Draft 1: March 2006 Make2Pack WBF Report

(Note: This effort is a work in progress and the material in this report is subject to change. To stay current use the WWW.Mak2Pack.org web site. This document is best used in conjunction with the Presentation from the 2006 WBF North American Conference available through the web site)

Background:

Make2Pack was conceived of in January 2004 and became a workgroup sponsored by the WBF in May 2004 as a joint effort between the WBF, OMAC OPW workgroup and ISA with it's first working meeting in August of 2004 at Pfizer's Peapack location. As of March 2006 the workgroup has held over 5000 meeting hours in various locations around the world with a roster of 194 people representing 96 companies.

Following is the description and objectives from the charter:

Make2Pack Project Description
The Make2Pack project will develop ***conceptual models and terminology*** for industrial automation that can be consistently applied to the total manufacturing process. This includes making/converting and packaging. The results will enable end users to reduce overall costs and enhance responsiveness to changing business requirements. Business owners universally recognize the need to overcome organizational and business process barriers (silos) within their manufacturing operations. Traditionally, technology has imposed artificial barriers or gaps between manufacturing operations. The goal of this project is to bridge these barriers and gaps by consciously identifying and applying underlying, unifying principles of manufacturing automation for the first time. The initial focus will be on packaging and converting machinery, and batch processing equipment. The result of this effort will provide mutual benefit for participating end users, machinery suppliers and automation technology providers. The result of this work will be increased modularity, reusability and a common structure which together will significantly streamline development, integration, and programming efforts for new initiatives as well as improving maintenance and support. There will be several example applications that can be used to guide implementations.

Objectives
• Harmonize OMAC PackML guidelines with ISA S-88.00.01 and .02, and the upcoming S-95 Part 3 standards. • Identify harmonization principles. • Develop recommendation for submitting PackML as an ISA and IEC standard. • Develop cross-reference naming conventions and models between S-88 and PackML. • Develop required definitions and models.

Goals:

The goals of Make2Pack are to improve the functionality of automated manufacturing and to reduce the costs in defining and delivering these automated systems. One of the approaches being used is to identify common methods of automation across all of manufacturing that encourage modular and reusable components. Mak2Pack has identified several examples where this has been accomplished on a non-standard basis (internal use by integrators and corporations) where the benefits have been documented and are significant, in some cases reducing the design and delivery effort by 80%. The Batch Process Industry has been applying modular techniques based upon several different understandings and interpretation of S88. This has lead to that industry realizing significant benefits and as the different approaches are reconciled the benefits continue to grow. Make2Pack is using the S88 series of standards and the work by the OMAC PackML workgroup as well as many other internally recognized standards as a basis for its work. In its efforts Make2Pack has derived several new concepts and is providing recommend clarity to the existing body's of work as a starting point for the next goal of delivering an international standard that supports automation approaches and techniques across all of manufacturing. This effort has been embraced by the ISA and its SP88 committee and will progress under several working group committees with some results ready for ballot expected in early 2007.

Make2Pack is developing examples which support all aspects of manufacturing that are useable as a method to effectively educate a wide audience about the basic concepts are also part of the goals. These examples are expected to be used throughout the industries that supply platforms, services and the end users of automation technologies to form a part of industry wide tools for understanding and clear communication.

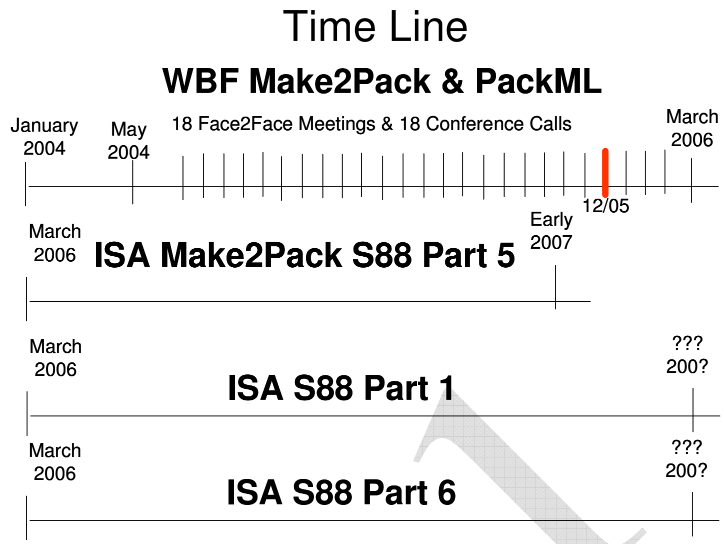
Another goal of Make2Pack is to develop methods to create libraries of commonly used automation components that users can specify and vendors can supply as an integral part of any automation application. These components will be able to communicate horizontally across equipment boundaries and vertically to business management systems without requiring any specialized customization providing common functionality across platforms.

In some industries manufacturing is viewed a not having the flexibility and agility to meet rapidly changing business requirements and is viewed as a tactical liability. Another goal of the Make2Pack effort is to provide the required flexibility and agility to manufacturing, and the engineering efforts that support it, such that it joins the other components that make up the business enterprise as valued strategic assets.

Make2Pack Current Status

As the Make2Pack work progresses, similar to what happened with the S88 Part 1 standard, this body of work is being used by suppliers and integrators to guide in the deliver of systems that can meet the intent of what the final standard will define. This

Draft



effort will continue as the ISA-SP88, Batch Control Part 5: Modular Concepts for Automated Control Systems working group and will continue to keep not only the Make2Pack name but many, if not all, the people who have contributed greatly to it and gain others with new perspective and abilities. As you will see there are several spin-off opportunities from Make2Pack learning that others will be taking for farther development, items such as ERP to Recipe communication and Recipe Phase to Equipment Phase communication and interaction and possible enhancements to the Part 1 standard itself.

The PackML State Model

Several years ago the OMAC OPW PackML work group developed a state model to describe the behavior and any packaging machine as shown in Figure 1: PackML State Model and the S88 Control Activity Model. This effort also defined detailed "TagNames" and data types for a method of intercommunication between applications for the implementation of this model. This work was originally believed to be analogues with the S88 Equipment Phase State Diagram which is also shown in Figure 1. It is important to note that the model was only valid when a packaging machine is in the Full Automation Mode of operation. Modes and their relationship to complex automation applications will be discussed later.

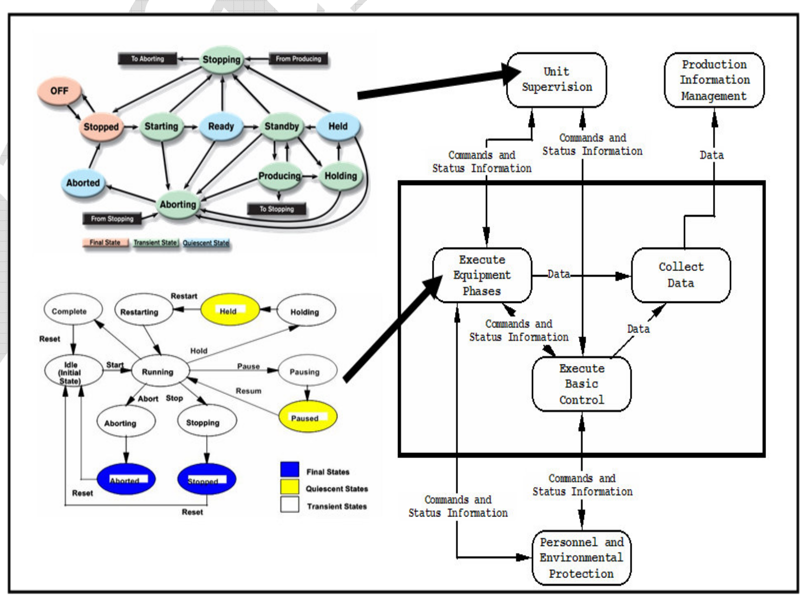


Figure 1: PackML State Model and the S88 Control Activity Model

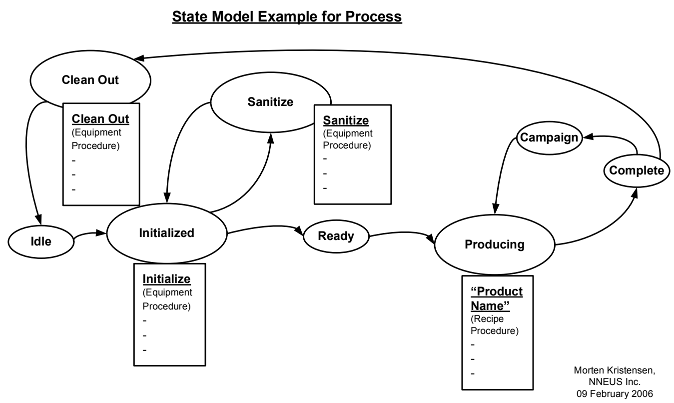


Figure 2: Possible Batch Unit State Model

During the effort to define what was different between the two state models the group became aware that each model was describing a different part of the control activity that occurs in manufacturing out side of process control as defined by S88. The PackML model represents what is occurring at the Unit Supervision activity and the S88 Equipment Phase Model represents an example of what the states that might exist within an Equipment Module executing a procedural equipment control action. In the end these two models are complementary and do not conflict at all. It also demonstrates an improvement area in S88 where it is possible to represent Unit Supervision using a state model as shown in Figure 2. This allows the intermixing of Recipe Equipment Procedures and Recipe Control Procedures. This approach allows for the farther separation between product and equipment that in many of today's approach have been forced into the Control Recipe.

PackML States as Operations

The concept of each state being an operation that is made up of Equipment Phase References is proving to be as powerful a concept in packaging machinery as it has in batch processes. Make2Pack is suggesting the models used to represent the differences between a control recipe and an equipment recipe be modified as shown in figure 3. In this figure the equipment operations are made up of an ordered list of Equipment Phase References, which is analogous to a Recipe Phase, that identifies the actual Equipment Phase that is to be interacted with to carry out the intended process function. In this representation there is no difference between the recipes other than residency.

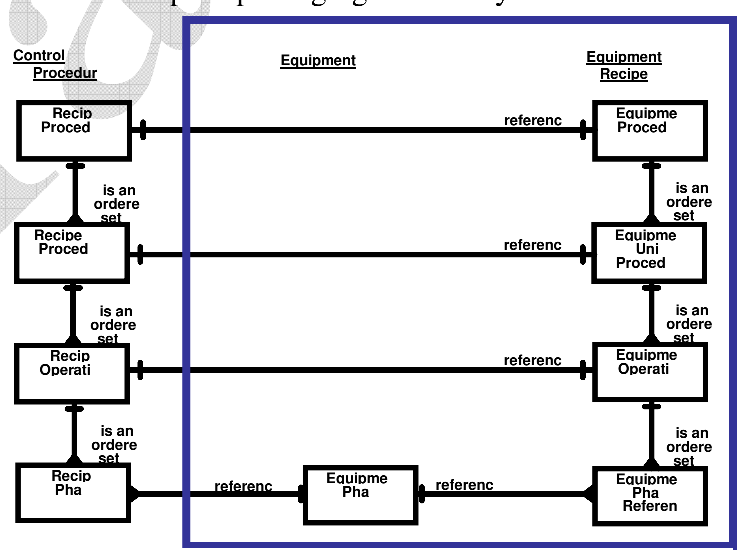


Figure 3: Equipment Phase Referance

Draft 1

Process Models

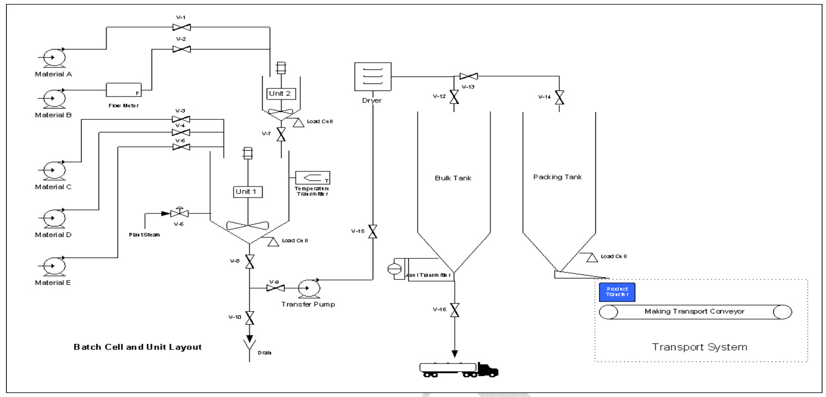


Figure 4: Batch Process

As part of the Make2Pack effort the need to develop different process models to help the group communicate became quickly evident. To that end the group developed three simple models, one for a Batch Process Figure 4), one for a Continuous Process (Figure xx) and one for a Form Fill Packaging Machine (Figure 5). The details about each model can be found in the Appendix. Using these models we are able to describe required functionality and how it can implemented using the automation techniques for that industry. This proved to be a very useful tool for understanding between our different groups. Our intent is to expand the use of these models in documentation and training so that they can form a commonly used set of process models that all can feel comfortable with when exploring these difficult concepts. The Make2Pack group also devised very simple prototype systems to demonstrate the feasibility of having different control vendors provident the same functionality for several basic automation components in the same way such that it looked and acted the same from the end users viewpoint. This in no way reduced the vendor's ability to provide significant IP to differentiate their platforms from others.

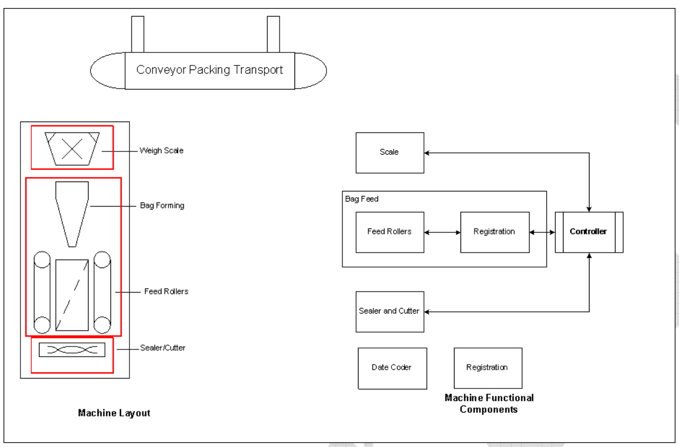


Figure 5: FFS Packaging Machine Model

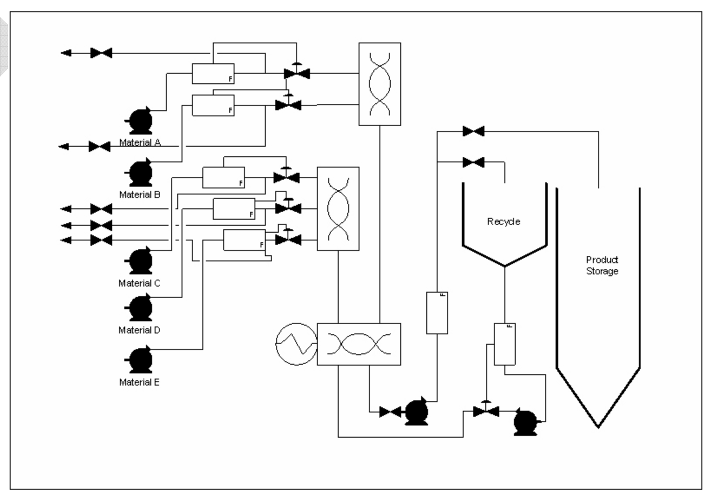


Figure 6:Continuous Model

Figure 6 represents a continuous process model that will be used to demonstrate how the modular techniques an be used to satisfy the automation needs of that discipline.

These models are being proposed as a basis for use in future programs as a method to connect with students as they learn all aspect of automation. Providing a common touch stone that will help them stay well grounded and more easily relate the subjects and how they interact.

Modular Decomposition in Packaging

After quite a bit of learning about the different disciplines and several iterations Make2Pack found they were able to apply the techniques used in batch process of decomposing equipment down to the smallest reasonable level, as detailed in the S88

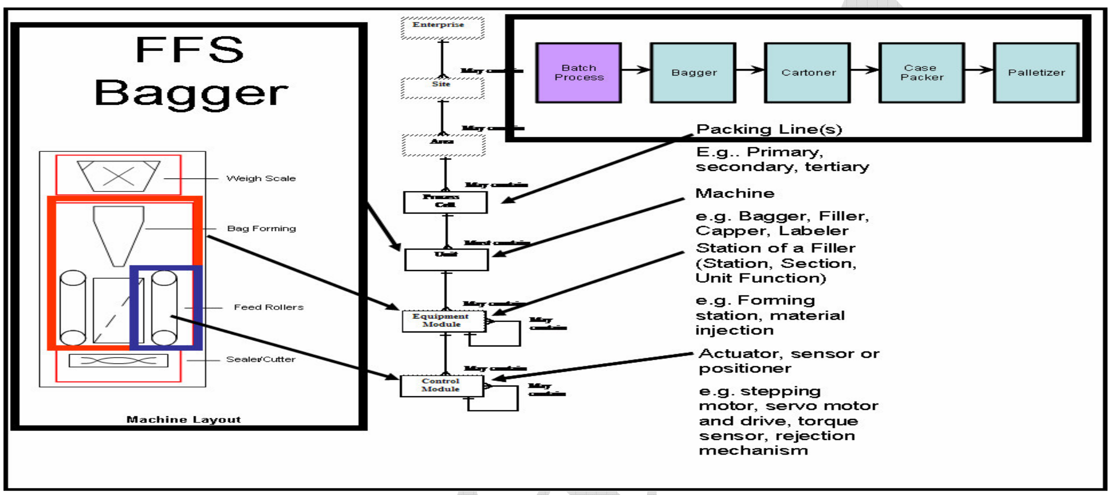
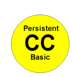


Figure 6a:FFS Packaging Machine Decomposed

Equipment Model, against a packaging machine. The Packaging line is analogues to the Batch Process Cell and individual packaging machines are analogous to batch units as shown in Figure 6a.

The FFS is farther decomposed into Equipment and Control Modules as shown. There is discussion within the Packaging Industry of how to apply these techniques against the physical design of a machine and not just the automation creating even more flexible systems.

Equipment Phase  and Control Module  Component of Control

The idea of bounding the software that provides the functionality of automation components of control and then associating that software with the physical components in such a way that the relationship is clear creates a modular relationship between the physical and logical. The orange hexagon is a representation of the software that is used to carry out the equipment procedural logic of the associated physical equipment. The yellow circle is a representation of the software that will carry out the basic control associated with the physical equipment being controlled. Continuing this decomposition at the Equipment Module and Control Modula level creating a hierarchy of components of control (see definitions and refer to the 2006 WBF Make2Pack Presentation) that will direct and command lower levels of components of control allow the creation of the simplest possible control functions. These simple components of control become reusable building blocks that are used to address the majority of automation requirements. Figure 8 represents the command and status structures that are currently thought to represent all basic and equipment phase component of control types. Creating the proper balance is of

Command	Status
Manual Mode Request	Manual Mode
Auto Mode Request	Auto Mode
Bypass Request	Bypassed
Controlled Request	Controlled
Acknowledge Alarm	Alarm Acknowledged
Reset	Reset
	Alarm Present
	Command Rejected

Figure 8: Component of Control Command and Status

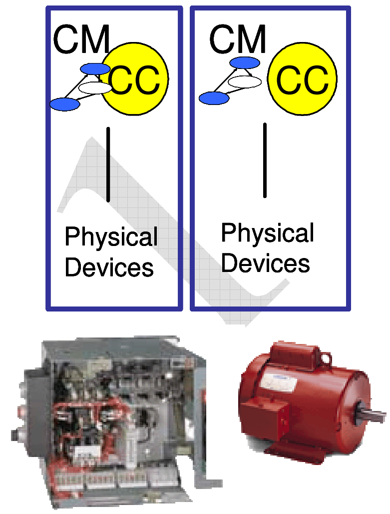


Figure 7:Two Control Modules

modularity requires a level of skill that today is quite subjective and the number of levels will vary. As the industry continues to evolve the Make2Pack library of components this will require less skill and become more of a configuration effort than a custom programming effort. Make2Pack is defining the method of communication between all automation components of control.

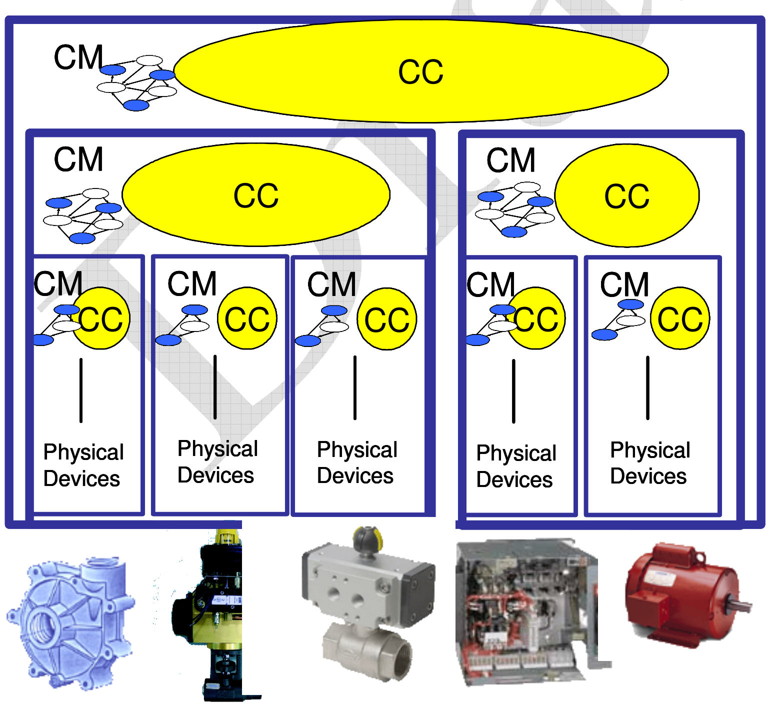


Figure 9:Three Levels of Control Modules

Figure 7 represents the lowest level of a hierarchy of supervisory Control Modules where a physical device (Per the S51.1 standard) is managed by it's Basic Component of control. In this example there are two CMs managing two physical Devices. The state diagrams for each of the CMs reflect what the physical device is actually doing and each CM can be commanded by other supervisory (Current thinking is these are Internal Agents that are operating within the normal designed automatic design of the automation). Or can be commanded by an External Agent if the CM is in the proper “Mode” to receive commands from outside of the normal designed activities due to an upset that was not planned for by the normal design. This external agent only has the ability to interact with the automation components of control as designated by the specification of the automation system. For detailed information on these concepts see the section on Modes and Agents. Figure 9 represents a three level hierarchy of modular Control Modules that make up a process function using several physical devices.

S88 does not require Equipment Phase to only have the example state diagram shown in the part 1 standard. Most systems available on the marked today only support the Equipment Phases functionality represented in the example and require it to be used in the created Equipment Phase Component of Control that it will interact with. The S88 standard actually encourages the creation of Equipment Phases with states that address the specific requirement that they are intended to satisfy as represented in Figure 10:Equipment Modules. These components of control can be and are commanded and directed by other components of control to carry out the design intent of the system.

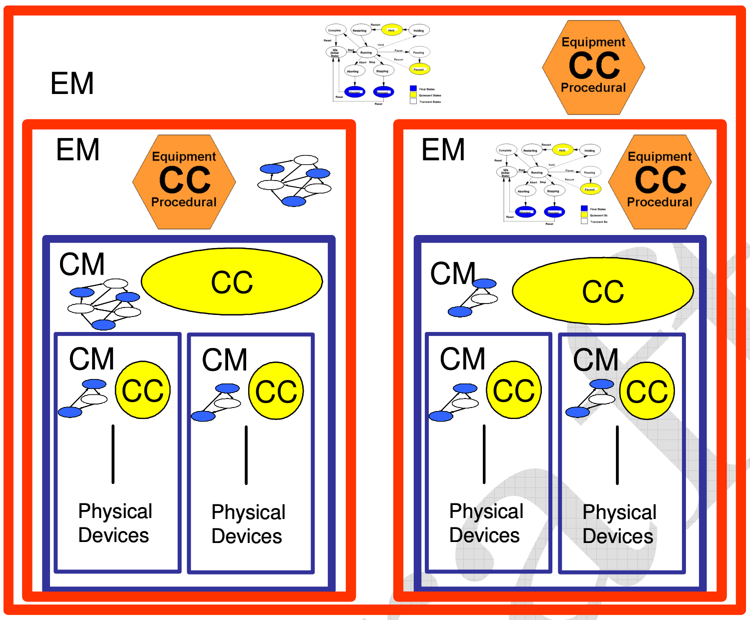
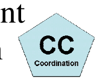
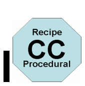


Figure 10:Equipment Modules

The types of control performed are in line with the four types defined by the S88 standard with the simplest example of Control Modules performing Basic Control and Equipment Modules performing Equipment Procedural Control and both performing Coordination Control (represented by a blue pentagon and sometimes referred to as arbitration) where necessary. Coordination Control at the control module level is generally simple first in first out type of equipment coordination. More complex forms, such as multi use and priority selection, might occur higher in the architecture with the Procedural Logic of the Equipment Phase being used to manage this or a higher level supervisory Control Module at the Unit Level. Very complex forms might be managed by other systems used to supervisory the unit or process cell. Some products that provide Recipe Components of Control also provide services that assist in managing complex equipment coordination requirements.



Recipe Components of Control



Each part of a recipe has a component of control associated with it that will manage its execution based upon the requirements of that recipe. The blue octagon represents the software that will mange the procedural part of recipe control. Any number of these components of control can be associated with any part of the recipe model or there can be one that manages all aspects of the recipe, this is up to the requirement of the user. This is where the Unit Supervision occurs and is where Recipe Procedural Control occurs. These components of control can be purchased products such as batch recipe sequencers that reside in a supervisory system or can be custom programmed and reside in the equipment control platform. In either case the functions performed are the same. This type of control occurs at the Unit/Machine and Process Cell/Packaging Line levels. Coordination Control occurs at these levels also as necessary.

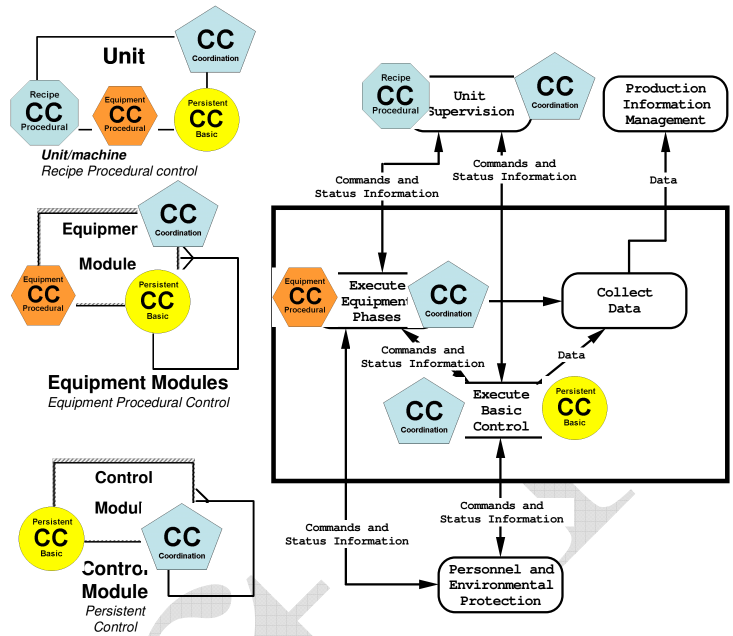


Figure 11:Super Set Software Residency

Software Residency

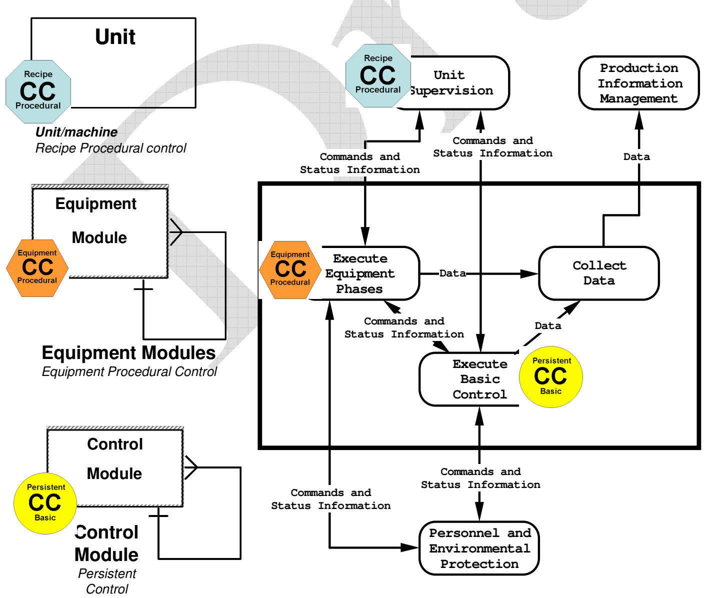


Figure 12:Simple Software Residency

It is important to be able to visualize and manage the actual software that makes up the entire automation used to control and manage manufacturing. S88 allows many different options in how to configure and construct the software used to carry out the intent of any application. With the great concept of collapsibility the approaches can and do become quite varied. Software in the automation field has been described as an odorless, colorless and formless element that in high concentrations may be hazardous to health and seems to be addictive. It has also been described as the soft nougat center that is found inside the hard shell of control and equipment modules. Managing the concentration of software used in automation by applying the modular concepts of S88 creates manageable and useable automation that consists of reusable components of control and can be used across the different automation needs found in manufacturing. Being able to visualize where that control resides within the physical world is also a powerful concept and helps manage the software and the equipment. Figure xx represents a supper set view of possible residency of the components of control that are possible under S88. Figure xx represents a simple view of a possible implementation that is easier to get the concept across to people newly exposed to the S88 Modular concepts. ,

Modes and Agents

March 02 2006 Make2Pack Modes of Operation Proposal

Josiah Hoskins

Review Modes in the S88 Standard

We will proceed as follows:

- The text of the Make2Pack Modes Proposal will be in Times New Roman font with a size of 12.
- Reference text from the S88 Standard will be in Arial font, with a size of 11. [1]
- Reference text from "Applying S88" will be in the Times New Roman font and referenced as noted. [2]
- We will first present the text from the S88 Standard relating to modes which will be followed by new interpretation and extension of the S88 Modes of Operation including supporting analysis and rationale.

Let us start with a review of the S88 Standard:

5.7 Modes and states

This section discusses the modes and states of equipment entities and of procedural elements. In the preceding sections, models describing equipment entities and procedural elements have been defined. In these models, transitions for procedural elements and for equipment entities occur within each hierarchical level. The status of equipment entities and of procedural elements may be described by their modes and states. Modes specify the manner in which these transitions take place; states specify their current status. Other resources, such as materials, may also have states.

5.7.1 Modes

Equipment entities and procedural elements may have modes. Example modes are described in this standard in relation to batch control. The mode of an equipment entity may be based on procedural elements or equipment entities utilizing basic control functions, depending on the main control characteristic of the entity.

This standard uses as examples three modes (automatic, semi-automatic and manual) for procedural elements, and two modes (automatic and manual) for equipment entities. Control modules contain basic control functions and will have automatic and manual modes, whereas a unit running procedural control would also have a semi-automatic mode.

This standard does not preclude additional modes or require the use of the modes defined here. The functionality of the modes presented is felt to be generally useful in most batch applications. By naming the modes and including them in the standard, a defined set of terms is documented that can be used when communicating on batch control issues.

A mode determines how equipment entities and procedural elements respond to commands and how they operate. In the case of procedural elements, the mode determines the way the procedure will progress and who can affect that progression. In the case of a control module, such as an automatic block valve, that contains basic control functions, the mode determines the mechanism used to drive the valve position and who/what, such as another device or an operator, may manipulate it to change its state.

For procedural elements, the mode determines the way the transitions are treated. In the automatic mode, the transitions take place without interruption when the transition conditions are fulfilled. In the semi-automatic mode, the procedure requires manual approval to proceed after the transition conditions are fulfilled. Skipping or re-executing one or more procedural elements, without changing their order, is usually allowed. In the manual mode, the procedural elements and their order of execution are specified manually.

For equipment entities containing basic control functions, the mode determines how their states may be manipulated. In automatic mode equipment entities are manipulated by their control algorithms and in manual mode the equipment entities are manipulated by an operator.

Table 1 lists possible behaviors and commands associated with the example modes.

Equipment entities or procedural elements may change mode. This change can occur if the conditional logic requirements for the change are met by internal logic or by an external command such as one generated by another procedural element or by an operator. A mode change takes place only when the conditions for the change request are met.

A change of mode in one equipment entity type or procedural element type may cause corresponding changes in other types. For example, putting a unit procedure to the *Semi-automatic* mode may cause all lower-level procedural elements in that unit to go to the *Semi-automatic* mode, or, a safety interlock trip may cause several control modules to go to the *Manual* mode with their outputs at minimum value. The propagation can be in either direction, from a higher level entity to a lower level entity, or conversely. This standard does not specify propagation rules.

Table 1 - Possible Implementations of example modes

Mode	Behavior	Command
Automatic (Procedural)	The transitions within a procedure are carried out without interruption as appropriate conditions are met.	Operators may pause the progression, but may not force transitions.
Automatic (Basic Control)	Equipment entities are manipulated by their control algorithm.	The equipment cannot be manipulated directly by the operator.
Semiautomatic (Procedural Only)	Transitions within a procedure are carried out on manual commands as appropriate conditions are met.	Operators may pause the progression or redirect the execution to an appropriate point. Transitions may not be forced.
Manual (Procedural)	The procedural elements within a procedure are executed in the order specified by an operator.	Operators may pause the progression or force transitions.
Manual (Basic Control)	Equipment entities are not manipulated by their control algorithm	Equipment entities may be manipulated directly by the operator.

New Make2Pack Mode Concepts

Get P. 77 from S88 Blue Book

New Concepts

- Introduced the concept that states and their transitions occur within a mode
- Introduced the concept of Mode Control that formalizes how mode transitions take place
- Introduced the concept of a Control System Boundary
- Introduced the concept Control Agents
- Introduced the concept that there are additional fundamental modes beyond the 3 fundamental modes noted by the S88 Standard.

Make2Pack additions to the S88 Definitions

- **Mode:**

  A **mode** is a named set of intended control system behaviors.

  a. For example, in Automatic Mode, it is expected that the machine will be performing a sequence of basic control actions without intervention by an external agent.

  b. States and transitions occur within a mode (or mode type)

  A mode does not specify or contain a procedure. For a given machine there may be multiple procedures that are capable of being executed in AUTO. For examples, there may be a Production procedure, Special Maintenance Procedure, etc. A mode may have no procedures, e.g., MANUAL.
  For a procedure to be valid within a certain mode there may be a requirement to organize it into particular states and transitions, e.g., an AUTO procedure must have Starting, Aborting, Running, … states and transactions – i.e., PACKML compliant procedures.
- **Mode Control:**

a. Control that is dedicated to changing modes

  - Mode change may be initiated by an external agent (e.g., the operator puts the mode in AUTO)
  - Mode change may be initiated by feedback from either procedural control or basic control (e.g., when a fault causes the system to drop out of AUTO)
  - In addition, there may be hierarchical mode considerations.

b. Control that is dedicated to determining what control system capabilities are available to the external agent. (we identify three paths, see figure)

  - Although, mode logic in current implementations is intertwined in both procedural and device logic this should not be confused with the intent of the logic which is to actualize the decision of the external agent with respect to mode.
  - Current implementations organize the objects around the device (e.g., a control module), hence the mode logic is embedded in each control module via branching.

- **(S88) Procedural Control:** Control that directs equipment-oriented actions to take place in an ordered sequence in order to carry out some process-oriented task.
- **(S88) Basic Control:** Control that is dedicated to establishing and maintaining a specific state of equipment or process condition.
  NOTE – Basic control may include regulatory control, interlocking, monitoring, exception handling, and discrete or sequential control.

  Basic control is a collection of capabilities that are exposed through the control system. These capabilities can be commanded to perform an action. For example, a clamp can be commanded to open; a pump can be commanded to turn on.
- **(S88) Procedure:** The strategy for carrying out a process. NOTE – In general, it refers to the strategy for making a batch within a process cell. It may also refer to a process that does not result in the production of product, such as a clean-in-place procedure.
- **Control Component:** The collection of Basic Control capabilities that is provided by a Unit, Equipment Module, or Control Module.

Make2Pack Control System / Mode Boundary

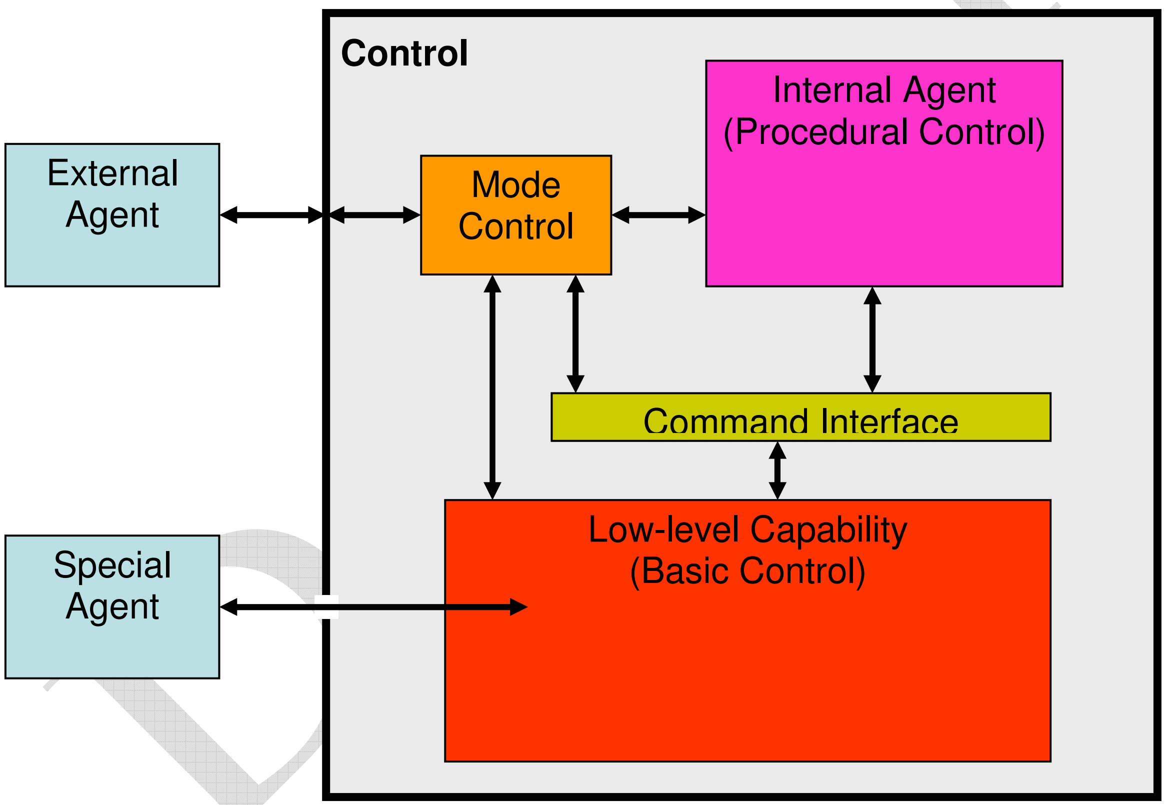
When developing a control system, there is a design boundary that specifies what will and will not be controlled by the control system.   Two different types of control are provided within this boundary; Mode Control, Procedural Control and Basic Control.

Make2Pack Control Agents

Another point introduced in the S88 mode discussion is the notion of "who is in control." To clarify this notion, Make2Pack introduces the term "Control Agent". Here we use the term agent to mean an entity that can take independent action. A control agent is an entity that can request mode changes. Subsequent to requesting a mode, if the request is granted by Mode Control, the control agent can command the mode's allowable capabilities such as commanding Basic Control capabilities or directly accessing the low-level software of the Control Components.

In the Figure below we identify three such agents: an Internal Agent that exists within the confines (or boundary) of the Control System and an External Agent and a Special Agent that exist outside of the Control System.

Figure 1: Relationship between External Agents, Mode Control, Procedural Control, and Basic Control



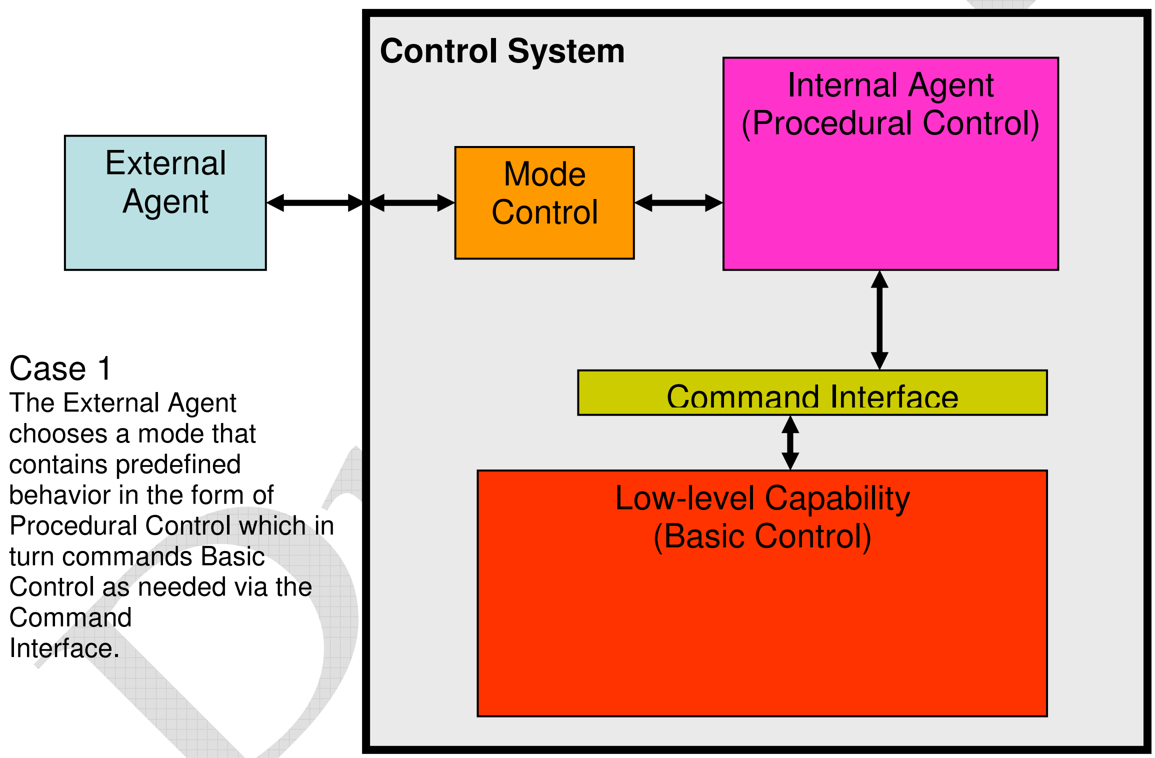
Make2Pack additions to the S88 Definitions regarding Control Agents

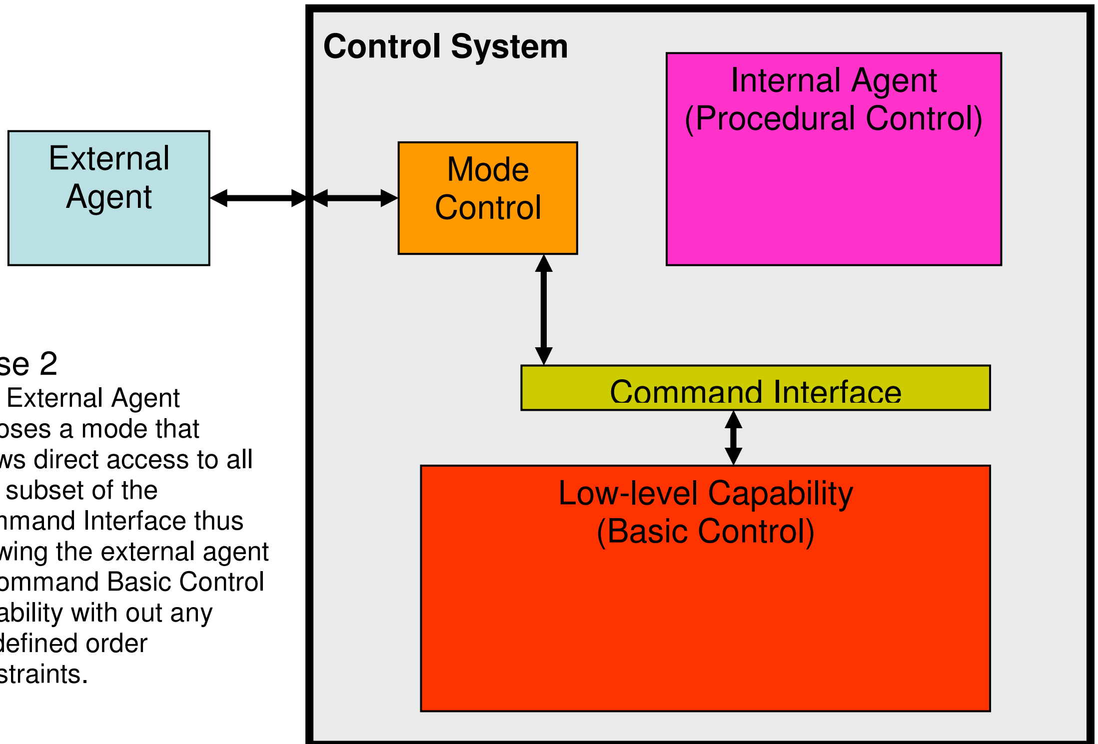
1) **Internal agents** execute equipment control procedures, unit procedures, operation procedures and phases. Using these control components internal agents can execute normal production, routine maintenance activities, and routine machine activities.
2) **External agents** command internal agents to perform execution of their prescribed controls components and they also can command basic control element. External agents can set a mode which in turn causes an internal agent to perform a set of prescribed activities (e.g., Auto.Production, Auto.Cleanout, Auto.Tipdress, Auto.RunOut, etc.). External agents can also set a mode in which they directly invoke

basic control actions to recover from unplanned or unexpected situations (e.g., Manual[Pump.ON], Manual[Pump.OFF], …). Manual activities are a way to address the need to compensate for unplanned or unexpected situation. These manual activities are more desirable and less risky than the below proposed Special Agents.

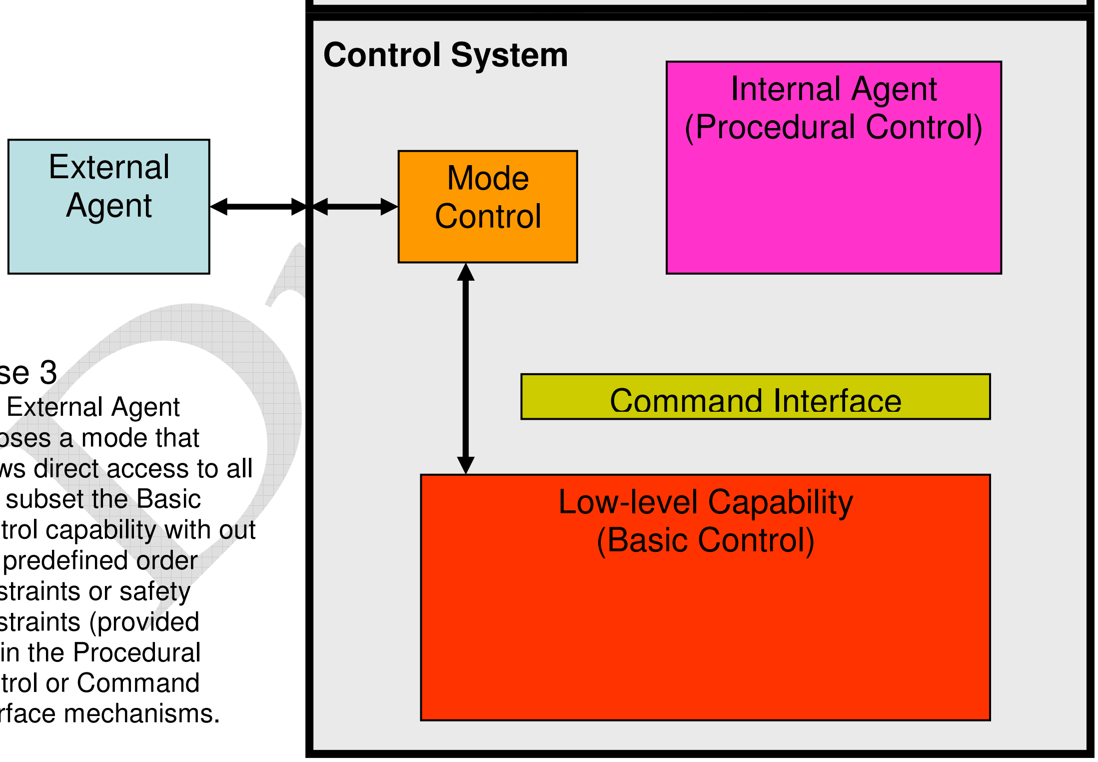
3) **Special (Secret) agents** ignore the normal control command structure and operate directly on the internal software of control components circumventing any of the normal safeguards. For example, using a programming tool to force and I/O bit. Special agents should only be used as a last resort.

Make2Pack Control Agent Cases

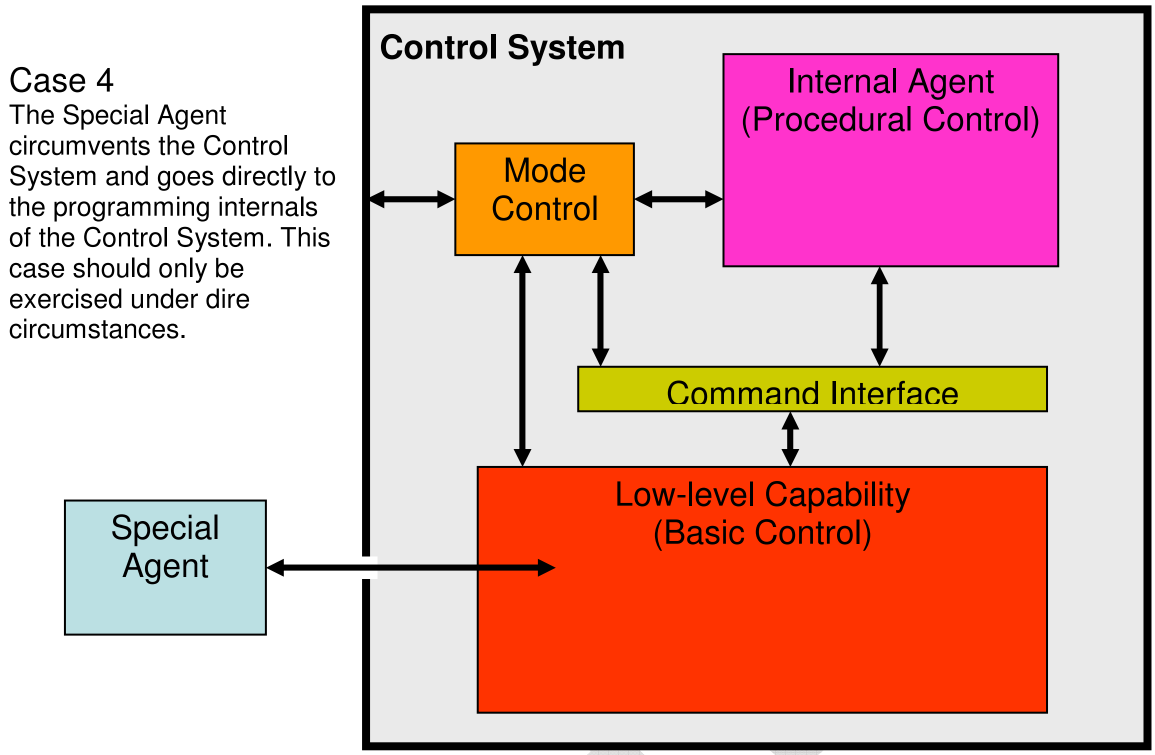




Case 2
The External Agent chooses a mode that allows direct access to all or a subset of the Command Interface thus allowing the external agent to command Basic Control capability with out any predefined order constraints.



Case 3
The External Agent chooses a mode that allows direct access to all or a subset the Basic Control capability with out any predefined order constraints or safety constraints (provided within the Procedural Control or Command Interface mechanisms.



The internal agent when asked invokes these capabilities according to a script(s) also known as procedures. For example, a starting script, a producing script, a held script, …

The external agent is outside the control system boundary and does not contain predefined scripts (and if they do we don't know about them and they are not part of the control system standard).

The S88 standard suggests three modes for procedural control components (*automatic*, *semiautomatic*, and *manual*) and two modes for basic control components (*automatic* and *manual*). [1]
For procedural control, the mode determines how the procedure progresses and who can affect that progression. Table 8.1 summarizes the modes for procedural Control Components. [1]
Procedural elements and equipment entities may change mode when a command is given by an operator or one is generated in another procedural element. A mode change can only occur when defined, required conditions for the change are met. Probably the most common way to change modes is via an External Agent command. [1]
A mode change in one procedural element may force a change of mode in others. If an operator changes a unit procedure mode from automatic to semiautomatic, it may be wise for him or her to propagate all dependent operations and phases to semiautomatic mode. The S88 standard recognizes that propagation can move from a higher-level entity to a lower-level entity or vice versa, but it does not specify any propagation rules. [1]

Table 8-1. Rewrite of the S88 Fundamental Modes in the Make2Pack Formalism

Mode	Purpose	Transition Behavior	Type Of Control	Control Agent	External Agent
Automatic	Production of product	Transitions within a procedure are carried out as necessary order and equipment conditions are met.	Procedural	Internal	External agent may start the procedure or they may abort the procedure.
Semiautomatic	Startup Or Debugging	Transitions within a procedure are carried out as necessary order and permissive conditions are met.	Procedural	Internal + External	External Agent may start the procedure or they may abort the procedure. External agent assumes role of the "next transition" permissive condition. The "next transition" can be any transition for which the necessary order and equipment conditions are met.
Manual	Debugging		Basic Control	External	External agent may execute any Basic Control Commands.

Draft

Proposed Make2Pack Modes of Operation

The first question that we ask is, "Are there other modes worth considering from a standards point of view as we expand our focus beyond batch control such that we begin to include discrete control. We will first introduce some additional "S88 type" modes that determine how procedural elements behave and respond to commands of an External Agent.

Mode	Purpose	Transition Behavior	Type Of Control	Control Agent	Role of External Agent
Automatic S88 *Auto.Production*	Production of product	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a)	Procedural	Internal	May command Automatic.Production mode or may command to leave Auto.Production
By Pass *Auto.ByPass*	Perform procedures similar to those in Automatic Mode, however without modifying the material or parts.	Transitions within a procedure are carried out as necessary order and permissive conditions are met.	Procedural	Internal	Operator may stop or start the progression.
Dry Run *Auto.DryRun*	Perform procedures similar to those in Automatic Mode, however ensuring that no actions are perform that will damage equipment if materials or parts are no present.	Transitions within a procedure are carried out as necessary order and permissive conditions are met.	Procedural	Internal	External agent may start the procedure or they may abort the procedure.
Tip Dress of Robots (example of the class of Special Function maintenance modes) *Auto.TipDress*	To clean the welding tips on robot arms. not part of the normal production sequence, in fact it halts the normal production sequence and supplies its own procedure.	Transitions within a procedure are carried out as necessary order and permissive conditions are met.		Procedural Logic	Operator may stop or start the progression.
Clean Out (another example of the class of Special Function maintenance modes)	not part of the normal production sequence, in fact it uses a different	Transitions within a procedure are carried out as		Procedural Logic	Operator may stop or start the progression.

Auto.CleanOut	procedure and material to clean the equipment to be ready for production	necessary order and permissive conditions are met.			
Automatic.Operator	Production of product	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a)	?	Internal + External	Operator may stop or start the progression. Operators may assume the role of the "next transition" permissive condition for one or more transitions, but not ALL (if all becomes Manual Lead Thru).
Automatic.SpecialFcn	Maintenance, repair, troubleshooting,…	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a)	Procedural	Internal	Operator may stop or start the progression.
Semiautomatic S88 ???	Troubleshooting	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a*)		Operator (2b)	Operator may stop or start the progression. Operator assumes role of the "next transition" permissive condition. The "next transition" can be any transition for which the necessary permissive conditions are met.
Manual.LeadThru (Similar to S88 Semiautomatic)	Troubleshooting	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a)		Operator (2b)	Operator may stop or start the progression. Operator assumes role of the "next transition" permissive condition. (the specified order of transitions may not be ignored, i.e., no skipped steps, etc.)
Manual.Override *(S88 Manual)*	Troubleshooting	Transitions within a procedure are executed in the order specified by an operator. (1c)		Operator (2b)	Operator may stop or start the progression. Operator may execute any transition in any order with no constraining permissive conditions.
Manual.Safe	Troubleshooting	Transitions within a		Operator (2b)	Operator may stop or start the progression.

		procedure are executed in the order specified by an operator, but permissive conditions must still be met. (1b)			Operator can choose any transition provided the necessary permissive conditions are met.

Table 8-1. Fundamental Make2Pack Modes

Mode	Purpose	Transition Behavior	Type of Control	Control Agent	Role of External Agent
Automatic S88 *Automatic.Production*	Production of product	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a)	Procedural	Internal	Operator may stop or start the progression.
Automatic.Operator	Production of product	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a)		Procedural Logic (2a) Operator (2b)	Operator may stop or start the progression. Operators may assume the role of the "next transition" permissive condition for one or more transitions, but not ALL (if all becomes Manual Lead Thru).
Automatic.SpecialFcn	Maintenance, repair, troubleshooting,…	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a)		Procedural Logic (2b)	Operator may stop or start the progression.
Semiautomatic S88 *???*	Troubleshooting	Transitions within a procedure are carried out as necessary order and permissive conditions are met. (1a*)		Operator (2b)	Operator may stop or start the progression. Operator assumes role of the "next transition" permissive condition. The "next transition" can be any transition for which the necessary permissive conditions are met.
Manual.LeadThru (Similar to S88 Semiautomatic)	Troubleshooting	Transitions within a procedure are carried out as necessary order and permissive conditions are met.		Operator (2b)	Operator may stop or start the progression. Operator assumes role of the "next transition" permissive condition. (the specified order of transitions may not be ignored, i.e., no skipped steps, etc.)

Draft

		(1a)			
Manual.Override *(S88 Manual)*	Troubleshooting	Transitions within a procedure are executed in the order specified by an operator. (1c)		Operator (2b)	Operator may stop or start the progression. Operator may execute any transition in any order with no constraining permissive conditions.
Manual.Safe	Troubleshooting	Transitions within a procedure are executed in the order specified by an operator, but permissive conditions must still be met. (1b)		Operator (2b)	Operator may stop or start the progression. Operator can choose any transition provided the necessary permissive conditions are met.

See the working document on Modes and Agents

S88 Equipment Model and how it fits to Define a Packaginf Machine:

Draft 1

Morten Kristensen, NNEUS Inc.
17 Feb 06

Make2Pack Definitions

S88 part 1 term	Make2Pack term	Make2Pack Description
Automatic Mode	Automatic Mode	The component does not respond to human commands. Responds only to commands from other logic while executing its own internal logic.
n/a	Bypassed Mode	The component lets input commands affect the outputs directly while bypassing the internal logic. This mode applies to device level control components only and can be selected in conjunction with automatic mode only.
n/a	Command	A signal to a component to change mode or state. Commands are generally momentary in nature and latched by the receiving component and reset when the command has been executed.
n/a	Control Agent	A Control Agent is an entity that can initiate procedural transitions or provide commands to basic control components. A Control Agent can be either external or internal: For basic control the control agent is always external and can be either another basic control component or and operator. For Procedural Elements the control agent will be internal in case of automatic mode while in manual, semiautomatic or bypass mode the control agent will be external and can be either another Procedural Element or an operator.
n/a	Control Component	Control software part of a Unit, Equipment Module or Control Module.

<table>
<tr><th>S88 part 1 term</th><th>Make2Pack term</th><th>Make2Pack Description</th></tr>
<tr><td>Control Module</td><td>Control Module</td><td>A Control Module is the lowest module in a physical model breakdown of a unit.
The term Control Module relates to the combination of (a) physical device(s) and the lowest level control component that controls this(these) device(s) to carry out a physical process action.
There may be control modules without directly associated physical devices. These control modules coordinate/supervise/sequence other control modules.

NOTE: The use of the term control module to describe the supervisory/sequencing/coordinating functions is proving confusing and difficult to convey the concepts. Needs further consideration.

Make2Pack examples of Control Modules:

• Servo
• Conveyor
• Pneumatic Cylinder with feed-back

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. A control module can also be made up of other control modules. For example, a header control module could be defined as a combination of several on/off automatic block valve control modules.</td></tr>
<tr><td>n/a</td><td>Cycling</td><td>The repetitive motion of a machine for discrete manufacturing. The machine typically performs 1 cycle for each processed item.</td></tr>
<tr><td>No formal definition</td><td>Device</td><td>An apparatus for performing a prescribed function [Definition from ANSI/ISA-51.1 – 1979 (R1993)]

Used here to describe elements of the physical environment that control systems connect to/control.</td></tr>
<tr><td>n/a</td><td>Environmental Safety</td><td>Safety measures to protect the production environment and/or the environment outside the production facility.

This is outside of the scope of this standard. Expectation is that these types of safety systems are managed as "External Agents" in relation to the automated controls.</td></tr>
</table>

S88 part 1 term	Make2Pack term	Make2Pack Description
Equipment Module	Equipment Module/Station	A group of equipment and its associated control located within the context of a machine or unit, designed and or arranged to perform a certain function. This can be equivalent to what is often called a Station in a discrete machine. The equipment module may be made up of control modules and subordinate equipment modules . • E.g. the capper station in bottle filling machine. S88: *Physically, the equipment module may be made up of control modules and subordinate equipment modules. An equipment module may be part of a unit or a stand-alone equipment grouping within a process cell. If engineered as a stand-alone equipment grouping, it can be an exclusive-use resource or a shared-use resource.*
Equipment Operation	Equipment Operation	S88: ***equipment operation:*** *An operation that is part of equipment control.*
Equipment Phase	Equipment Phase	Code in equipment control that issues commands to other control components in order to affect the process as intended. An equipment phase will only be active while it is carrying out its task. When complete the equipment phase becomes idle. NOTE: In general equipment phase that are commanded by recipe phases use the S88 example state diagram. NOTE: It is possible to have equipment phases that are directed by other control components that do not use the S88 example state diagram. S88: ***equipment phase:*** *A phase that is part of equipment control.*
Equipment Procedure	Equipment Procedure	S88: ***equipment procedure:*** *A procedure that is part of equipment control.* NOTE: An equipment procedure has the same construct as a recipe procedure. The only difference is the residence of the procedure instructions.
n/a	Equipment Safety	Safety measures (interlocks) to protect the production equipment itself from control system actions that might damage the equipment. These measures are decided by business rules. They may be implemented as “Internal Agents” and/or “External Agents” using the guidance provided by these standards.

S88 part 1 term	Make2Pack term	Make2Pack Description
Equipment Unit Procedure	Equipment Unit Procedure	S88: ***equipment unit procedure:*** *A unit procedure that is part of equipment control.*
n/a	External Agent	A Control Agent that acts externally to a control component.
n/a	Incremental Jogging (Inching)	Human controlled incremental advancement of a line, machine/unit or lower level module in its cycle.
n/a	Internal Agent	A Control Agent that acts internally within a control component. This will only apply to procedural elements in automatic mode.
n/a	Jogging	Human controlled cycling of a line, machine/unit or lower level module. One step at a time in its sequence.
Manual Mode	Manual Mode	The component responds to human commands to activate its outputs and does not execute its own internal logic. For procedural components manual mode overrides the procedure and human commands affect the process directly. Manual mode is optional.
n/a	Mechatronics	Object oriented machine design. In object oriented machine design an object is the combination of hardware and logic.
Mode	Mode	Production Lines, Units/Machines, and all underlying control components can have modes. The mode determines how the components react to command inputs. . Modes apply to 2 different types of entities: 1. Equipment Entities 2. Procedural Elements The definition of mode for the 2 types of entities are slightly different: 1. Equipment Entities: Equipment Entities are the lowest level control component and the only type of component that connect to the physical devices. Equipment entities do not have procedural control. There are only 2 modes for equipment entities: a. Manual: The logic within the control component is not active. The physical device is controlled directly by the operator or other external control agent.

S88 part 1 term	Make2Pack term	Make2Pack Description
		b. Automatic: The logic within the control component is active. The logic controls the physical device based on inputs from other control components. 2. Procedural Elements: Procedural Elements execute an ordered sequence of actions to affect the process. There are 4 modes associated with procedural elements: a. Manual: Transitions within the procedure are carried out in the order specified by an operator or other external control agent. b. Automatic: Transitions within the procedure are carried out without interruption as necessary conditions are met. Operators or other external control agents may pause the progression but may not force transitions. c. Semi Automatic: Transitions within a procedure are carried out on commands from an operator or other external control agents as necessary conditions are met. d. Bypass: The internal sequence logic in the element is disregarded and outputs are directly manipulated by another procedural element (external control agent). NOTE: For discrete machines certain predefined ways of operating the machines are traditionally called modes. For example: o Dry Run o By Pass o Tip Dress of Robots o Clean Out In the context of this standard these will not be called modes. These will be procedures operating under 1 of the 4 modes described above. S88: ***mode:*** *The manner in which the transition of sequential functions are carried out within a procedural element or the accessibility for manipulating the states of equipment entities manually or by other types of control.*

S88 part 1 term	Make2Pack term	Make2Pack Description
Operation	Operation	A procedural element that provides lower level procedural control within a Unit. S88: ***operation:*** *A procedural element defining an independent processing activity consisting of the algorithm necessary for the initiation, organization, and control of phases.* *An operation is an ordered set of phases that defines a major processing sequence that takes the material being processed from one state to another, usually involving a chemical or physical change. It is often desirable to locate operation boundaries at points in the procedure where normal processing can safely be suspended.* *Examples of operations include the following:* *— Preparation: Pull a vacuum on the reactor and coat the walls with antifoulant.* *— Charge: Add demineralized water and surfactants.— React: Add VCM and catalyst, heat, and wait for the reactor pressure to drop.*
Personnel Safety	Personnel Safety	Safety measures to protect personnel working in the production facility from safety hazards. This is outside of the scope of this standard. Expectation is that these types of safety systems are managed as "External Agents" in relation to the automated controls.
Phase	Phase	A procedural element that provides the lowest level procedural control within a Unit/Machine Module. NOTE: There is a one to one relationship between a recipe phase and an equipment phase. S88: ***phase:*** *The lowest level of procedural element in the procedural control model.* *The smallest element of procedural control that can accomplish a process-oriented task is a phase. A phase may be subdivided into smaller parts.*

S88 part 1 term	Make2Pack term	Make2Pack Description
Procedure	Procedure	Procedural element that controls a Process Cell or Production Line. A Procedure is a series of steps to be performed in order to reach a goal. It is also a strategy because there is no higher level procedure. The Procedure for a discrete Production Line relates to the higher level process of producing goods, e. g. how to get the line ready, how to get into the producing state or how to clean out the line. The Procedure does not relate to the sequential cycling motion that the line equipment performs in order to process each discrete component. This cycling motion is part of Basic Control that takes place within Control Modules. S88: ***procedure:*** *The strategy for carrying out a process.* *NOTE— In general, it refers to the strategy for making a batch within a process cell. It may also refer to a process that does not result in the production of product, such as a clean-in-place procedure.* *The procedure is the highest level in the hierarchy and defines the strategy for carrying out a major processing action such as making a batch. It is defined in terms of an ordered set of unit procedures. An example of a procedure is "Make PVC."*
n/a	Producing	The state of a Production Line or Machine/Unit when product is being produced in automatic mode.
Process Cell	Production Line	A collection of one or more machines, linked together, to perform one or multiple tasks of the process for one or more products in a defined sequence. • Continuous Process (e.g. forming line in the food industry) • Converting Line (e.g. paper, fibers) • Discrete Manufacturing (e.g. assembly) • Packaging Line (from filling to secondary and tertiary packaging) S88: *A process cell contains all of the units, equipment modules, and control modules required to make one or more batches/lots*.
Product Safety	Product Safety	Safety measures (interlocks) to protect the product from control system actions that might damage or impair the quality of the product. These measures are decided by business rules. They may be implemented as “Internal Agents” and/or “External Agents” using the guidance provided by these standards.

S88 part 1 term	Make2Pack term	Make2Pack Description
Recipe	Recipe	A combination of parameter values and procedural instructions in the form of Procedures, Unit Procedures, Operations and/or Phases that uniquely defines how the Production Line or Unit is going to produce a specific product. S88: ***recipe:*** *The necessary set of information that uniquely defines the production requirements for a specific product.*
n/a	Recipe Phase Control Component	Control Component that executes a recipe phase by coordinating Equipment Phases. The Recipe Phase Control Component can be executed in a supervisory system, such as an HMI, SCADA or Batch Sequencer, or it can be embedded in the equipment controller.
Semiautomatic Mode	Semiautomatic Mode	The component responds to human commands in lieu of commands from other logic while still executing its own internal logic. Procedural components will typically advance to the next step in its sequence at human command. In some applications this mode may be known as Maintenance.

S88 part 1 term	Make2Pack term	Make2Pack Description
State	State	The operating conditions of an component, whether it is a lower level control module or a higher level procedural component. States are only relevant if the component is in automatic or semiautomatic mode. A state can be stationary or transitional. A transitional state is a temporary state that the component is in while moving from one stationary state to another. Examples of stationary states are: • Stopped • Producing • Cleaning • Cycling • Halted Examples of transitional states are: • Starting • Stopping S88: ***state:*** *The condition of an equipment entity or of a procedural element at a given time.* *NOTE — The number of possible states and their names vary for equipment and for procedural elements.*
n/a	Supervisory Control Module	Control Module that coordinates/sequences lower level Control Modules. The Supervisory Control Module in general does not connect directly to physical devices like lower level Control Modules.

S88 part 1 term	**Make2Pack term**	**Make2Pack Description**
Unit	Unit (Machine)	In packaging, the unit corresponds to the logical grouping of mechanical and electrical assemblies that historically have been called machines. The term unit may apply to single function machine (filler, capper) or a multifunctional machine (monoblock filler/capper or any other configuration that combines functions within a single machine frame and control system). A multifunctional machine/unit can perform some or all functions of a packaging line, corresponding to process cell, that perform some or all of the functions of primary, secondary and tertiary packaging. A multifunctional machine may be logically broken down into several units corresponding to the individual functions. S88: *A unit is made up of equipment modules and control modules. The modules that make up the unit may be configured as part of the unit or may be acquired temporarily to carry out specific tasks.*
Unit Procedure	Unit Procedure	Procedural element for the highest level procedural control within a Unit. S88: ***unit procedure:*** *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 operations.* *A unit procedure consists of an ordered set of operations that cause a contiguous production sequence to take place within a unit. Only one operation is presumed to be active in a unit at any time. An operation is carried to completion in a single unit. However, multiple unit procedures of one procedure may run concurrently, each in different units. Examples of unit procedures include the following:* — *Polymerize VCM.* — *Recover residual VCM.* — *Dry PVC.*
n/a	Unit/Machine Control Module	Supervisory Control Module on the unit level that handles general unit functionality, (e.g. reset, mode propagation, alarm management)