

**Presented at the
World Batch Forum
North American Conference
San Diego, CA
April 1999**



**107 S. Southgate Drive
Chandler, Arizona 85226-3222
480-893-8803
Fax 480-893-7775
E-mail: info@wbf.org
www.wbf.org**

What Does a Procedure Look Like?

The ISA S88.02 Recipe Representation Format

David Emerson
Yokogawa Corporation of America
2155 Chenault Drive Suite 401
Carrollton, TX 75006
(972) 417-2753

KEY WORDS

ISA, SP88, S88.01, S88.02, ISA-TR88.0.03-1996, Sequential Function Chart, SFC, Procedure Function Chart, PFC

ABSTRACT

The S88.02 draft standard defines a means for depicting procedural control in master and control recipes. This depiction method is called a Procedural Function Chart (PFC). This standard depiction is based on the requirements of S88.01, technical report ISA-TR88.0.03-1996, user requirements, previous standards and commonly used procedure depiction methods in use today. PFCs can provide a common basis for the exchange of recipes, reduce the learning curve between control systems, and establish a common ground for improved human communication. This paper provides a brief overview of the development of PFCs, the rules for their use and some of the benefits expected by their adoption in the industry.

INTRODUCTION

S88.01 defines the term procedure as “The strategy for carrying out a process.”¹ The term procedure is used throughout the part 1 document, but the rules for specifying a procedure are not defined. Although this left a large “hole” in part 1, it was done intentionally due to the varying opinions represented on the SP88 committee. The committee recognized that this was a very important issue, and agreed on two

actions to reach resolution. First, the committee agreed that upon completion of part 1 a technical report would be generated to discuss “possible formats for recipe procedures”². Secondly, it was agreed that part 2 of the standard would define a method for recipe depiction. The technical report, designated ISA-TR88.0.03-1996 Possible Recipe Procedure Presentation Formats, was issued in 1997. The part 2 work continued after issuance of the technical report and is now contained in section 6 of S88.02 draft 13 titled “Language Guidelines”.

Section 6 of S88.02 defines a standard method for depicting procedural logic in master and control recipes. This method has been called a Procedure Function Chart (PFC). The PFC notation was created to meet requirements agreed upon in one of the first part 2 committee meetings. The PFC notation was developed using elements of the three formats discussed in the technical report; list, Gantt and Sequential Function Chart (SFC). At first glance, the PFC notation may appear to be the same as for SFCs. However there are subtle yet significant differences. The differences are necessary to meet the requirements of recipe procedural control as opposed to sequential control execution and documentation.

S88.01 states that a recipe contains five categories of information (i.e. header, equipment requirements, formula, procedure and the “ever ambiguous” other information). Primarily the method chosen for recipe presentation in S88.02 only deals with the procedure category since this is the “glue” that combines the other categories, gives a recipe “depth” (i.e. a multi-level hierarchy of recipe procedural elements) and must contain logic. The other four categories are only mentioned briefly in the draft, and are not intended to be standardized beyond the requirement that their relation to a specific element or symbol in a procedure “be clearly indicated and consistent within each application”³.

The PFC notation is intended to be a standard means for depicting procedural logic in recipes. It is recognized that alternative methods will be used, since depending upon the characteristics of a procedure (e.g. size, complexity, and user needs) alternative methods may be more appropriate. The benefit of a standard depiction method is that it will enable data exchange of recipes, reduce the learning curve for users between systems and establish common ground for communication among users and suppliers.

REQUIREMENTS FOR PROCEDURAL LOGIC DEPICTION

The first order of business in developing the recipe representation method was to agree on a set of requirements the method should meet. The resulting list is included in section 6.11 of ISA-dS88.02-1999 Draft 13 and is repeated below:

- Simple to follow: Easy for people to understand
- Easy to build: Few syntax requirements and symbols to learn
- Clearly defined boundaries: Standardized graphical symbol for Start and End
- Unambiguous depiction of execution order: Sequence, parallelism, selection (divergence) and convergence
- Expression of coordination relationships: Material transfers, Wait for, Synchronize, etc.
- Hierarchical level: Standardized symbols for Procedure, Unit Procedure, Operation, Phase

- Existence of levels: Standardized graphical symbol to show possible decomposition of an element of the hierarchy
- Applicable to master recipes and control recipes
- Applicable to all levels: Similar set of symbols and rules at all levels in a recipe
- Independent of media: Equally usable and understandable whether implemented with pencil and paper or with full-color animated computer graphics.

Some of the requirements are easily quantifiable (i.e. standardized symbols for procedure, unit procedure, operation and phase) while other are subjective (i.e. easy for people to understand).

Implied by the requirement for an unambiguous depiction of execution order is the need for scalability. In this context scalability means the range of complexity procedures may have. In the very simplest cases a procedure may have one Recipe Procedural Element (RPE), or a simple list of them to be performed sequentially when the previous is completed. In complex cases conditional logic and timing constraints may be needed.

Not listed in these requirements, yet recognized by the committee, was the need to have a standard procedure notation for data exchange using the table structures in section 5 of ISA-dS88.02-1999 Draft 13. The table structures in section 5 and the data model in section 4 both support the PFC notation in a generalized manner. If for no other reason, a standard procedure depiction method is required in order to enable the exchange of a recipe.

COMPARISON OF EXISTING AND PROPOSED METHODS

After agreeing on the requirements for a depiction method the committee surveyed existing methods, considered the technical report and examined other standards. During this process it was recognized that the technical report provided an accurate analysis of the options, including most methods in use in industry and provided by commercially available batch control systems.

The list method, shown in two different presentation formats in Figures 1 and 2, has the advantages of being easy to visualize and “unmistakably precise”⁴. Historically, it should be noted that most recipe procedures were in a textual or list format. These recipes were often a numbered list that instructed an operator on which steps to perform. In this format an operator would commonly finish one item and then proceed with the next. However a list is useful only in simple situations. When conditional logic and/or timing and coordination actions and visualizations are needed the textual approach quickly becomes confusing and error prone.

| ID | Phase | Phase Parameters | | |
|----|--------------|------------------|-----------|-----------------|
| | | Ingredient | Set Point | Other Parameter |
| 1 | Fill | Water | 1000 kg | |
| 2 | Add Manually | Salt | 50 kg | |
| 3 | Heat | Steam | 50 °C | |
| 4 | Add Manually | Sugar | 30 kg | |

Figure 1 - Procedure in Table Format

1. Fill with 1000 kg Water
2. Manually add 50 kg Salt
3. Heat with steam to 50 °C
4. Manually add 30 kg sugar

Figure 2 - Procedure in Text List Format

Gantt charts are useful for depicting the progression of activities over time. They can also be used to show multiple levels of sub-activities. This can be a good fit for recipe depiction in that a recipe procedure consists of one or more unit procedures that essentially operate independently with coordination points. Figure 3 shows an example with four unit procedures in a Gantt chart.

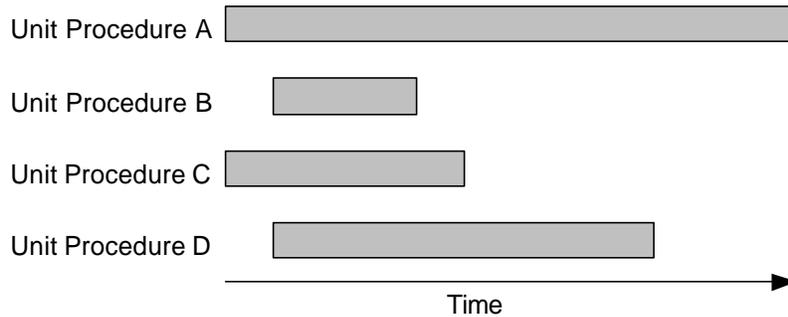
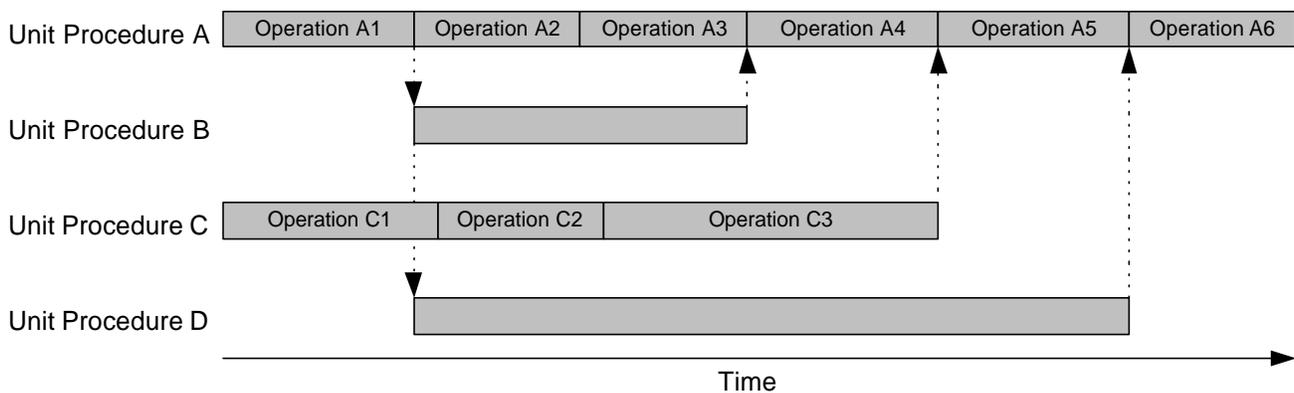


Figure 3 Recipe Procedure in Gantt Chart Format

This could be extended to show coordination between the unit procedures as well as operations inside the unit procedures as shown in Figure 4.



**Figure 4 -
Gantt Chart with Unit Procedure Coordination and Multiple Levels of Procedural Elements**

However, when complex conditional logic is added the Gantt chart becomes a cumbersome tool for depiction. An analogy in scheduling systems is the use of PERT charts to show this type of information along with the often-complex case of multiple predecessors and successors. While PERT charts are a very powerful tool and required for complex scheduling packages they are not easily viewed or understood by users of the schedule.

So, while Gantt charts provide a powerful method to depict unit procedures in a time series, something that is missing in the list method, they do not provide for adequate specification and depiction of conditional logic.

The third method discussed in the technical report is the use of Sequential Function Charts (SFCs) as defined in IEC 61131-3. SFCs have gained widespread acceptance in the process industries and provide a powerful means to specify conditional logic, something lacking in the list and Gantt chart methods. It was noted that many of the recently developed batch systems use SFCs as a means for procedure depiction. An example of an operation's procedure in SFC format is shown in Figure 5

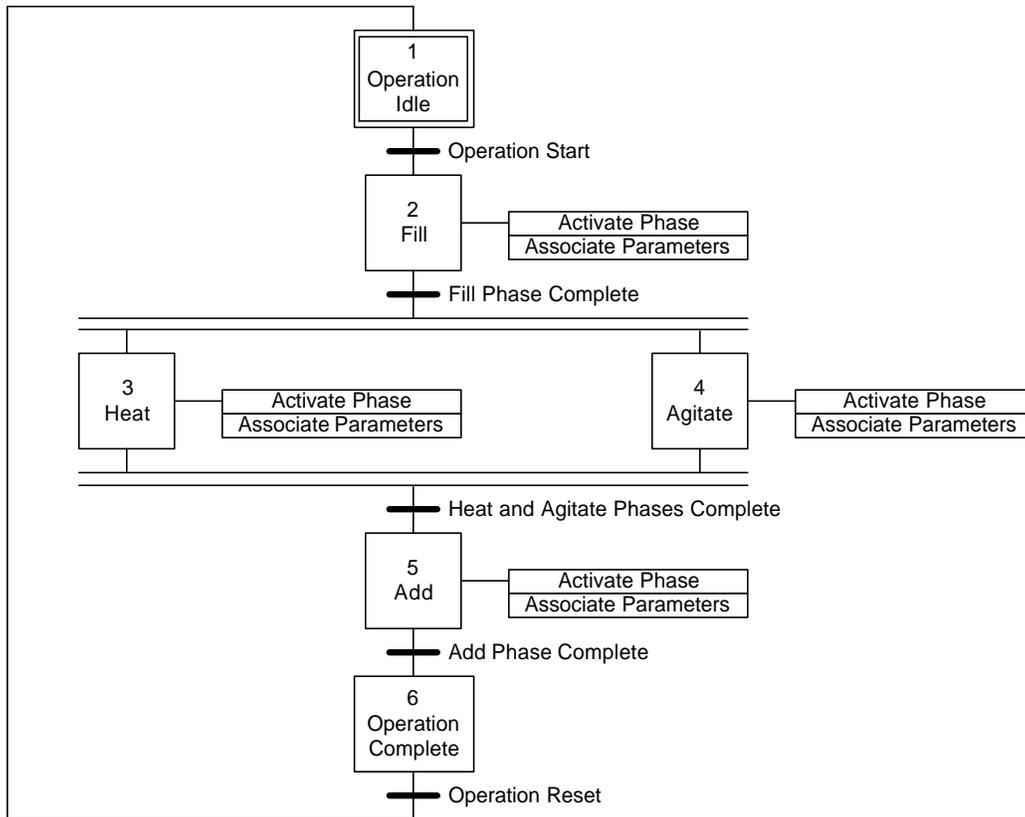


Figure 5 - Operation Procedure in SFC Format

In this example steps have been used to provide operation control states, idle and complete. These operation control steps do not have actions. Other steps have been used to activate recipe phases. Each recipe phase in this example would correspond to an equipment phase.

There are many styles of SFCs. In this example a loop back is shown from the Operation Complete step to the Operation Idle step. The loop back could be omitted along with the Operation Reset transition. In that type of implementation the Operation Complete step could be used as the ending point for the SFC.

Figure 6 contains a recipe procedure in SFC format. This has also been drawn with the loop back, although control recipes/batches are not typically repeated. To reflect this the transition after the Batch Complete step has a condition of false, thereby preventing repetition of the procedure.

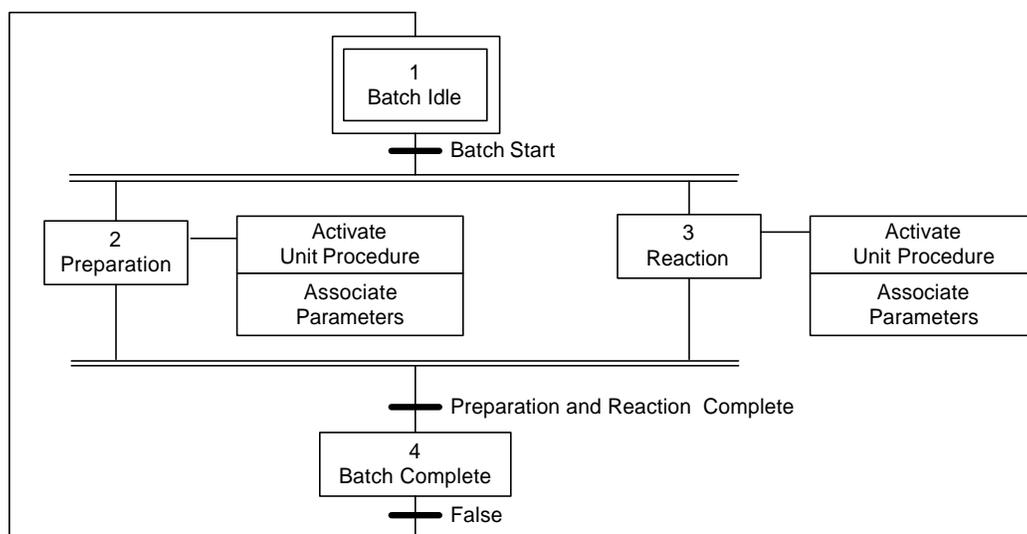


Figure 6 Recipe Procedure in SFC Format

SFCs provide a suitable tool for depicting the conditional logic that is often needed at the operation level. In contrast to this, using SFCs at the recipe procedure level results in a diagram that, while technically correct, does not provide a great deal of useful information. Figure 6 shows 2 unit procedures running simultaneously. However, there is no information provided regarding material flow, synchronization between unit procedures and the general timing of the unit procedures relative to each other. The SFC depiction in Figure 6 does not show this because of the need to place all the unit procedures inside the same simultaneous sequence structure (the horizontal double lines). Also Figure 6 does not show the coordination timing between unit procedures. For example, the Reaction unit procedure is not intended to start until the Preparation unit procedure is about half done. This information is not visible using the SFC format.

The technical report concluded that the SFC method, while supporting complex conditional logic presentation, “does not allow the user to easily visualize the relationships between unit procedures”⁵.

Two issues not addressed by the technical report but extremely critical for recipe depiction are the use of multiple levels of procedures in a hierarchy and the separation of the procedure execution in a recipe from the equipment procedural element execution.

During this evaluation, other work was also examined. Karl-Erik Arzen and Charlotta Johnsson have published works regarding High-Level Grafchart and how it can be used for recipe depiction^{6 7 8}. Also, work has been underway for some time to revise IEC 60848-1988 to specify how multiple levels of function charts may be used⁹. This includes the definition of structures such as pit steps, pit transitions and macro steps to deal with some of the issues of multiple level charts. These works deal with the issue of multiple levels of charts which was inadequately handled in the earlier definitions of Function Charts¹⁰ and Sequential Function Charts¹¹. For this reason the resulting PFC notation bears a resemblance to High-Level Grafcharts and the proposed revision to IEC 60848: 1988 being prepared by IEC 3B/WG14.

DEVELOPMENT OF PROCEDURE FUNCTION CHARTS

The result of the evaluation was the conclusion that no existing method met the requirements for all procedural levels in a recipe and for all ranges of procedure complexity. However it was agreed that each of the three methods discussed in the technical report had strengths that if combined could contribute to a new method. The committee also recognized that a method similar to the then proposed IEC 60848 macro step revision would be needed to permit multiple levels of procedures in a recipe.

The PFC notation was developed and revised over a four-year period. The work often proceeded down paths that were reversed and involved false starts. Whenever new people joined the working group, or the working group reported to the full committee, there was a great deal of discussion and consideration of alternatives. The realization occurred that it would be impossible to develop a method that exceeded all the requirements in everyone's minds. Therefore tradeoffs and compromises would be required to reach consensus on a depiction method.

Although it is a simplification of the PFC notation development process it is helpful to look at some of the elements of its derivation.

1. The list format's influence can be seen in the use of the implicit transition (discussed later in this paper) that permit a simple list of recipe procedural elements (e.g. unit procedures, operations and phases) to be made.
2. The Gantt chart format's ability to provide a sense of timing and synchronization on a simple scale is reflected in the optional ability to draw RPE's different lengths to indicate relative timing.
3. The influence of SFCs is obvious to anyone familiar with that method. The use of the powerful sequence selection and simultaneous sequence constructs was used as well as the step-transition-step constraint. However (as discussed later) there are significant differences not as visible between PFCs and SFCs.
4. Although the term "macro step" is not used the concept was put forth in the 1995 proposed amendments to IEC 60848: 1988 is reflected in the PFC notation. The requirement for the separation of recipe and equipment logic led to the creation of a special case for the activation and evaluation of transitions.
5. Special process needs such as equipment allocation, material transfers and the synchronization of asynchronous activities are reflected in the PFC notation.
6. The belief that recipe procedures must show the orchestration of relatively independent unit procedures led to the depiction of unit procedure boundaries even when lower level operations or phases are shown inside the unit procedures.
7. There is an inherent difference between procedural logic in a recipe and sequential logic in an equipment entity. The equipment logic, no matter the language, must always be responsible for terminating itself. The equipment entity's decision may be based on outside signals with no internal filter. However the ability for equipment logic to perform housekeeping or other activities once signaled to terminate is critical. This is the line of reasoning that causes the PFC RPE-transition relationship to behave differently than the SFC's step-transition relationship.

PROCEDURE FUNCTION CHART NOTATION

The PFC notation is defined in ISA-dS88.02-1999 Draft 13 at the time this paper was written. It is entirely possible that the notation will be modified before release as an approved standard since it will undergo comment and balloting as part of the ANSI/ISA and IEC standards development processes. Therefore the explanation of the PFC notation that follows should be checked against later drafts and the final standard when it is released for future reference.

A full attempt to completely explain the PFC notation is not attempted in this paper (that would be a worthy goal of future publications). Instead, a brief overview of the notation and some examples are provided.

Figure 7 shows the symbols used in the PFC notation. These symbols are combined to create PFCs.

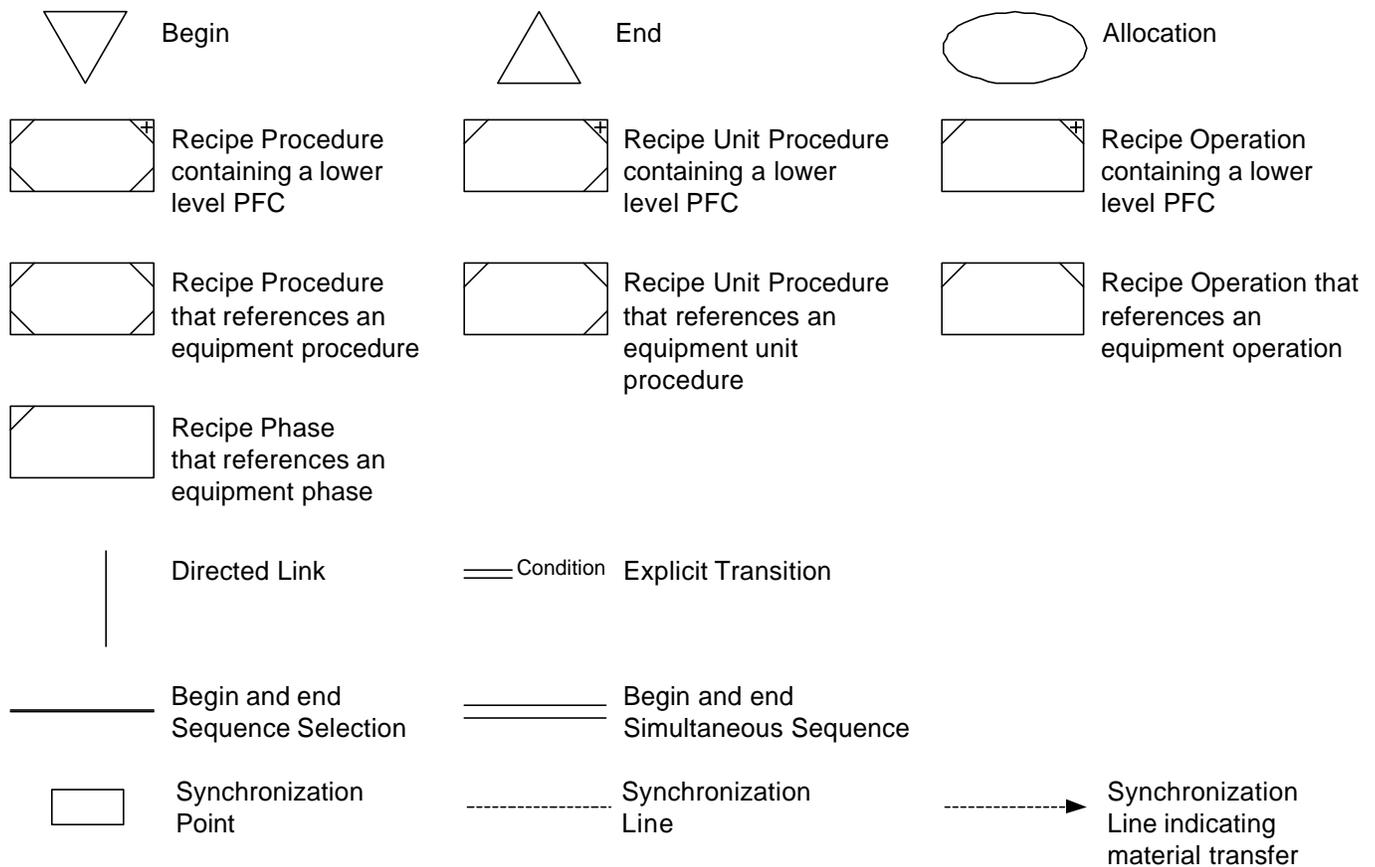


Figure 7 - Procedure Function Chart Symbol Key

PROCEDURE FUNCTION CHART EXAMPLES

Using the PFC symbols, a simple operation containing two phases can be depicted as shown Figure 8. In this chart the begin symbol indicates where execution of the PFC commences. There can only be one begin symbol in a PFC. The directed links connect the various symbols and provide the order of execution for the chart.

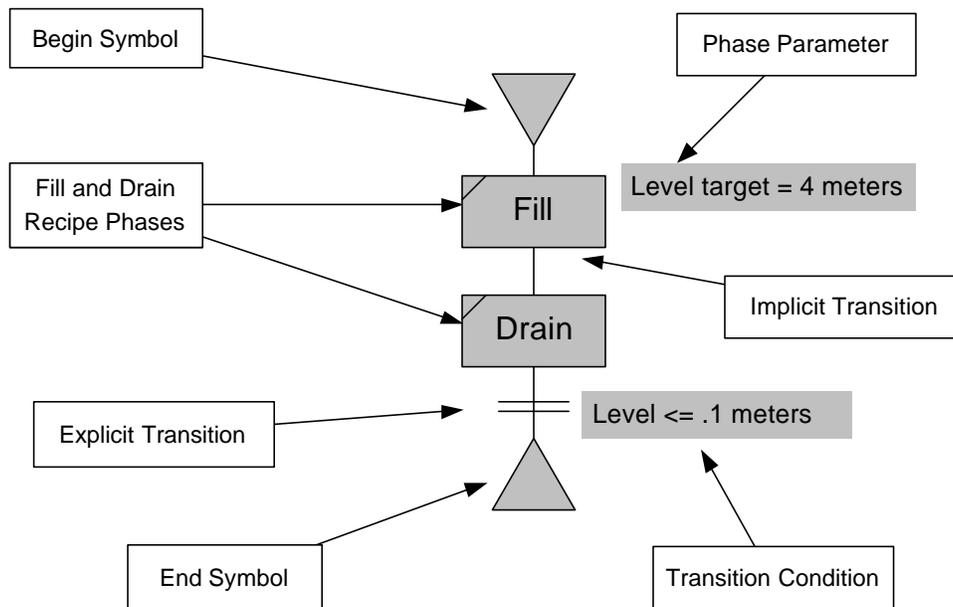


Figure 8 Operation Procedure Function Chart

The Fill and Drain recipe phases reference equipment phases that are typically implemented as programming logic in a controller. These two recipe phases are depicted differently to illustrate the use of implicit and explicit transitions.

The Fill phase uses an implicit transition. The Fill equipment phase is programmed to transition to the complete state when it has accomplished its goal. In this, case the goal is to fill the vessel to a level of 4 meters. The goal has been defined using a parameter instead of a transition condition so there is no need to depict a transition following the recipe phase. When transitions are not depicted they are called an implicit transition meaning that the transition has not been shown since it is implied to be true when the preceeding phase has finished execution. Yet, the transition is implied in order to maintain the “step-transition-step” requirement specified in IEC 1131-3 clause 2.6.5, Rules of evolution. When an implicit transition is used its condition may not contain any logic. In this case when the Fill equipment phase transitions to the complete state¹², the Drain phase is immediately started.

Figure 9 and Figure 10 expand upon the concept of implicit and explicit transitions. In essence the implicit transition is a convenience so that explicit transitions containing the condition “equipment phase complete” do not have to be placed in PFCs. In most batch applications equipment phases are programmed by an engineer to be started based on the execution of a recipe procedure and then run to

completion using any formula/parameter values appropriate. This is a very critical issue in batch control, the equipment procedural element should always be the ultimate decision-maker as to when it transitions to the complete state. Of course, this may include a programmed deference to an explicit transition in a PFC. However, this is only done as a conscience decision by the programmer, not a desire by a recipe author to arbitrarily cause an equipment procedural element to complete. A major difference between this and the use of Sequential Function Charts is that in SFCs once the transition is evaluated true, the preceding step stops instantly and there is no opportunity for housekeeping. This sudden termination of equipment phases is not suitable for batch applications. Anecdotally, in its survey of existing commercially available batch systems the committee found that many systems using SFCs for procedures had found this problem and worked around it, essentially deviating from IEC 61131-3 Sequential Function Charts in order to meet the needs of batch processing.

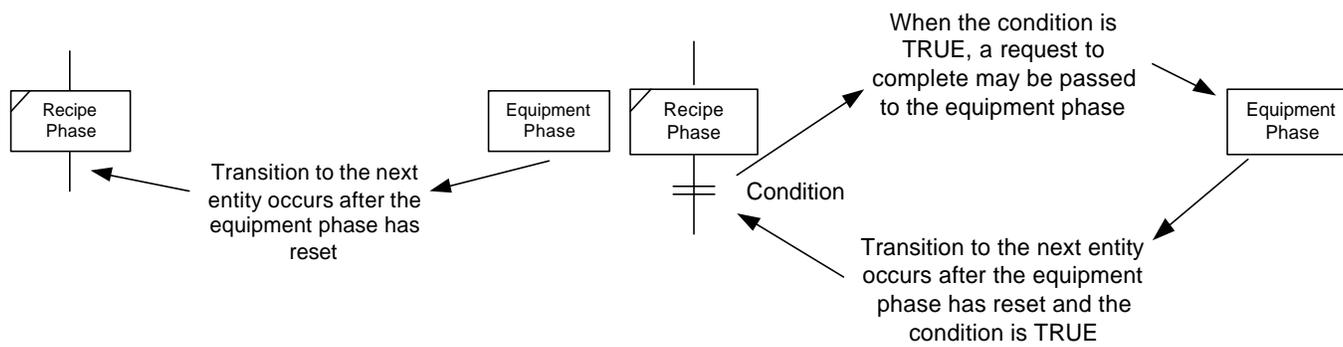


Figure 9 Implicit Transition

Figure 10 Explicit Transition

Referring back to Figure 8, if a transition were displayed after the Fill phase it may have a condition of “true”, “level target reached”, or “fill phase complete or reset”. If it is felt that a transition would be helpful for depiction purposes it may be included as an explicit transition.

An explicit transition is shown following the Drain recipe phase. In this case, the Drain equipment phase is programmed to drain material from the vessel. Once the draining process has started the equipment phase logic does not close the drain valve until it has been directed to do so as a result of the recipe PFC execution. In this case the explicit transition’s condition of “Level <= .1 meter” is what will trigger this notification once it is evaluated as true. When the Drain equipment phase logic receives this notification it closes the drain valve, performs any housekeeping desired, and transitions to the complete state. Once the equipment phases’ complete state is reached execution in the PFC proceeds past the explicit transition to the end symbol causing the operation to complete.

As can be seen by this example the choice between implicit and explicit transitions is an end-user preference. The same logic may be implemented using parameters with implicit transitions or using explicit transitions with a transition condition. The standard supports both styles.

It should be noted that the coloring in the PFCs in this paper and the depiction of the Fill recipe phase’s parameter are for presentation purposes. The coloring is not part of the standard. Regarding the depiction of formula/parameters, the standard states that they shall be displayed as appropriate and always associated with a recipe procedural element. So although this depiction is not shown in the standard it meets the requirements of the standard.

Wrapping up the operation PFC is the end symbol. Like the begin symbol there can only be one per chart. The single end symbols means that when sequence selections or simultaneous sequences are used all of the possible paths must converge back to a single path to reach the single end symbol. The begin and end symbols are not executed, they are merely placeholders.

Building up from the operation PFC, Figure 11 contains a PFC for a unit procedure. At this level in a recipe there is the possibility that a procedural element (i.e. operation) could either contain a PFC (i.e. an ordered set of recipe procedural elements) or directly reference an equipment procedural element (i.e. an equipment operation). To provide an indication of what a procedural element represents a “+” sign is placed in the upper right hand corner of a procedural element when it encapsulates a PFC. For example the Initialize operation references an equipment operation, it does not contain recipe phases. The other recipe operations all encapsulate PFCs, which in this case all contain recipe phases.

The use of implicit transitions after the Initialize and Charge operations is done for conciseness (at the decision of the recipe author and the engineering staff that wrote the operation logic) and makes the first part of the unit procedure PFC read like a list.

The explicit transitions below the sequence selection symbol are required, implicit transitions may not be used in this case.

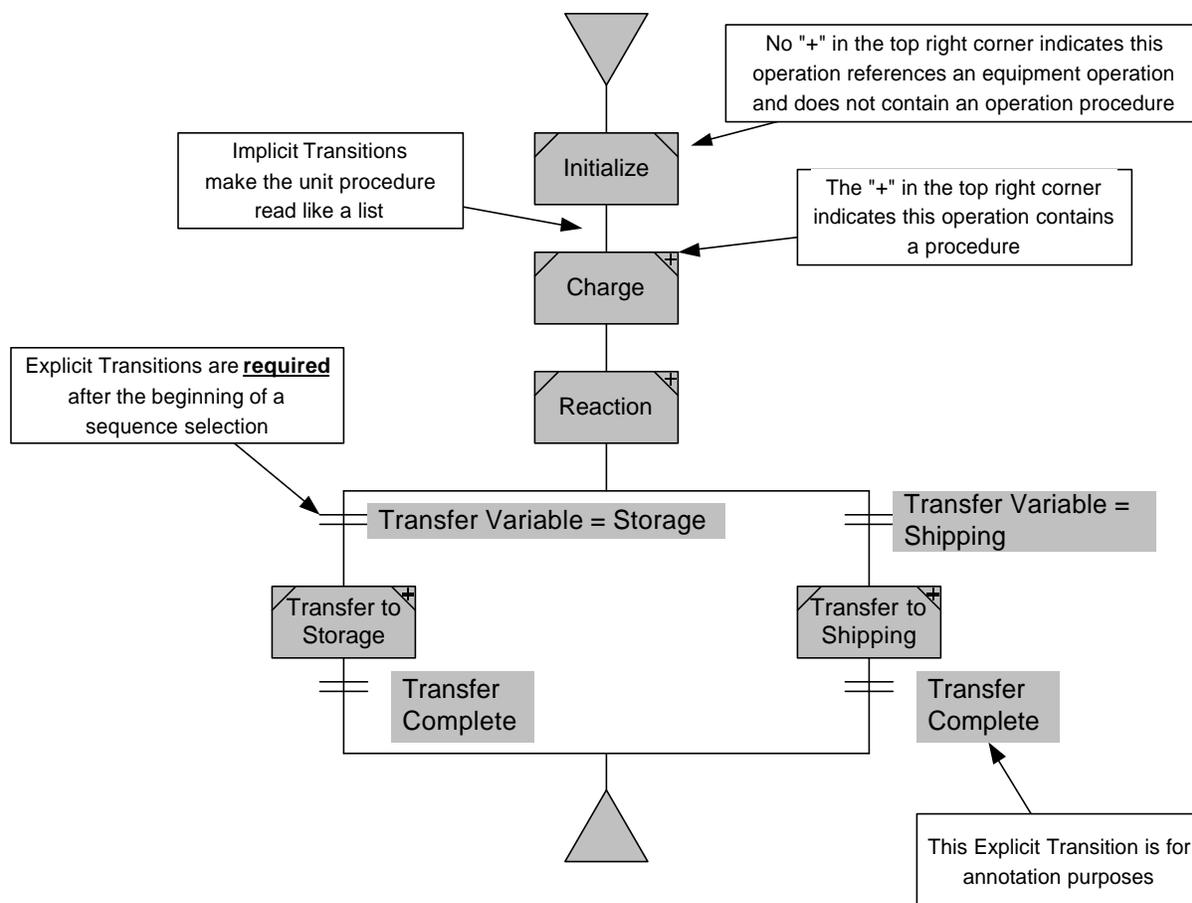


Figure 11 Unit Procedure PFC

The needs for depiction of the unit procedure and operation PFCs are not very different. However, the needs for depiction of recipe procedure PFCs are significantly different from these two. Recipe procedures orchestrate the execution of asynchronous activities (i.e. unit procedures) that have points of synchronization, material transfer and equipment requirements. Also, as the top level in a recipe's procedure they need to provide as much information at a high level of abstraction as possible. Figure 12 contains an example of a simple recipe procedure. There are two unit procedures, Preparation and Reaction that run simultaneously.

The allocation symbols contain the equipment requirements for the following unit procedure. The allocation symbol's shape and purpose has been standardized in S88.02. However the contents have been left unstructured. It is intended to contain equipment allocation rules for the unit procedure. In this example, the allocation rules are a list of the possible units that may be used by the unit procedure. When an explicit transition is placed immediately after an allocation symbol it represents the start condition for the following unit procedure. In this example the Preparation unit procedure starts as soon as the allocation symbol is executed while the Reaction unit procedure requires an operator input before it starts.

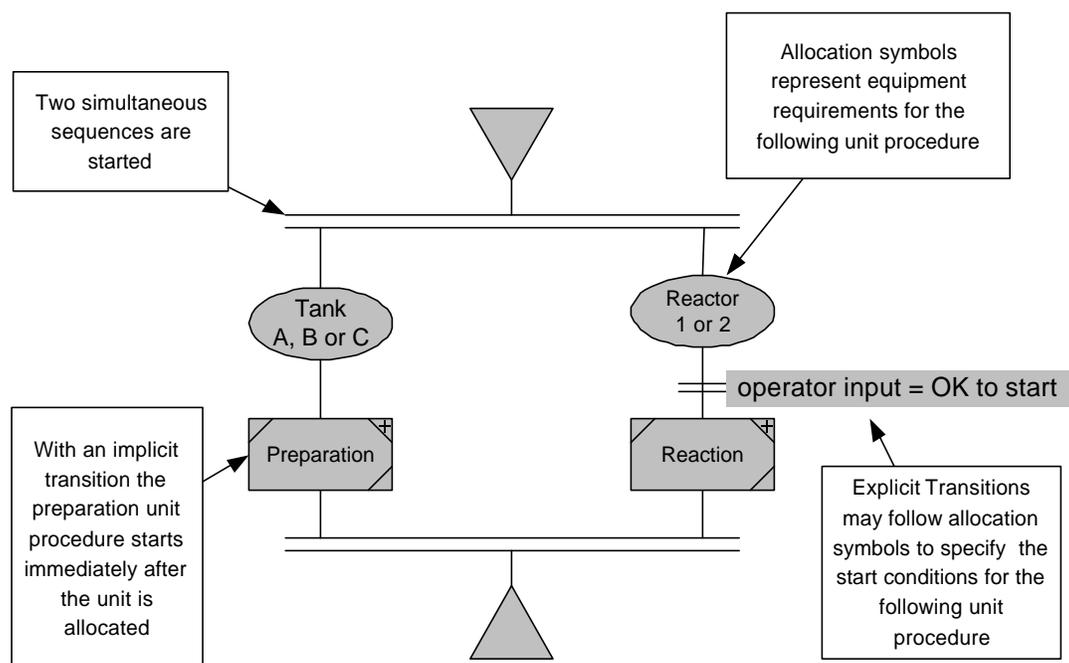


Figure 12 Recipe Procedure PFC

Once both unit procedures have completed, the simultaneous sequence end symbol causes the convergence of the two active sequences and the end symbol is reached, thereby ending recipe execution.

Although Figure 12 indicates that the two unit procedures run simultaneously there is much information about the recipe procedure that is not depicted. In fact, using a depiction like Figure 12 on a larger scale usually results in all the unit procedures in a recipe being placed inside the same set of simultaneous sequence structures and this only weakens the usefulness of this depiction method. Also, any synchronization and/or the movement of material between units is not depicted. The first instinctive

thought to solve the first issue, that of all unit procedures in the same simultaneous sequence structure, is to depict the unit procedures serially, as in a list. Figure 13 provides an example of this. Unfortunately this method does not work. The problem is that when presented serially the first unit procedure must be complete before the second may have its equipment allocated and be started. This makes it very difficult to perform material transfers.

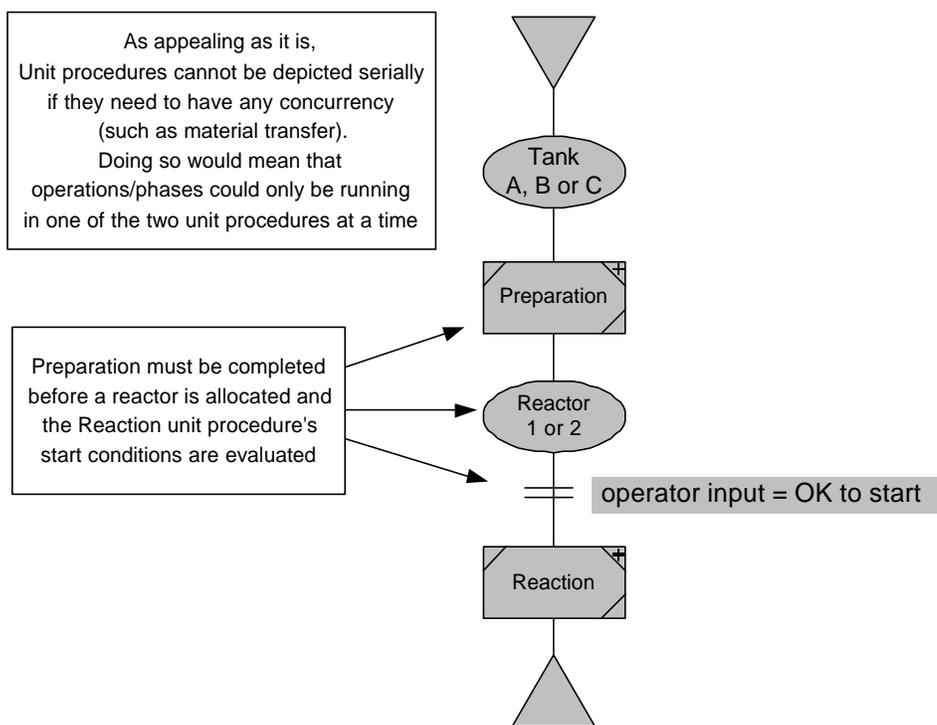


Figure 13 Recipe Procedure PFC Depicted Serially

With this option ruled out, the challenge is to make the one large simultaneous sequence structure viable. Applying Gantt chart concepts of relative time in the vertical axis does this. It should be noted that this involves relative time since master recipes, and the portions of control recipes not yet executed, do not contain absolute timing information, such as how long an operation takes to run. So for master recipe procedure depiction the sizing of the procedural elements (i.e. unit procedures) is purely relative and intended as an annotation. Figure 14 shows the same recipe procedure as in Figure 12, only drawn to show the relative relationships and synchronization points.

The sizing of the two unit procedures is intended to indicate that Preparation runs first and at some point during its execution the allocation rules for the Reaction unit procedure are executed. After the Reaction allocation rules are processed and the start condition of the operator input is satisfied the unit procedure starts. Then at some point near the end of Preparation and near the beginning of Reaction a material transfer occurs that sends material from the Preparation tank to the Reactor for use by Reaction. After the transfer Preparation continues to run for some time and Reaction runs for a longer time. When both unit procedures are complete the recipe ends.

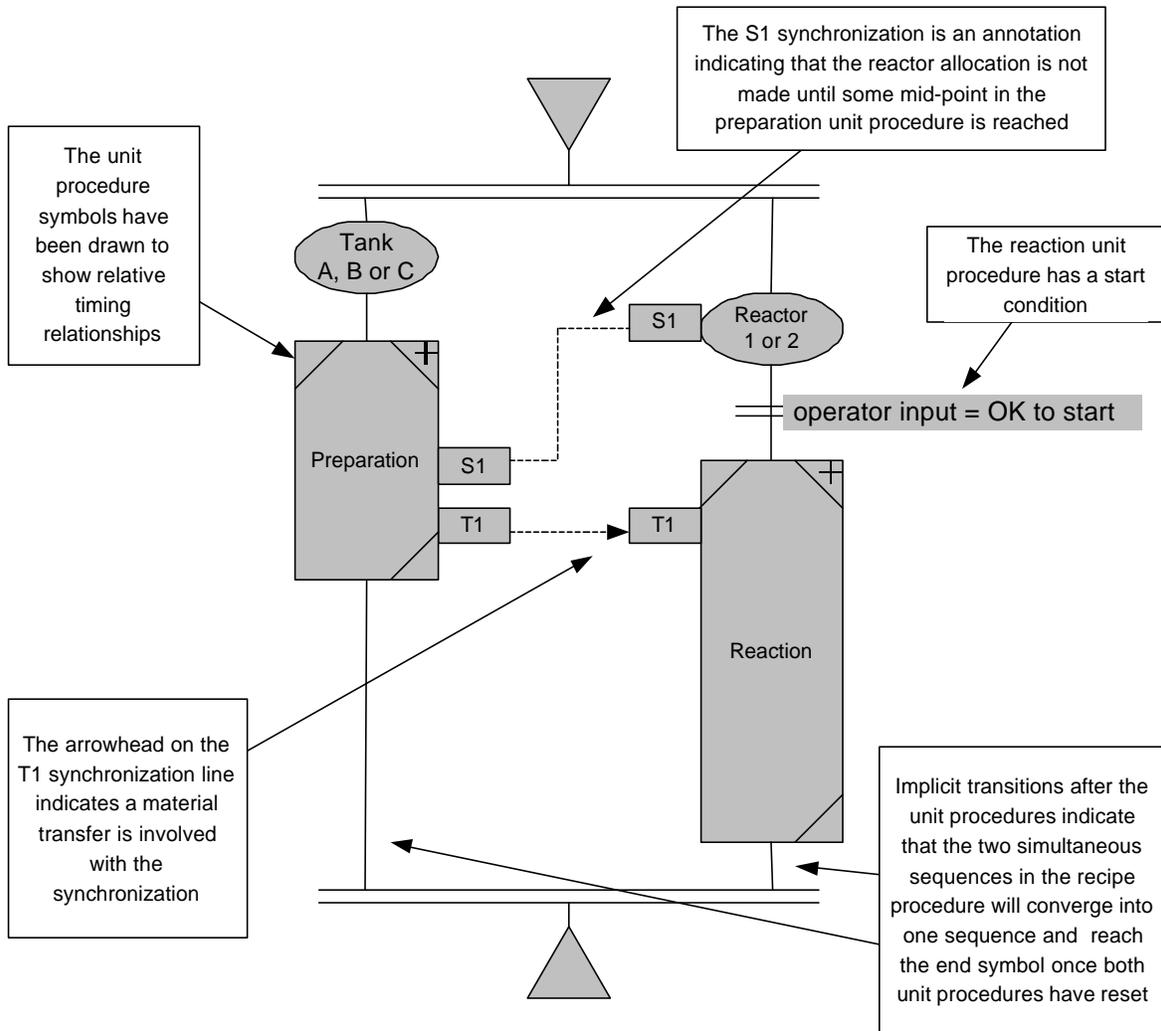


Figure 14 Recipe Procedure PFC Showing Relative Relationships

Although not specific about events, Figure 14 does provide more information than Figure 12 for the same example. To carry this further it is possible to show multiple levels of PFCs on one PFC. Figure 15 provides an example of this. In Figure 15 the unit procedure symbols have been expanded further and the unit procedure PFCs they encapsulate are shown inside the symbols. When this is done it can be seen that Preparation and Reaction each involves four operations. The S1 synchronization point is revealed to involve the sample taken from the preparation tank. Likewise the material transfer T1 is achieved through operations “Transfer to Reactor” and “Receive from Prep.”

The intention of the standard is to permit the types of visual expansion shown in Figure 14 and Figure 15 when it is desired. There is no requirement to always provide this depiction style. The minimum recipe procedure depiction is that of Figure 12.

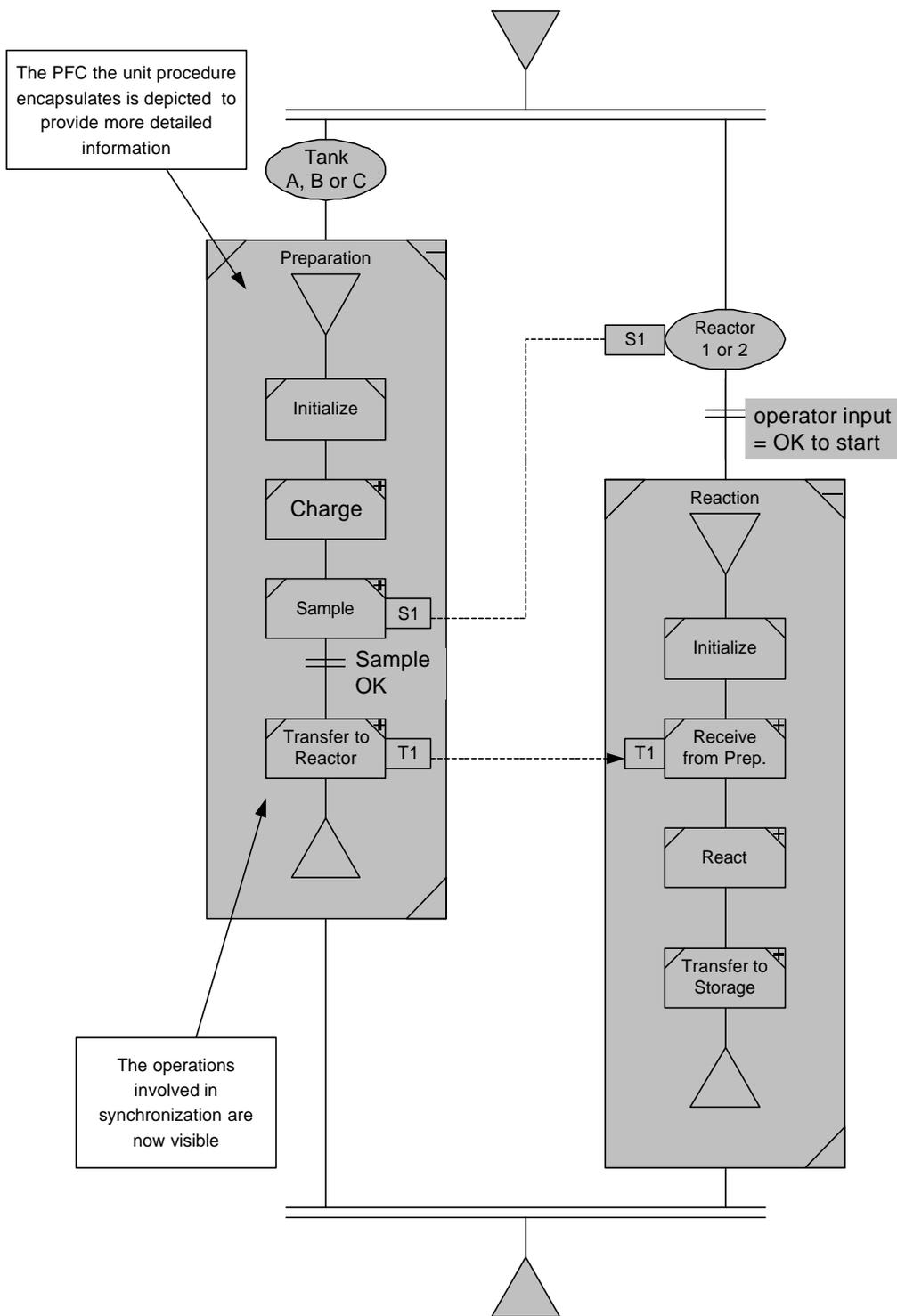


Figure 15 Two Levels of Procedural Elements Depicted in One PFC

ADDITIONAL PFC RULES

The PFC notation is intended to foster the exchange of recipe data between systems and make the learning of each new batch system easier. However, it is recognized that no paradigm is permanent and that evolution and innovation are constants. Therefore, the standard permits the PFC notation to be extended. The only requirement in this area is that extensions should be clearly defined.

As mentioned earlier the procedure is the “glue” that holds the different categories of information in a recipe together. The examples show an instance of formula/parameter depiction, and the allocation symbols provides for equipment requirement depiction. Depiction of header and “other” information is left flexible as deemed appropriate for the application.

SUMMARY

The PFC notation provides a standard, vendor independent, method for procedure depiction. It is clearly vendor independent at this time since it is based on multiple methods and standards, and deviates from them sufficiently so that no vendor is favored. A mix of vendors and users were involved in the development process resulting in a notation that represents a consensus of all parties involved.

It is hoped that the PFC notation will:

- Provide the basis for a standard data structure thereby enabling recipes to be exchanged between systems,
- Permit more effective communication during all phases of projects,
- Lessen the learning curve for recipe authors, operators and other recipe and control system practitioners when introduced to various systems.

¹ ANSI/ISA-S88.01-1995, Batch control- Part 1: Models and terminology, clause 3.37.

² ISA-TR88.0.03-1996, clause 2.

³ ISA-dS88.02-1998, Draft 13.

⁴ ISA-TR88.0.03-1996, clause 5.1.

⁵ ISA-TR88.0.03-1996, clause 7.

⁶ Object-oriented SFC and ISA-S88.01 recipes, Karl-Erik Arzen and Charlotta Johnsson, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, at the 1996 World Batch Forum Conference.

⁷ Grafchart and Batch Recipe Structures, Charlotta Johnsson, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, at the 1998 World Batch Forum Conference.

⁸ Recipe-Based Batch Control Using High-Level Grafchart, Licentiate Thesis by Charlotta Johnsson, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, June 1997.

⁹ IEC 3B/138/CD, Amendment No. 1 to IEC 848 (1988): Preparation of function charts for control systems, 1995-02-03.

¹⁰ IEC 60848: 1988, Preparation of function charts for control systems.

¹¹ IEC 61131-3, Programmable controllers – Part 3: Programming languages

¹² ANSI/ISA-S88.01-1995, Batch control- Part 1: Models and terminology, Figure 18 – State transition diagram for example states for procedural elements. The Part 1 example states are used throughout this paper.