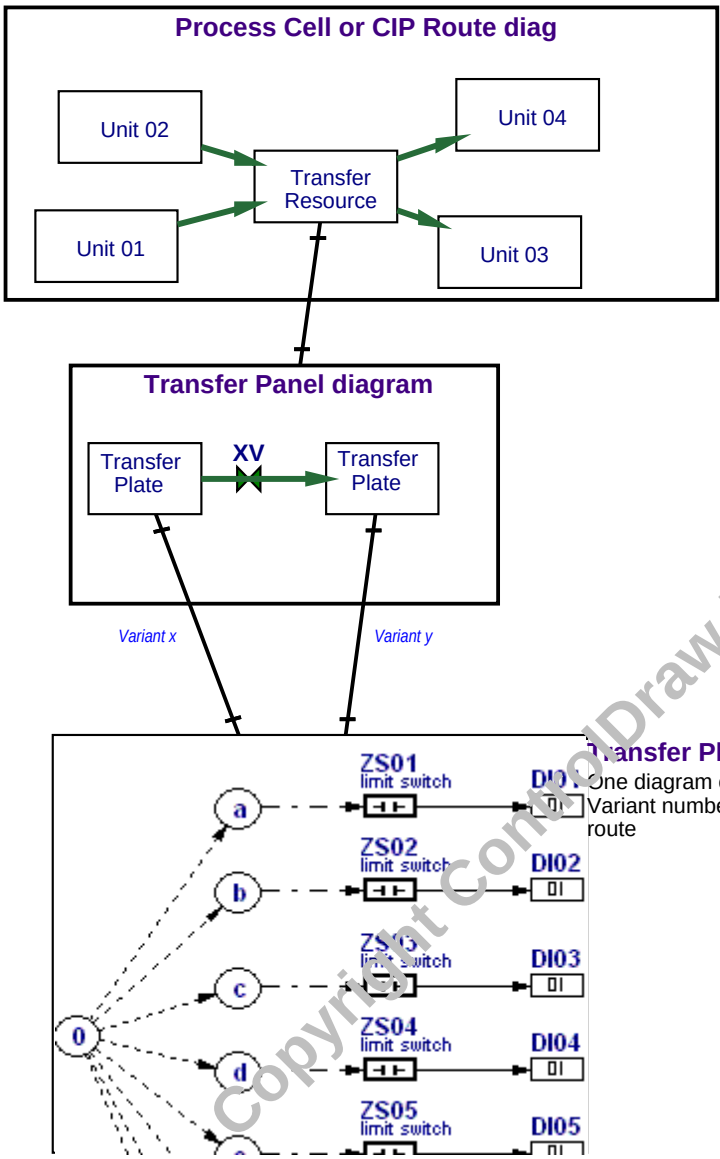
Diagram Version: 143 Diagram 34 of 43



Description for Diagram 35 - Transfer Panel objects

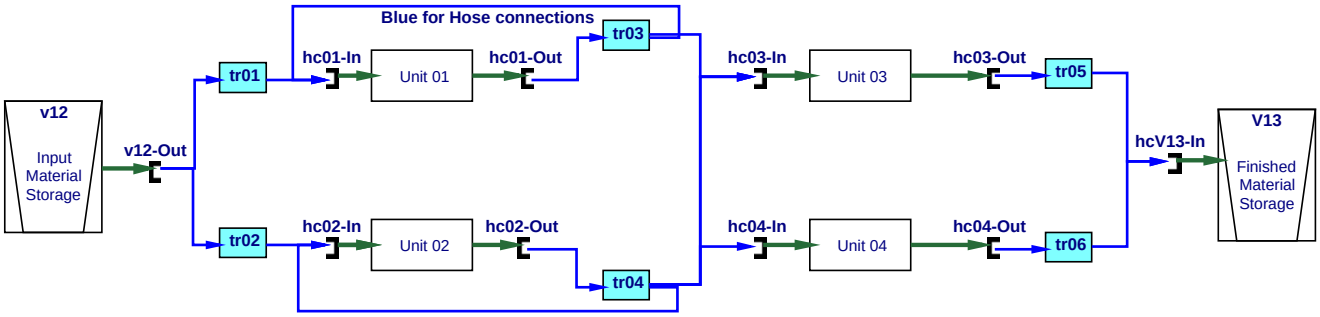
The boundaries of each unit do not include the transfer panels.
Where a TP is shown with a unit on a P&ID then that is made separate. This improves modularisation.
This diagram shows how

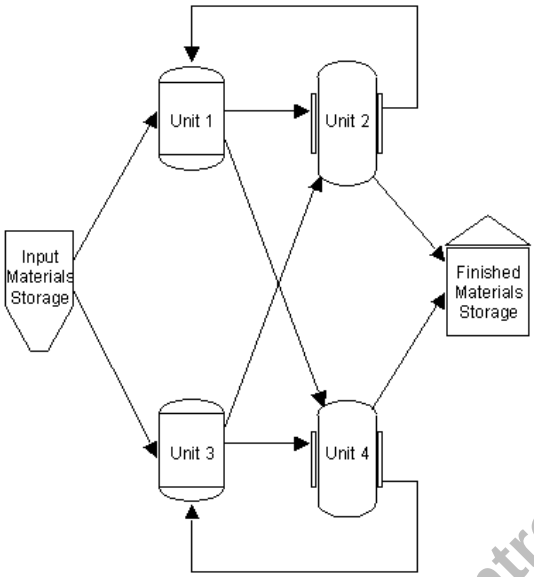
Diagram 35 - Transfer Panel objects
Diagram Version: 96 **Diagram 35 of 43**



Transfer Plate Diagram
One diagram covers all
Variant number selects number of route

Diagram 36 - Transfer Explanation workspace
Diagram Version: 143 Diagram 36 of 43

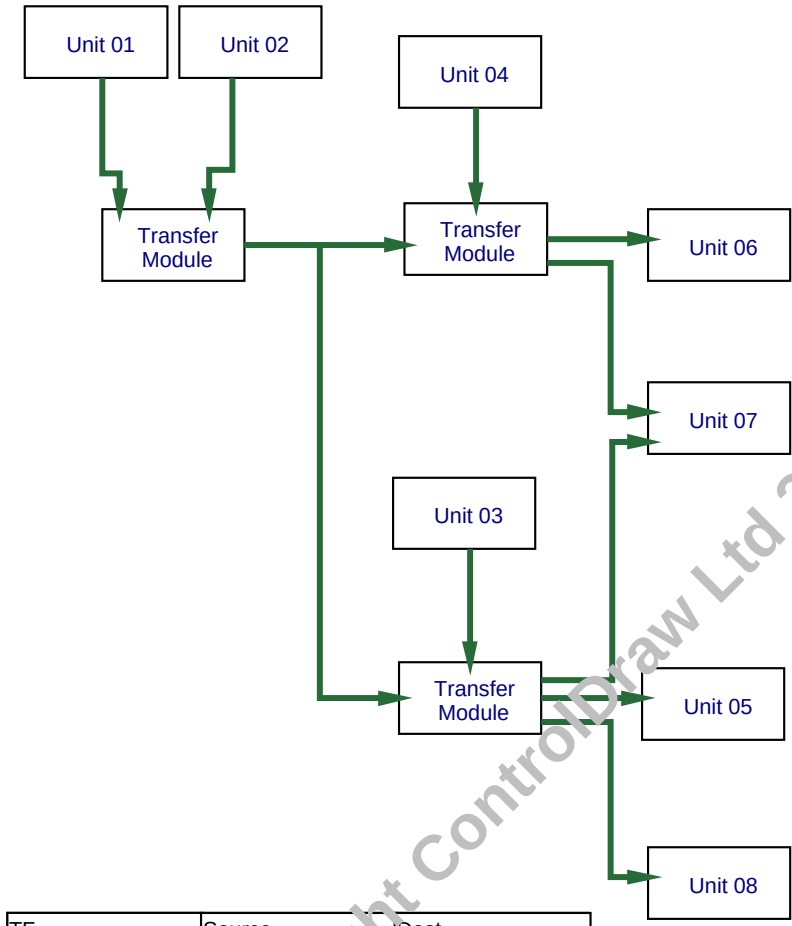




TF	Source	Dest
tr01	v12-Out	hc01-In
tr02	v12-Out	hc02-In
tr03	hc01-Out	hc01-In
	hc01-Out	hc03-In
tr04	hc02-Out	hc02-In
	hc02-Out	hc04-In
	hc02-Out	hc03-In
	hc02-Out	hc03-In
tr05	hc03-Out	hcV13-In
tr06	hc04-Out	hcV13-In

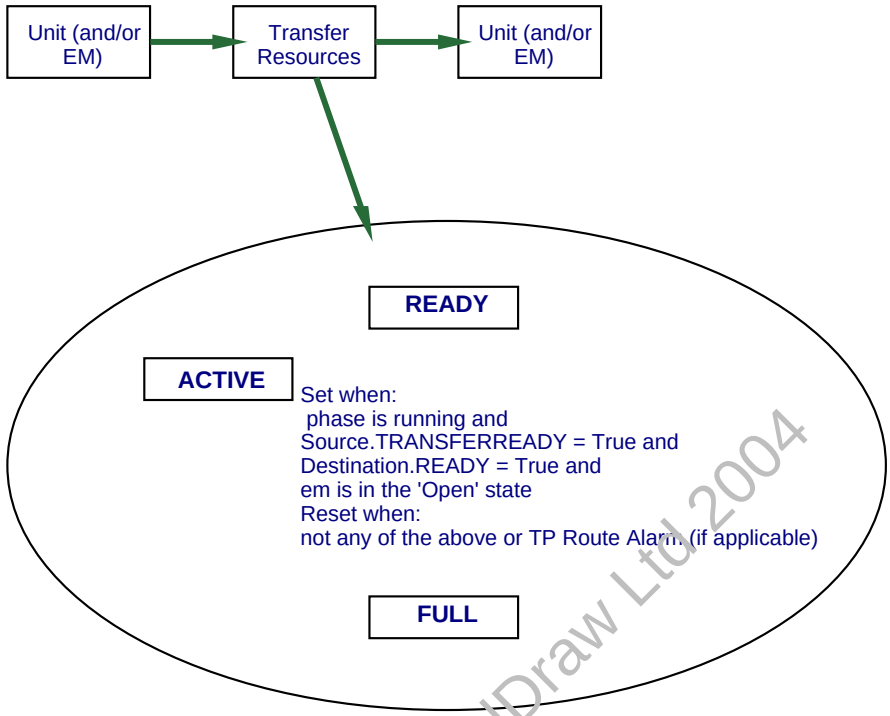
Diagram 37 - Process Cell and transfers
Diagram Version: 116 Diagram 37 of 43

Process Cell or CIP Route diag



TF	Source	Dest
Transfer Module	Unit 08	Unit 07
	Unit 03	Unit 07
	Transfer Module	Unit 07
	Unit 04	Unit 07
	Transfer Module	Unit 07
	Unit 01	Transfer Module
	Unit 02	Transfer Module
	Unit 08	Unit 05
	Unit 03	Unit 05
	Transfer Module	Unit 05
	Unit 01	Transfer Module
	Unit 02	Transfer Module
	Unit 04	Unit 06
	Transfer Module	Unit 06

Diagram 38 - Transfers
Diagram Version: 24 **Diagram 38 of 43**

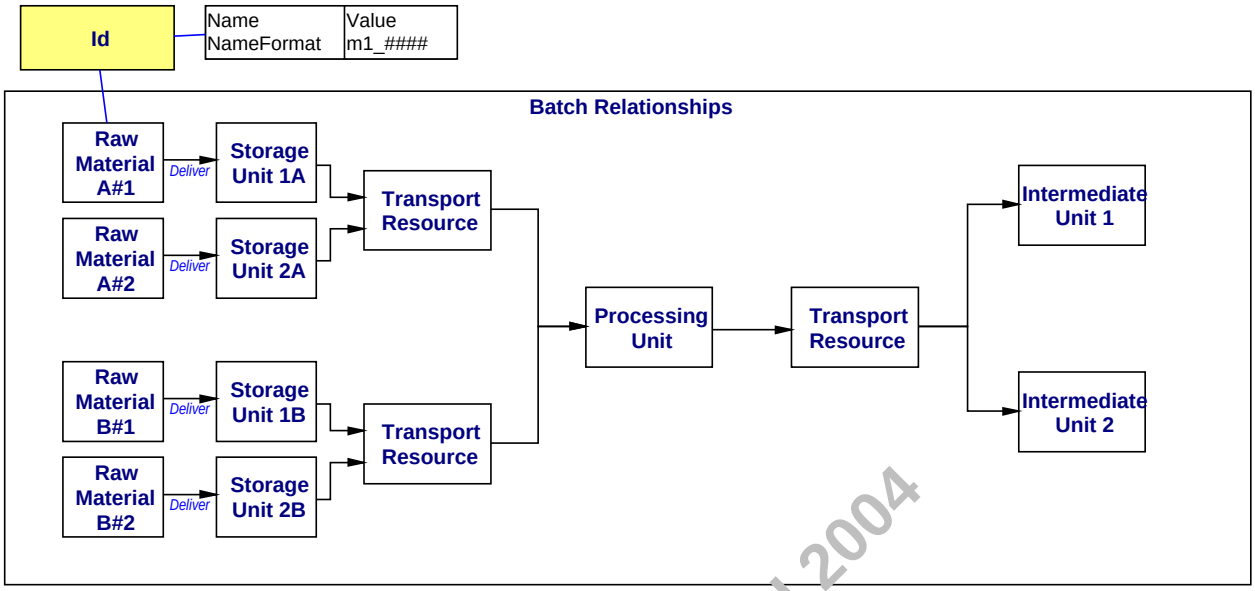


Description for Diagram 39 - Batch Relationships

This shows an example of a diagram that illlustrates the relationships between batches at they pass from Unit to Unit

Diagram 39 - Batch Relationships

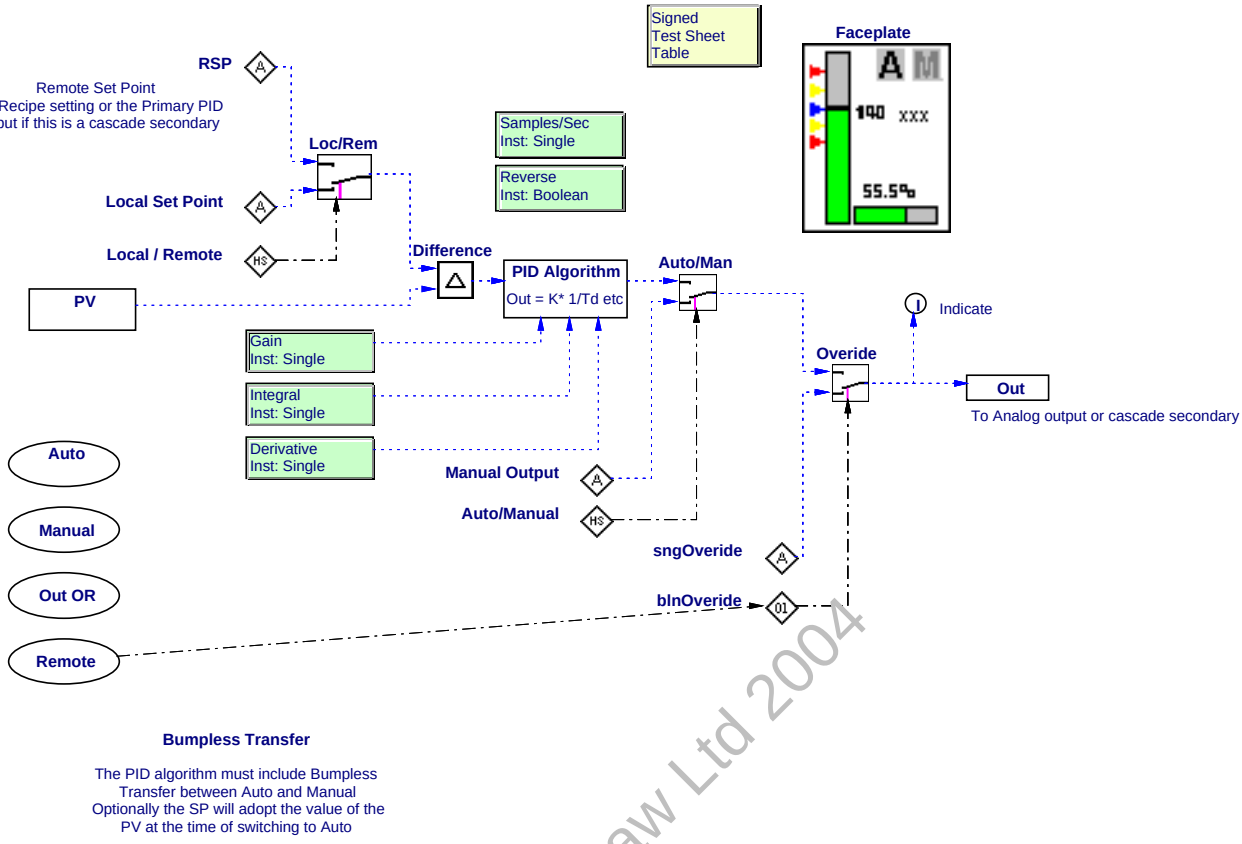
Diagram Version: 84 **Diagram 39 of 43**



Notes

This is a type of diagram that has never been developed. Potentially it could provide a better top level of the physical mode, with ID objects (special?) and queries to generate a batch structure table!

Diagram 40 - Standard PID Loop
Diagram Version: 70 RefVers: 539 Diagram 40 of 43



Bumpless Transfer

The PID algorithm must include Bumpless Transfer between Auto and Manual
Optionally the SP will adopt the value of the PV at the time of switching to Auto

Description for Diagram 41 - On Off Valve

Features of the Standard ControlDraw Valve Class

Per S88, A control module is typically a collection of sensors, actuators, other control modules, and associated processing equipment that, from the point of view of control, is operated as a single entity.....for example...(an).. on/off automatic block valve control modules.

The ControlDraw standard valve class is a a 2 position valve with optional position feedback switches, that is operated via commands from automatic or manual logic depending on it's mode.

Features of the Standard ControlDraw On Off Valve

Variants

Cover the instrumentation choices for a valve

The Valve Driver

Travel is timed in each direction. Timeout causes Fail to Open or Fail to Close
Standard Auto/Manual.

Alarms

Fail

Modes

Interlock

Overrides

If set to the limit switch is ignored

Valve status

This is used by control logic when it needs to know what the valve is up to.
0 = Closed
1 = Open
2 = Moving
3 = Failed

Travel Timers

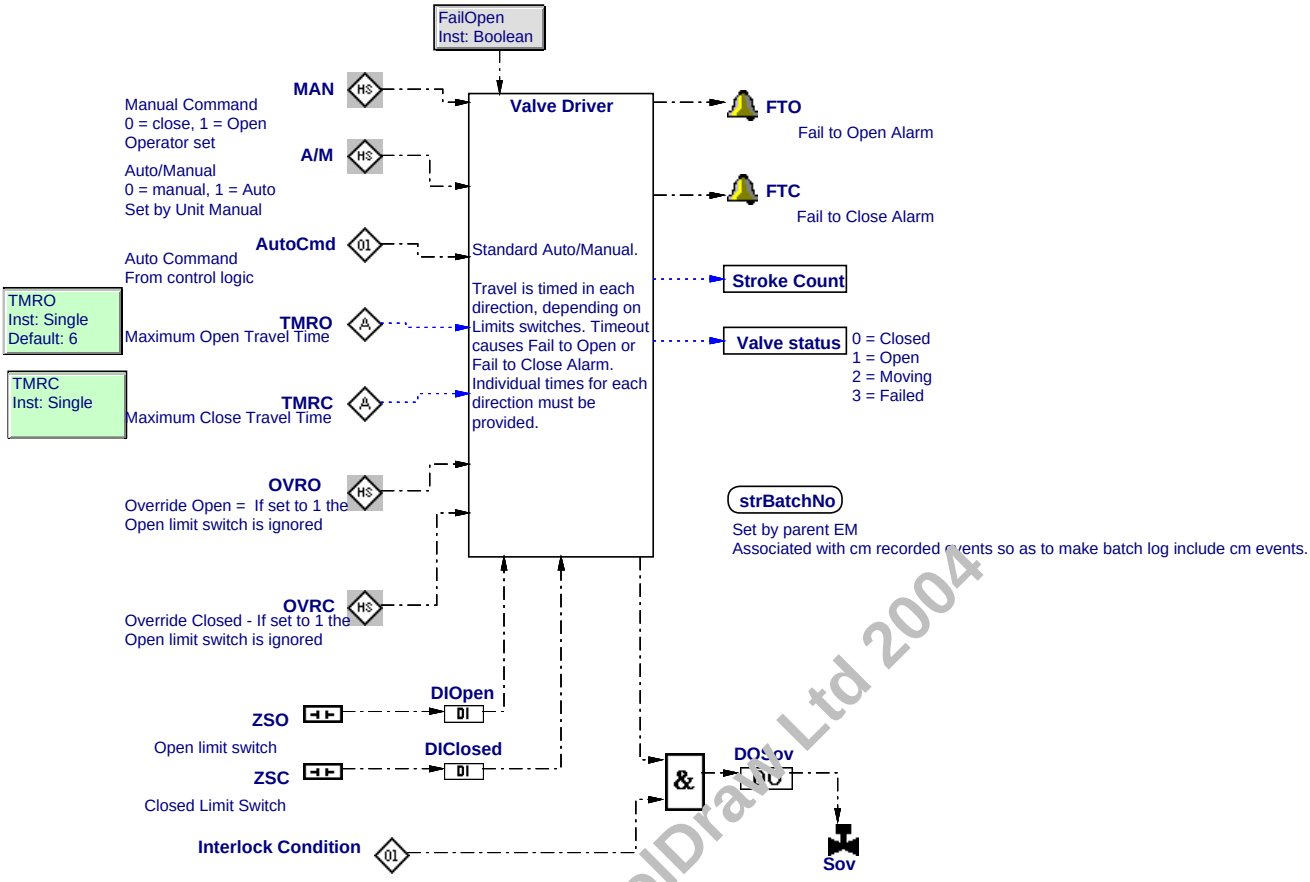
The

Stroke Count

Typically maintenance functions need to have a measure of the stress on a device, this provides a suitable count. It is best done in basic control as it is a simple function that can be handled at a low level

Diagram 41 - On Off Valve

Diagram Version: 110 RefVers: 570 **Diagram 41 of 43**

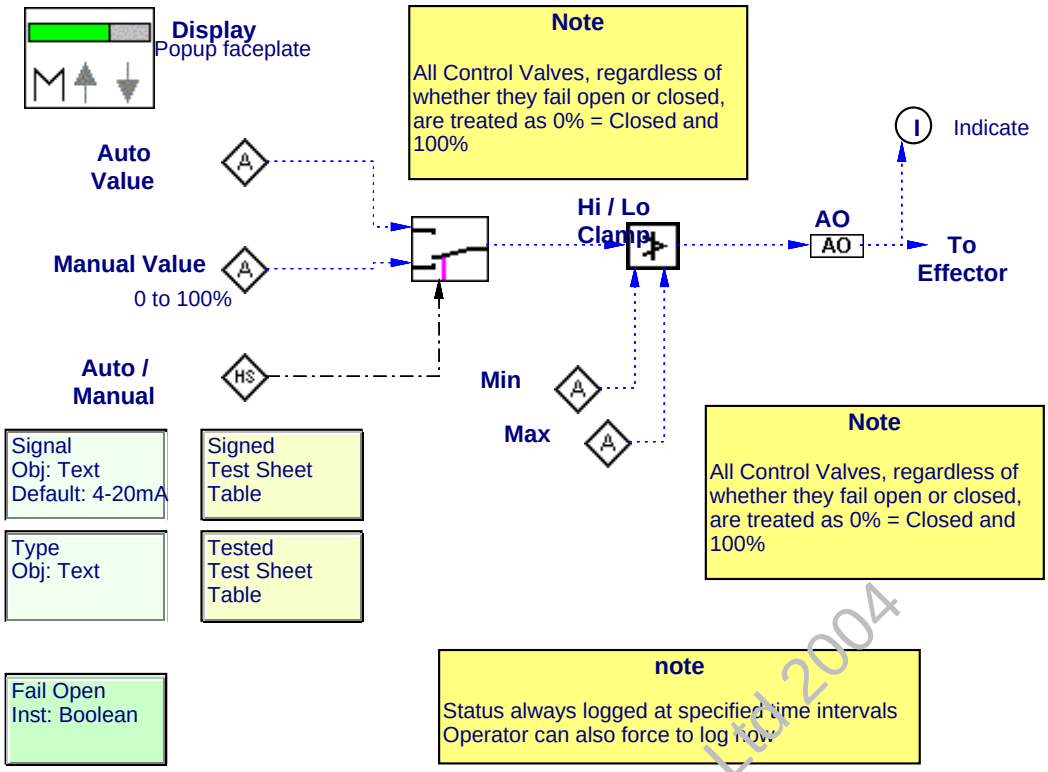


List of Variants

	1-Open LS	2-Close LS	3-2 Limit Switches	4-No limit switches
DIClosed	{Excluded}			{Excluded}
DIOpen		{Excluded}		{Excluded}
FTC	{Excluded}			{Excluded}
FTO		{Excluded}		{Excluded}
OVRC	{Excluded}	OVR		{Excluded}
OVRO	OVR	{Excluded}		{Excluded}
TMRC	{Excluded}	TMR		{Excluded}
TMRO	TMR	{Excluded}		{Excluded}
ZSC	{Excluded}			{Excluded}
ZSO		{Excluded}		{Excluded}

Diagram 42 - Control Valve

Diagram Version: 57 RefVers: 498 **Diagram 42 of 43**

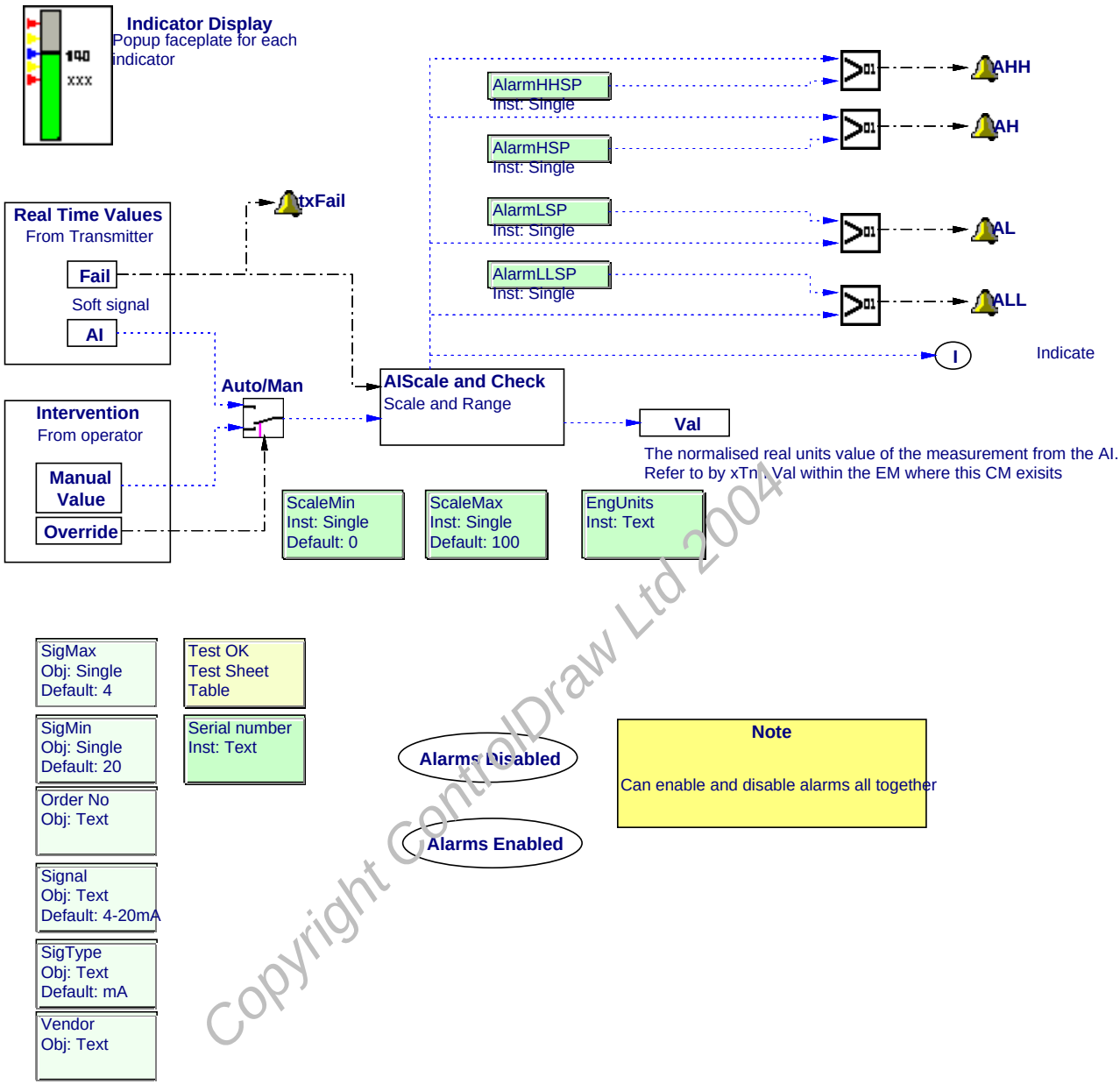


Description for Diagram 43 - Analog Input from Transmitter

No text, you can write some here

Diagram 43 - Analog Input from Transmitter

Diagram Version: 82 RefVers: 550 **Diagram 43 of 43**



Data Report: Classes

S88 Def	Class	Tag	Include	Description
Alarm	Alarm	al	☑	Indicates an alarm event
Allocation	PFC Acquire / Release	acq	☑	A form of coordination control that assigns a resource to a batch or unit.
Area	Plant Area	pa	☑	A component of a batch manufacturing site that is identified by physical, geographical, or logical segmentation within the site. NOTE - An area may contain process cells, units, equipment modules, and control modules.
Batch	Container	ct	☐	A generic class for container such as mobile vessels.
Batch Journal	Batch Journal Entry	bj	☑	An element to be included in a Batch Report
Batch Journal	Batch Report	br	☑	The extraction of data related to one or more batches.
Common Resource	Transfer Class	tc	☑	A transfer route
Control Module	Control Module	cm	☑	A generic type of Control Module.
Control Module	Effector Analog	ee	☑	A type of Control Module that drives an analog effector, such as a Control Valve
Control Module	EffectorDiscrete	ed	☑	A type of Control Module that drives an boolean effector, such as a Lamp
Control Module	Interlock Control Module	il	☑	Shows a low level interlock, such as a pump trip, that does not depend on procedural control.
Control Module	Logic Function	lg	☐	A discrete logic function
Control Module	Loop Function	lf	☐	A continuous control function
Control Module	Measurement Analog	me	☑	A continuous measurement such a pressure, flow, temperature etc
Control Module	Measurement Switch	ms	☑	A discrete measurment such a level switch.
Control Module	Motor	mtr	☑	Control module for Motors
Control Module	PID Control Loop	pid	☑	A Proportional, Integral and Derivate controller. Standard PCS function.
Control Module	Software Driver Module	sd	☑	The software object associated with a control module
Control Module	Software Object	so	☑	A generic class for software entities
Control Module	Valve	vv	☑	Control module for valves
Control System	Control System	cs	☑	An attribute of the physical control system, such as an input or output.
Control System	Control System Node	cns	☑	A part of the physical control system, such as a controller or workstation.
Equipment Module	Equipment Module	em	☑	Equipment module A functional group of equipment that can carry out a finite number of specific minor processing activities.
Equipment Procedure	Equipment Procedure	er	☐	not used
Header	Document Reference	dr	☐	A reference to a document that is not contained in a model, such as a P&ID or Process Description
None	Action	AC	☐	A generic class for general use
None	Datastore	ds	☐	A generic data storing object
None	Entity	en	☐	A generic type for such things as software structure diagrams
None	Equipment	eq	☐	Used for physical items such as vessels
None	Interface	if	☑	A object that is for linking between modules that need to communicate, for example for transfers.
None	MemoryDiscrete	bln	☐	A Discrete (boolean) data storing object, used for local variable in modules
None	MemoryInteger	int	☐	An Integer data storing object, used for local variable in modules
None	MemoryReal	sng	☐	A Real data storing object, used for local variable in modules
None	MemoryString	str	☐	A String data storing object, used for local variable in modules
None	None	none	☐	Typically used for notes
None	Operator Command	co	☐	An event generated by an operator
None	Operator Display	od	☐	A display element for the operator
None	Operator Entry	oe	☐	A parameter enterednby an operator

S88 Def	Class	Tag	Include	Description
None	Person	pe	☐	Used for Notes
None	Transfer Segment	ts	☑	A part of a transfer route
None	Transition	tr	☐	A discrete event
Operation	Operation	op	☑	A procedural element defining an independent processing activity consisting of the algorithm necessary for the initiation, organization, and control of phases.
Phase	EM State change phase	emst	☐	A sequence that runs in order to change the state of an equipment module.
Phase	EMSubState	emsb	☐	A sub state
Phase	Phase	ph	☑	The lowest level of procedural element in the procedural control model.
Process Action	Phase/Op Step	ps	☐	A step in a sequential process
Process Cell	Cell Common Resource	cc	☑	A Container for Common resources that are contained in an Area
Process Cell	Process Cell	pc	☑	A container for the Units, Recipes and Resources in a cell.
Process In-Out	Process Material	pm	☐	Raw material or other resource
Process Parameter	Equipment Parameter	ev	☑	This is an S88 process parameter that is equipment dependant and not product dependant
Process parameter	Recipe Formula Value	rf	☑	Information that is needed to that is dependant on the product.
Recipe	Recipe	re	☑	The necessary set of information that uniquely defines the production requirements for a specific product.
Recipe Procedure	Recipe Procedure	fp	☑	The part of a recipe that defines the strategy for producing a batch.
Site	Site	si	☑	Site A component of a batch manufacturing enterprise that is identified by physical, geographical, or logical segmentation within the enterprise. NOTE - A site may contain areas, process cells, units, equipment modules, and control modules.
State	State	st	☐	The condition of an equipment entity or of a procedural element at a given time.
Unit	Common Resource	cr	☑	Common resource A resource that can provide services to more than one requester. NOTE - Common resources are identified as either exclusive-use resources or shared-use resources (3.22 and 3.54).
Unit	Unit	un	☑	Unit A collection of associated control modules and/or equipment modules and other process equipment in which one or more major processing activities can be conducted. Units are presumed to operate on only one batch at a time.
Unit Procedure	Unit Procedure	up	☑	A strategy for carrying out a contiguous process within a unit. It consists of contiguous operations and the algorithm necessary for the initiation, organization, and control of those operation
Wiring	Wiring	wr	☑	

Data Report: S88 Index