S88 Batch Standard

An esoteric standard, used as a formal communications tool for batch control.

Manufacturing Operations

- **Continuous**
- **Discrete**
- **Batch**

Continuous
- **Make "stuff"**
- **Long cycle time**
- **Process rarely "shut-down"**
- **Few procedural changes**

Example: crude oil refining

Discrete
- **Make "things"**
- **Short cycle time**
- **Process up and down at will**
- **Lot size of one**

Example: automotive parts, gears, etc

Batch
- **Make "discrete quantities of stuff"**
- **Medium cycle time**
- **Process runs in "batches"**
- **Mixes attributes of continuous and discrete**

Example: Pharmaceuticals, food processing, chemical processing

S88 is a way of thinking of Batch Control, not the hardware, rather a set of terms and philosophy.

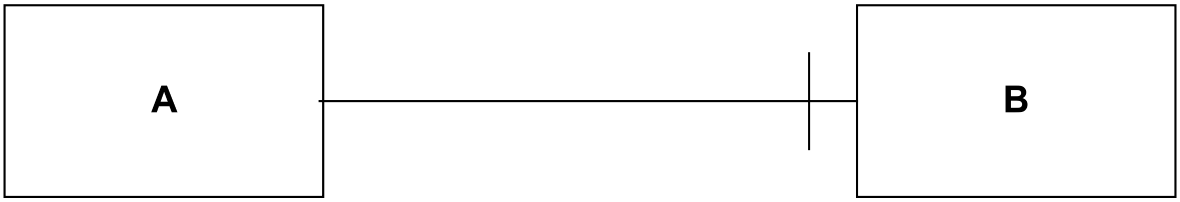
Why a standard? To address control problems with batch control:

No Universal model for batch control
Difficulty in communicating terms, Example *tune*
Difficulty integrating different manufactures
Batch Control Configurations too difficult

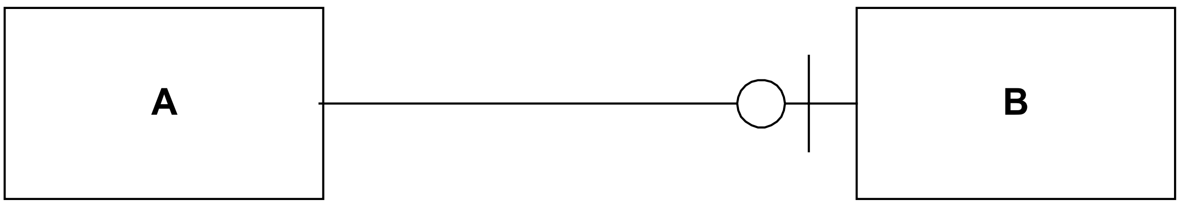
ISA SP88 Committee
The First Standard: (American National Standards Institute) ANSI / ISA S88.01-1995
Identified Common Models and Terminology for Batch Control Systems
Introduced Concept of Modular Batch Automation (MBA)

Isolates equipment from recipes
Track historical data
Recovery from abnormal events
Gather requirements
Provides modularity
Facilitate validation

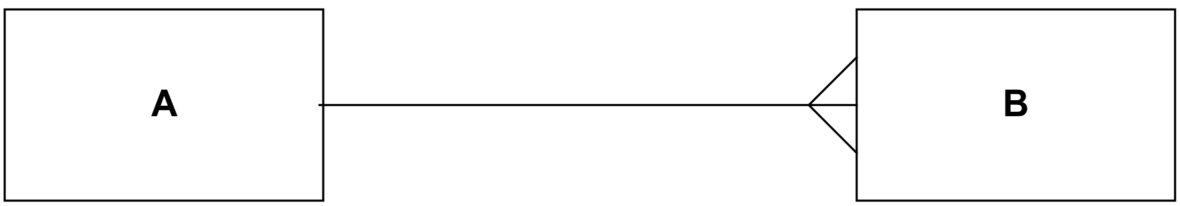
Entity-Relationship Diagrams, describing objects in a system and how they interact:



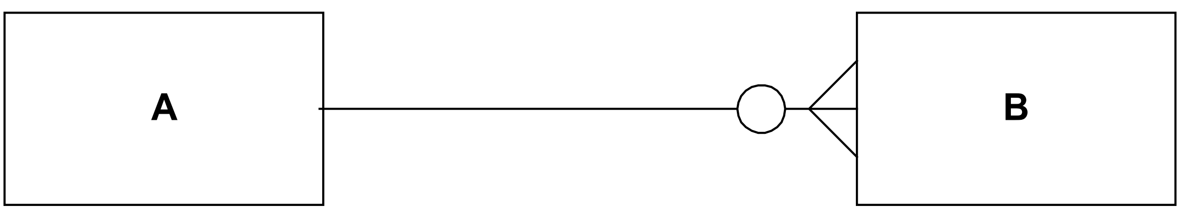
A has one and only one occurrence of B



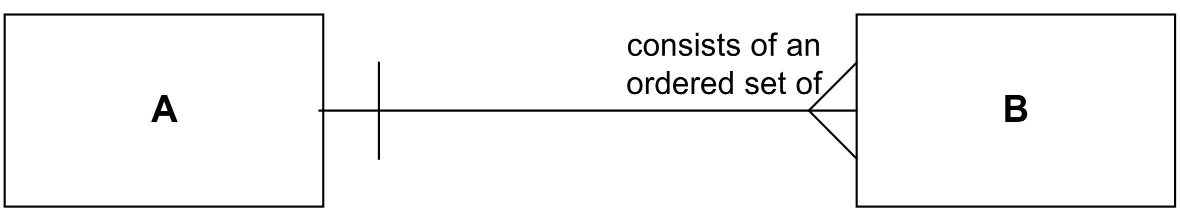
A has zero or one occurrence of B



A has one or more occurrences of B



A has zero, one or more occurrences of B

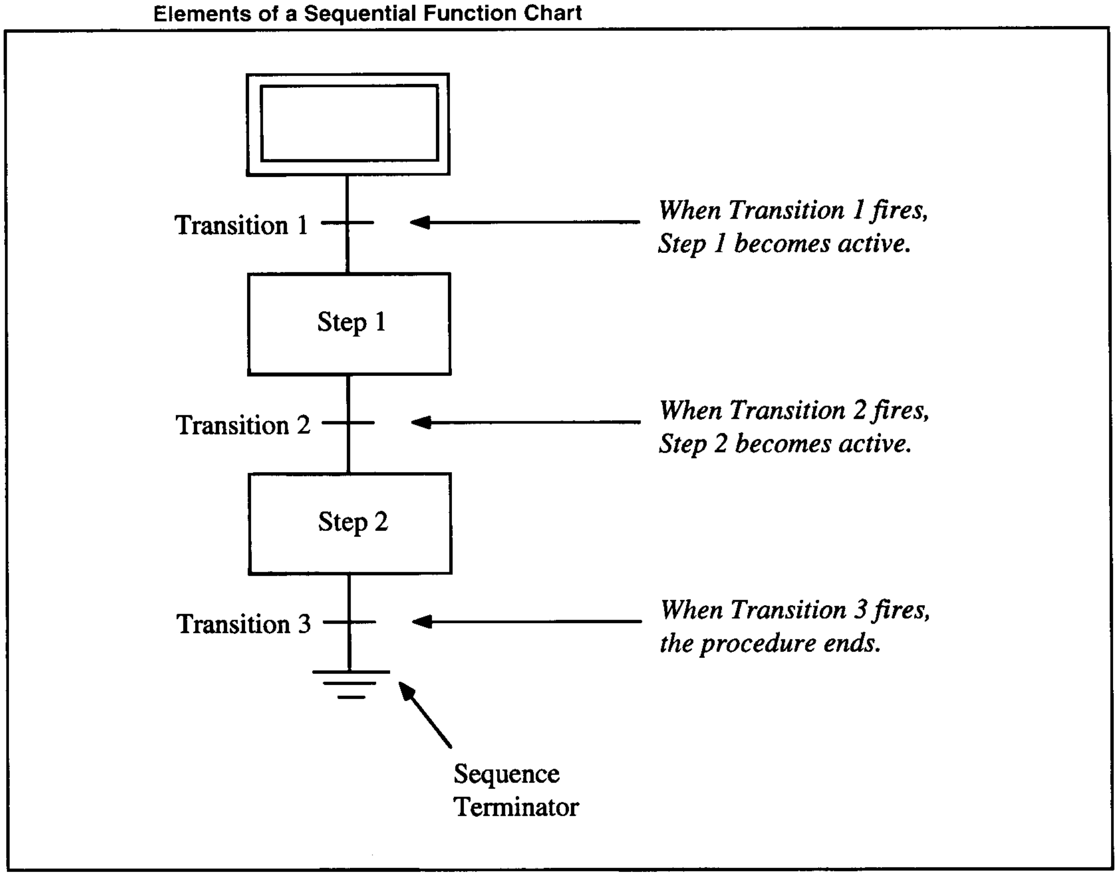


Associations: with descriptive labels; A consists of an ordered set of B

Sequential Function Charts (SFC)

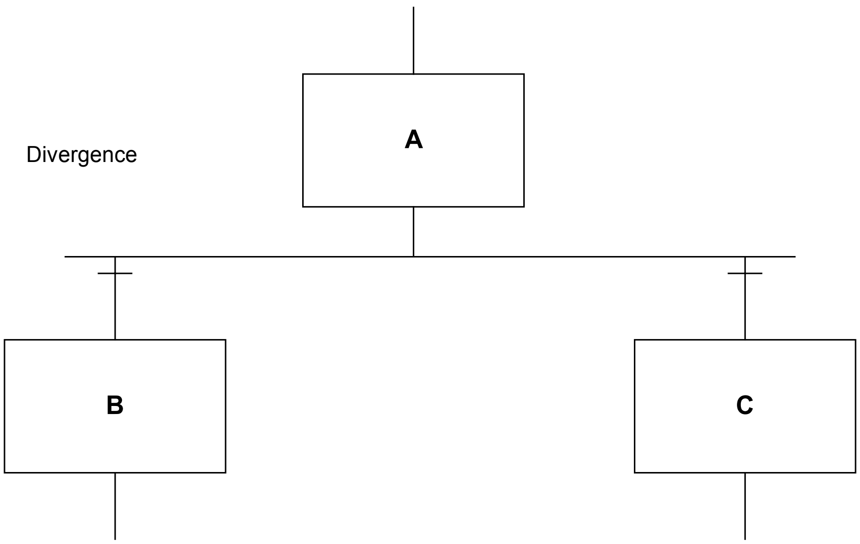
SFC is a programming language or nomenclature and documentation to represent executable procedures. *Also called Flowcharting*

Elements of a Sequential Function Chart

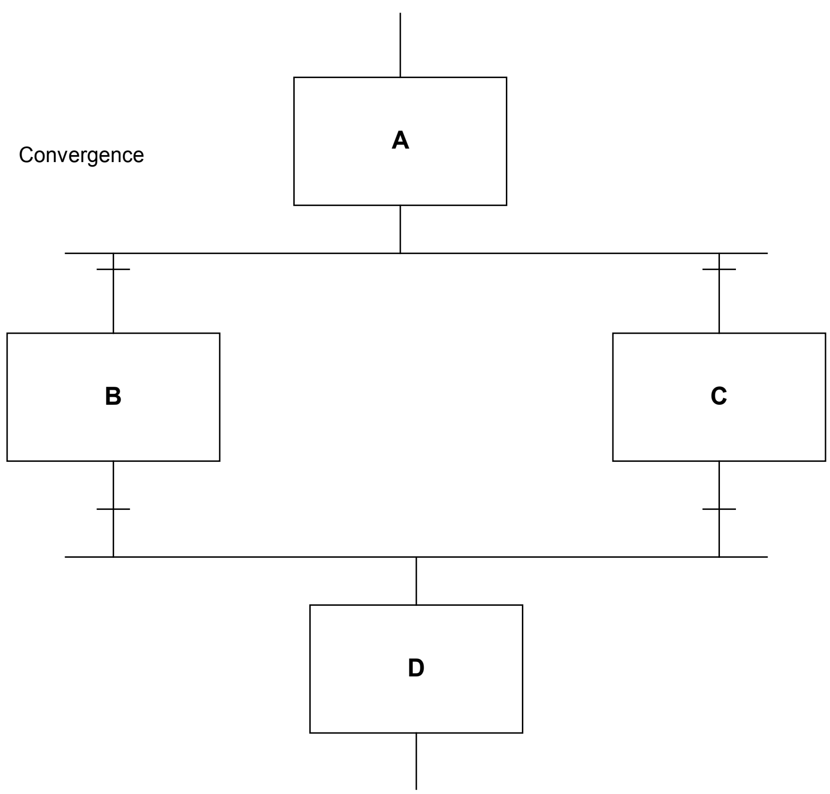


Double box is the initial step horizontal line shows transition, with description. Terminator is the ground symbol.

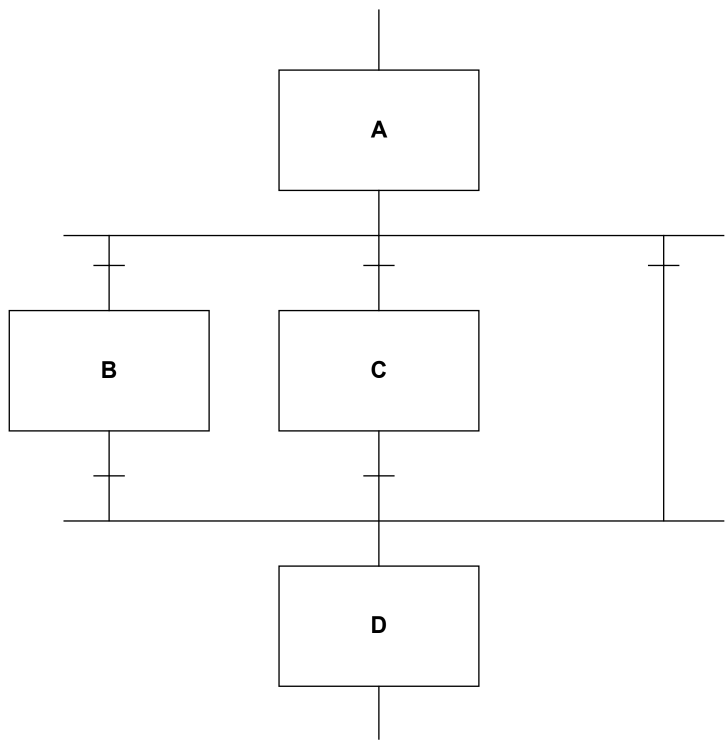
Note: In flow charting, the diamond is generally used as a decision point, yet I have not seen it used in any document describing S88.



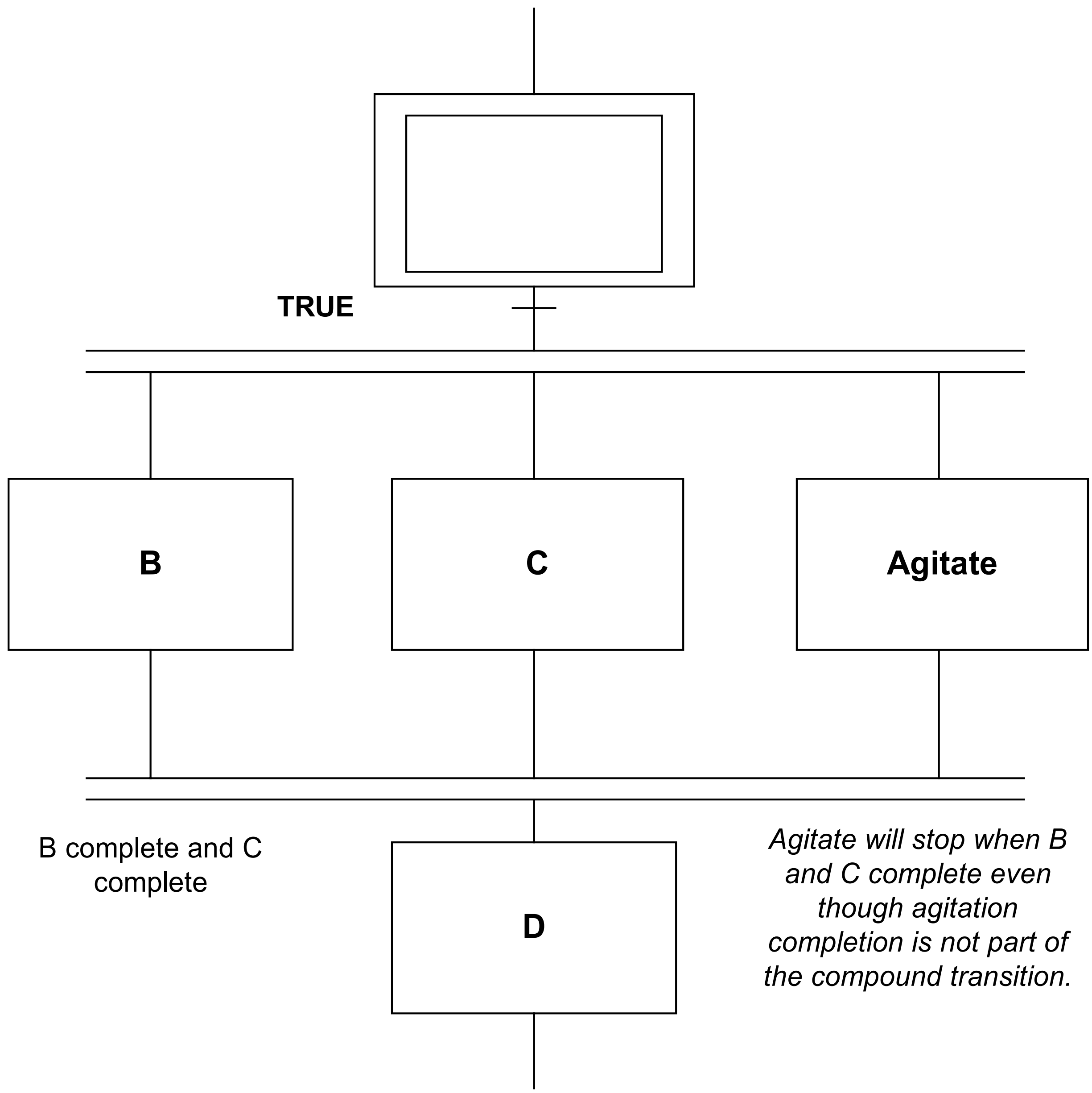
When A is complete, select either the path to B or C.



When A is complete, select either the path to B or C, then execute D.



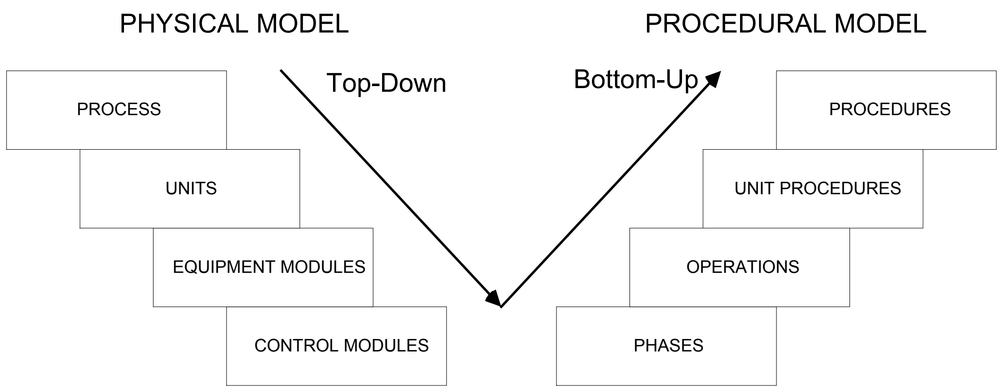
Skipping steps: When A is complete, select either the path to B or C, or skip to execute D.



Simultaneous Convergence

Mix Making Example
Tank and Valve diagram; Use a Process flow diagram, make several copies of each drawing and highlight the product flow. Very useful for complicated batch processes

The Physical and the Procedural Models

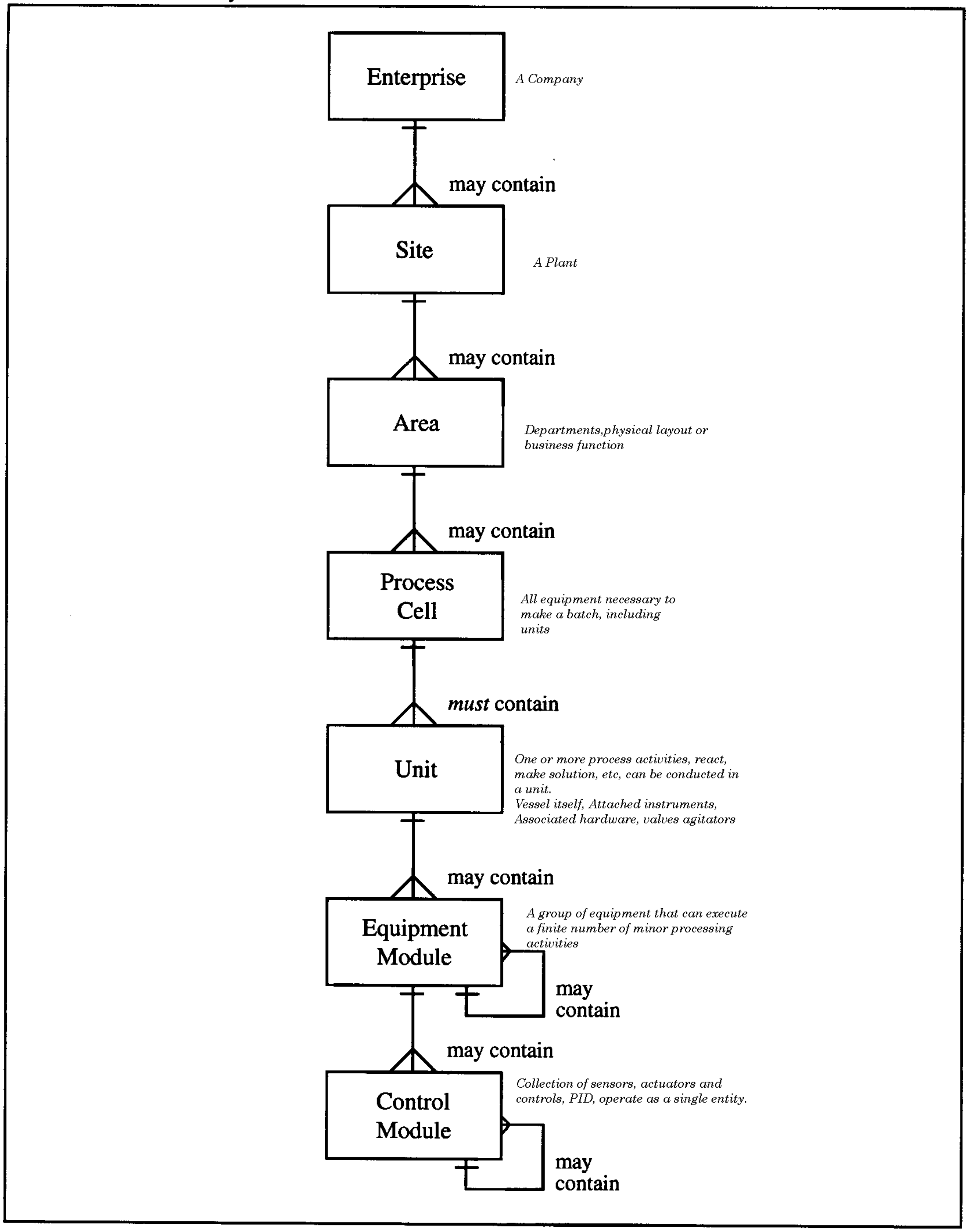


How to make the product? (Recipe)
What is the physical equipment to make the product?
How the process is run (Control activities)

S88 defines several "Models" that describe the equipment and how it operates
Major division between Physical Model and the Procedural Model

The Physical Model

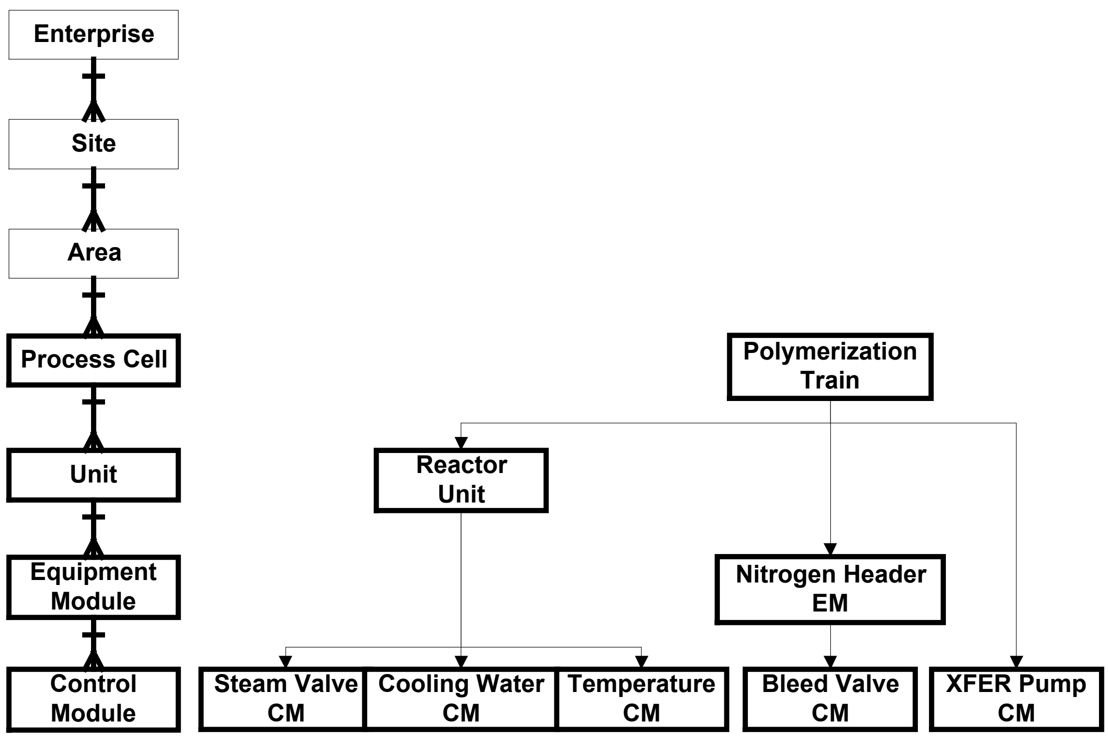
S88.01 Physical Model



S88.01 says:

"All control related sections of the standard assume that the process cell in question (both physical equipment and related control activities) has been subdivided into well defined equipment entities such as units, equipment modules, and control modules. Effective subdivision of the process cell into well defined equipment entities is a complex activity, highly dependent on the individual requirements of the specific environment in which the batch process exists. Inconsistent or inappropriate equipment subdivisions can compromise the effectiveness of the modular approach to recipes suggested by this standard.

Subdivision of the process cell requires a clear understanding of the purpose of the process cell's equipment. Such understanding allows the identification of equipment entities that must work together to serve an identifiable processing purpose"



The Physical Model

A *Process Cell* defines the span of logical control of one set of process equipment within an area that can be used to manufacture batch product

The domain for a batch control system is the *Process Cell*
Sometimes a cell is called a train

A *Unit*

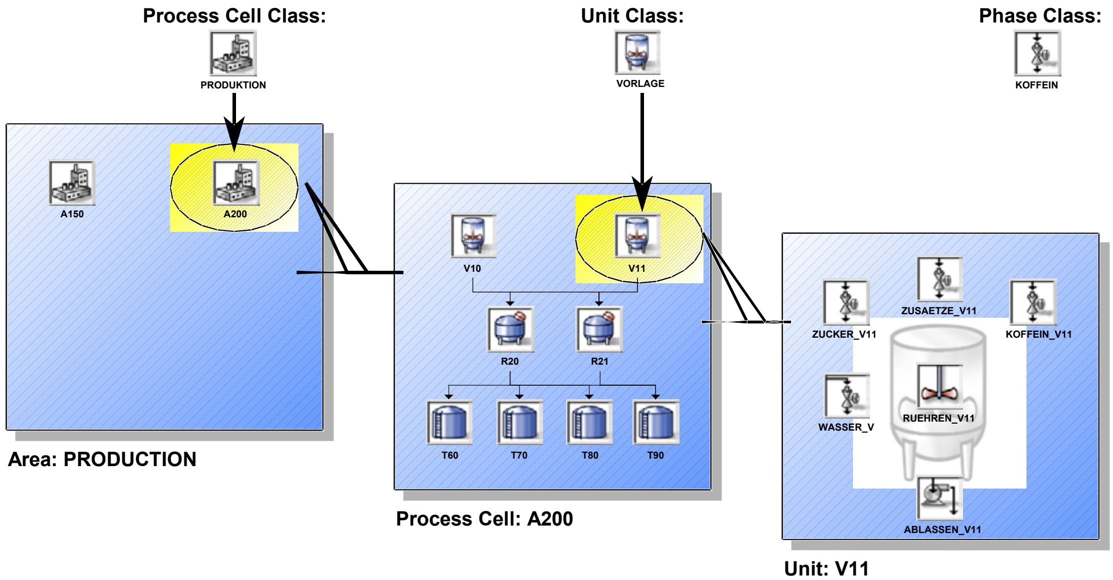
- **May contain a flexible amount of equipment, equipment modules and control modules**
- **Operates on all or part of the batch**
- **Only one batch at a time**
- **Cannot acquire another unit**
- **May operate independent of other units.**

An *Equipment Module*

- **Consists of equipment, control modules and other equipment modules**
- **Contain all equipment and control functions necessary to perform its process function**
- **Usually centered around a fixed piece of equipment, for example, Heat Exchanger, Weigh Tank, Agitator**
- **The scope of the equipment module is defined by the finite processing activity it is designed to carry out.**
- **May be part of a Process Cell, Unit, or another Equipment Module.**

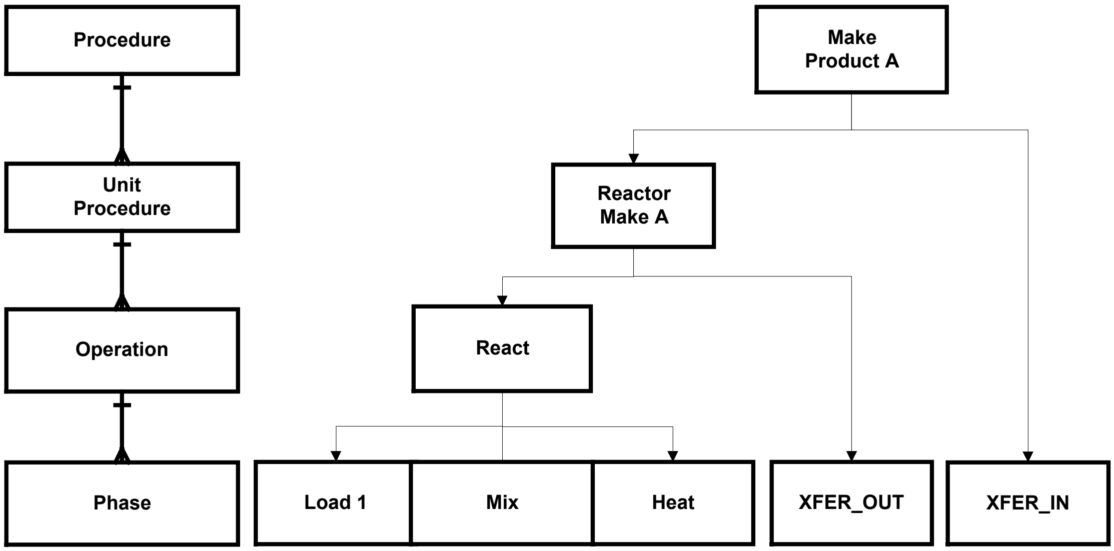
A *Control Module*

- **Control modules provide the interface to basic control**
- **Basic control may include regulatory control, or sequential control.**
- **Sequential control may involve state based control and exception monitoring/handling**
- **May be part of a Process Cell, Unit, Equipment Module, or another Control Module**

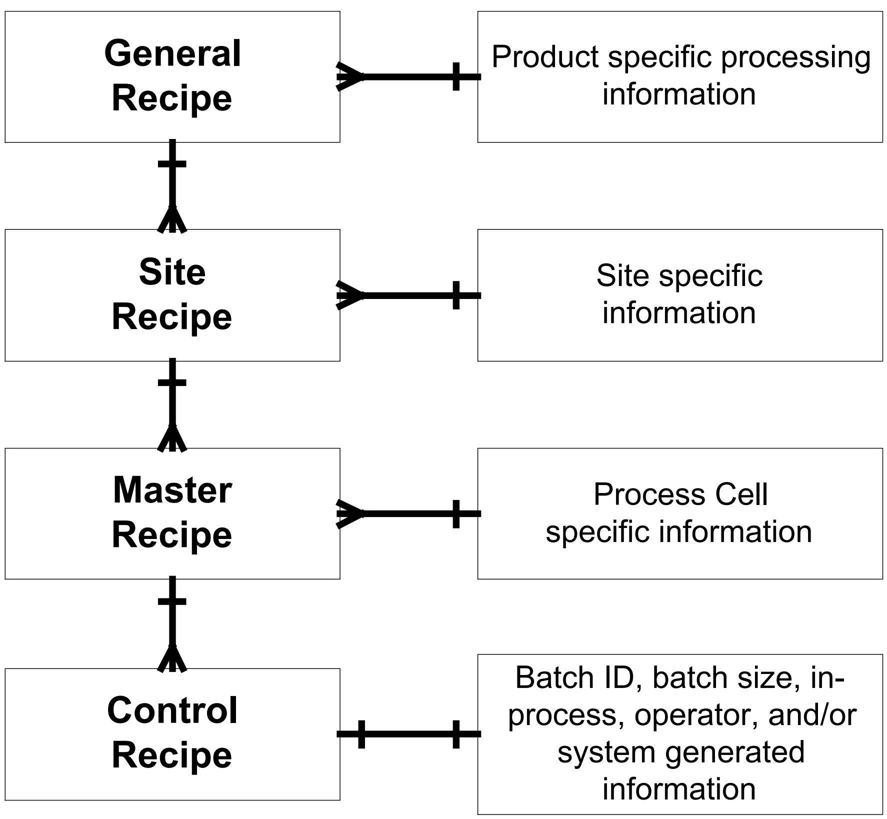


In this diagram, the cell is the production train, within the cell are the process units. Units are not necessarily “chemical unit operations”, S88 unit may run many different “chemical unit operations”, Example: Reaction and crystallize within the same vessel.

The Procedural Model



Based on the concept of a *Recipe*



General Recipe – Corporate planning investment decisions, Research, *Intellectual Property*

Site Recipe – In local language derived from the general recipe.

Master Recipe – Targeted to a specific cell. Engineering related, specific agitation, metering details, control requirements

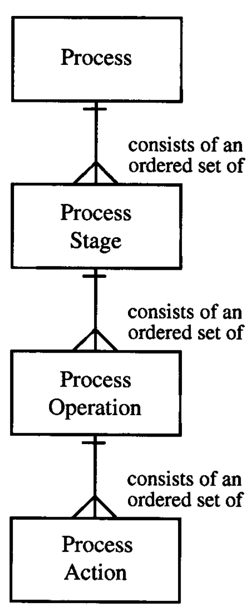
Control Recipe – Used to create a specific batch, has Batch ID, operator and other information unique to that specific batch.
Example: You can track, through the Lot number on a pill bottle, where and when it was made, all the details about manufacturing, operators on duty etc. *All the things that make the FDA great.*

Recipe Contents

Header	**Administration information and process summary**
Equipment Requirements	**Information about specific hardware to make the batch**
Procedure	**Defines the strategy for carrying out a process**
Formula	**Recipe formula, chemicals, process parameters and inputs**
Other Information	**MSDS (Material Safety Data Sheets), Regulatory, FDA GMP etc.**

Recipe Procedures defined by two models, Process Model and Procedural Control Model
Procedural Model

General and Site Recipe Procedures Are Based on the Process Model



Process Stages for PVC

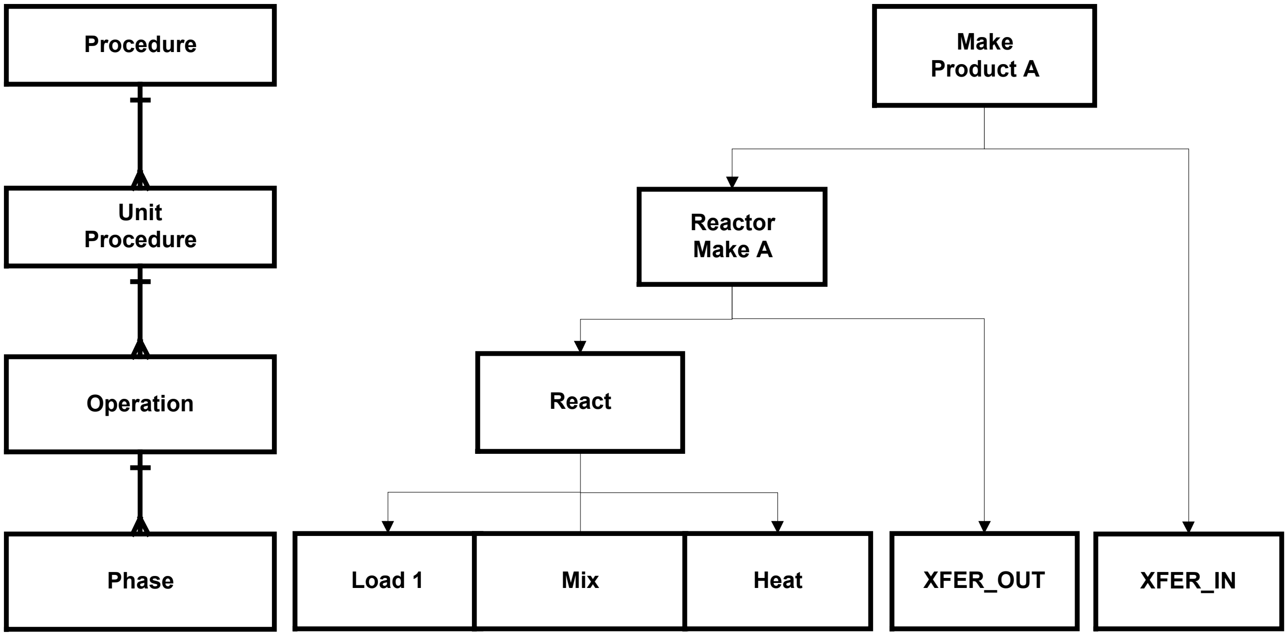
Process Stage	Description
Polymerize	Polymerize vinyl chloride monomer.
Recover	Recover residual vinyl chloride monomer that did not polymerize.
Dry	Dry the polyvinyl chloride.

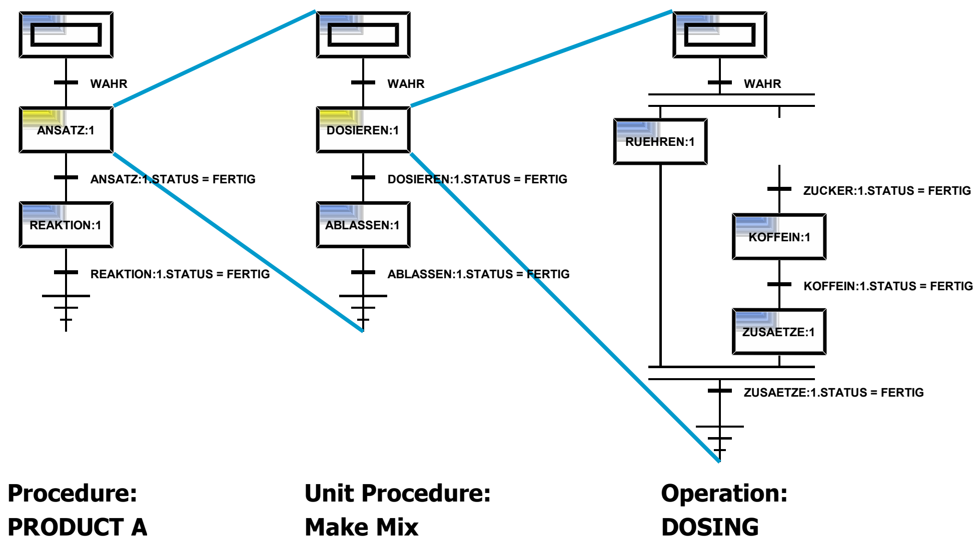
Operation of a Process Stage

Process Operation	Description
Prepare Reactor	Evacuate the reactor to remove oxygen.
Charge	Add demineralized water and surfactants.
React	Add vinyl chloride monomer and catalyst, heat 60°C, and hold until reactor pressure decreases.

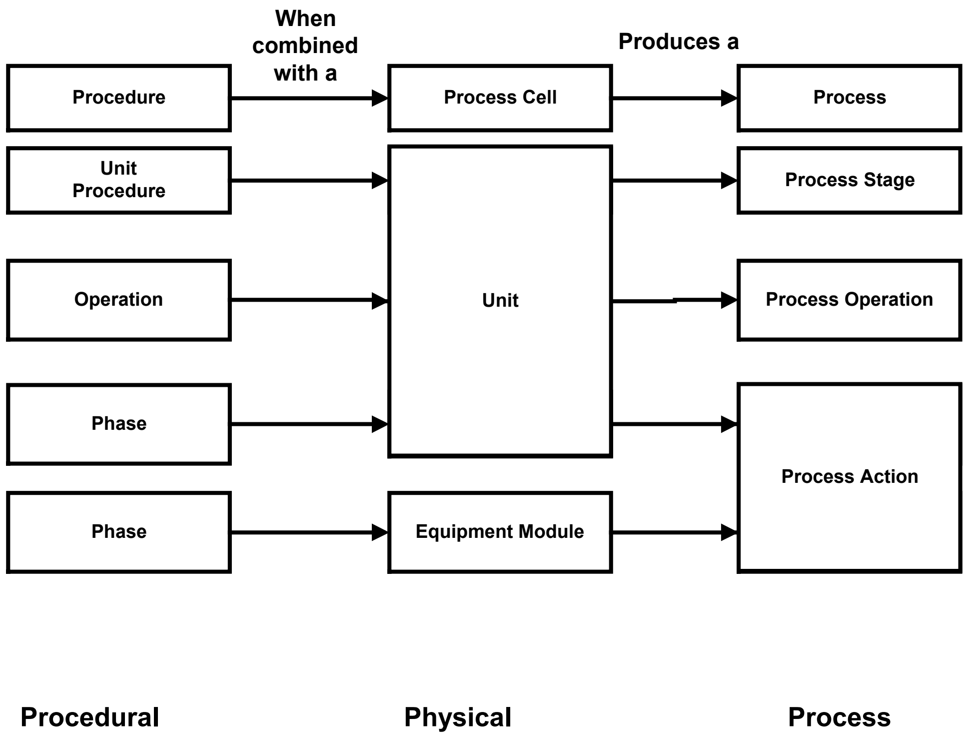
Actions in a Process Stage

Process Action	Description
Add	Add required amount of vinyl chloride monomer to reactor.
Add	Add required amount of catalyst to reactor.
Heat	Heat reactor contents to 60°C.
Hold	Hold reactor contents at 60°C until reactor pressure decreases.





Procedural Control Recipes are grouped as shown above. The “Phase” not shown
A phase is the smallest element of control. Usually executed in a PLC or DCS.
A phase can operate one or more pieces of equipment.



Linking the Physical, Procedural Control and Process Models

Modes of Operation

"The manner in which the transitions of sequential functions are carried out within a procedural element or the accessibility for manipulating the states of equipment manually or by other types of control."

Possible Implementations of Suggested Modes for Procedural Elements

Mode	Behavior	Command
Automatic	Transitions within a procedure are carried out without interruption as necessary conditions are met.	Operators may pause the progression but may not force transitions.
Semiautomatic	Transitions within a procedure are carried out on manual commands as necessary conditions are met.	Operators may pause the progression and redirect the execution to an appropriate point, but operators may not force transitions.
Manual	Procedural elements within a procedure are executed in the order specified by an operator.	Operators may pause the progression and may force transitions.

Possible Implementations of Suggested Modes for Equipment Entities

Mode	Behavior	Command
Automatic	Equipment entities are manipulated by a control algorithm.	Equipment entities cannot be controlled by an operator.
Manual	Equipment entities are not manipulated by a control algorithm.	Equipment entities can be controlled directly by an operator.

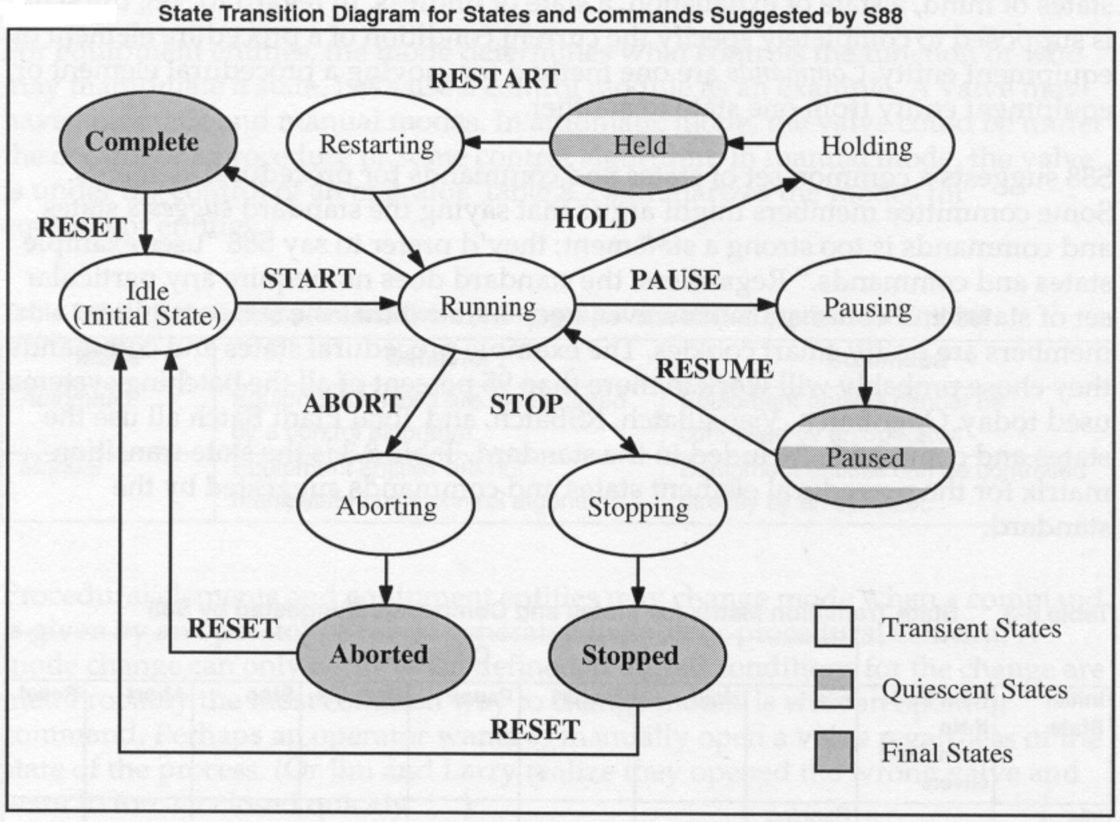
States and Commands

A State completely specifies the current condition of a procedural element or equipment entity.
A Command is a method of moving a procedural element or equipment entity from one state to another.

State Transition Matrix for States and Commands Suggested by S88

Initial State	Next State if No Command Given	Command: Start	Hold	Restart	Pause	Resume	Stop	Abort	Reset
Idle		Running							
Running	Complete		Holding		Pausing		Stopping	Aborting	
Complete									Idle
Holding	Held						Stopping	Aborting	
Held				Restarting			Stopping	Aborting	
Restarting	Running		Holding				Stopping	Aborting	
Pausing	Paused		Holding				Stopping	Aborting	
Paused			Holding			Running	Stopping	Aborting	
Stopping	Stopped							Aborting	
Stopped								Aborting	Idle
Aborting	Aborted								
Aborted									Idle

State Transition Diagram for States and Commands Suggested by S88



Procedural States Suggested by S88

Idle	The procedural element is waiting for a **START** command that will cause a transition to the *Running* state.
Running	Normal operation.
Complete	Normal operation has run to a normal completion. The procedural element is now waiting for a **RESET** command that will cause a transition to the *Idle* state.
Holding	The procedural element has received a **HOLD** command and is executing its separate *Holding* logic to put the procedural element into a known condition. Once the *Holding* logic completes, the procedural element transitions automatically to the *Held* state. If no special sequencing is required to place the procedural element into a known condition, the procedural element transitions immediately to the *Held* state.
Held	The procedural element has completed its *Holding* logic and has been placed into a known or planned condition. The procedural element is now waiting for a command to proceed. This state is usually for longer-term batching interruptions.
Restarting	The procedural element has received a **RESTART** command while in the *Held* state and is executing its *Restarting* logic in order to return to the *Running* state. If no restarting sequencing is required, then the procedural element transitions immediately to the *Running* state.
Pausing	The procedural element has received a **PAUSE** command. This will cause the procedural element to stop at the next defined safe or stable location in its *Running* logic. Once the defined safe or stable location is reached, the state automatically transitions to *Paused.*
Paused	State reached once the procedural element has reached the next defined safe or stable location after a **PAUSE** command. A **RESUME** command causes a transition to the *Running* state, resuming normal operation immediately following the defined safe or stable location. This state is usually for shorter-term batching interruptions.
Stopping	The procedural element has received a **STOP** command and is executing its *Stopping* logic, which sequences a controlled, normal stop. If no stopping sequencing is required, then the procedural element transitions immediately to the *Stopped* state.
Stopped	The procedural element has completed its *Stopping* logic and is waiting for a **RESET** command to transition to the *Idle* state.
Aborting	The procedural element has received an **ABORT** command and is executing its *Aborting* logic, which sequences a quicker, but not necessarily controlled, abnormal stop. If no aborting sequencing is required, then the procedural element transitions immediately to the *Aborted* state.
Aborted	The procedural element has completed its *Aborting* logic and is waiting for a **RESET** command to transition to the *Idle* state.

Procedural Commands Suggested by S88

Start	Orders the procedural element to execute its normal *Running* logic. Only valid while the procedural element is in the *Idle* state.
Hold	Orders the procedural element to execute its *Holding* logic. Only valid while the procedural element is in the *Running*, *Pausing*, *Paused*, or *Restarting* states.
Restart	Orders the procedural element to execute its *Restarting* logic to safely return to the *Running* state. Only valid while the procedural element is in the *Held* state.
Pause	Orders the procedural element to pause at the next programmed pause transition within its normal *Running* logic and await a **RESUME** command before proceeding. Only valid in the *Running* state.
Resume	Orders the procedural element to resume execution in its normal *Running* logic after it has paused at a programmed transition as a result of either a **PAUSE** command or a *Semiautomatic* mode. Only valid in the *Paused* state.
Stop	Orders the procedural element to execute its *Stopping* logic. Only valid while the procedural element is in the *Running*, *Pausing*, *Paused*, *Holding*, *Held*, or *Restarting* states.
Abort	Orders the procedural element to execute its *Aborting* logic. Valid while the procedural element is in every state *except Idle*, *Complete*, *Aborting*, and *Aborted*.
Reset	Orders the procedural element to transition to the *Idle* state. Only valid while the procedural element is in the *Complete*, *Stopped*, and *Aborted* states.

The present participle form, ___ing, is usually a separate block of code in a phase. The execution of this code is transitory. Completing this state results in placing the system in a quiescent or final state. Note that "running" is considered transitory.

General Comments, Experience

When designing these systems, the database or noun, the Physical Model description is the most important. It is usually easier to change some phase logic than it is to add or delete a Control or Equipment module. This is very important when a compiled language is used for generating the code.

Americans tend to emphasize the verbs and not the nouns. Example: German is very noun orientated language. This results in descriptive verse that appears "foreign" to Americans. I have found that translation of operating instructions makes control definition more difficult.

S88, by design, separates the equipment from the recipe, assumes a "pots and pan" approach. Example: Very few cookbooks describe in detail the exact design of a pot or pan. S88 defines equipment in a generic form. For many processes, this will not completely describe the whole process. In many industries, the processing equipment design is integral and unique to the process i.e. it cannot be made any other way. Example Electronic grade silicon makes use of very specialized machines and equipment. Design of this equipment is considered intellectual property to the company. S88 can be used to describe the batch control description, but it is not complete enough to describe the whole process.