PROGRAMA DE PESQUISA E DESENVOLVIMENTO TECNOLÓGICO CICLO 2003-2004

MODERNIZAÇÃO DA ÁREA DE AUTOMAÇÃO DE PROCESSOS DAS USINAS DE BALBINA E SAMUEL

PLANO DIRETOR DE AUTOMAÇÃO PARA BALBINA
(VERSÃO FINAL)

Alberto José Álvares

Brasília, 20 Março de 2008

UNIVERSIDADE DE BRASILIA

UNIVERSIDADE DE BRASÍLIA

PROGRAMA DE PESQUISA E DESENVOLVIMENTO TECNOLÓGICO
CICLO 2003/2004

Nº DO CONTRATO: 4500052325 **Nº DO PROJETO: 128**
INÍCIO: 28/09/2005 **DURAÇÃO: 24 meses**

A. Instituição executora: Universidade de Brasília

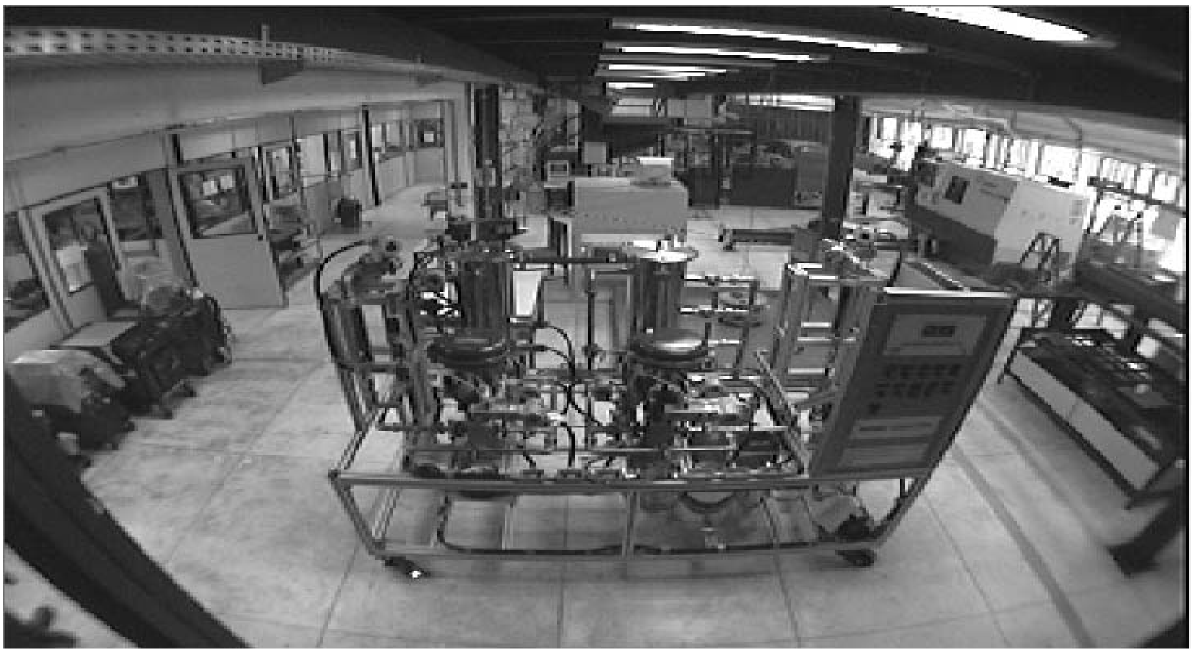
B. Título do Projeto:

Modernização da Área de Automação de Processos das Usinas Hidroelétricas de Balbina e Samuel

C. Coordenador do Projeto:

Nome:	Alberto José Álvares				
Fone:	(61)-3307-2314	Cel:	(61)-99679435	E-Mail:	alvares@AlvaresTech.com

Produto Ano2 - Etapa 2: Plano Diretor de Automação para Balbina
(Março de 2008)



Sumário

1. INTRODUÇÃO

A Manaus Energia é a empresa de geração, transmissão e distribuição de energia elétrica na cidade de Manaus. É responsável pelo fornecimento da energia elétrica na cidade de Manaus. Dentre diversos sistemas de geração da empresa a usina hidroelétrica de Balbina tem capacidade total de geração de 250 Mwatts composto de 5 geradores hidroelétricas de 50 Mwatts cada.

O presente Plano Diretor de Automação tem como objetivo definir metas para produção da energia elétrica de boa qualidade e de baixo custo, através da utilização de ferramentas modernas de automação composto de hardware, software e recursos humanos especializados em manutenção, operação e administração em geração de energia elétrica na usina hidroelétrica.

As usinas hidroelétricas existentes no mundo sofreram grandes alterações na sua forma de manutenção, operação e administração no decorrer dos anos. Logo no início do funcionamento das usinas hidroelétricas do mundo, tanto a manutenção, operação e administração eram baseadas exclusivamente nas sensibilidades humanas dos operadores, dos técnicos de manutenção e do pessoal administrativo.

Alguns fatos tecnológicos tais como a evolução da tecnologia de semicondutores que possibilitou o surgimento de sistema de processamento de informações de baixo custo baseado nos microprocessadores e microcontroladores capazes de tomar decisões, aplicados no controle das máquinas, a evolução da instrumentação eletrônica e sensores cada vez mais inteligentes, a evolução da tecnologia de redes de computadores confiáveis e de alta velocidade que possibilitou a comunicação de dados entre os sensores e o sistema de processamento, e o software de processamento de informações baseado nos algoritmos inteligentes tais como a lógica Fuzzy, redes neurais, sistema especialista, etc. fizeram com que a usina hidroelétrica tenha uma eficiência maior na geração da energia elétrica de boa qualidade.

Assim, na área técnica, a experiência do operador eficiente pode-se transformar em algoritmo de decisão dentro do computador com a vantagem de poder processar grande quantidade de informações provindas dos sensores instalados nas máquinas de forma ininterrupta, tomando decisões rápidas e precisas, auxiliando na operação. A monitoração contínua das diversas variáveis da máquina tais como a temperatura e a pressão do óleo de lubrificação dos mancais, a temperatura do gerador, a densidade do óleo de lubrificação para mensurar o nível de contaminação do mesmo, a temperatura dos mancais, a vibração do eixo e do estator, a quantidade de energia gerada, a temperatura do transformador elevador, a tensão e a corrente no transformador, etc. fornecem informações que quando processados em conjunto, conhecido como "fusão de sensores", são capazes de fornecer informações quanto a probabilidade de ocorrência de falhas antes que a situação se torne crítica a ponto de causar severos danos nas máquinas e possibilitar a parada e manutenção programada. Em outras palavras, através da utilização de sistemas de monitoração on-line das variáveis da turbina e do gerador, podem-se eliminar manutenções de rotina e permitir ao usuário, trabalhar no equipamento somente quando o sistema de monitoramento indicar que tal serviço está sendo necessário. A monitoração on-line é útil também na operação para verificar em tempo real, a quantidade ótima de energia a ser gerada, baseada nas variáveis de decisão tais como o estado da máquina, o nível da água do reservatório, etc.

A vantagem da monitoração on-line das variáveis das máquinas não é somente nesse aspecto técnico, mas também no aspecto da administração que possibilita o uso da ferramenta MES (Manufacturing Execution System) que integra as informações do campo com o sistema corporativo para auxiliar nas tomadas de decisões administrativas, permitindo acompanhamento on-line do processo produtivo.

Nessa óptica, está em andamento, na UHE Balbina, a implantação do sistema de automação com objetivo de aumentar a produtividade que significa a geração de energia elétrica de boa qualidade e de custo reduzido.

Como a estratégia inicial, foram instalados sensores instrumentalizados eletronicamente com recursos não somente de realizar medições das grandezas físicas tais como a temperatura, a pressão, a vibração etc., mas também capaz de auto-diagnosticar o seu próprio estado da saúde e enviar dados digitalizados de medição via rede de computadores para que o computador possa processar informações em tempo real.

Um dos objetivos do processamento das informações no computador é a predição da manutenção antes que ocorram falhas que resultem em paradas inesperadas. Para que isso seja possível, torna-se necessário o software inteligente capaz de tomar decisões baseadas nas evoluções dos valores das variáveis fornecidos por sensores no decorrer do tempo de operação. Isto significa que quanto maior a quantidade de variáveis monitoradas, maior será a qualidade da decisão.

O software que processa informações será baseado no sistema especialista onde o conhecimento do operador experiente estará presente.

A automação que é pretendida não se resume somente na predição da manutenção, mas também deve abranger o gerenciamento da operação assim como auxiliar na decisão administrativa. Atualmente, existem diversos software disponíveis no mercado que dá suporte ao gerenciamento de operação, manutenção e administração. A utilização desses softwares está planejada para dar suporte a gerenciamento.

Será implantado o KPI (*Key Performance Indicators*) como o indicador de desempenho.

2. DESCRIÇÃO TÉCNICA DO SISTEMA

Como o presente documento é dirigido aos colaboradores da Manaus Energia, a descrição técnica do sistema atual da UHE Balbina foi colocada no Apêndice A, pois acredita-se que a maioria dos leitores já tenha conhecimento do sistema.

3. DESCRIÇÃO DO SISTEMA DE AUTOMAÇÃO DA UHE DE BALBINA

O projeto de modernização executado em Balbina utilizou-se de uma arquitetura e de tecnologias no estado da arte, ou seja, as opções tecnológicas foram bastante acertadas para monitoração das grandezas dos ativos de geração e transmissão através do uso de instrumentação inteligente baseado em tecnologia FieldBus Foundation, sistema de controle e supervisão (operação da Usina) baseado em tecnologia Rockwell, sistemas de Execução de Manufatura (MES) Rockwell, sistema de Manutenção Preditiva de Balbina (SIMPREBAL) e o sistema de planejamento da Manutenção Maximo, certamente irão contribuir para melhoria dos índices de performance (MTBR, qualidade da energia, manutenibilidade, confiabilidade) de Balbina. A figura 1.1 apresenta o modelo hierárquico de automação adotado em Balbina.

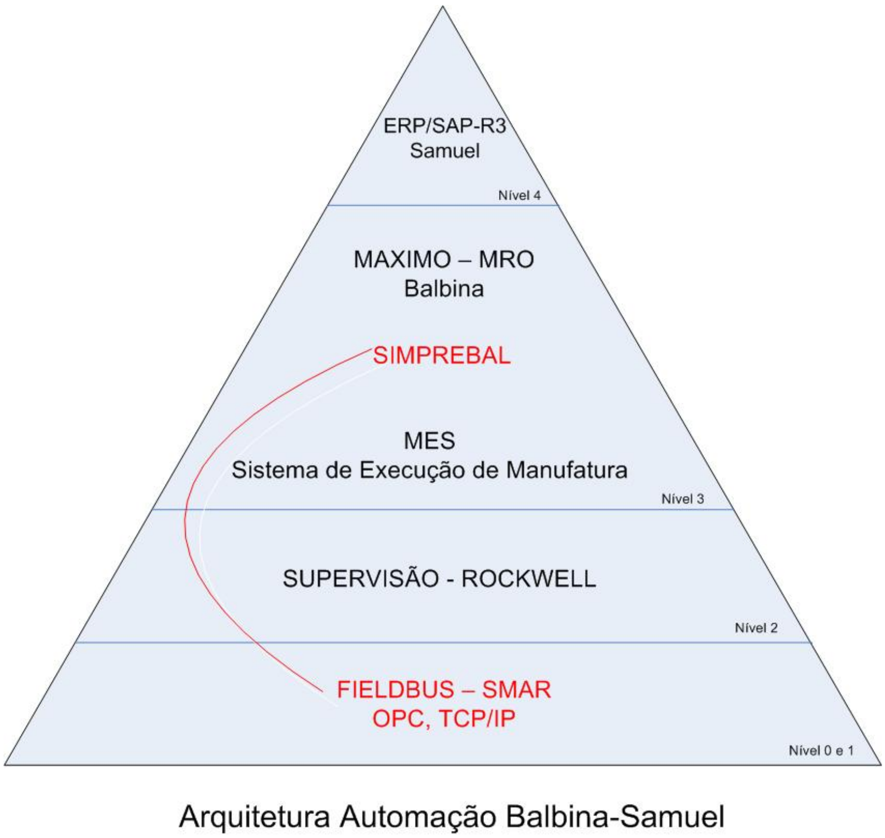


Figura 1.1 Modelo Hierárquico de Balbina

O modelo hierárquico de Balbina é baseado nas normas ISA-88 e ISA-95. ISA-95 define cinco níveis hierárquicos em companhias industriais: Nível 0, 1, 2, 3 e 4. O foco de ISA-88 é até o nível 2. ISA-95 focaliza sobre os níveis 3 e 4.

O Nível 0, 1 e 2 são níveis de controle de processo. Seu objetivo é o controle do equipamento, que é responsável pela execução do processo de produção de um ou mais produtos.

O Nível 3 pode ser chamado de nível atividades do MES (Sistema de Execução de Manufatura). O nível 3 consiste de muitas atividades que devem ser executadas para preparar, monitorar e completar o processo de produção que é executado nos níveis 0, 1 e 2. Por exemplo, atividades como operação e planejamento da manutenção da Usina de Balbina são executados pelas atividades MES no nível 3, assim como, a manutenção baseada em condição. O MES permite a integração do chão-de-fábrica (níveis 0, 1 e 2) com os níveis de gerenciamento da produção e da empresa (níveis 3 e 4). O MES é a interface entre os níveis de processo e o nível de gestão da empresa.

O nível mais superior (nível 4) deve ser chamado de sistema ERP (Planejamento de Recursos da Empresa). Neste nível atividades de logística e financeira são executadas. Estas atividades não estão diretamente relacionadas a produção. Por exemplo planejamento estratégico, vendas e mercado.

ISA-88 define o modelo Físico, o qual estrutura a Empresa hierarquicamente do maior nível para o mais baixo nível: Empresa, Sítio, Área, Célula de Processo, Unidade, Módulo Equipamento e Módulo Controle. ISA-88 é especialmente focada no nível de Controle de Processo e nos níveis mais baixos. ISA-95 é focada na fronteira entre os níveis de Área e de Sítio.

ISA-88 (Apêndice B) é um padrão para indústrias de transformação em lote. Ela também pode ser aplicada em processos discretos e contínuos que requerem um certo grau de flexibilidade. Os modelos ISA-88 e terminologia são desenvolvidos para o controle da produção. Portanto, produto e produção são o foco de ISA-88.

ISA-95 é o padrão internacional para a integração das empresas e dos sistemas de controle. ISA-95 consiste de modelos e terminologia que podem ser usadas para determinar qual a informação tem que ser trocadas entre os sistemas de vendas, finanças, logística e sistemas de produção, manutenção e qualidade. Esta informação está estruturada em modelos UML (Linguagem de Modelagem Unificada), que são a base para o desenvolvimento de interfaces normalizadas entre sistemas ERP e MES. O padrão ISA-95 pode ser usado para diversos fins, por exemplo, como um guia para a definição dos requisitos do usuário, para a seleção de fornecedores de MES como base para o desenvolvimento de sistemas MES e banco de dados.

ISA-95 é baseada na estrutura hierárquica de ISA-88 de ativos físicos de uma empresa industrial e adota a terminologia ISA-88. Entretanto ISSA-88 foca na Célula do Processo e ISA-95 foca no Sítio e na Área. Para ISA-95 os níveis de Células de Processo e Unidade são apenas de interesse se é necessário a troca de informações entre a Empresa e o sistema de controle sobre estes níveis. Pode ser necessário para o sistema de logística da empresa (Maximo), ter informações sobre o Células de Processo, para poder agendar a longo prazo atividades no processo, como pode ocorrer no caso da manutenção programada.

As soluções e tecnologias adotadas em Balbina são baseadas nos modelos e padrões ISA-88 e ISA-95 e certamente colocarão a Usina como uma referência internacional na área de automação de processos, pois as opções tecnológicas adotadas são o estado da arte em automação de processos. A arquitetura do sistema em implantação deverá ser referência para o Setor Elétrico Brasileiro e possibilitará a implantação de políticas de manutenção baseada em condição utilizando-se das variáveis já monitoradas pela instrumentação inteligente implantada em Balbina, através da tecnologia OPC (OLE for Process Control) que disponibiliza as TAGs geradas pela instrumentação Foundation FieldBus.

O sistema de automação da UHE Balbina em implantação utiliza os seguintes subsistemas:

a) Sistema de sensores para monitoração das variáveis da máquina;
b) Sistema de comunicação (rede) para transmissão de dados medidos e entre os computadores;
c) Sistema de processamento das informações provenientes das máquinas (hardware e software);
d) Recursos Humanos para operação, manutenção e administração.

3.1 SISTEMA DE SENSORES PARA MONITORAÇÃO DAS VARIÁVEIS DAS MÁQUINAS

A automação industrial de qualquer natureza necessita de sistema de sensoriamento para adquirir informações atuais sobre variáveis envolvidas no processo. No presente caso são: temperatura do estator, do óleo de lubrificação dos mancais, dos mancais do gerador e da turbina; a pressão e a vazão do óleo de lubrificação dos mancais, a quantidade de energia elétrica gerada, a tensão e a corrente do estator e do transformador elevador, a posição das palhetas do distribuidor e das pás do hélice, a densidade do óleo de lubrificação, a vibração do eixo da turbina e do gerador, o valor do entreferro do gerador, etc. Para cada nível de automação pretendido é necessário um conjunto de sensores para monitoração das variáveis envolvidas no processo de automação.

Como a primeira etapa de automação, foram instalados os sensores de temperatura, pressão e de densidade. Na etapa seguinte está planejada instalação de sensores de vibração. Os sensores para medir entreferro do estator, sensor para medir a espessura da lamina de óleo nos mancais deverão ser planejados futuramente.

3.2 SISTEMA DE COMUNICAÇÃO (REDE) PARA TRANSMISSÃO DE DADOS MEDIDOS E ENTRE OS COMPUTADORES

As redes de comunicações que serão utilizadas no presente sistema de automação foram escolhidas de acordo com as suas características dos equipamentos, ou seja, cada grupo de equipamentos que compõem o sistema de automação foi projetado com o tipo de rede de comunicação mais apropriado considerando a quantidade de dados a serem transmitidos, a segurança, a robustez, etc.

O documento (Configuração do Sistema – Usina Hidroelétrica de Balbina) ilustra a arquitetura do sistema de automação. Nele pode ser visto vários tipos de rede de comunicação: Devicenet, controlnet, profibus, foundation fieldbus, modbus e ethernet.

A seguir serão descritas algumas características de cada uma dessas redes de comunicações.

3.2.1 DeviceNet

A rede devicenet é uma rede de baixo nível aberta que realiza conexões entre os dispositivos industriais simples (tais como os sensores e os atuadores) e equipamentos de nível mais alto (tais como Controladores Lógicos Programáveis e computadores). A devicenet usa o protocolo industrial comum (CIP) e possibilita o controle, configuração, aquisição de dados dos dispositivos industriais. A devicenet é uma rede flexível que trabalha com equipamentos de diferentes fabricantes, ou seja, equipamentos industriais fabricados por diferentes fabricantes estabelecem comunicações entre si através da devicenet.

No presente caso é utilizado para comunicação entre os sensores e equipamentos de tratamento de sinais desses sensores.

3.2.2 Controlnet

A rede controlnet é uma rede aberta e avançada, adequada nas demandas de tempo real, aplicação do controle com elevada velocidade de comunicação. A controlnet usa o protocolo industrial comum (CIP) e combina a funcionalidade de uma rede de I/O e de uma rede par-à-par que possibilita o desempenho de alta velocidade para ambas as funções.

3.2.3 Profibus (Process Field Bus)

Profibus é o sistema de comunicação industrial para automação de manufatura mais difundida na Europa, com forte crescimento em muitas outras áreas de aplicações. Profibus é suportado por Siemens e promovido pela Profibus User Organization (PNO). Os produtos com Profibus são certificados por PNO garantindo compatibilidade mundial.

Profibus é uma rede MultiMaster. Dependendo da variação do protocolo selecionado, a rede suporta comunicação Máster/Slave, Máster/Máster e Slave/Slave. Profibus é otimizada para transmitir os dados cíclicos do processo entre Máster e Slave rápida e eficientemente.

A velocidade de transmissão pode ser selecionada entre 9,6 kbits/s até 12 Mbits/s. Pode conectar 126 estações por rede.

3.2.4 Foundation Fieldbus

A Foundation Fieldbus é um sistema de comunicação totalmente digital, em série e bidirecional que conecta equipamentos “Fieldbus” tais como sensores, atuadores e controladores. O Fieldbus é uma rede local (LAN) para automação e instrumentação de controle de processos, com capacidade de distribuir o controle no campo.

A Foundation Fieldbus mantém muitas das características operacionais do sistema analógico 4-20 mA, tais como uma interface física padronizada da fiação, os dispositivos alimentados por um único par de fios e as opções de segurança intrínseca, mas oferece uma série de benefícios adicionais ao usuários.

Os benefícios da Fieldbus são Interoperabilidade, Dados de processo mais completos, Vista expandida do processo, Melhor segurança da planta, Manutenção proativa mais fácil, Redução de custo de fiação e de manutenção. É um barramento de campo usado principalmente para instrumentação, destacando-se a parte o sistemas de gestão de ativos Assetview. É pouco utilizada para controle. A DeviceNet e Profibus é mais adequado para aplicações de controle.

3.2.5 Modbus

Modbus é um protocolo de comunicação de dados utilizados em sistemas de automação industrial. Criado na década de 1970 pela Modicon. É um dos mais antigos utilizados em redes de Controladores Lógicos Programáveis (PLC) para aquisição de sinais de instrumentos e comandar atuadores. A Modicon colocou as especificações e normas que definem o Modbus em domínio público. Por esta razão, é utilizado em milhares de equipamentos existentes e é uma das soluções de rede mais baratas a serem utilizadas em automação industrial.

O Modbus utiliza o RS-232, RS-485 ou Ethernet como meio físico. O mecanismo de controle de acesso é mestre-escravo. A estação mestra (geralmente um PLC) envia mensagens solicitando dos escravos que enviem os dados lidos pela instrumentação ou envia sinais a serem escritos nas saídas para o controle dos atuadores. O protocolo possui comandos para envio de dados discretos (entradas e saídas digitais) ou numéricos (entradas e saídas analógicas).

Em redes seriais baseadas em RS-485 ou RS-232 o Modbus pode ter duas variações: RTU e ASCII. No modo RTU os dados são transmitidos em formato binário de oito bits, permitindo a compactação dos dados em pequenos pacotes. RTU é a sigla de Remote Terminal Unit. No modo ASCII os dados são codificados em caracteres ASCII de sete bits para serem transmitidos. Apesar de gerar mensagens legíveis para pessoas, este modo consome mais recursos da rede.

3.2.6 Ethernet

Ethernet é uma tecnologia de interconexão para redes locais (LAN) baseada no envio de pacotes. Ela define cabeamento e sinais elétricos para camada física e formato de pacotes e protocolos para camada de controle de acesso ao meio (Media Access Control – MAC) do modelo OSI. A Ethernet foi padronizado pelo IEEE como 802.3. A partir dos anos 90, ela vem sendo a tecnologia de LAN mais amplamente utilizada.

Ethernet é baseada na idéia de pontos da rede enviando mensagens, no que é essencialmente semelhante a um sistema de rádio, cativo entre um cabo comum ou canal, às vezes chamado de éter (referência obliqua ao éter luminoso). Cada ponto tem uma chave de 48 bits globalmente única, conhecida como endereço MAC, para assegurar que todos os sistemas em uma Ethernet tenham endereços distintos.

4. SISTEMA DE PROCESSAMENTO DE INFORMAÇÕES PROVENIENTES DOS SENSORES.

O sistema de processamento recebe as informações do sensor via rede e toma decisões quanto às ações que devem ser realizadas. As decisões quanto às ações que devem ser realizadas são baseadas na ferramenta conhecida como a solução MES (Manufacturing Execution System).

A solução MES foi desenvolvida e aprimorada no decorrer dos anos, num panorama onde o setor industrial vem sofrendo constantes pressões para alcançar a excelência operacional, objetivando garantir sua competitividade. Algumas das principais pressões observadas são: redução de custo de produção, gerenciamento pró-ativo da manufatura, cumprimento das datas planejadas e redução da variabilidade.

Para que tais pressões sejam suportadas, é necessário que sejam geradas informações e indicadores confiáveis. Como essas informações devem ser traduzidas em resultados financeiros, a qualidade das informações é crítica para uma ação efetiva de redução de custos. Além disso, torna-se vital que as informações sejam geradas de forma rápida, preferencialmente em tempo real, e que haja ferramentas de análise adequadas. Devido ao dinamismo da planta, mix de produtos, fluxo de materiais entre as linhas de produção, paradas de equipamentos, dentre outros fatores, torna-se também importante uma visão clara e integrada da situação de toda a planta e, conseqüentemente, um conhecimento das capacidades e restrições produtivas atuais.

Soluções MES (Manufacturing Execution System) são sistemas de informações operacionais com comunicação bidirecional entre chão de fábrica e sistemas corporativos que visam apoiar de forma efetiva as intenções estratégicas relacionadas direta ou indiretamente com as operações de manufatura.

Através da utilização da MES as empresas podem medir e controlar as atividades críticas de produção. Alguns dos benefícios da utilização da MES são: aumento da rastreabilidade, produtividade e qualidade. Outras funções servidas pela MES são aumento da rastreabilidade dos equipamentos, genealogia dos produtos, rastreamento de trabalho, gerenciamento do inventário, redução de custos, monitoramento dos defeitos e soluções, cálculo de índice de desempenho (KPI), alarme e outras soluções variadas.

No presente caso de automação será utilizada a solução MES, fornecida pela empresa Rockwell em integração com o sistema de gerenciamento da manutenção e o software MAXIMO, fornecida pela MRO Software. Além disso, será utilizado um software desenvolvido pela equipe coordenada por Prof. Dr. Alberto Álvares da Universidade de Brasília para predição de manutenção utilizando sistema especialista e lógica Fuzzy denominado de SIMPREBAL (Sistema de Inteligente Manutenção Preditiva de Balbina).

4.1 MES Rockwell

O objetivo estratégico da MES da Rockwell é a integração bidirecional com o chão-de-fábrica fazendo coleta automatizado das variáveis das máquinas eliminando anotações manuais, o que resulta na maior confiabilidade e maior velocidade, e utilizando esses dados para gerenciamento em tempo real com acesso remoto através da Web, emitindo relatório via e-mail. Podem gerar relatórios sobre as informações de Performance e Eficiência, Motivo de paradas e eventos da linha, Comparativo entre capacidade real e produzidas, relatório de produção e produtividade, histórico de variáveis do processo produtivo, monitoramento de unidades, etc.

As interligações das partes componentes para realizar estas tarefas são mostradas na figura 4.1.

4.2 GERENCIAMENTO ESTRATÉGICO DE MANUTENÇÃO MRO MAXIMO

O gerenciamento estratégico de manutenção será realizado através do software MAXIMO da empresa MRO Software em parceria com Rockwell possibilitando:

- Gestão dos ativos de produção;
- Geração automática de ordens de manutenção;
- Migração automática das ordens de manutenção;

obtendo os seguintes resultados:

- Otimização das Manutenções Corretivas, reduzindo perdas na produção;
- Aumento significativo nas ordens de manutenção atendidas;
- Atualização de dados ON-LINE no histórico dos equipamentos devido automação da coleta de dados;
- Otimização de pessoal devido a redução de atuação humana na coleta de dados;
- Maior confiabilidade nas informações;
- Ampliação das Manutenções Preditivas para mais equipamentos, utilizando a plataforma de coleta de dados;
- Visibilidade de plano de produção para apoio a programação das ordens de manutenção, possibilitada pela integração com sistema de produção;
- Melhor adequação entre o plano de produção e plano de manutenção;
- Redução da documentação em papel.

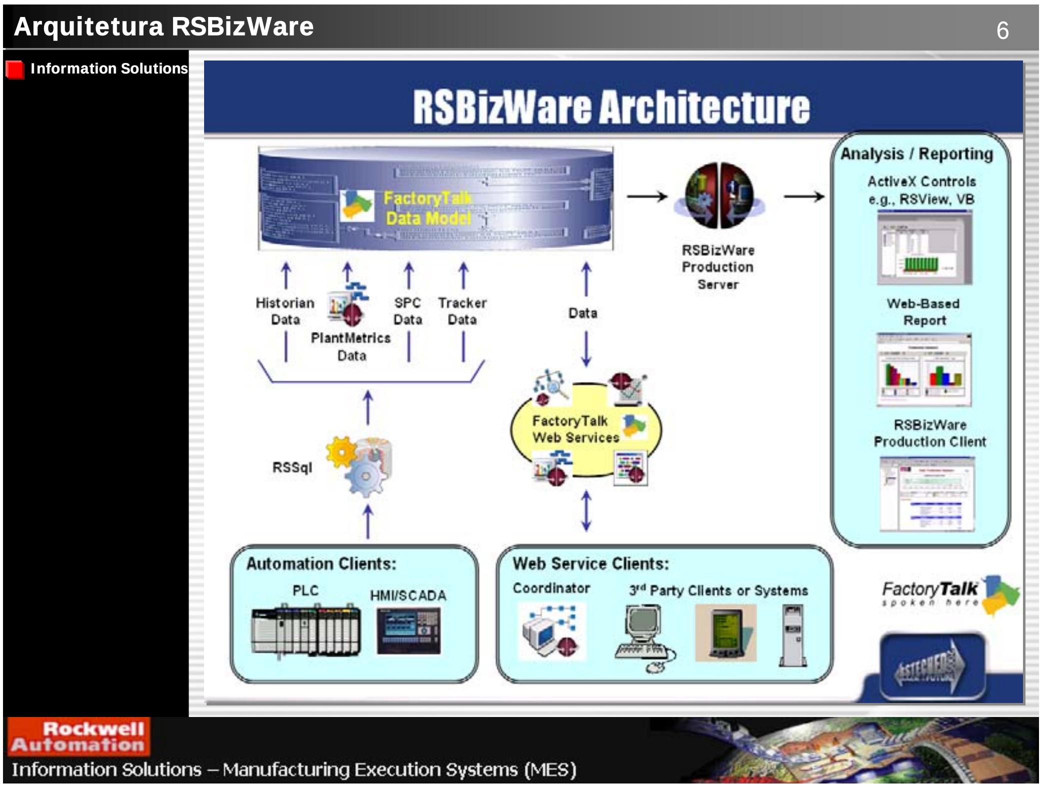


Figura 4.1: Arquitetura da RSBizWare

As características operacionais do MAXIMO são:

- Facilidade de uso;
- Sistema Full Web desenvolvido em JAWA;
- Integração com Automação/Chão de fábrica;
- IPC – busca por imagem;
- Hand Helds – comunicação remota;
- Facilidade de integração através do MAXIMO Enterprise Adapter;

Os maiores detalhes sobre os requisitos técnicos do MAXIMO, assim como as plataformas necessárias para execução do sistema na Manaus Energia, podem ser vistos no documento "MAXIMO – Documento de Referência".

4.3 SIMPREBAL

SIMPREBAL consiste num sistema de manutenção inteligente para auxiliar a tomada de decisão. A metodologia de desenvolvimento deste sistema é baseada em conceitos de manutenção centrada em confiabilidade (MCC), sendo utilizada para analisar as funções, os modos e efeitos de falhas (FMEA) das Unidades Geradoras Hidráulicas de Balbina a partir das grandezas monitoradas pelo sistema de supervisão e controle da usina. A análise está focada no sistema da turbina, entretanto, a metodologia proposta é genérica, podendo ser utilizada indistintamente em qualquer outro sistema, como, por exemplo, no sistema do gerador.

4.3.1 CONCEITOS BÁSICOS DA MCC

Originária da indústria aeronáutica americana, e adotada pelas indústrias nuclear e elétrica mundiais, a MCC é hoje aplicada em muitos outros setores modernos da economia, inclusive o terciário e de serviço. Entre as tecnologias contemporâneas de manutenção, a MCC tem expandido sua aplicação a praticamente todos os ramos de atividade humana, onde haja necessidade de manter o funcionamento de ativos físicos ou processos.

A literatura aponta a MCC como uma ferramenta de manutenção, que visa racionalizar e sistematizar a determinação das tarefas adequadas a serem adotadas no plano de manutenção, bem como garantir a confiabilidade e a segurança operacional dos equipamentos e instalações ao menor custo. Nesses termos, para Branco Filho (2000, p.41), a MCC "com sua ênfase em otimização, documentação, rastreabilidade e continuidade está sintonizada com as mudanças gerenciais que vêm se processando ultimamente na indústria em geral".

De acordo com Smith (1992), a MCC tem o propósito de "preservar as funções do sistema, identificar os modos de falha que afetam essas funções, determinar a importância das falhas funcionais [...] e selecionar as tarefas aplicáveis e efetivas na prevenção das falhas" (p.51).

4.3.2 MODELO DE REFERÊNCIA USADO PARA MANUTENÇÃO BASEADA EM CONDIÇÃO: OSA-CBM

Será utilizada como referência para o desenvolvimento do sistema de manutenção inteligente baseado em condição a arquitetura OSA-CBM (Open System Architecture for Condition Based Maintenance) descrita na URL http://www.osacbm.org.

A arquitetura OSA-CBM consiste em sete camadas (Fig. 4.2). A noção de uma arquitetura estendida em camadas, usada aqui, é consistente com a usada por Buschman (1996). Uma camada é vista como uma coleção de tarefas semelhantes ou funções em níveis diferentes de abstração.

As camadas hierárquicas representam uma transição lógica ou um fluxo da saída dos sensores para a camada de tomada de decisão, através das camadas intermediárias. A camada de apresentação é uma exceção dentro da arquitetura, pois permite comunicação ponto-a-ponto entre esta camada e qualquer outra.

A seguir são apresentadas as sete camadas e como estas camadas se ajustam ao sistema SIMPREBAL:

Módulo de sensor: A camada de módulo de sensor consiste no transdutor e elementos de aquisição de dados. O transdutor converte alguns estímulos em energia elétrica ou óptica. A aquisição de dados é a formatação de produção analógica do transdutor para um formato digital. Neste módulo serão utilizados os sensores e conversores Foundation FieldBus da Smar, System 302. Também é previsto o uso de sensores da Rockwell para monitoração de vibração, por exemplo.

Processamento de sinal: A camada de processamento de sinais processa os dados digitais do módulo de sensor de modo a convertê-los numa forma específica capaz de representar a grandeza física que está sendo monitorada e, caso necessário, efetuar cálculos matemáticos sobre ela. Além disso, esta camada é responsável por processar os parâmetros e comandos enviados pelo usuário ao módulo sensor. Contribuições para esta camada podem incluir tipo de módulo de sensor e informação de local e dados de calibração. Também são utilizados a instrumentação FieldBus e Rockwell. No caso do sistema de monitoração de vibração da Rockwell, o sistema tem incorporado funcionalidades para tratamento de sinais usando Transformada Rápida de Fourier TRF.

Figura 4.2 – As sete camadas da arquitetura OSA-CIM.

Monitoração de Condição: Esta camada determina a condição da planta, de seus subsistemas, ou componentes (excede limiar, ciclo de tensão, condição operacional, métrica de uso) baseado em algoritmos, sistemas inteligentes e na saída dos módulos de sensores e processamento de sinal. Esta camada pode fazer uso de histórico de condição local e fornece parâmetros para o modelo. Duas dimensões devem ser consideradas. Na primeira o conversor FieldBus e o sistema Assetview da Smar fornecem análises relativa a condição da instrumentação FieldBus. Na segunda dimensão o sistema SIMPREBAL fornece a condição do sistema monitorado.

Avaliação de estado: A camada de avaliação de estado determina o estado do sistema, subsistemas ou componentes monitorados baseado na saída do modulo de monitoração de condição, das condições históricas e de valores de referência. A saída desta camada é um índice de estado do equipamento monitorado. Duas dimensões também devem ser consideradas. Na primeira o conversor FieldBus e o sistema Assetview da Smar fornecem a avaliação do estado da instrumentação FieldBus. Na segunda dimensão o sistema SIMPREBAL fornece a avaliação do estado do sistema monitorado.

Prognósticos: A camada de prognósticos considera a avaliação de estado do sistema, subsistema, ou componente, o escalonamento empregado (predição de uso - cargas e duração) e capacidade do modelo/raciocínio que pode predizer o estado do equipamento, com uma determinada precisão. Lee et al. (2004) apresentam seu sistema WatchDog Agent que implementa dezenas de ferramentas e algoritmos de prognósticos, baseados em Transformada de Fourier, Modelo Auto-regressivo, Lógica Fuzzy, Redes Neurais, entre outros. O sistema SIMPREBAL irá utilizar abordagem baseada em Sistemas Inteligentes – Sistemas Especialistas e Lógica Fuzzy – que serão desenvolvidos usando as ferramentas Jess, FuzzyJess e o MatLab .

Tomada de decisão: A camada de tomada de decisão integra informações necessárias para que se tome uma determinada atitude diante de uma condição específica do sistema. É baseada em informações sobre a avaliação do estado do sistema, subsistemas ou componentes, bem como em uma noção a respeito da severidade, urgência e importância de se tomar certa decisão. A tomada de decisão envolve uma avaliação rigorosa das conseqüências de tal decisão, além de exigências de missão e incentivos financeiros. Provê indicando ações e alternativas com as implicações de cada alternativa. O sistema SIMPREBAL irá realizar a tomada de decisão baseado na sua base de conhecimento e relacionamento entre a árvore de faltas/falhas e a árvore de sintomas, complementado pelas informações de inspeções simplificadas e detalhadas realizada pelo sistema de Manutenção Centrada em Confiabilidade (MCC).

Apresentação: A camada de apresentação suporta a interface com o usuário. Permite a monitoração, por meio de visualização dos dados medidos pelo módulo sensor, e o controle do sistema, através de parâmetros enviados pelo usuário ao módulo sensor. Esta camada provê independência para as aplicações em relação às diferentes formas de representação dos dados, bem como, fornece inteligibilidade aos resultados produzidos pelas camadas inferiores, e fornece também a interface homem/máquina. A camada de apresentação do sistema SIMPREBAL será desenvolvida para GUI (Interface Gráfica com o Usuário) baseada em browser (Netscape, Mozilla, IExplore, entre outros) usando html, XML, javascript e applets (Java).

5. RECURSOS HUMANOS

Sempre que há implementação de novos sistemas deverá ter treinamento do pessoal para que possam operar eficientemente. No presente caso, deverá haver treinamento do pessoal para operação do sistema, e principalmente para manutenção de redes de computadores já que a rede de computadores exerce uma função essencial neste sistema. Este aspecto já está previsto nos planos da Manaus Energia sendo elogiável todas as ações em execução para treinamento e reciclagem do pessoal de manutenção e operação, como a montagem de um laboratório baseado na Planta Didática III (PD-III) da Smar e todas as ações de capacitação de pessoal que estão em execução.

A dimensão de recursos humano no Plano Diretor de Automação é fundamental para o sucesso e manutenção do sistema de automação implantado. Nota-se a preocupação da gerência do projeto com os aspectos e impactos da introdução desta nova tecnologia na Usina de Balbina. Deve ficar claro para o pessoal de chão-de-fábrica que toda a tecnologia incorporada à Usina deve ser vista na realidade como uma poderosa ferramenta que permitirá o gerenciamento otimizado da planta de Balbina. É fundamental para o sucesso do processo de modernização de Balbina que esta nova cultura seja absorvida por todo o pessoal envolvido em atividades de operação e manutenção da Usina.

O sistema de automação deve ser visto como um aliado pelo pessoal de manutenção e operação. O sistema de automação é aberto e permitirá que o mesmo evolua a cada dia melhorando os indicadores de desempenho da Usina, desde que haja um comprometimento e envolvimento do corpo técnico da Usina, não enxergando o sistema de automação como um concorrente e sim como um aliado, que irá facilitar a gestão dos ativos de Balbina para obtenção de índices de desempenho cada vez melhores.

6. ASPECTOS TECNOLÓGICOS

Na implementação de um sistema de automação eficiente, não basta instalar novo sistema de sensoriamento e implementar novo sistema de processamento. É necessário que haja levantamento dos pontos de defeitos freqüentes, pois mesmo que o novo sistema de manutenção venha a detectar previamente, se a freqüência de manutenção for alta, prejudica a produção.

Como novo sistema de controle, serão instalados dois sistemas de PLC em paralelo sendo que na situação normal apenas um deles estará ativo para realizar o controle de cinco máquinas (no sistema atual um PLC controla uma máquina). Quando esse sistema parar por alguma razão, o outro assumirá a função dele. Assim, o sistema antigo de reserva, implementado através de lógica a relé será desativado.

Através da análise do ponto de manutenção freqüente, foi detectado que o trocador de calor do óleo de refrigeração da máquina 5 precisa ter uma solução mais radical, ou seja, precisa resolver a fonte do problema que é a qualidade da água de refrigeração que está causando excesso de material incrustado em pouco tempo.

Um outro ponto que será substituído por uma tecnologia mais moderna são os sistemas de partida das motobombas para pressurizar óleo de lubrificação. Atualmente existem 123 motobombas, sendo que um sistema de partida para duas motobombas, implementados a lógica de relé utilizando contactores. Como a vida útil dos contactores não é longa, atualmente necessita de manutenção freqüente. Por esse motivo serão instalados “soft start” para motobombas de potência mais alta e relé a semicondutor com proteção contra excesso de potência E3 Plus para substituir o atual sistema de partida de demais motobombas.

7. ATIVIDADES PARA IMPLEMENTAÇÃO DA AUTOMAÇÃO

A implementação da automação na UHE de Balbina seguiu os seguinte passos estando na fase final de implantação do sistema de supervisão controle da Rockwell:

- Seleção dos equipamentos para automação (já realizada), sendo os sensores de pressão e temperatura são de SMAR, o controlador (PLC) da Rockwell.

- Instalação dos sensores e controladores – concluída.

- Programação do PLC- já se encontra basicamente concluída pela Rockwell e já foi testado na plataforma SMAR.

- Treinamento do pessoal técnico – Com a implementação do sistema novo é necessário que haja treinamento do pessoal técnico tanto de operação quanto de manutenção e administração.

8. CONFIABILIDADE E BACKUP DE BANCO DE DADOS

Quando tenta avaliar sistema automatizado, o ponto principal é a sua robustez e funcionalidade. Neste aspecto, o sistema a ser implementado com duas unidades de PLC (sistema redundante) para controlar o sistema é uma estratégia que aumenta a confiabilidade do sistema. Entretanto, para que a comutação de um PLC para outro de reserva seja feita de forma tranqüila, é necessário que haja treinamento freqüente, ou seja, é necessário realizar esta atividade de comutação do PLC com certa freqüência, mesmo que o PLC ativo não esteja com defeito.

Outro ponto que deve ser alertado é quanto ao armazenamento de dados, que serão coletados automaticamente, em forma de arquivo informatizado. Como esses dados eram anotados em papel, o método de gerenciamento desses dados requer certos cuidados, pois arquivos informatizados tem a facilidade de acesso, por outro lado os mesmos poderão ser perdidos (deletados) com facilidade. Para evitar que isso aconteça é necessário que haja "backup" freqüente de arquivo de dados, devendo-se adotar uma política de backups de banco de dados.

9. ATIVIDADES FUTURAS

Será implantado o KPI (Key Performance Indicators) na UHE da Balbina, como a metodologia para avaliação de desempenho, onde o sistema automatizado de monitoração exerce um papel fundamental.

10. INTRODUÇÃO KPI

Muitas pessoas utilizam o termo KPI (Key Performance Indicators) e métricas (ou indicadores) de desempenho indistintamente. Isso é errado, pois um KPI é uma métrica, mas nem toda métrica é um KPI. De acordo com Eckerson (2006), A diferença básica é que um KPI sempre deverá refletir direcionadores (drivers) de valor estratégico enquanto que uma métrica apenas representa uma medida de desempenho de uma atividade.

KPIs são "veículos de comunicação". Permitem que os executivos do alto escalão comuniquem a missão e visão da empresa aos mais baixos níveis hierárquicos, envolvendo diretamente todos os colaboradores na realização dos objetivos estratégicos da empresa.

Na logística e no supply chain management existem inúmeras possibilidades de composição de KPIs e por isso, seguir as recomendações aqui relatadas é crucial para o sucesso da área, e conseqüentemente da organização.

Bons KPIs devem apresentar as 10 características citadas a seguir, enquanto que simples métricas ou indicadores de desempenho apresentarão parte delas.

#1: KPIs devem refletir direcionadores (drivers) de valor estratégico
KPIs refletem e medem direcionadores-chave de valor. Direcionadores de valor representam atividades que quando executadas corretamente, garantem o sucesso futuro da organização. Direcionadores de valor movimentam a empresa para a direção correta, impulsionando-a para alcançar seus objetivos organizacionais e resultados financeiros. Exemplos de direcionadores de valor podem ser "alta satisfação do Cliente" ou "excelência na qualidade de um produto".

#2: KPIs são definidos por Executivos
Executivos definem os direcionadores de valor em reuniões de planejamento, onde determinam o direcionamento estratégico da empresa, no curto, médio e longo prazos. Para extrair o máximo desses direcionadores de valor, os executivos necessitam definir como eles querem mensurar o desempenho organizacional através desses direcionadores. Infelizmente, muitos executivos terminam o seu planejamento estratégico antes de definir e validar essas medidas de desempenho. Não podemos gerenciar aquilo que não é medido.

#3: KPIs devem fluir ao longo da empresa
Qualquer grupo, em qualquer nível hierárquico, de qualquer organização, é gerenciado por um "executivo". Esses executivos podem ser conhecidos por "presidentes divisionais", "diretores", "gerentes" ou "supervisores", entre diversos outros termos. Como "executivos", também necessitam realizar sessões de planejamento estratégico e identificar direcionadores-chave de valor, objetivos e planos para o seu grupo. Devido ao fato de cada "executivo", em cada nível hierárquico, basear-se em direcionadores-chave repassados por seus superiores, acaba ocorrendo um "efeito cascata", fazendo com que seja refletido nos KPIs dos mais baixos níveis organizacionais os valores propostos pelos altos executivos da empresa.

#4: KPIs são baseados em padrões corporativos
A única forma de fazer com que um KPI flua ao longo de uma organização é através da criação de padrões de medição. Isso é muito difícil e trabalhoso, mas não impossível.

#5: KPIs são baseados em dados válidos

Antes de optar pela utilização de um determinado KPI é necessário saber se a informação existe e qual a sua precisão (acuracidade). Freqüentemente a resposta não é positiva. Em alguns casos as empresas optam por investir em sistemas que possam capturar as informações necessárias. Em outros casos preferem revisar o KPI.

#6: KPIs devem ser fáceis de serem compreendidos
Um dos grandes problemas dos KPIs é que existem muitos deles. Como resultados disso, eles perdem o poder de atrair a atenção dos empregados e de modificar o seu comportamento. De acordo com uma pesquisa da TDWI (The Data Warehousing Information), uma organização deveria ter, na média, sete KPIs por usuário. Mais do que isso torna difícil para os empregados tomarem a decisão requisitada para cada um deles.
Além disso, os KPIs devem ser de fácil entendimento. Os colaboradores devem saber como calculá-lo e principalmente, o que fazer (e o que não fazer) para alcançar as metas pretendidas. Treinamentos e reuniões de acompanhamento são necessárias para o perfeito entendimento. Medidas de desempenho sem reuniões são inúteis.

#7: KPIs são sempre relevantes
Para garantir que os KPIs possam, continuamente, melhorar a performance da empresa, será preciso auditar os KPIs para avaliar o seu uso e relevância. Se um KPI não está sendo utilizado, ele deverá

ser re-escrito ou mesmo descartado. Muitos KPIs têm um ciclo de vida e uma vez criados, energizam a força de trabalho, produzindo resultados incríveis. Com o passar do tempo perdem o seu valor, e provavelmente, tenham que ser redesenhados. Muitas organizações realizam revisões a cada 4 ou 6 meses para avaliar a eficiência de seus KPIs.

#8: KPIs proporcionam contexto
Métricas sempre são expressas em número que refletem o desempenho. Mas um KPI coloca essa performance em um contexto. Ele avalia o desempenho em função de expectativas. O contexto é proporcionado através de limites, metas, benchmarks, etc. KPIs devem indicar a direção da performance, como acima, abaixo ou estático.

#9: KPIs criam "empowerment" nos usuários
É muito comum dizer que aquilo que não é medido não é gerenciado. Mas é também verdadeiro que não pode ser medido aquilo que não possa ser recompensado. Para serem efetivos, os KPIs devem ter uma recompensa atrelada a eles. Cerca de 40% das empresas pesquisadas pela TDWI (The Data Warehousing Information) nos EUA informaram ter reestruturado seus sistemas de incentivos com a implementação de KPIs.

#10: KPIs conduzem a ações positivas
KPIs não podem ser criados de forma isolada. Devem gerar ações de melhoria conjuntamente. Objetivos antagônicos poderão enfraquecer KPIs e colocar em risco a realização de importantes objetivos estratégicos da empresa.

11. METODOLOGIA DE DEFINIÇÃO E DESENVOLVIMENTO DE KPIs

KPIs são medições quantificadas que antecipadamente refletem os fatores críticos de sucesso. Sejam quais forem, eles devem ser selecionados e devem ajudar a organização a definir e medir seu progresso em direção as suas metas organizacionais. Um bom KPI deve refletir a missão, a visão e os valores da empresa.

A missão da Manaus Energia é: “atender, com excelência, o mercado de energia elétrica na sua área de atuação, contribuindo para o desenvolvimento da Amazônia”. Sua visão é: “ser reconhecida como uma empresa de excelência no negócio de energia elétrica”. E seus valores são: “foco nos clientes; valorização das pessoas; responsabilidade social; comportamento ético; respeito ao meio ambiente; segurança e excelência na gestão” (Manaus Energia, 2007).

A norma internacional ISA-95, da Instrumentation, Systems and Automation Society, estabelece uma série de instruções que visam determinar o grau de sucesso alcançado ou previsto para a operação de uma empresa no atendimento aos seus objetivos principais (Dufort, 2006).

De acordo com esta norma, para que se possa atender aos objetivos principais e escolher corretamente o posicionamento estratégico da empresa deve-se definir à mesma uma cadeia de agregação de valor. Segundo Dufort (2006), Os procedimentos básicos necessários para construção da cadeia de agregação de valor incluem:

- Compreender completamente as necessidades e expectativas dos clientes-alvo;

- Documentar tais necessidades e expectativas;
- Identificar fatores críticos de sucesso e métricas relevantes que se adequam às necessidades e expectativas dos clientes;
- Priorizar ações (ou projetos) utilizando-se de um gráfico de valor. O gráfico de valor quantifica benefícios versus risco associado;
- Definir e desenvolver KPIs apropriados, estabelecer metas e cronogramas que garantam o cumprimento das metas;
- Definir um sistema de aquisição de dados capaz de padronizar e sistematizar as informações operacionais compatibilizando-as com as métricas das cadeias de agregação de valor.
- Medir, visualizar e analisar os KPIs e comparar os valores medidos com as metas estabelecidas;
- Revisar periodicamente com os clientes suas necessidades e expectativas;
- Ajustar os KPIs sempre que necessário, adequando-os às eventuais novas necessidades dos clientes, ou quando há alterações nos modelos de negócio ou de mercado.

Em sistemas de potência, a avaliação de desempenho medido através de índices ou indicadores de desempenho, baseados nas normas ISA-95, pode ser desenvolvida tendo como referência a seguinte cadeia de agregação de valor (Bernini, 2006):

- Excelência de serviço;
- Excelência operacional;
- Excelência de segurança;
- Eficácia de custos (Borges, 2005).

A partir dos requisitos supracitados, são definidos diferentes KPIs com diferentes enfoques, conforme descrito nas seções subseqüentes.

11.1 KPIs PARA EXCELÊNCIA DE SERVIÇO

A excelência de serviço consiste em determinar quais são os atributos do serviço que de fato são determinantes para a satisfação dos clientes. No caso de geração de energia elétrica, a excelência de serviço, conforme definida por Borges (2005), consiste no atendimento à demanda elétrica dos consumidores com 3 atributos:

1. Regime contínuo
2. Boa qualidade
3. Margem de segurança conveniente

11.1.1 REGIME CONTÍNUO

Segundo os indicadores de desempenho padronizados pelo Operador Nacional de Sistemas (ONS), a continuidade de fornecimento de energia elétrica pode ser contabilizada por meio da duração das interrupções forçadas de suprimento.

Interrupções forçadas consistem na retirada de uma unidade geradora de serviço, em condições não programadas, e geralmente resultante da ocorrência de uma condição de emergência que impõe que a mesma seja desligada manualmente ou automaticamente para evitar a sua danificação e/ou outras conseqüências ao sistema elétrico e/ou riscos humanos. Na usina de Balbina todas as interrupções forçadas são realizadas automaticamente por atuação do sistema de proteção e são comumente denominadas de TRIP.

Define-se o índice Duração das Interrupções Forçadas (DINT), para cada unidade geradora, como sendo o somatório da duração das interrupções forçadas (TRIPs), dado em minutos por mês.

$$DINT = \sum_i D_{TRIP\,i}$$

onde,

$D_{trip\,i}$ = Duração do TRIP associado ao evento i de uma determinada máquina, em minutos.

O índice de duração das interrupções forçadas deve ser calculado pelo intervalo de tempo decorrido entre o instante de acionamento dos relés de parada por defeito hidráulico, elétrico ou mecânico (5HX1, 5EX1, 5MX1, respectivamente) e o instante de religamento da máquina.

O Operador Nacional de Sistemas determina ainda, conforme detalhado nos Procedimentos de Rede – módulo 22, que tal índice, supramencionado, deve ser calculado mensalmente e registrado em relatório. Os indicadores serão apurados por causa e origem, devendo ser coletadas, para cada evento de TRIP, as seguintes informações:

a) Dia do desligamento;
b) Hora do início do desligamento;
c) Hora do fim do desligamento;
d) Origem do desligamento;
e) Identificação do equipamento associado à origem do evento;
f) Tipo do evento (hidráulico, elétrico ou mecânico);
g) Carga interrompida (MW);
h) Severidade do desligamento.

A severidade do desligamento, também denominada pelo ONS de indicador Sistema-Minuto (SM), é definida pela seguinte equação:

$$SM = \frac{Carga\ Interrompida_i\ (MW)\ x\ D_{TRIP\,i}}{Demanda\ máxima\ instantânea\ verificada\ no\ período\ (MW)}$$

onde,

$Carga\ Interrompida_i$ = Carga interrompida em decorrência do TRIP i.

$D_{TRIP\,i}$ = Duração do TRIP em estudo, definido anteriormente, medido em minutos;

Demanda máxima instantânea verificada no período = Demanda máxima coincidente (MW), verificada no dia da perturbação, do(s) Agente(s) de Distribuição afetado(s).

Quando não se dispõe da demanda máxima coincidente, utiliza-se a não coincidente, ou seja, o somatório das demandas máximas dos agentes afetados.

O grau de severidade dos desligamentos forçados é classificado da seguinte forma:

i. Grau 1 (baixo): $1 \leq SM < 10$;
ii. Grau 2 (médio): $10 \leq SM < 100$;
iii. Grau 3 (alto): $100 \leq SM < 1000$.

11.1.2 QUALIDADE

A qualidade de serviço pode ser medida em função da freqüência das interrupções forçadas. Quanto maior o número de ocorrências de interrupções, obviamente menor será o fator de qualidade no fornecimento de energia elétrica.

Segundo o ONS, a freqüência das interrupções forçadas (FINT) é definida, para cada unidade geradora, como sendo o somatório mensal do número de ocorrências de TRIP, e é dada em ocorrências por mês.

$$FINT = \sum_i N^\circ_{TRIP\,i}$$

onde,

$N^\circ_{trip\,i}$ = Número de ocorrências de cada TRIP de uma determinada máquina.

11.1.3 MARGEM DE SEGURANÇA CONVENIENTE

A avaliação da margem de segurança do sistema de geração tem como objetivo garantir o suprimento adequado tendo em vista a disponibilidade de capacidade de geração. A capacidade da geração é a capacidade instalada que deve ser planejada antes da necessidade. A reserva estática é um produto do planejamento gerencial e deve ser suficiente para permitir manutenção nas unidades de geração, saídas programadas e não programadas, e crescimento não previsto da carga.

Os critérios utilizados para determinação da reserva estática são baseados em avaliações dos históricos de crescimento da demanda (fornecidos pelo Operador Nacional de Sistemas), bem como dos históricos de desligamento forçado e do planejamento de manutenção programada. A partir destes dados, é previsto anualmente o número esperado de déficit de energia de modo que este valor jamais seja maior do que 5% da demanda.

Por meio dos históricos de desligamento forçado e do gráfico anual de demanda é possível estabelecer métricas de desempenho tais como porcentagem de energia não suprida (ENS) por máquina, definida a seguir:

$$ENS = \frac{\sum_i \left(Carga\ Interrompida_i\ (MW) \times D_{TRIP\,i}(horas)\right)}{Demanda\ anual\ de\ potência\ (MW) \times horas\ do\ ano}$$

Neste caso, a demanda anual de potência deve ser considerada para cada máquina.

11.2 KPIs PARA EXCELÊNCIA OPERACIONAL

A excelência operacional de empresas do setor elétrico consiste em atender a demanda de energia dos consumidores da maneira mais econômica possível, dentro de padrões ótimos de continuidade, qualidade e segurança.

Por padrões ótimos de continuidade, entende-se maximizar a disponibilidade das máquinas, isto é, maximizar a probabilidade das máquinas estarem em condições para entrar em funcionamento quando forem solicitadas num determinado instante (Monchy, 1989). E, por padrões ótimos de qualidade e segurança, entende-se maximizar a confiabilidade das máquinas, isto é, maximizar a probabilidade das máquinas desempenharem satisfatoriamente suas funções requeridas, sob condições de operação estabelecidas, por um período de tempo pré-determinado (Freitas, 1997). Portanto, serão analisadas a seguir métricas para estes dois fatores:

1. Disponibilidade
2. Confiabilidade

11.2.1 DISPONIBILIDADE

A disponibilidade das máquinas pode ser medida por meio de registros do número de paradas forçadas por TRIP ou para manutenção corretiva emergencial e do número de paradas programadas para manutenção preventiva. Vale ressaltar que as paradas estratégicas, devido à baixa demanda de carga, não entram neste cálculo, ou seja, tais paradas não são efeitos de indisponibilidade das máquinas.

Assim sendo, a disponibilidade mensal de cada unidade geradora, conforme definido pela gerência de operação e manutenção da geração hidráulica (COGH) da Eletronorte, pode ser dada por:

$$DISP = \frac{Hmês - Hparada}{Hmês}$$

onde,

$Hmês$ = Horas do mês

$Hparada$ = Horas de máquina parada (não incluindo paradas estratégicas devido à baixa demanda)

As horas de máquina parada, $Hparada$, correspondem à soma dos tempos medidos entre o acionamento dos relés de seqüência de parada e o acionamento do relé seqüência de partida para uma determinada máquina, ou seja, é a soma dos tempos decorridos entre cada instante de parada e cada respectivo instante de religamento da máquina.

Os relés de seqüência de parada são:

- 5HX1: Seqüência de parada normal por defeito hidráulico;
- 5EX1: Seqüência de parada normal por defeito elétrico;
- 5MX1: Seqüência de parada normal por defeito mecânico;
- 94PX1: Seqüência de parada parcial voluntária.

E o relé que sinaliza seqüência de partida da máquina é 94SDX4.

Deve-se ter atenção especial para as paradas parciais voluntárias (relé 94PX1), pois há necessidade de se distinguirem as paradas voluntárias para manutenção corretiva ou preventiva (conforme descrito nas ordens de serviço) e paradas estratégicas por baixa demanda de carga, uma vez que estas últimas devem ser desconsideradas no cálculo.

11.2.2 CONFIABILIDADE

A teoria de confiabilidade lida, trata e modela fontes de incerteza no processo, tais como o tempo de ocorrência de falhas, o tempo de reparo de falhas, entrada em serviço de novas obras, a freqüência de ocorrência de eventos de falha, etc. Para tanto requer a utilização de técnicas probabilísticas.

De acordo com Borges (2006), os requisitos mínimos para se realizar um estudo de confiabilidade são:

- Modelo matemático apropriado para o problema em questão;
- Índices de risco apropriados que meçam de forma apropriada a adequação;
- Dados estatísticos de falha e operação dos componentes e sistemas para possibilitar a estimação da confiabilidade preditiva.

11.2.2.1 MODELO MATEMÁTICO DO SISTEMA

O comportamento dinâmico dos componentes das unidades geradoras pode ser modelado por meio de um processo estocástico denominado Processo Não-Homogêneo de Poisson (NHPP). A motivação para se utilizar esse processo é que os equipamentos em questão são submetidos a um ciclo de falhas e reparos, ou seja, caracterizam sistemas reparáveis. Dessa maneira, a análise será realizada considerando a eficiência da manutenção para reparos mínimos, após os quais o equipamento retorna a uma condição imediatamente anterior à falha.

O processo não-homogêneo de Poisson é definido por:

$$P(n) = \frac{[\Lambda(t)]^n}{n!}.e^{-\Lambda(t)}$$

onde,

P(n) = Probabilidade de ocorrência de *n* falhas no tempo *t*

$\Lambda(t)$ = Número esperado de falhas até o tempo *t*

A probabilidade de não ocorrer falhas no tempo *t*, chamada de Confiabilidade do Componente *R(t)*, é dada por:

$$P(0) = R(t) = e^{-\Lambda(t)}$$

O número de esperado de falhas, Λ(t), descreve a quantidade de falhas esperada cumulativa até o tempo t. Seja, u(t) a função de intensidade de falhas, Λ(t) satisfaz:

$$\Lambda(t) = \int_0^t u(s)ds$$

De acordo com o modelo Crow/AMSAA (1979) de análise do crescimento da confiabilidade para componentes reparáveis, a função u(t) se comporta como uma distribuição Weibull conforme mostrado a seguir:

$$u(t) = \lambda.\beta.t^{\beta-1}$$

E, portanto,

$$\Lambda(t) = \int_0^t u(s)ds = \lambda.t^{\beta}$$

Esta função tem dois parâmetros, λ e β. λ mede a escala ou magnitude da taxa de falhas e β mede a tendência da taxa de falhas acumulada sobre as horas de operação do produto. Com relação ao fator β, temos que:

- Se β =1, a intensidade de falhas dos equipamentos é constante. Pode ser uma indicação que modos de falhas múltiplas estão presentes. Uma taxa de falhas constante pode indicar que as falhas são provocadas por agentes externos, tais como: uso inadequado do equipamento ou técnicas inadequadas de manutenção.

- Se β >1, a intensidade de falhas dos equipamentos está aumentando com o tempo. Neste caso, é caracterizado modo de falhas por desgaste.

- E se β <1 a intensidade de falha está diminuindo. Isto pode ser causado pela eficácia das ações corretivas ou pelo fenômeno da "mortalidade infantil" de determinados equipamentos.

Substituindo a função de intensidade de falhas Λ(t) na função de confiabilidade R(t), tem-se que a confiabilidade de cada componente das unidades geradoras pode ser descrita por:

$$R(t) = e^{-\lambda.t^{\beta}}$$

Para se calcular a confiabilidade dos componentes resta então estimar os parâmetros λ e β em função do número de falhas observado em cada componente e do instante em que ocorreu cada falha. O método de estimação dos parâmetros que será aqui adotado é denominado método de máxima verossimilhança. A equação de estimação dos parâmetros é mostrada a seguir:

$$\hat{\beta} = \frac{n}{\sum_{i=1}^{n} \ln\left(\frac{T}{Ti}\right)}$$

$$\hat{\lambda} = \frac{n}{T^{\beta}}$$

Sendo que,

n = número de falhas observado no período *T*.

Ti = instante em que ocorreu a falha *i*, em horas.

A confiabilidade dos componentes vem sendo calculada de maneira mais simples pela Eletronorte, através do INFO_OPR, sob a forma de taxa de falha, conforme será mostrado na seção 4.

A partir da confiabilidade dos componentes de cada unidade geradora, pode-se calcular a confiabilidade geral das unidades geradoras conforme apresentado na seção a seguir.

11.2.2.2 ÍNDICES DE RISCO

Os índices de risco utilizados para cada componente são produtos da análise FMEA que foi desenvolvida. Trata-se do número de prioridade de risco (NPR), valor calculado pelo produto dos índices Severidade x Ocorrência x Detecção. É utilizado para a priorização da tomada de ação. É uma maneira prática de priorizar certas falhas e avaliar quais providências devem ser tomadas primeiramente.

A confiabilidade de uma unidade geradora (CONF) é definida então como sendo a média ponderada das confiabilidades de cada um dos componentes, sendo que o fator de ponderação é o número de prioridade de risco associado a cada componente. Em forma algébrica, tem-se:

$$CONF = \frac{\sum_{i=1}^{n} NPR_i . R_i(T)}{\sum_{i=1}^{n} NPR_i}$$

A medição de confiabilidade, assim como o FMEA, deve ser atualizada anualmente, e o número de falhas n e o período de tempo de tempo T, dado em horas, considerado para estimar os parâmetros $\hat{\beta}$ e $\hat{\lambda}$ devem ser acumulados a cada ocorrência de falhas, enquanto o FMEA deve corresponder aos últimos 3 anos de operação. Deste modo, o fator CONF representará uma média móvel de confiabilidade referente aos últimos 3 anos, e deverá ser calculado a cada ano para cada máquina.

Para os componentes que não apresentarem falhas no período considerado, o valor de confiabilidade deve ser fornecido pelo fabricante, ou estimados por meio da experiência dos operadores ou através de ensaios e testes de vida acelerados.

11.3 KPIs PARA EXCELÊNCIA DE SEGURANÇA

Os KPIs para excelência de segurança são métricas que tratam da prevenção de acidentes de trabalho. A palavra “acidente” leva ao entendimento de um evento que ocorre sem que possa ser evitado. Na verdade, somente muito raramente isto acontece; na maioria das vezes, os “acidentes” decorrem de falhas facilmente perceptíveis e que podem, se lhes for dada a devida atenção e providências tomadas, evitar que o “acidente” venha a ocorrer (Bernini, 2006).

As métricas de documentação de acidentes são altamente estocásticas e impossíveis de serem automatizadas, ou seja, devem ser registradas e calculadas manualmente. Tais métricas incluem:

1. Taxa de freqüência de acidentes
2. Taxa de severidade de acidentes

11.3.1 FREQÜÊNCIA

O índice de freqüência de acidentes (FACID) é uma medida padrão internacional que reflete o número de acidentes em relação ao número de horas trabalhadas em um ano. Este índice, que no caso da Manaus Energia, pode incluir trabalhadores diretos e indiretos, é medido a partir de uma fórmula obtida multiplicando-se a quantidade de acidentes por 200 mil, e dividindo-os pelo número de horas trabalhadas (Cibié, 2005).

$$FACID = \frac{N°acidentes \times 200.000}{horas\ trabalhadas}$$

11.3.2 SEVERIDADE

O índice de severidade de acidentes (SACID) reflete o número de dias perdidos devido a acidentes de trabalho, é também é uma medida padrão internacional. Ele é medido a partir da seguinte fórmula: o Índice de Severidade de Acidentes de Trabalho é igual à quantidade de dias perdidos devido a acidentes de trabalho multiplicados por 200 mil e divididos pela quantidade de horas trabalhadas (Cibié, 2005).

$$SACID = \frac{dias\ perdidos \times 200.000}{horas\ trabalhadas}$$

11.4 KPIs PARA AVALIAÇÃO DE CUSTOS

Os aspectos de baixos custos da energia elétrica e o alto nível de confiabilidade estão freqüentemente em conflito direto. O estabelecimento do custo da confiabilidade é uma tarefa subjetiva e difícil, uma avaliação direta não é suficiente. Uma alternativa utilizada é avaliar os impactos financeiros incorridos devido a interrupções no suprimento de energia elétrica. Para tanto, é necessário conhecer o custo da energia produzida.

O preço do custo da energia produzida obtém-se pela divisão da soma de todas as despesas anuais, calculadas considerando um prazo de amortização, pela produção também anual de energia. Desta forma, será obtido o preço do custo do MWh produzido na saída dos bornes do transformador elevador da usina. Na comercialização desta energia, a este preço do MWh deverá ser acrescida a tarifa de transporte da energia até o consumidor final, além das perdas de energia no sistema de transmissão. De acordo com o critério estabelecido pelo Mercado Atacadista de Energia (MAE), estes acréscimos devem ser rateados de maneira proporcional para cada categoria, ou seja, 50% para a geração e 50% para o consumo, conforme previsto pela ANEEL, contrato CUST N° 37/1999.

Enfim, a receita prevista para a geração de energia elétrica é descrita por:

$$receita = valor\ do\ MWh\ vendido\ ao\ MAE - custo\ \ final\ do\ MWh$$

sendo que o custo final do MWh é:

$$custo\ \ final\ do\ MWh = despesas\ anuais + 50\%.tarifa\ \ de\ \ tranporte\ \ de\ \ energia$$
$$+ 50\%.perdas\ na\ transmissão$$

A perda de receita anual em função da indisponibilidade das máquinas (PREC) pode ser quantificada percentualmente, por máquina, conforme a equação a seguir:

$$PREC = receita \times ENS$$

onde,
ENS = índice de energia não suprida por máquina, conforme definido anteriormente.

Desta forma, esta perda de receita passa a ser considerada com um índice de sensibilidade para a seleção de melhores investimentos em geração, isto é, um índice de atratividade capaz de fornecer respaldo à avaliação de retorno econômico dos investimentos aplicados.

12. INDICADORES DE DESEMPENHO DEFINIDOS PELO SISTEMA INFO_OPR

O INFO_OPR (informativo operacional) é um sistema de gerenciamento de informações utilizado pela Eletronorte em seus centros de operação. Trata-se de um programa desenvolvido no COT (Centro de Informação e Análise da Transmissão) e está em funcionamento há nove anos, proporcionando uma redução de tempo considerável na transferência de informações entre servidores, uma vez que anteriormente estas informações chegavam através de arquivos em planilhas diversas e via fax, ocasionando re-digitação de entrada de dados e perda de informação.

Algumas características do programa:

1. Recebe e armazena dados de interrupção de energia, carga e hidrológicos das regionais de transmissão e sistemas isolados da Eletronorte;
2. Apresenta relatórios diários da Operação (RDO) e relatórios diários de Interrupção (RDI);
3. Executa cálculo dos índices de desempenho dos sistemas e das instalações;
4. Executa cálculo da parcela variável;
5. Fornece gráficos padrões e personalizados diários, mensais e anuais, permitindo comparação com valores de referência;
6. Disponibiliza diagramas unifilares dos sistemas elétricos, com recursos de "zoom" e impressão;
7. Permite a transferência de informações pelo correio eletrônico;
8. Permite plena exportação de todos os dados para planilhas Excel, definidas pelo usuário, que poderá utilizá-las para gráficos e usos específicos;
9. Disponibiliza cadastros de equipamentos.
10. Distribui documentos de operação.

O sistema tem uma base de dados centralizada no COT, em Brasília, com informações operacionais de todas as unidades Regionais do sistema de transmissão interligado e dos sistemas isolados.

Os indicadores de desempenho do INFO_OPR foram definidos de acordo com as especificações da ONS – Operador Nacional do Sistema Elétrico, Manual de Procedimentos de

Rede, Módulo 16 – Acompanhamento da Manutenção, Sub-módulo 2.7 – Indicadores de Desempenho para Acompanhamento da Manutenção de 23 de julho de 2001.

São estes:

- Disponibilidade [DISP];
- Taxa de Desligamento Forçado [TDF];
- Tempo Médio de Reparo da Função [TMRF];
- Indisponibilidade para manutenção programada [INDISPMP];
- Indisponibilidade para manutenção forçada [INDISPMF];
- Taxa de Falha [TF].

A forma de cálculo de cada um destes indicadores é mostrada adiante (fonte: Silva Filho, 2002).

12.1 FORMULAÇÃO DOS ÍNDICES CALCULADOS

A seguir são apresentados a formulação dos índices sugeridos para implementação mostrando como calculá-los.

12.1.1 DISP – DISPONIBILIDADE DE EQUIPAMENTOS

$$DISP = \frac{\frac{EXT.LT}{100} \times HD}{\frac{EXT.LT}{100} \times HP} \times 100 \qquad \text{(Para Linhas)}$$

$$DISP = \frac{POT \times HD}{POT \times HP} \times 100 \qquad \text{(Para Geradores)}$$

$$DISP = \frac{HD}{HP} \times 100 \qquad \text{(Outros)}$$

Terminologia:

POT: Potência ativa do gerador em MW.

EXT.LT: Extensão da Linha em Quilômetros.

HP: Horas do Período (total de horas do período considerado).

HD = HP – HI: Horas Disponíveis (Somatório dos tempos, em horas, que o equipamento ou instalação está apto a operar com ou sem restrições).

HI: Horas Indisponíveis, com Classes de Rede de Operação iguais à:

- AIP Desl. por atuação indevida proteção
- DES Desl. causa externa sem retorno à operação.
- DIS Desl. causa interna sem retorno operação
- DPM Desl. prog. p'melhorias e modif. Rede Básica
- DSO Desl. Iniciado e suspenso pelo ONS/COR
- DST Desl. solic. ONS/Emp. por motivo segur. terceiros
- DVA Desl. por Vandalismo
- EME Desl. emerg. p/evitar riscos de vida humana
- EMR Desl. emerg. p/evitar riscos de danific. equipam.
- FLH Desli. falha humana acidental
- PMC Desl. Programado manobras operacionais (Corretivo)
- PMV Desl. Programado manobras operacionais (Preventivo)
- PRA Desl. Prog. Excedente ao previsto
- PRC Desl. Programado corretivo
- PRV Desl. Programado preventivo
- UMO Desli. Urgência Manobras Oper. não aceito como programado
- UMP Desli. Urgência Manobras Oper. aceito como programado
- URG Desl. Solic. pela Empresa ao ONS
- URP Desli. Urgência aceito como programado

Agregação Temporal : Mensal

Adimensional

Agregação :

DISP-ELN	Eletronorte
DISP-SIS	Sistema
DISP-CA	Capacitor (Por faixa de tensão)
DISP-CE	Compensador Estático
DISP-CS	Compensador Síncrono
DISP-G	Unidades Geradoras
DISP-L	Linha (Por faixa de tensão)
DISP-RE	Reator (Por faixa de tensão)
DISP-T	Transformador (Por faixa de tensão)

Por Equipamento

Fonte: ONS - Operador nacional do Sistema Elétrico Procedimentos de Rede Módulo 16 - Acompanhamento da Manutenção Sub-módulo 2.7 - Indicadores de Desempenho para Acompanhamento da Manutenção 23 de julho de 2001.

12.1.2 DREQ E FREQ – DURAÇÃO E FREQÜÊNCIA EQUIVALENTE

$$DREQ = \frac{Duração \times Potência\ Interrompida}{Demanda\ Máxima}$$

DREQ corresponde ao tempo de energia não suprida com relação à demanda máxima

$$FREQ = \frac{Potência\ Interrompida}{Demanda\ Máxima}$$

FREQ corresponde ao percentual da demanda máxima que foi interrompido.

Terminologia:

Potência Interrompida (MW)

Demanda Máxima(MWh/h)

Duração: Duração em horas das interrupções com Classes de Rede de Operação iguais à:

- AIP Desl. por atuação indevida proteção
- DAI Desl. ocasionados por atuação indevida ONS
- DDO Desl. Causa Externa, porém disponível à operação
- DES Desl. causa externa sem retorno à operação.
- DFM Desl. força maior
- DIS Desl. causa interna sem retorno operação
- DPM Desl. prog. p'melhorias e modif. Rede Básica
- DRO Desl. por razão ou conveniência operativa
- DSO Desl. Iniciado e suspenso pelo ONS/COR
- DST Desl. solic. ONS/Emp. por motivo segur. Terceiros
- DVA Desl. por Vandalismo
- EME Desl. emerg. p/evitar riscos de vida humana
- EMR Desl. emerg. p/evitar riscos de danific. equipam.
- FLH Desli. falha humana acidental
- PMC Desl. Programado manobras operacionais (Corretivo)
- PMV Desl. Programado manobras operacionais (Preventivo)
- PRA Desl. Prog. Excedente ao previsto
- PRC Desl. Programado corretivo
- PRV Desl. Programado preventivo
- UMO Desli. Urgência Manobras Oper. não aceito como programado
- UMP Desli. Urgência Manobras Oper. aceito como programado
- URG Desl. Solic. pela Empresa ao NOS
- URP Desli. Urgência aceito como programado

Agregação Temporal : Mensal

Unidade Dimensional para DREQ: horas

Unidade Dimensional para FREQ: adimensional

Agregação :
Por Sistema
Por Ponto de Suprimento

Fonte : ONS - Operador nacional do Sistema Elétrico Manual de Procedimentos da Operação Volume III - Normas e Metodologias da Operação, Tomo II, Janeiro de 1999.

12.1.3 DST – DESEMPENHO DO SISTEMA DE TRANSMISSÃO

$$DST = \frac{\text{Receita Permitida no mês} - \text{PV no mês}}{\text{Receita Permitida no mês}} \times 100$$

Terminologia:

PV: Parcela variável (será definida posteriormente)

Agregação Temporal : Mensal

Adimensional

Agregação :
DST-ELN Eletronorte
DST-SIS Sistema

Fonte: COTC – Divisão de Contratos de Operação

12.1.4 INDISPMF – INDISPONIBILIDADE PARA MANUTENÇÃO FORÇADA DE EQUIPAMENTOS

$$INDISPMF = \frac{\frac{EXT.LT}{100} \times HIMF}{\frac{EXT.LT}{100} \times HP} \times 100 \qquad \text{(Para Linhas)}$$

$$INDISPMF = \frac{POT \times HIMF}{POT \times HP} \times 100 \qquad \text{(Para Geradores)}$$

$$INDISPMF = \frac{HIMF}{HP} \times 100 \qquad \text{(Outros)}$$

Terminologia:

HIMF Horas Indisponíveis devido a Manutenção Forçada, com Classes de Rede de Operação iguais à:

- AIP Desl. por atuação indevida proteção
- DES Desl. causa externa sem retorno à operação.
- DIS Desl. causa interna sem retorno operação
- DVA Desl. por Vandalismo
- FLH Desli. falha humana acidental
- UMO Desli. Urgência Manobras Oper. não aceito como programado
- URG Desl. Solic. pela Empresa ao ONS

Agregação Temporal : Mensal

Adimensional

Agregação :

INDISPMF-ELN	Eletronorte
INDISPMF-SIS	Sistema
INDISPMF-CA	Capacitor (Por faixa de tensão)
INDISPMF-CE	Compensador Estático
INDISPMF-CS	Compensador Síncrono
INDISPMF-G	Unidades Geradoras
INDISPMF-L	Linha (Por faixa de tensão)
INDISPMF-RE	Reator (Por faixa de tensão)
INDISPMF-T	Transformador (Por faixa de tensão)

Por Equipamento

Fonte: ONS - Operador nacional do Sistema Elétrico Procedimentos de Rede Módulo 16 - Acompanhamento da Manutenção Sub-módulo 2.7 - Indicadores de Desempenho para Acompanhamento da Manutenção 23 de julho de 2001.

12.1.5 INDISPMP – INDISPONIBILIDADE PARA MANUTENÇÃO PROGRAMADA DE EQUIPAMENTOS

$$INDISPMP = \frac{\frac{EXT.LT}{100} \times HIMP}{\frac{EXT.LT}{100} \times HP} \times 100 \qquad \text{(Para Linhas)}$$

$$INDISPMP = \frac{POT \times HIMP}{POT \times HP} \times 100 \qquad \text{(Para Geradores)}$$

$$INDISPMP = \frac{HIMP}{HP} \times 100 \qquad \text{(Outros)}$$

Terminologia:

HIMP: Horas Indisponíveis devido a Manutenção Programada, com Classes de Rede de Operação iguais à :

- DPM Desl. prog. p'melhorias e modif. Rede Básica
- DSO Desl. iniciado e suspenso pelo ONS/COR
- PMC Desl. Programado manobras operacionais (Corretivo)
- PMV Desl. Programado manobras operacionais (Preventivo)
- PRA Desl. Prog. Excedente ao previsto
- PRC Desl. Programado corretivo
- PRV Desl. Programado preventivo
- UMP Desl. Urgência Manobras Oper. aceito como programado

- URP Desl. Urgência aceito como programado

Agregação Temporal: Mensal

Adimensional

Agregação :

INDISPMP-ELN	Eletronorte	
INDISPMP-SIS	Sistema	
INDISPMP-CA	Capacitor (Por faixa de tensão)	
INDISPMP-CE	Compensador Estático	
INDISPMP-CS	Compensador Síncrono	
INDISPMP-G	Unidades Geradoras	
INDISPMP-L	Linha	(Por faixa de tensão)
INDISPMP-RE	Reator	(Por faixa de tensão)
INDISPMP-T	Transformador	(Por faixa de tensão)

Por Equipamento

Fonte: ONS - Operador nacional do Sistema Elétrico
Procedimentos de Rede
Módulo 16 - Acompanhamento da Manutenção
Sub-módulo 2.7 - Indicadores de Desempenho para Acompanhamento da Manutenção 23 de julho de 2001.

12.1.6 TDF – TAXA DE DESLIGAMENTO FORÇADO

$$TDF = \frac{NDF}{\frac{EXT.LT}{100} \times HS} \times 8760 \qquad \text{(Para Linhas)}$$

$$TDF = \frac{NDF}{POT \times HS} \times 8760 \qquad \text{(Para Geradores)}$$

$$TDF = \frac{NDF}{HS} \times 8760 \qquad \text{(Outros)}$$

Terminologia:

NDF: Número de Desligamentos Forçados, com Classes de Rede de Operação iguais à:

- AIP Desl. por atuação indevida proteção
- DES Desl. causa externa sem retorno à operação.
- DIS Desl. causa interna sem retorno operação
- DVA Desl. por Vandalismo
- EMR Desl. emerg. p/evitar riscos de danific. equipam.
- UMO Desli. Urgência Manobras Oper. não aceito como programado

- URG Desl. Solic. pela Empresa ao ONS

HP: Horas do Período

HS = HP - HFS: Horas de Serviço

HFS Horas Fora de Serviço com Classes de Rede Operação iguais à:

- AIP Desl. por atuação indevida proteção
- DAI Desl. ocasionados por atuação indevida NOS
- DDO Desl. Causa Externa, porém disponível à operação
- DES Desl. causa externa sem retorno à operação.
- DFM Desl. força maior
- DFP Desl.Falha Equip. por não aprovação de Prog.
- DIR Desl.causa interna. Duração <= que 1 minuto.
- DIS Desl. causa interna sem retorno operação
- DPM Desl. prog. p'melhorias e modif. Rede Básica
- DRO Desl. por razão ou conveniência operativa
- DSO Desl. Iniciado e suspenso pelo ONS/COR
- DST Desl. solic. ONS/Emp. por motivo segur. Terceiros
- DVA Desl. por Vandalismo
- EME Desl. emerg. p/evitar riscos de vida humana
- EMR Desl. emerg. p/evitar riscos de danific. equipam.
- FLH Desli. falha humana acidental
- LIO equip. Liberado para Energização
- NAP Não atuação da proteção
- PMC Desl. Programado manobras operacionais (Corretivo)
- PMV Desl. Programado manobras operacionais (Preventivo)
- PRA Desl. Prog. Excedente ao previsto
- PRC Desl. Programado corretivo
- PRV Desl. Programado preventivo
- UMO Desl. Urgência Manobras Oper. não aceito como programado
- UMP Desl. Urgência Manobras Oper. aceito como programado
- URG Desl. Solic. pela Empresa ao NOS
- URP Desl. Urgência aceito como programado

Agregação Temporal: Anual

Unidade Dimensional: Número de Desligamentos Forçados / Ano

Agregação:
TDF-ELN Eletronorte
TDF-SIS Sistema
TDF-CA Capacitor (Por faixa de tensão)
TDF-CE Compensador Estático
TDF-CS Compensador Síncrono
TDF-G Unidades Geradoras
TDF-L Linha (Por faixa de tensão)
TDF-RE Reator (Por faixa de tensão)
TDF-T Transformador (Por faixa de tensão)
Por Equipamento

Fonte: ONS - Operador nacional do Sistema Elétrico Procedimentos de Rede Módulo 16 - Acompanhamento da Manutenção Sub-módulo 2.7 - Indicadores de Desempenho para Acompanhamento da Manutenção 23 de julho de 2001.

12.1.7 TF – TAXA DE FALHAS

$$TF = \frac{NF\ Anual}{\frac{EXT.LT}{100} \times HS} \times 8760 \qquad \text{(Para Linhas)}$$

$$TF = \frac{NF\ Anual}{POT \times HS} \times 8760 \qquad \text{(Para Geradores)}$$

$$TF = \frac{NF\ Anual}{HS} \times 8760 \qquad \text{(Para Equipamentos)}$$

Terminologia:

NF Anual Número de Falhas no Ano com Classe Rede Operação iguais à:

- AIP Desl. por atuação indevida proteção
- DES Desl. causa externa sem retorno à operação.
- DFP Desl. falha equipamento
- DIS Desl. causa interna sem retorno operação
- DVA Desl. por Vandalismo
- EMR Desl. emerg. p/evitar riscos de danific. equipam.
- UMO Desl. Urgência Manobras Oper. não aceito como programado
- UMP Desl. Urgência Manobras Oper. aceito como programado
- URG Desl. Solic. pela Empresa ao ONS
- URP Desl. Urgência aceito como programado

Agregação Temporal : Anual

Unidade Dimensional : Número de Falhas / Ano

Agregação :

TF-ELN	Eletronorte
TF-SIS	Sistema
TF-CA	Capacitor (Por faixa de tensão)
TF-CE	Compensador Estático
TF-CS	Compensador Síncrono
TF-G	Unidades Geradoras
TF-L	Linha (Por faixa de tensão)
TF-RE	Reator (Por faixa de tensão)

TF-T Transformador (Por faixa de tensão)
Por Equipamento

Fonte: ONS - Operador nacional do Sistema Elétrico Procedimentos de Rede Módulo 16 - Acompanhamento da Manutenção Sub-módulo 2.7 - Indicadores de Desempenho para Acompanhamento da Manutenção 23 de julho de 2001.

12.1.8 TMRF – TEMPO MÉDIO DE REPARO DA FUNÇÃO

$$TMRF = \frac{HIR}{NF}$$

Terminologia:

NF Número de Falhas

HIR Horas Indisponíveis para Operação e entregue para a Manutenção

Ambos com Classes de Rede de Operação iguais à:

- AIP Desl. por atuação indevida proteção
- DES Desl. causa externa sem retorno à operação.
- DIS Desl. causa interna sem retorno operação
- DVA Desl. por Vandalismo
- EMR Desl. emerg. p/evitar riscos de danific. equipam.
- UMO Desli. Urgência Manobras Oper. não aceito como programado
- URG Desl. Solic. pela Empresa ao CNOS (Centro Nacional de Operação de Sistemas)

Agregação Temporal : Mensal

Unidade Dimensional : hora

Agregação:
TMRF-ELN Eletronorte
TMRF-SIS Sistema
TMRF-CA Capacitor (Por faixa de tensão)
TMRF-CE Compensador Estático
TMRF-CS Compensador Síncrono
TMRF-G Unidades Geradoras
TMRF-L Linha (Por faixa de tensão)
TMRF-RE Reator (Por faixa de tensão)
TMRF-T Transformador (Por faixa de tensão)
Por Equipamento

Fonte: ONS - Operador nacional do Sistema Elétrico Procedimentos de Rede Módulo 16 - Acompanhamento da Manutenção Sub-módulo 2.7 - Indicadores de Desempenho para Acompanhamento da Manutenção 23 de julho de 2001.

12.1.9 PV – PARCELA VARIÁVEL

Fonte : Manual de Procedimento de Operação, Módulo 2.4, MTD 031, ONS, de 01/03/1999.

- AEN Atraso entrada operação melhorias na Rede Básica
- AIP Desl. por atuação indevida proteção
- DES Desl. causa externa sem retorno à operação.
- DIS Desl. causa interna sem retorno operação.
- DVA Desl. por Vandalismo.
- EMR Desl. emerg. p/evitar riscos de danific. equipam.
- FLH Desl. falha humana acidental
- PMC Desl. Programado manobras operacionais (Corretivo)
- PMV Desl. Programado manobras operacionais (Preventivo)
- PRA Desl. Prog. Excedente ao previsto
- PRC Desl. Programado corretivo
- PRV Desl. Programado preventivo
- REO Em operação c/restrição operativa não progr. (CPST)
- REP Em operação c/restrição operativa programada (CPST)
- SRO Em operação c/sobrecarga não progr. (CPST)
- SRP Em operação c/sobrecarga programada (CPST)
- UMO Desl. Urgência Manobras Oper. não aceito como programado
- UMP Desl. Urgência Manobras Oper. aceito como programado
- URG Desl. Solic. pela Empresa ao ONS
- URP Desli. Urgência aceito como programado

13. MÉTRICAS DE DESEMPENHO (KPIs) SUGERIDAS

Sugere-se, então, que sejam adotadas as seguintes métricas de desempenho (KPIs) classificadas conforme o enfoque:

- KPIs para excelência de serviço:

  $KPI_1 = DINT$ (direcionador de continuidade)

  $KPI_2 = SM$ (direcionador de severidade das interrupções)

  $KPI_3 = FINT$ (direcionador de qualidade)

  $KPI_4 = ENS$ (direcionador de energia não suprida)

- KPIs para excelência operacional:

  $KPI_5 = DISP$ (direcionador de disponibilidade)

$KPI_6 = CONF$ (direcionador de confiabilidade, pode ser calculado estatísticamente, como demonstrado na subseção 3.2.2, ou através dos tempos médios entre falhas, como é feito no INFO_OPR)

- KPIs para excelência de segurança:
  $KPI_7 = FACID$ (direcionador de freqüência de acidentes de trabalho)
  $KPI_8 = SACID$ (direcionador de severidade de acidentes de trabalho)

- KPIs para avaliação de custos:
  $KPI_9 = PREC$ (direcionador de perda de receita em função da indisponibilidade das máquinas)

A tabela a seguir mostra os requisitos necessários para a medição de cada um destes KPIs:

KPI	REQUISITOS PARA MEDIÇÃO E CÁLCULO
KPI_1	**Armazenar mensalmente uma base de dados contendo os instantes de tempo de parada por TRIP e religamento das máquinas.**
KPI_2	**Registrar no sistema de execução de manufatura da usina MES, a cada TRIP, a carga interrompida.** **Verificar, pela monitoração on line, a demanda máxima do dia da interrupção.**
KPI_3	**Acumular mensalmente o número de ocorrências de TRIP.**
KPI_4	**Verificar anualmente nos registros a carga interrompida e a duração de cada interrupção.** **Verificar a demanda anual.**
KPI_5	**Armazenar mensalmente uma base de dados contendo os instantes de tempo de parada e religamento das máquinas, bem como o motivo de cada parada (elétrica, mecânica, hidráulica, voluntária para manutenção ou voluntária estratégica).**
KPI_6	**Armazenar anualmente o número de falhas de cada componente e o instante de tempo em que cada falha foi detectada.** **Manter anualmente atualizado o FMEA dos componentes.**
KPI_7	**Registrar anualmente o número de acidentes ocorridos**
KPI_8	**Registrar anualmente o número de dias perdidos devido a acidentes de trabalho**
KPI_9	**Calcular minuciosamente todas as despesas anuais.** **Calcular as tarifas de transporte de energia** **Calcular as perdas de energia na linha de transmissão** **Registrar o valor do MWh comercializado**

Vale ressaltar que devem ser estabelecidas ***metas para cada KPI as quais representarão o objetivo estratégico da usina***. *Desta forma os KPIs apresentados se tornarão importantes padrões de medição os quais auxiliarão as tomadas de decisões gerenciais na UHE Balbina.*

O SIMPREBAL faz o cálculo de alguns KPIs associados a falhas em equipamentos que estão sendo monitorados, onde os dados para cálculo destes KPIs são armazenados no banco de dados do SIMPREBAL. Para a implementação de outros KPIs é necessário obter dados do sistema de operação com relação a desligamentos, tempos de parada de máquinas e equipamento, religamentos, entre outros. Sem os dados de operação não é possível realizar estes cálculos. Assim é necesário fazer a integração ou incluir os dados através de formulários que irão alimentar o banco de dados SIMPREBAL.

14. PLANO DE AÇÕES PARA IMPLANTAÇÃO DE KPIs EM BALBINA

Medir e utilizar corretamente os KPIs, associados a objetivos específicos aliados aos objetivos corporativos globais da Manaus Energia, pode melhorar a confiabilidade e a efetividade dos processos de trabalho e das tomadas de decisões estratégicas na Usina Hidrelétrica de Balbina, aproveitando assim as oportunidades oferecidas pela nova tecnologia adotada no sistema de Automação de Balbina, que certamente é um grande diferencial no Setor Elétrico Brasileiro.

As métricas fazem parte da vida de qualquer pessoa. Nas plantas indústrias do setor elétrico, petroquímico, metal-mecânico, entre outros, muitas medições ou indicadores-chave de desempenho (KPIs) podem ser capturados e registrados habitualmente. Tais medições podem tratar assuntos como segurança (número de acidentes), meio ambiente (número de iniciativas de preservação), custos (% de orçamento gasto com manutenção), produção (% de produção efetivada versus produção planejada).

Várias empresas coletam e reportam rotineiramente medidas de confiabilidade ou de desempenho na gestão de ativos. Ao coletar estes dados deve-se ter em mente três importantes perguntas:

- Por que estamos medindo?
- O que está sendo medido?
- E qual é a fonte os dados?

A resposta a estas perguntas viabiliza o desenvolvimento de uma estratégia focada no desempenho da gestão de ativos. Em Balbina será utilizado o software Maximo associado ao sistema MES da Rockwell no planejamento de manutenção (Manutenção Preventiva) que também executa atividades associadas à gestão de ativos da Usina de Balbina.

Se a meta é utilizar KPIs para identificar oportunidades de melhorias nos processos de trabalho que estão aliadas aos objetivos corporativos, então, primeiramente, os objetivos em questão devem ser compreendidos. Examine o que está sendo medido atualmente. Está de acordo com os objetivos da Manaus Energia ? Um problema comum a muitas organizações é que os processos de negócio são segmentados, e muitos departamentos ou grupos colecionam silos de informação os quais produzem métricas apenas por causa da medição, e não como um catalisador para mudar ou dar um sinal de alerta no sentido de identificar estratégias falhas ou ineficientes.

Considere a abordagem básica em malha fechada mostrada na Figura 14.1 para alinhar os KPIs às visões e estratégias corporativas. Os três elementos primários deste processo são:

- Perspectivas;
- objetivos e medidas;
- mapa estratégico.

Perspectivas são conjuntos de pontos de vista acerca de uma estratégia. Elas podem ser classificadas em quatro áreas primárias:

- empresa;
- equipamento;
- processo;
- aprendizagem.

Os objetivos específicos e as métricas de um KPI são formulados a partir de uma meta especificada. O alinhamento dos objetivos específicos aos objetivos corporativos é crítico. Cada um destes KPIs, já definidos, deve ser observável, confiável, mensurável e específico. O KPI deve ser capaz de traduzir resultados e ser contabilizado por eles, porque medições tornam-se basicamente inúteis se não puderem ser contabilizadas ou se não permitirem que alguém direcione os processos de trabalho aos objetivos não-alcançados.

O mapa estratégico é um esquema visual de todas as perspectivas (objetivos) e medidas específicas. Se cada medida não está diretamente ligada a um objetivo, e se um objetivo não está diretamente ligado a uma perspectiva, então por que se mede? Este fato indica que se está medindo simplesmente por causa da medida, e tal medição não tem nenhuma ligação ou alinhamento com as metas globais da companhia.

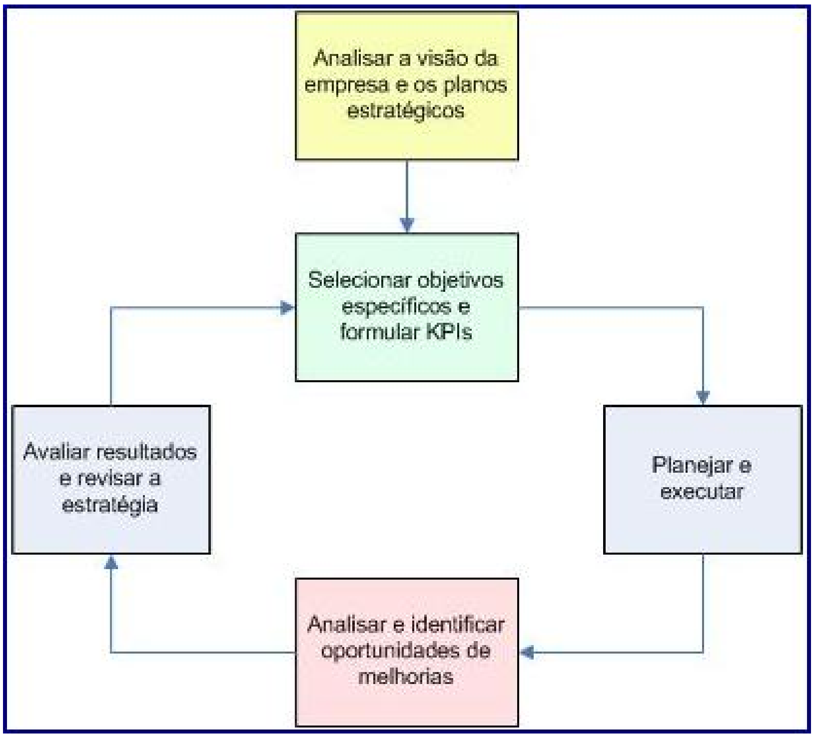


Figura 14.1 *Processo em malha-fechada para definição dos indicadores-chave de desempenho*

14.1 Perspectivas

Todas as empresas têm objetivos básicos de alto-nível que desejam alcançar. Eles normalmente estão relacionados a operar com segurança e ter responsabilidade com o meio ambiente.

Normalmente, a razão primária pela qual se abre um negócio é obter retorno do investimento empregado, ou melhor, alcançar rentabilidade. Deste modo, da perspectiva da companhia o objetivo financeiro está relacionado com obter retorno do capital investido. Os objetivos de segurança e ambientais são minimizar ou eliminar incidentes relacionados à segurança do trabalho ou ambientais. Estas são perspectivas corporativas com objetivos específicos.

Da perspectiva dos equipamentos, os objetivos normalmente estão relacionados à confiabilidade, disponibilidade, mantenabilidade, tempo, e tempo de manutenção (ou tempo de funcionamento). Da perspectiva do processo, os objetivos estão relacionados a melhorar os processos de trabalho de manutenção e operação, e manter e operar ativos. Adicionalmente, da perspectiva da aprendizagem, os objetivos estão relacionados à informação e treinamento de pessoal.

Assim, por que nós medimos? Uma primeira resposta é para alinhar os indicadores-chave de desempenho aos objetivos, os quais podem ser classificados em perspectivas corporativas, de equipamento, de processo e de aprendizagem. Como mencionado, esses itens, que possuem uma ligação direta com objetivos específicos que são derivados das quatro perspectivas, são o que se deve medir. Um mapa estratégico serve como uma excelente ferramenta para visualizar as quatro perspectivas, objetivos específicos, e as medidas ou KPIs que estão ligados aos objetivos. Veja o exemplo na Figura 2 (Página seguinte).

Note as quatro perspectivas: corporação, equipamento, processo, e aprendizagem. Os dois objetivos corporativos neste exemplo são "Melhorar a utilização dos recursos" e "minimizar as perdas nos lucros" (ou talvez melhor exposto, "maximizar a rentabilidade"). Estes são os dois objetivos corporativos de alto-nível.

Vale observar também como os objetivos estão listados em cada uma das perspectivas e como estão todos conectados (fig. 14.2).

Existem muitas medidas específicas para cada objetivo. Por exemplo, na perspectiva do equipamento, há um objetivo de "minimizar o tempo de reparo" e o KPI específico é medir o tempo médio de reparo (MTTR). Esta medida sozinha afeta o resultado de outros oito objetivos. E, no final das contas, está conectado aos dois objetivos corporativos globais de alto-nível. Um mapa estratégico pode ser uma grande ferramenta de planejamento. Ele pode responder a pergunta básica sobre o que deve ser medido para alcançar os benefícios desejados. Resumidamente, o referido mapa (fig. 14. 2):

- Enfoca os objetivos corporativos primários para assegurar sua conexão com os objetivos específicos;
- Aponta recursos para viabilizar cada objetivo e cada KPI;
- Aponta medidas próprias para cada objetivo;
- Identifica processos de trabalho que exigem melhorias para alcançar os objetivos esperados e provê uma visão global dos itens que podem exercer maior impacto sobre os objetivos corporativos.

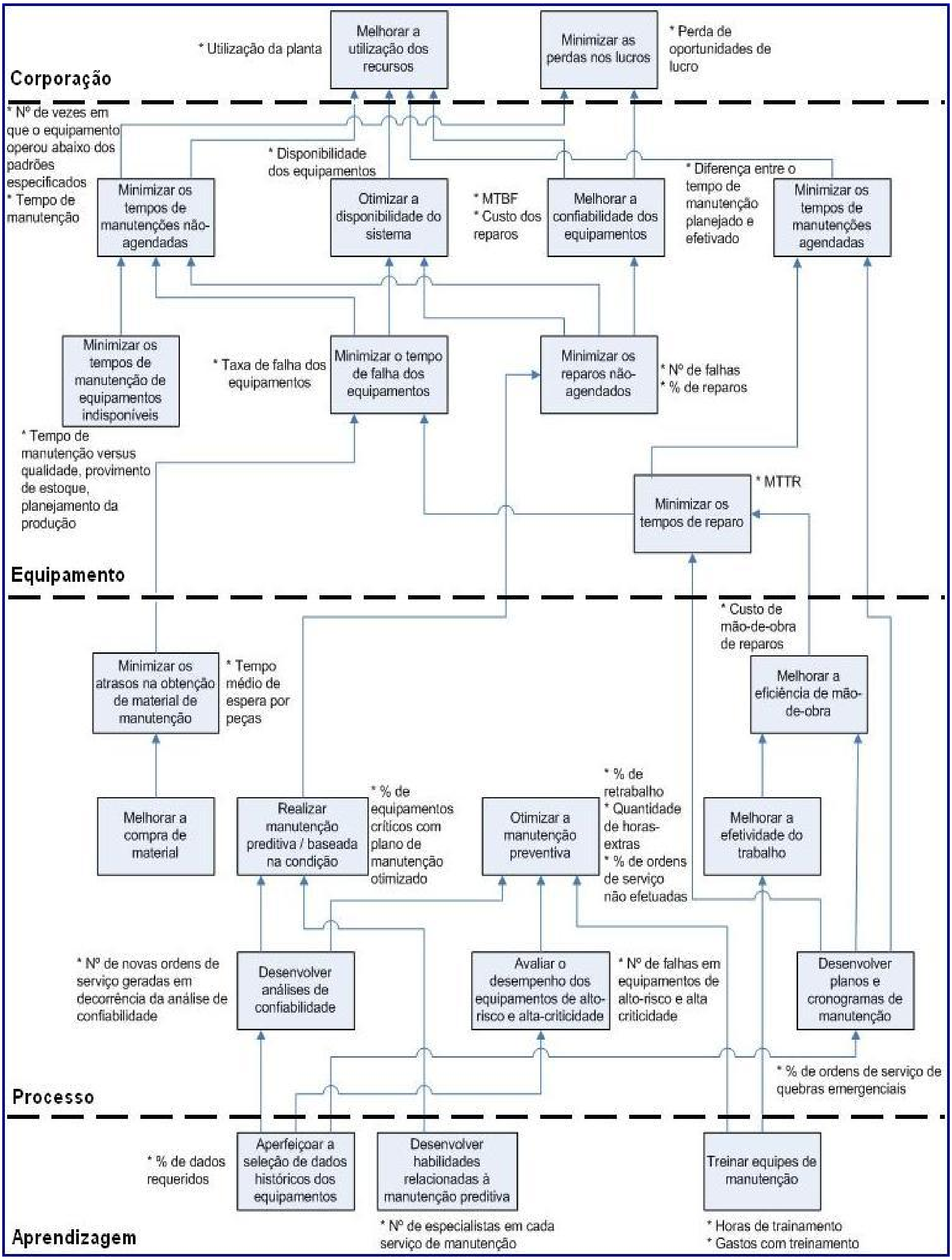


Figura 14.2 *Mapa estratégico - Quatro perspectivas: Corporação, Equipamento, Processo, Aprendizagem.*

14.2 Fonte dos dados

Como na maioria das empresas, os dados geralmente são dispostos em centenas de "silos" ao longo da instalação, sem mencionar todo o aparato: planilhas eletrônicas, escrivaninhas de arquivos e relatórios, engenharia central, sistemas de gestão da manutenção, e muitas outras soluções pontuais isoladas. E mesmo se for feito um ótimo trabalho de identificar os objetivos conectados à perspectiva corporativa, uma importante questão a se fazer é: como os dados serão razoavelmente coletados em uma base rotineira? Sistemas de gestão de desempenho de ativos podem facilitar o interfaceamento entre as várias soluções pontuais e os sistemas de gestão de manutenção. Eles podem também fornecer soluções em relação ao modo de captura e integração dos dados de processo ou operacionais, baseados em eventos, que direcionam as informações dos indicadores-chave de desempenho.

As rotinas que informam e automatizam os indicadores-chave de desempenho provêem gerenciamento, engenharia da confiabilidade, ou análises de gestão do desempenho de ativos juntamente com os dados coletados. Desta forma, é gasto menos tempo coletando e mais tempo aplicando os resultados da coleta para alcançar os benefícios já discutidos.

Sem um local central para coletar, armazenar, e registrar os dados indicadores-chave de desempenho, seria extremamente difícil executar estas tarefas. Os dados devem ser precisos, confiáveis, e oportunos para fazer a diferença no desenvolvimento de estratégias, corporativas ou departamentais, direcionadas à melhoria de desempenho dos ativos.

Assim, ter um sistema que agrega todos os dados necessários num local central é uma recomendação óbvia. Caso contrário, será gasto bastante tempo na coleta e verificação dos dados e tempo insuficiente na análise e aquisição do valor efetivo e utilização deste valor nas medições.

14.3 APM e KPIs

Algumas considerações devem ser feitas sobre como os KPIs se encaixam numa abordagem geral de gestão de desempenho de ativos. A gestão de desempenho de ativos (APM - Asset Performance Management) possui três componentes principais (fig. 14.3):

- Estratégia;
- Execução;
- Avaliação.

Se os direcionadores de negócio são os objetivos corporativos, então as estratégias devem ser desenvolvidas baseando-se nestes direcionadores. Portanto, a APM pode ser descrita como:

- A criação de estratégias que visam preservar e melhorar o desempenho dos ativos;
- A execução dessas estratégias, incluindo a geração de KPIs;
- Atribuição de critérios para iniciar uma análise minuciosa do desempenho de ativos;
- Seleção de ferramentas de apoio à decisão para avaliar o desempenho dos ativos;
- Implementação de estratégias revisadas quando a abordagem inicial ou atualmente empregada não alcançar os resultados desejados.

Indicadores-chave de desempenho são uma parte primária de qualquer iniciativa de APM. Selecionar medições apropriadas para alcançar objetivos específicos pode ser um processo trabalhoso. Alguns pontos fundamentais que devem ser levados em consideração quando se seleciona uma determinada métrica incluem:

- Qual é o impacto desta medida sobre o objetivo específico ?
- Quem, ou que grupo(s), é responsável pelo o sucesso ou ineficácia em se alcançar o objetivo desta medida ?

Figura 14.3 *Estratégia – Execução – Avaliação: utilizando KPIs em APM*

- Quais processos de trabalho devem estar em funcionamento para se alcançar o objetivo de um determinado KPI?
- Qual é a freqüência de medição necessária?
- De onde vem o dado medido? Ele é preciso?
- Como os dados serão utilizados para revisar e alterar estratégias ou processos de trabalho ineficazes?

14.4 Contabilidade

"Não se pode administrar aquilo que não se pode medir" [Tom Peters, In Search of Excellence; Warner Books, Nova Iorque, 1988].

Se você alguma vez já esteve envolvido numa empresa, companhia ou iniciativa local, o que se deseja sempre saber é por onde começar. Nas iniciativas de gestão da confiabilidade ou gestão do desempenho de ativos, o termo "padronização" parece sempre estar presente.

Qual é o status atual dos seus programas?
Em que estamos gastando muito dinheiro?
Que ativos estão falhando?
Com que freqüência eles falham?
Onde estão os dados?
E a mais importante questão: Quão precisos são os dados?

Medições são necessárias para administrar suas estratégias, executar suas ações, e avaliar seu desempenho. Se você não realiza medições ativamente, então existem chances dos seus esforços, ou suas falhas, não serem contabilizados.

Contabilidade é um ingrediente chave para se alcançar sucesso. As companhias que não mantém funcionários que contabilizam, provavelmente, não alcançam o sucesso desejado. Os indicadores-chave de desempenho direcionam o comportamento e são catalisadores para o sucesso. A contabilidade normalmente conduz desempenhos favoráveis.

14.5 Estudo de Caso Como Referência Para Aplicar em Balbina

Uma indústria petroquímica com aproximadamente 50.000 ativos possuía um programa de manutenção preventiva e preditiva bastante rígido. Um objetivo corporativo primário da companhia era alcançar um lucro líquido de, no mínimo 12%, mas a empresa amargou a marca de somente 8% nos últimos anos. Falhas evitáveis e perdas na produção de alguns itens foram ainda prevalecentes.

A companhia começou então a focar seus esforços na confiabilidade para que fosse possível alcançar o lucro esperado. Um programa de manutenção preventiva foi instituído de acordo com a localização funcional dos equipamentos e agendado no sistema de gerenciamento de manutenção da empresa.

Os itens de *manutenção preventiva*, tais como coleta e análise de dados de vibração, inspeção infravermelha do câmbio, medição ultrasônica de espessura, e amostragem e análise de óleo foram executados habitualmente.

A instalação contratou *peritos no assunto e estabeleceu programas de manutenção*. Como resultado, a ***disponibilidade das máquinas*** aumentou de um grau de **desempenho baixo** para o **mais alto grau possível**.

Entretanto, falhas críticas nas máquinas continuaram a ocorrer aleatória e inesperadamente. As metas de produção foram afetadas e as metas de lucro continuaram inalcançadas.

Apesar do trabalho dos especialistas e do foco na implementação de ***programas de manutenção preventiva e preditiva***, ***falhas inesperadas estavam afetando a confiabilidade do sistema e influenciando negativamente as metas de produção***.

Esta indústria era reconhecida como um estabelecimento de trabalho altamente inovador. Contudo, um problema comum com os programas de manutenção em desenvolvimento é que eles nunca são projetados corretamente na primeira tentativa e cerca de 40 a 60% das tarefas de manutenção propostas seguem um propósito bastante incipiente [J M Moubray, Reliability-Centered Maintenance; Butterworth-Heinemann, Oxford, 1997].

Isto retrata um assunto muito significante sobre como melhorar a produtividade apesar de não haver um planejamento perfeito de manutenção e o agendamento da produção mostrar ineficiências no próprio programa de manutenção. Alcançar 100% de coesão com um programa que possui 50% de utilidade e 50% de desperdício pode não ser uma boa estratégia de administração de ativos [Steve Turner, PM Optimization – Maintenance Analysis of the Future, www.pmoptimization.com, 2001].

Falhas evitáveis continuam ocorrendo. Este fato foi rapidamente determinado como sendo o “coração do problema”. Mas como o uso de KPIs pode ajudar esta companhia?

A gerencia de produção era a principal incumbida de alcançar o esperado lucro de 12%. A fim de alcançar este objetivo, ela rapidamente percebeu que a maneira de fazê-lo não seria somente aumentando contratação de pessoal de operação, mas sim, inserindo todos os funcionários neste processo. Então ela começou a se assegurar de que todos os funcionários seguissem esta métrica em seus departamentos.

A gerência aceitou a responsabilidade e começou a contabilizar e perseguir o lucro de 12%, mas para alcançar aquele resultado final, ela precisou de que todos os departamentos também seguissem esta meta. A gerência percebeu que a única maneira de realizar aquilo era tornando todos co-responsáveis.

Dentro de alguns meses e após muitas discussões e muitas apresentações para os chefes de departamento, os *KPIs começaram a aparecer em cada gestão departamental*. Juntamente com a contabilidade vem a responsabilidade, e neste caso, a responsabilidade de se alcançar o objetivo corporativo. *Os processos departamentais estavam totalmente conectados ao objetivo corporativo.*

Foi desenvolvida uma estratégia que empregava KPIs para gerenciar a efetividade das estratégias de manutenção preventiva e preditiva. Sabendo quando realizar a manutenção preventiva e com que freqüência capturar as informações de manutenção preditiva em cada ativo, de fato, as ações tornaram-se mais efetivas e puderam-se eliminar as falhas evitáveis que estavam ocorrendo.

14.5.1 Como Utilizar KPIs Alinhado às Estratégias de Manutenção

Se o lucro era um objetivo corporativo fundamental, então como esta companhia pôde alinhar suas estratégias de manutenção preventiva para alcançar o objetivo? Como os KPIs foram utilizados neste contexto?

Esta companhia decidiu incorporar uma estratégia de gestão de desempenho de ativos primeiramente definindo quais ativos eram considerados críticos para se alcançar as metas de produção. Eles decidiram primeiramente concentrar seus esforços nos ativos que exerciam impacto sobre negócio e que os estavam impedindo de alcançar os 12% de lucro. Apenas 8% dos 50.000 ativos totais, ou 4000 ativos, foram identificados. Isto incluía ativos giratórios, fixos, elétricos e de instrumentação.

Depois que os ativos altamente críticos foram identificados, *programas de manutenção preventiva e preditiva específicos foram desenvolvidos e executados preferencialmente uma vez por mês, pois a filosofia "uma vez por" já era a estratégia anteriormente empregada.*

Os KPIs foram introduzidos como um esforço para se integrar os objetivos corporativos aos ativos relacionados. Criaram-se métricas tais como:

- ✓ número de falhas em equipamentos de alta criticidade;
- ✓ porcentagem de equipamentos críticos com estudo de manutenção preventiva completo e otimizado.

Como se pode notar, estas métricas não são as métricas que muitas companhias tipicamente adotam. As métricas foram desenvolvidas para englobar uma população de 4000 ativos altamente críticos que estavam impedindo a realização de 12% de lucro. O foco tinha agora se alterado. Ao invés de se desenvolver esforços de manutenção preventiva e preditiva em todos os 50.000 ativos, tais esforços foram concentrados nos 4000 ativos altamente críticos.

Recursos foram destinados para empregar metodologias para aperfeiçoar os planos de manutenção preventiva para cada um dos 4000 ativos. Quando falhas ocorriam em algum dos equipamentos não-críticos da instalação era dada uma atenção significativamente menor.

Operadores, técnicos de manutenção, engenheiros de processo e administradores estavam redirecionando sua alocação de recursos nos 4000 ativos de alta criticidade.

Os KPIs ditavam quais ativos receberiam maior atenção baseando-se na conseqüência e impacto de uma determinada falha. Esta estratégia prática e fácil de implementar utilizando KPIs conduziram a realização dos desejados 12% de lucro.

14.5.2 Benefícios da gestão do desempenho de ativos (APM)

Os benefícios desta estratégia incluem o entendimento de uma meta ou objetivo comum e uma conexão entre estratégias e recursos. O lucro de 12% agora não é só um objetivo corporativo, ele tem um significado real para os funcionários da instalação. Além disso, quando os objetivos não são alcançados, é estabelecido um processo para analisar os resultados obtidos e determinar se as estratégias relacionadas ao ativo específico, cuja falha foi observada, precisam ser alteradas. **Verifica-se em funcionamento uma malha-fechada contínua de processos de negócio.**

Para realizar as tarefas descritas anteriormente, um sistema de APM também foi implementado. Para despender uma quantia correta de tempo analisando os eventos de falha em ativos, especialmente os ativos de alta-criticidade, é necessário manter os dados num local central e detalhar os eventos dos ativos de modo a facilitar a obtenção de diagnósticos e outras informações relevantes. Os sistemas de APM provêem:

- Um processo em malha-fechada de definição de estratégias, avaliação do desempenho destas estratégias, análise das falhas nos ativos objetivando identificar se a estratégia inicial foi ineficiente e se requer mudanças;
- Utilização de um repositório central de toda informação técnica e detalhada sobre os eventos dos ativos;
- Eliminação das barreiras entre os departamentos (todos têm um objetivo comum);
- Utilização de ferramentas estatísticas para avaliar o desempenho dos ativos e auxiliar na compreensão dos fracassos e sucessos anteriores;
- Retorno do valor corporativo (a meta organizacional foi alcançada).

14.5.3 KPIs inseridos no sistema de APM

Quando KPIs são utilizados numa estratégia de APM, surge uma interface entre o sistema de APM e o sistema de gestão de ativos (sistema de gestão da manutenção). Os dados estão automaticamente fluindo entre esses dois sistemas, gerando indicadores-chave de desempenho. Num exemplo específico para uma bomba (PMP-101) a estratégia empregada obtém determinadas tarefas do sistema de APM e automaticamente as insere no sistema de gestão da manutenção.

Tipicamente, elas são programadas como ordens de serviço. Neste processo, todos os dados de ordens de serviço são capturados e enviados ao sistema de APM e é realizada uma coleta detalhada de dados com respeito a qualquer evento associado à PMP-101. Estes dados são habitualmente utilizados na captura de KPIs os quais definem um conjunto de critérios que automaticamente notificam ao usuário a necessidade de uma eventual análise. Neste exemplo, a referida eventual análise poderia ocorrer quando um ativo altamente crítico sofrer uma determinada falha. A implementação do sistema de APM tornou a iniciativa muito mais fácil de se realizar e o ajuste de estratégias mais simples de se desenvolver.

14.6 Roteiro Para Implantação KPIs

Os indicadores-chave de desempenho influenciam as organizações:

- Fornecendo informações para *auxiliar* os administradores a tomarem decisões corretas;
- *Mantendo* o negócio sob controle;
- *Possibilitando* análises quantitativas;
- *Provendo* uma avaliação contínua e compreensiva das pessoas, dos processos e do desempenho da produção global da empresa.

Os direcionadores do negócio ditam a estratégia global. Dependendo de qual for o direcionador do negócio a estratégia será diferente, em alguns casos, de maneira bastante significativa. A seguir há uma descrição resumida de como os KPIs podem ser implementados para coordenar as decisões estratégicas:

1. Comece com uma compreensão clara dos direcionadores do negócio da companhia;

2. Identifique as perspectivas primárias: corporativas, de equipamento, processo e aprendizagem;

3. Defina os objetivos específicos e relacione os KPIs aos objetivos;

4. Assegure-se de que há recurso alocado para a implementação dos KPIs e desenvolve senso de responsabilidade para alcançar as metas desejadas;

5. Quando as metas não forem alcançadas, avalie a estratégia ineficiente ou o processo de trabalho empregado e defina os recursos requeridos para alterar a estratégia ineficiente. (Algumas vezes os custos podem ser considerados muito elevados e as metas precisarão ser modificadas para objetivos mais realísticos).

O novo sistema de automação de Balbina oferece as condições ideais para a implantação de KPIs. O sistema de automação já armazena em banco de dados (Maximo, MES e Simprebal) uma grande quantidade de dados sobre os ativos, sobre falhas, manutenção corretiva e preventiva, manutenção baseada em condição, operação, entre outros, que poderão ser utilizados na geração dos KPIs que irão auxiliar na gestão dos ativos de Balbina.

O sistema de manutenção preditiva de Balbina (SIMPREBAL) dá os primeiros passos na utilização de KPIs pois gera relatórios calculando alguma métricas de importância para a gestão de manutenção de Balbina.

O Sistema Inteligente de Manutenção Preditiva de Balbina (SIMPREBAL) fornece 9 tipos de KPIs, sendo que 3 deles estão relacionados à excelência de serviço, 5 estão relacionados à excelência operacional e o KPI restante mede a confiabilidade das informações geradas pelo próprio SIMPREBAL. Tais KPIs estão descritos nas seções a seguir.

EXCELÊNCIA DE SERVIÇO

A excelência de serviço consiste em determinar quais são os atributos do serviço que de fato são determinantes para a satisfação dos clientes. No caso de geração de energia elétrica, a excelência de serviço, consiste no atendimento à demanda elétrica dos consumidores com 3 atributos:

1. Regime contínuo
2. Boa qualidade
3. Margem de segurança conveniente

Os atributos mencionados podem ser expressos por meio da análise de 3 KPIs, calculados para cada dispositivo de campo:

- Número de ocorrências de ALERTAS;
- Número de ocorrências de ALARMES;
- Número de ocorrências de TRIPS.

EXCELÊNCIA OPERACIONAL

A excelência operacional consiste em identificar oportunidades de melhorias nos processos de trabalho sob a perspectiva dos equipamentos. Isto que significa maximizar a confiabilidade, disponibilidade e mantenabilidade dos equipamentos, e minimizar o tempo médio entre falhas, e tempo de manutenção.

Os 5 KPIs definidos para excelência operacional são:

- Número de ocorrências de falhas;
- Taxa de falhas;
- Tempo médio entre falhas (MTBF);
- Tempo médio de reparo (MTTR);
- Número de prioridade de risco dos equipamentos.

CONFIABILIDADE DO SIMPREBAL

Por fim, a confiabilidade do SIMPREBAL diz respeito à sua capacidade de produzir inferências corretas (diagnósticos e tomadas de decisão) acerca de uma determinada ocorrência de anomalia no sistema. O KPI para medição da confiabilidade do SIMPREBAL é:

- Porcentagem de decisões acertadas com relação a uma determinada falha: para que o cálculo deste KPI seja possível é necessário que, a cada ocorrência de falha identificada pelo SIMPREBAL o operador acesse a tela de análise dos modos e efeitos de falhas e indique se a tomada de decisão relacionada àquela falha condiz ou não com a realidade.

15. REFERÊNCIAS

BERNINI, E.; SOARES, B. AES Brazil Business Review, AES Corporation, 28/06/2006. Disponível em < http://library.corporate-ir.net/library/76/761/76149/items/203397/AESBrazil.pdf > Acesso em: 28/03/2007.

BORGES, C. L. T. Confiabilidade de sistemas de potência. Parte I. Programa de Engenharia Elétrica, COPPE/UFRJ, 2005.

BORGES, C. L. T. Confiabilidade de Sistemas de potência. In: Custos da Confiabilidade, CPE 754. Programa de Engenharia Elétrica, COPPE/UFRJ, 2005.

CIBIÉ, E. B. Saúde e segurança ocupacional. Grupo Nueva, MASISA, 2005. [on line] Disponível em < http://stage.masisa.com/Content.aspx?content=75&oldidioma=3&menu=81&site=4&lang=3&idioma=3 > Acesso em: 02/04/2007.

CROW, L. H. Reliability analysis for complex repairable systems. ReliaSoft Corporation, 1974. Disponível no endereço < http://www.weibull.com/RelGrowthWeb/Crow-AMSAA_(N.H.P.P.).htm > Acesso em: 29/03/2007.

DUFORT, Y. C. ISA-95-Based operations and KPI metrics assessment and analysis. White paper 24, MESA, ISA, Invensys Wonderware, 28/11/2006. [on line] Disponível no endereço < http://us.wonderware.com/NR/rdonlyres/46DFB778-2AF3-4437-A151-ACA0C0424505/0/MESAWhitepaper_Nov2806.pdf > Acesso em: 28/03/2007.

ECKERSON, W. Ten characteristics of a good KPI. The Data Warehousing Institute, 2006.

ELETRONORTE - CENTRAIS ELÉTRICAS DO NORTE DO BRASIL S/A. Programa de manutenção planejada, gerência de engenharia de operação e manutenção da geração hidráulica. Brasília: Centro de Análise e Monitoração de Máquinas do COGH, 2006.

FREITAS, M. A.; ENRICO, A. C. Confiabilidade: análise de tempo de falha e testes de vida acelerados. Belo Horizonte: Fundação Christiano Ottoni, Escola de Engenharia da Universidade Federal de Minas Gerais, 1997. (Série Ferramentas da Qualidade, Volume 12)

HELMAN, H.; ANDERY, P. R. P. Análise de falhas (aplicação dos métodos de FMEA – FTA). Belo Horizonte: Fundação Christiano Ottoni, Escola de Engenharia da Universidade Federal de Minas Gerais, 1995. (Série Ferramentas da Qualidade – Volume 11)

MANAUS ENERGIA. Disponível em < http://www.manausenergia.gov.br >. Acesso em: 20/03/2007.

MONCHY, F. A Função manutenção: formação para a gerência da manutenção Industrial. São Paulo: Durban, Brasileira, 1989. (Série Tecnologias)

MOUBRAY, J. RCM II: Reliability centered maintenance. 2ª ed. New York: Industrial Press Inc, 1997.

ONS – OPERADOR NACIONAL DO SISTEMA ELÉTRICO, Procedimentos de Rede, Módulo 02 – Padrões de Desempenho da Rede Básica. Disponível em < http://www.ons.org.br >. Em vigência a partir de 24/12/2002.

ONS – OPERADOR NACIONAL DO SISTEMA ELÉTRICO, Procedimentos de Rede, Módulo 16 – Acompanhamento da Manutenção, [on line] Disponível em < http://www.ons.org.br >. Em vigência a partir de 24/12/2002.

ONS – OPERADOR NACIONAL DO SISTEMA ELÉTRICO, Procedimentos de Rede, Módulo 22 – Análise de Ocorrências e Perturbações, [on line] Disponível em < http://www.ons.org.br >. Em vigência a partir de 24/12/2002.

SCARPIN, C. A. Análise sistêmica de falhas. São Paulo: Desenvolvimento Gerencial, 1999.

SILVA FILHO, M. N. Sistema de gerenciamento de informações: informativo operacional (INFO_OPR)**,** Programa de Pós-Graduação em Engenharia da Energia, Universidade Federal de Itajubá, Julho de 2002. Disponível em < ftp://fmc2.graco.unb.br/simprebal-edgar-rosi-rodrigo/Rodrigo/dissertação_INFO_OPR.doc >. Acesso em 04/04/2007.

SIQUEIRA, I. P. Manutenção centrada na confiabilidade: manual de implementação. Rio de Janeiro: Qualitymark, 2005.

APÊNDICE A

DESCRIÇÃO TÉCNICA DE CADA TURBINA E GERADOR

O grupo turbina hidráulica e o gerador tem o eixo rotativo comum posicionado em vertical, sendo o gerador na parte superior e turbina hidráulica na parte inferior (figura 1).

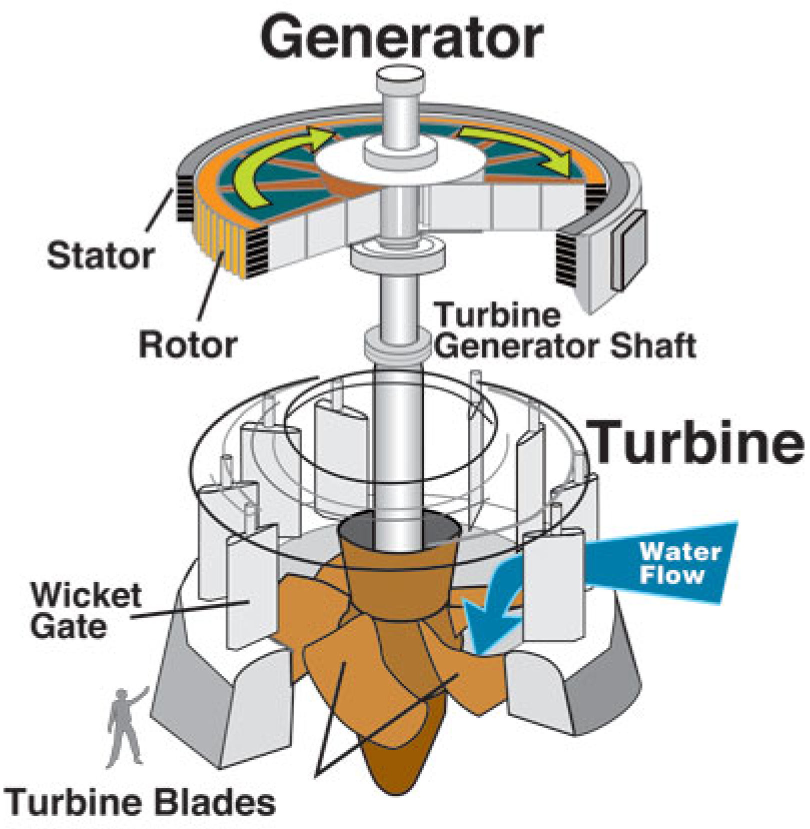


FIGURA 1: ILUSTRAÇÃO DO GRUPO TURBINA GERADOR

A força da água canalizada em movimento aciona a turbina. Como esta força é variável dependendo do nível de água da represa e, por outro lado, a força necessária para girar o eixo do gerador depende da carga (consumo de energia elétrica), torna-se necessário um sistema mecânico que controla o fluxo da água.

SISTEMA DE CONTROLE DE FLUXO DE ÁGUA

O sistema mecânico que controla o fluxo de água é composto de dois cilindros de acionamento hidráulico e um anel circular que transformam o movimento deslizante dos cilindros em movimento circular do anel e o sistema de links que distribui o movimento circular do anel nas 24 laminas fazendo com que o movimento deslizante dos cilindros hidráulicos resulte em alteração de ângulos das laminas do distribuidor. As figuras 2 e 3 mostram esse sistema mecânico de controle e distribuição de fluxo da água. A situação mostrada na figura 3 é quando um dos cilindros está totalmente estendido e o outro totalmente recuado.

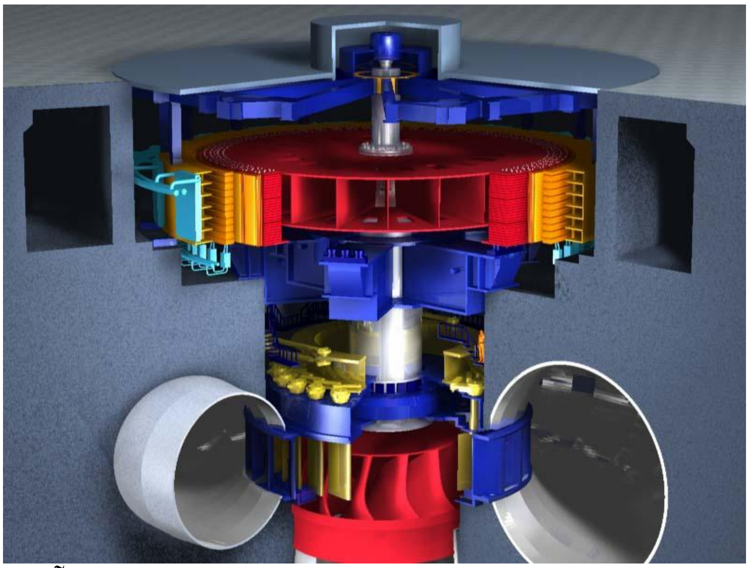


FIGURA 2: ILUSTRAÇÃO DA TURBINA E GERADOR. O MECANISMO QUE CONTROLA O ÂNGULO DAS LAMINAS É VISTO EM AMARELO.

FIGURA 3: MECANISMO DE ANEL E LINKS PARA ALTERAR ÂNGULOS DAS PÁS DO HÉLICE DA TURBINA.

A figura 4 mostra outro cilindro, totalmente recuado. Estes dispositivos mecânicos que é um conjunto de laminas verticais formando circulo têm grau de liberdade rotativo em torno do eixo vertical formando um determinado ângulo em relação às linhas radiais do eixo da turbina. Assim ao alterar o ângulo destas laminas, altera o fluxo de água na turbina. As laminas têm, também, a função de distribuir uniformemente o fluxo de água nas pás do hélice da turbina.

FIGURA 4: UM DOS CILINDROS TOTALMENTE RECUADO

Num dos cilindros hidráulicos está instalado um mecanismo que transmite o movimento deslizante do cilindro em movimento rotativo do sensor de posição para informar a posição atual do cilindro ao controlador (pode ser observado, na figura 5, uma barra cinza que começa no centro da figura continuando para o lado direito superior onde está instalado o sensor).

FIGURA 5: MECANISMO QUE TRANSFORMA MOVIMENTO DESLIZANTE EM ROTATIVO E MECANISMO PARA SENSOR DE POSIÇÃO.

O acionamento desses cilindros é feito por óleo pressurizado a 40 bar utilizando-se o ar pressurizado, ou seja, como pode ser observado na figura 6, dois tanques em formato de balão com comunicação entre si, sendo um deles pintado de azul e amarelo contém óleo com ar pressurizado e no outro, somente o ar pressurizado. Suponho que esse método de pressurização do óleo de acionamento através do ar pressurizado é pela necessidade de manter a pressão do óleo em 40 bar sem ter que acionar permanentemente o motobomba aproveitando a característica de compressibilidade do ar.

FIGURA 6: DOIS TANQUES CILINDRICOS SENDO UM DELES CONTENDO ÓEO COM COLCHÃO DE AR (PINTADO DE AZUL E AMARELO) E UM OUTRO SOMENTE DE AR PRESSURIZADO.

No tanque que contem óleo está instalado indicador de nível de óleo (figura 9) com seguintes marcas:
LC – nível muito alto óleo ; ação : aciona alarme
CJ - nível alto de óleo: ação: injeta ar
CK – nível normal de óleo
CL – nível baixo de óleo; ação: ligar segunda bomba reserva de ar
LD1 – nível muito baixo de óleo; ação: aciona alarme
LD2 – nível muito baixo de óleo; ação: desliga máquina (trípe)
LE- nível muito muito baixo de óleo; ação: fecha comporta.

FIGURA 7: INDICADOR DE NÍVEL DE ÓLEO INSTALADO NA PARTE LATERAL DO TANQUE CILINDRICO DE ÓLEO COM COLCHÃO DE AR.

SISTEMA DE CONTROLE DE POSIÇÃO ANGULAR DAS PÁS DO HÉLICE DA TURBINA.

As pás do hélice da turbina hidráulica desta usina têm ângulos reguláveis através do servomotor hidráulico e dependendo dos ângulos das pás do hélice, muda a força de atuação da água na rotação da turbina. Assim, a ação conjunta – controle do fluxo da água e o controle dos ângulos das pás – resulta na regulação de velocidade do eixo da turbina, mantendo-a em 105,88 rpm, mesmo que haja a variação da carga.

Para isso está instalado o sensor de velocidade angular no eixo do gerador. Não foi possível obter informações sobre o método de regulação de velocidade, mas como a carga varia continuamente, acredita-se que a regulação seja do tipo contínuo, ou seja, tão logo detectado o desvio na velocidade angular do eixo do gerador, o sistema de regulação atua diminuindo ou aumentando o fluxo da água, controlando a posição do cilindro hidráulico que posiciona as laminas do distribuidor e o outro servomotor altera os ângulos das pás do hélice para manter a velocidade angular do eixo do gerador em 105,88 rpm.

O sistema mecânico de mudança de ângulo das pás do hélice da turbina é acionado através do óleo pressurizado por duas motobombas AF e AE instalados na parte superior do tanque retangular de óleo sem pressão. A figura 8 mostra a vista de cima desse tanque com duas motobombas. As quatro vias de canalização de óleo (sendo duas de retorno) entram pela parte superior do eixo do gerador que é vazado e alcança o mecanismo de acionamento que altera os ângulos das pás do hélice da turbina (figura 9).

FIGURA 8: TANQUE DE ÓLEO EM PRESSÃO E DUAS MOTOBOMBAS PARA ACIONAMENTO DOS ÂNGULOS DAS PÁS DO HELICE DA TURBINA

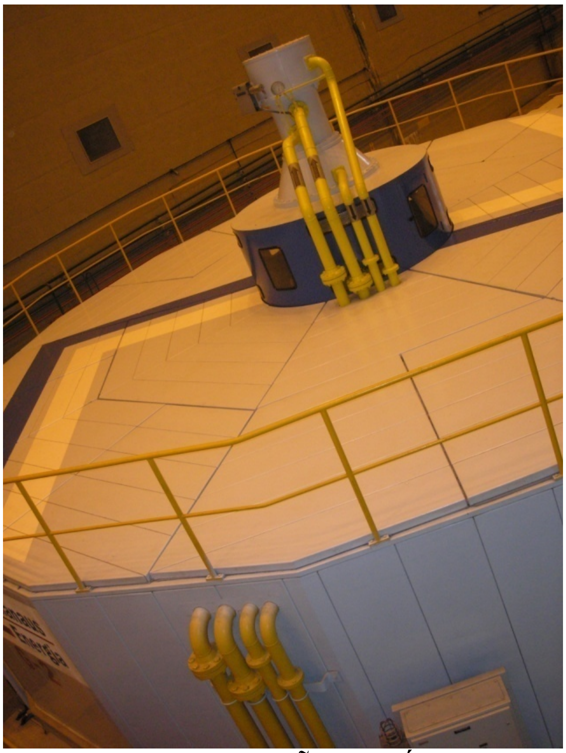


FIGURA 9: QUATRO VIAS DE CANALIZAÇÃO DE ÓLEO PARA ACIONAMENTO DOS ÂNGULOS DAS PÁS DO HELICE DA TURBINA ENTRANDO PELA PARTE SUPERIOR DO GERADOR

Neste tanque de óleo sem pressão estão instalados dois trocadores de calor (figura 10 e 11) e um medidor de temperatura instrumentalizado (figura 12). Está instalado também um monitor de nível de óleo rudimentar (uma janela com material transparente através do qual pode-se ver o nível de óleo do tanque). Nesta janela há indicações de três níveis de óleo – alto, normal e baixo (figura 13). Está instalado também o sensor de nível que é uma bóia.

FIGURA 10 : TANQUE RETANGULAR DE ÓLEO COM DUAS BOMBAS HIDRÁULICAS E TROCADOR DE CALOR.

FIGURA 11: TROCADOR DE CALOR INSTALADO NO TANQUE RETANGULAR.

FIGURA 12: SENSOR DE TEMPERATURA INSTALADO ENTRE OS TROCADORES DE CALOR.

FIGURA 13: MONITOR E NIVEL DE ÓLEO NO TANQUE RETANGULAR DE ÓLEO

SENSOR DE VIBRAÇÃO.

Quando a carga aumenta acima de um determinado valor, o eixo do gerador entra em vibração. Quando esta vibração atinge um determinado valor, torna-se necessário desligar a carga para não causar danos a turbina. Para detectar esta vibração, está instalado o sensor óptico de vibração no eixo. Pela observação realizada, acredita-se que o sensor de vibração é na verdade, sensor de distancia tipo óptico que monitora a distancia entre o sensor e o eixo em movimento rotativo, ou seja, a vibração é monitorada através da variação da distancia entre o sensor e a superfície do eixo rotativo. Este sensor foi instalado experimentalmente, mas notou-se que para se ter um diagnóstico do estado do gerador é necessário que tenham outros sensores para complementar a informação.

SISTEMA DE FRENAGEM.

Quando a carga é diminuída bruscamente ou retirada subitamente por algum motivo, o eixo do gerador fica livre e aumenta a rotação. Conforme o aumento do valor da velocidade angular deve-se tomar providencias mais radicais, que é chamado de “trip”, que é desligamento da carga, quando a velocidade atinge 115 rpm (1º estágio). Quando a velocidade atinge “sobre velocidade” de 165 rpm (2º estágio) o eixo deve ser freado. O sistema de frenagem é pneumático e pela sua importância estão instalados dois sensores de pressão e o gerador só dá partida após estes sensores indicarem a pressão suficiente para frenagem.

Este sistema é composto de 2 sensores de pressão e 1 sistema de atuação de freio pneumático. Atualmente este sistema não está integrado no supervisor, mas deverá ser automatizado. No sistema atual o gerador entra em funcionamento somente após a certificação da pressão pneumática do sistema de frenagem.

FIGURA 14: TANQUE DE AR PRESSURIZADO PARA SISTEMA DE FRENAGEM DA TURBINA E INDICADOR DE PRESSÃO.

SISTEMA DE RESFRIAMENTO DO GERADOR

O resfriamento do estator é através dos 8 radiadores instalados no compartimento próximo ao estator (figura 15).

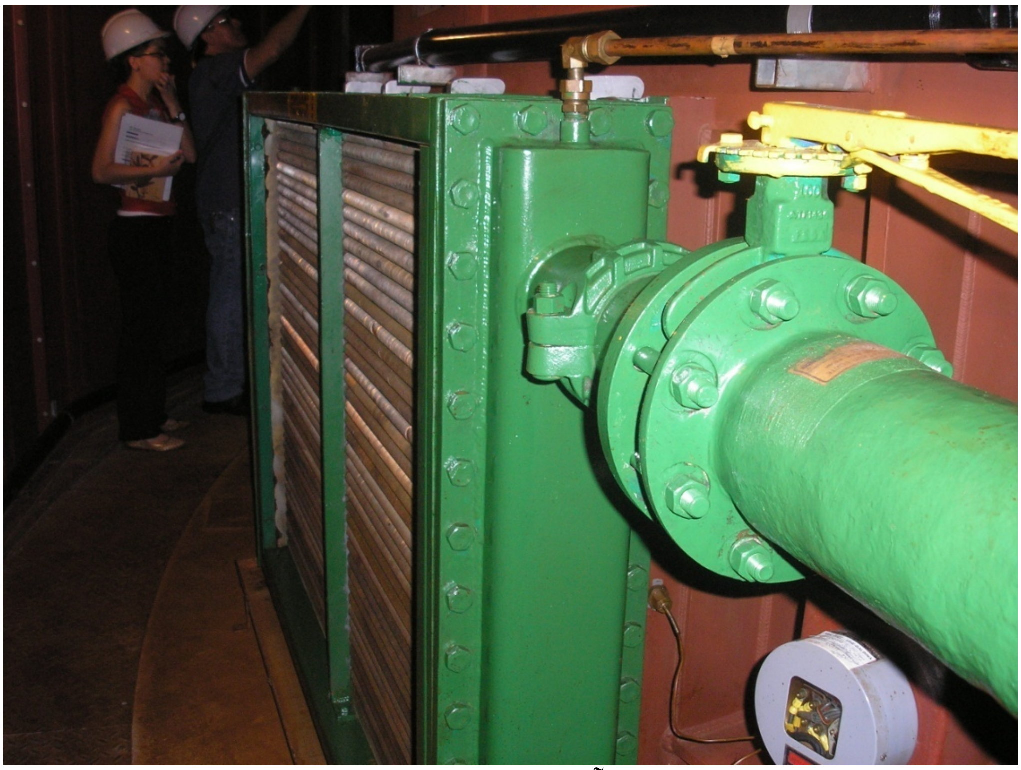


FIGURA 15: RADIADOR PARA REFRIGERAÇÃO DO ESTATOR. EXISTEM OITO UNIDADES EM VOLTA DO ESTATOR

RACK DE TRANSMISSORES

Nas partes laterais do gerador estão instalados dois racks de transmissores que medem temperaturas de patins, metal e óleo do rotor e do estator, temperatura do óleo de regulação e temperatura da água do sistema de regulação da velocidade.

SISTEMA DE VEDAÇÃO DA ÁGUA NO MANCAL INFERIOR

A parte inferior do mancal inferior do eixo do conjunto gerador-turbina (figura 16) está em contato com água canalizada com pressão (é a própria água que aciona a turbina) e com a pressão, essa água poderá infiltrar no mancal. Se isso acontecer, torna-se necessário trocar o óleo de lubrificação do mancal e no pior caso, a substituição do elemento. Para evitar que isso aconteça é injetada água com pressão maior que a água que vem canalizada da represa na direção oposta para evitar que a água infiltre no mancal. Mas mesmo com esta estratégia algumas vezes a água da represa penetra e imunda o mancal e a parte inferior do eixo . (Obs.: Esta foto é situação normal, ou seja, não há água inundando o eixo). Para monitorar este estado serão instalados transmissores de vazão e de pressão.

FIGURA 16: PARTE INFERIOR DO EIXO DA PARTE EXPOSTA DO EIXO DE TURBINA. ALGUMAS VEZES A ÁGUA INVADE O COMPARTIMENTO CILINDRICO VISTO NA PARTE INFERIOR DA FOTO.

BARRAMENTO DE COBRE PROTEGIDO (SAÍDA DA ENERGIA ELÉTRICA)

São três dutos de ar seco em cujo interior encontra-se barramento de cobre (figura 17). O interior do duto é pressurizado com ar seco para evitar a entrada de umidade que poderá afetar o cobre. Um sistema de controle regula a pressão interna do ar seco no interior dos dutos. A pressão deve ser mantida entre 5 a 20 milibar. Este sistema de controle deverá ser digitalizado via supervisão.

FIGURA 17: BARRAMENTO DE COBRE PROTEGIDO COM AR SECO PRESSURIZADO

SISTEMA DE FILTRAGEM DE ÁGUA CANALIZADA PARA ACIONAMENTO DA TURBINA (FIGURAS 18,19 e 20)

Este sistema tem como objetivo reter as impurezas da água que vem canalizada da represa para evitar que as impurezas desgastem as pás da turbina. Neste sistema, a monitoração do estado do filtro é realizada através da diferença de pressão, ou seja, mede-se a pressão da água antes e depois do filtro e quando esta diferença atinge acima de um determinado valor, realiza-se o procedimento de limpeza do filtro. Estes procedimentos de leitura das pressões e limpeza do filtro são manuais e estão previstas as automações desses processos, instalando os transmissores.

FIGURA 18: DUTO DE ENTRADA DO FILTRO DA ÁGUA CANALIZADO DA REPRESA E O FILTRO.

FIGURA 19: FILTRO PARA RETER AS IMPUREZAS DA ÁGUA DA REPRESA.

FIGURA 20: DUTO DE SAIDA DO FILTRO. PODE-SE VER OS DOIS INDICADORES DE PRESSÃO AO LADO DO PAINEL ELÉTRICO.

MEDIDOR DE DENSIDADE DO ÓLEO

O medidor de densidade do óleo para lubrificação do mancal combinado (guia e escora) tem como objetivo monitorar se o óleo de lubrificação está ou não contaminado com água. Este medidor já se encontra digitalizado no supervisor (figuras 21 e 22).

FIGURA 21: MEDIDOR DE DENSIDADE DO ÓLEO.

FIGURA 22: TRANSMISSOR INSTALADO NO MEDIDOR DE DENSIDADE DO ÓLEO.

MANCAIS NO GERADOR E NA TRUBINA

A figura 23 mostra a localização dos mancais no gerador e na turbina e sensores instalados.

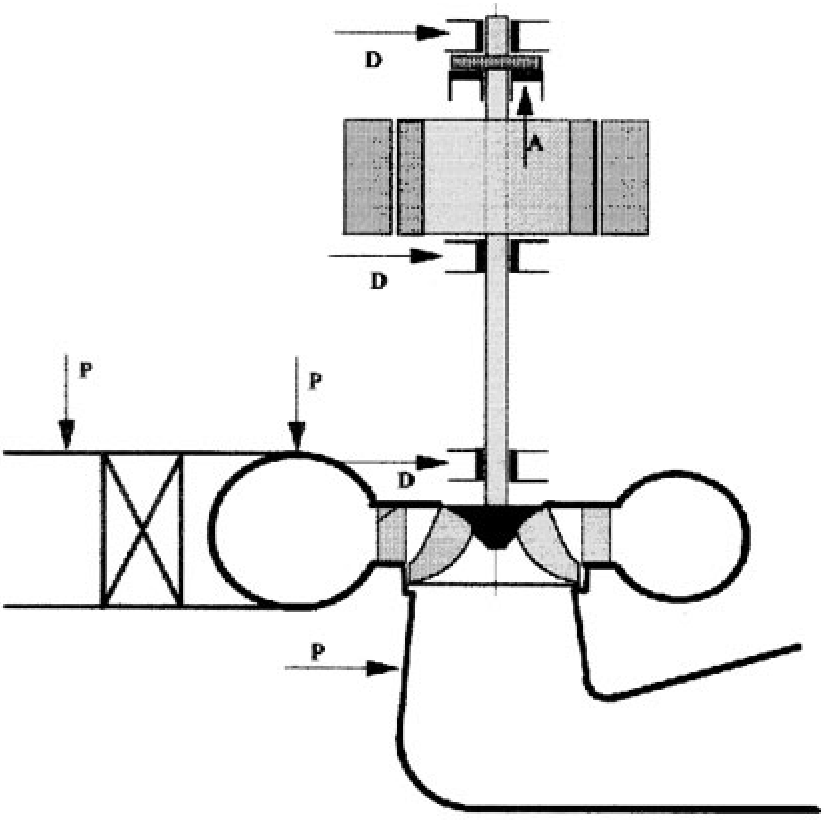


FIGURA 23: DESENHO DO GERADOR E TRUBINA COM A LOCALIZAÇÃO DOS MANCAIS E SENSORES.

Estão instalados dois sensores de proximidade a 90º nos mancais radiais e um no mancal axial para detectar o deslocamento do eixo.
Os mancais, conforme ilustrados na figura 23, são:
Mancal guia superior do gerador
Mancal de escora
Mancal guia inferior do gerador
Mancal guia da turbina.

Os sensores de pressão estão instalados na Tubulação de sucção, no Caracol (caixa espiral) e na Tubulação de descarga.

SISTEMA DE LUBRIFICAÇÃO DO MANCAL GUIA SUPERIOR (GERADOR) (FIGURA 24,25 E 26)

A lubrificação do mancal é realizada através do óleo pressurizado por bomba hidráulica. Neste sistema é monitorado os seguintes dados:
Fluxo de óleo (vazão) e a falta de fluxo causa “trip” ;
Pressão da bomba;
Temperatura antes e depois do trocador de calor;
Temperatura da água de refrigeração.

FIGURA 24: SISTEMA DE CIRCULAÇÃO DE ÓLEO PARA LUBRIFICAÇÃO DO MANCAU GUIA SUPERIOR DO GERADOR.

FIGURA 25: SISTEMA DE LUBRIFICAÇÃO DO MANCAL GUIA SUPERIOR (TROCADOR DE CALOR NO CENTRO DA FOTO)

FIGURA 26: BOMBAS HIDRÁULICAS PARA CIRCULAÇÃO DO ÓLEO DE LUBRIFICAÇÃO DO MANCAL GUIA SUPERIOR.

SISTEMA DE LUBRIFIAÇÃO DO MANCAL COMBINADO

O sistema de lubrificação do mancal combinado é similar ao do mancal guia superior. A figura 27 mostra duas motobombas para enviar o óleo para mancal, a figura 28 mostra os filtros, as figuras 29, 30 e 31 mostram os medidores de pressão de entrada e saída dos filtros e a figura 32 mostra o trocador de calor.

FIGURA 27: BOMBAS DE ÓLEO PARA LUBRIFICAÇÃO DO MANCAL GUIA E DE ESCORA COMBINADOS.

FIGURA 28: FILTROS DE ÓLEO PARA SISTEMA DE LUBRIFICAÇÃO DO MANCAL GUIA E DE ESCORA COMBINADOS. OS MEDIDORES DE PRESSÃO ESTÃO ENTRE DOIS FILTROS

FIGURA 29: MEDIDOR DE PRESSÃO DO ÓLEO NA SAIDA DO FILTRO

FIGURA 30: MEDIDOR DE PRESSÃO DO ÓLEO NA ENTRADA DO FILTRO

FIGURA 31: MEDIDORES DE PRESSÃO DO ÓLEO NA ENTRADA E SAIDA DO FILTRO (A ESQUERDA É DE SAIDA)

Neste sistema de lubrificação existe um sistema de injeção de óleo para criar um filme de óleo no mancal combinado antes da partida da máquina (figura 32).

FIGURA 32: SISTEMA DE INJEÇÃO DE ÓLEO PARA CRIAR FILME DE ÓLEO NO MANCAL COMBINADO ANTES DA PARTIDA DA MÁQUINA

FIGURA 33: TROCADOR DE CALOR DO ÓLEO DE LUBRIFICAÇÃO DO MANCAL COMBINADO

MEDIDOR DE PRESSÃO DA ÁGUA NO INTERIOR DO TUBO DE SUCÇÃO DA TURBINA

O medidor de pressão da água no interior do tubo de sucção da turbina será automatizado.

SISTEMA DE MANUTENÇÃO

O sistema de manutenção é composto de diversos subsistemas:

Monitor de caixa espiral: Caixa espiral é uma tubulação de forma toroidal que envolve a região do rotor. É a parte integrante da estrutura civil da usina. É fabricado com chapas de aço carbono soldados com segmentos. A caixa espiral conecta-se ao conduto forçado na seção de entrada e ao pré-distribuidor na seção de saída. Na ocasião da manutenção este monitor indica se a caixa está vazia ou não. (figura 34)

FIGURA 34: MONITOR DE CAIXA ESPIRAL

COMPRESSOR DE REGULAÇÃO (figura 35 E 36)

São os compressores que pressurizam o ar em 50 bar para ser enviado no sistema de colchão de ar do tanque de óleo que por sua vez aciona o sistema mecânico de controle de fluxo da água canalizada que aciona a turbina. Os 4 sensores de pressão do compressor serão digitalizados.

FIGURA 35: COMPRESSOR DE REGULAÇÃO

FIGURA 36: TANQUE DE AR PRESSURIZADO DE REGULAÇÃO

COMPRESSOR DE AR DE SERVIÇO
Será instalado transmissor para supervisório.

MEDIDOR DE NÍVEL PARA RESERVATÓRIO DA ÁGUA DA SELAGEM (VEDAÇÃO)
Atualmente existem 2 reservatório de água para vedação do mancal inferior do eixo de turbina-gerador. Serão instalados dois transmissores de nível para supervisório.

CENTRAL HIDRÁULCIO DAS TOMADAS DE ÁGUA
O central hidráulico de tomadas de água é composto de 3 seções de 11,80 m de altura por 4m de largura. A queda d'água é de no mínimo 22 metros.

CENTRALINA HIDRÁULICA DA COMPORTACentralina (figuras 37 e 38)
A centralina hidráulica da comporta é composto de tanque de óleo e bomba de óleo. Este sistema controla o movimento da comporta. Será instalado sensor de nível de óleo e pressão de óleo.

FIGURA 37: TANQUE DE ÓLEO DA CENTALINA HIDRÁULICA


FIGURA 38: BOMBA DE ÓLEO DA CENTRALINA HIDRÁULICA

APÊNDICE B: ISA-88

AMERICAN NATIONAL STANDARD

ANSI/ISA–88.01–1995

Formerly ANSI/ISA–S88.01–1995

Batch Control Part 1: Models and Terminology

Approved 23 October 1995

ISA–The Instrumentation, Systems, and Automation Society

ANSI/ISA-88.01-1995, Batch Control, Part 1: Models and Terminology

ISBN: 1-55617-562-0

ISA
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, North Carolina 27709

Preface

This preface as well as all footnotes and annexes are included for informational purposes and are not part of ANSI/ISA-88.01-1995.

This standard has been prepared as part of the service of the ISA, the international society for measurement and control, toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms and asks that they be addressed to the Secretary, Standards and Practices Board; ISA; 67 Alexander Drive; P. O. Box 12277; Research Triangle Park, NC 27709; Telephone (919) 990-9227; Fax (919) 549-8288;e-mail: standards@isa.org.

The ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards, recommended practices, and technical reports. The Department is further aware of the benefits to USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, this Department will endeavor to introduce SI-acceptable metric units in all new and revised standards to the greatest extent possible. *The Metric Practice Guide*, which has been published by the Institute of Electrical and Electronics Engineers as ANSI/IEEE Std. 268-1992, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

It is the policy of ISA to encourage and welcome the participation of all concerned individuals and interests in the development of ISA standards, recommended practices, and technical reports. Participation in the ISA standards-making process by an individual in no way constitutes endorsement by the employer of that individual, of ISA, or of any of the standards, recommended practices, and technical reports that ISA develops.

This standard is structured to follow the IEC guidelines. Therefore, the first three sections discuss the *Scope* of the standard, *Normative References, and Definitions*, in that order.

Section 4 is entitled *Batch Processes and Equipment*. The intent of this section is to discuss batch processing and the batch manufacturing plant. Things that are involved in batch manufacturing (e.g., batch process classification, equipment, and processes) are described in this section. The models and terminology defined in this section provide a foundation for understanding the application of batch control to the batch manufacturing plant in Sections 5 and 6.

Section 5 is entitled *Batch Control Concepts*. The intent of this section is to discuss key aspects of batch processing and batch manufacturing plants. This is where control is finally introduced to physical equipment, and the concept of equipment entities is introduced. Recipes are introduced in Section 5. The concepts of Allocation and Arbitration, Modes and States, and Exception Handling are introduced in this section so that they can be applied to the discussions in Section 6.

Section 6 is entitled *Batch Control Activities and Functions*. The intent of the models and terminology introduced in this section is to establish the necessary control activities that are needed to address the diverse control requirements of batch manufacturing. The concept of a Control Activity Model is introduced in this section. Each control activity from the Control Activity Model is discussed in terms of the individual control functions that are needed to address the batch processing, manufacturing, and control requirements of the previous two sections. Note that there will be no attempt to define compliance requirements within this section since the overall purpose of this standard is to define a common approach to defining and modeling batch processes and their associated controls.

This standard (Part 1, Models and Terminology) is intended for people who are

— involved in designing and/or operating batch manufacturing plants;

— responsible for specifying controls and the associated application programs for batch manufacturing plants; or

— involved in the design and marketing of products in the area of batch control.

The following people served as active members of ISA Committee SP88:

NAME	COMPANY
L. Craig, Chairman	Rohm and Haas Company
*R. Mergen, Past Chairman	The Lubrizol Corporation
*T. Fisher, Past Chairman/Editor	The Lubrizol Corporation
C. Gross, Past Managing Director	Dow Chemical Company
*M. Albano	Honeywell, Inc.
A. Aujesky	ICI Australia Engineering
*J. Barrault	Siemens
G. Barron	John Brown Engineers & Constructors BV
R. Baxter	Eastman Kodak Company
*D. Brandl	Groupe Schneider
B. Braunstein	Exxon Chemical Company
*E. Bristol	The Foxboro Company
M. Bruns	Hoechst AG
R. Bullotta	WonderWare
*H. Burns	Fisher • Rosemount
G. Carlo-Stella	Batch Systems International
*B. Casey	Groupe Schneider
D. Chappell	Procter & Gamble Company
*L. Charpentier	GSE Process Solutions
*T. Crowl	Moore Products Company
*B. Cubizolles	Siemens
M. Dawson	Smith-Kline Beecham
*K. Dittman	Johnson Yokogawa
*D. Dodd	The Foxboro Company
*S. Duff	Moore Products Company
*C. Eaves	Intellution, Inc.
*D. Edwards	Johnson Yokogawa
*D. Emerson	GSE Process Solutions
*S. Farmer	ABB Process Automation, Inc.
G. Felton	Ashland Chemical
H. Fittler	Honeywell Regelsysteme
*A. Ghosh	The Foxboro Company
*P. Gustafson	Hartmann & Braun

*One vote per company

NAME	COMPANY
*R. Hall	PID, Inc.
*W. Hawkins	Fisher • Rosemount
N. Haxthausen	Novo-Nordisk Engineering
*C. Hertz	Bailey Controls Company
S. Hjelmager	CRI Industrial Systems A/S
T. Hoekstra	Yokogawa Europe
*T. Hollowell	Fisher • Rosemount
*D. Hornbeck	Allen-Bradley Co.
*D. Imming	Fisher • Rosemount
*S. Jayanthi	Intellution, Inc.
*B. Jensen	Johnson Yokogawa
T. Jonsson	ABB Automation AB
H. Kayser	Consultant
G. Klipfel	Upjohn Company
*D. Leach	Air Products & Chemicals Company
T. Leffert	3M
*B. Lightle	Allen-Bradley Company
*W. Loner	Bailey Controls Company
R. Lotz	Consultant
B. Lozier	Pacific Access Computer
*D. Macias	Fisher • Rosemount
S. Mallaband	Bass Brewers, Ltd.
*E. Massey	Honeywell, Inc.
W. McFarlane	Valmet Automation, Inc.
*N. Meierhoefer	Hartmann & Braun
*T. Müller-Heinzerling	Siemens
L. Natiello	Kraft General Foods
K. Ng	Office of Naval Research
*P. Nowicki	ABB Process Automation, Inc.
A. Pampel	A. F. Pampel Consulting
*A. Pawlus	Honeywell, Inc.
*S. Prichard	Fisher • Rosemount
A. Rabinowitz	Consultant
H. Rosenof	Gensym Corporation
*J. Ruhe	Bailey Controls Company
P. Saebye	CRI Industrial Systems A/S
*M. Saucier	PID, Inc.
*C. Schmidt	Siemens
*J. Schwatmann	Siemens
J. Shaffer	Consultant
R. Shilts	Modicon AEG
E. Smith	Good Manufacturing Practices, Inc.
*K. Spencer	ABB Process Automation, Inc.
*J. Sten	E. I. du Pont de Nemours & Company (Ret.)
*F. Sutter	Fisher • Rosemount
*B. Sykes	Johnson Yokogawa
R. Thome	Merck
T. Tom	Elsag Bailey

*One vote per company

NAME	COMPANY
J. Unger	Chesebrough-Pond's (Unilever)
*J. Vardy	The Foxboro Company
J. Verhulst	Biogen, Inc.
J. Via III	Alcon Labs
*N. Vroom	Honeywell, Inc.
*H. Wähner	Hartmann & Braun
*M. Warburton	ABB Process Automation, Inc.
R. Watson	Mettler-Toledo, Inc.
*A. Webster	E. I. du Pont de Nemours & Company
A. Weidenbach	Eastman Chemical Company
*S. Whitman	Johnson Yokogawa
*E. Whitmer	Honeywell, Inc.
*G. Wilcox	Air Products & Chemicals Company

This published standard was approved for publication by the ISA Standards and Practices Board in February 1995.

*One vote per company

Contents

Annexes

Figures

Tables

Foreword

1) The formal decisions or agreements of the IEC on technical matters, prepared by technical committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.

2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.

3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules insofar as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rules should, as far as possible, be clearly indicated in the latter.

4) The IEC has not laid down any procedure concerning marking as an indication of approval and has no responsibility when an item of equipment is declared to comply with one of its recommendations.

This part of this International Standard has been prepared by IEC/SC65A/WG11 and ISA SP88.

It forms part 1 of a series, the other part being Part 2: Data structures and guidelines for languages.

Annex A forms an integral part of this part of this international standard. Refer to Annex A for an explanation of the format and general associations used in creating the diagrams in this international standard. Annex B is for information only.

Introduction

This part of this international standard on Batch Control provides standard models and terminology for defining the control requirements for batch manufacturing plants. The models and terminology defined in this standard

— emphasize good practices for the design and operation of batch manufacturing plants;

— can be used to improve control of batch manufacturing plants; and

— can be applied regardless of the degree of automation.

Specifically, this standard provides a standard terminology and a consistent set of concepts and models for batch manufacturing plants and batch control that will improve communications between all parties involved; and that will

— reduce the user's time to reach full production levels for new products;

— enable vendors to supply appropriate tools for implementing batch control;

— enable users to better identify their needs;

— make recipe development straightforward enough to be accomplished without the services of a control systems engineer;

— reduce the cost of automating batch processes; and

— reduce life-cycle engineering efforts.

It is not the intent of this standard to

— suggest that there is only one way to implement or apply batch control;

— force users to abandon their current way of dealing with their batch processes; or

— restrict development in the area of batch control.

The models presented in this standard are presumed to be complete as indicated. However, they may be collapsed and expanded as described below. The unit and the control module levels may not be omitted from the physical model. The master recipe and the control recipe may not be omitted from the recipe types model. Specific rules for collapsing and expanding these models are not covered in this standard.

— Collapsing: Elements in the models may be omitted as long as the model remains consistent, and the functions of the element removed are taken into account.

— Expanding: Elements may be added to the modules. When they are added between related elements, the integrity of the original relationship should be maintained.

1 Scope

This part of the standard on Batch Control defines reference models for batch control as used in the process industries and terminology that helps explain the relationships between these models and terms. This standard may not apply to all batch control applications.

2 Normative references

The following normative documents contain provisions, which through reference in this text, constitute provisions of this part of this standard. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this part of this standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid normative documents.

IEC 848: 1988, *Preparation of function charts for control systems*

NOTE – Structures defined in IEC 848 may be useful in the definition of procedural control, and in particular in the definition of a phase.

IEC 902: 1987, *Industrial-process measurement and control — Terms and definitions*

NOTE – Definitions found in IEC 902 were used as a basis for definitions in this standard. Where necessary, the specific connotation of terms used in batch control were included as definitions in this standard.

3 Definitions

For the purposes of this part of this international standard, the following definitions apply.

3.1 allocation: A form of coordination control that assigns a resource to a batch or unit.

NOTE – An allocation can be for the entire resource or for portions of a resource.

3.2 arbitration: A form of coordination control that determines how a resource should be allocated when there are more requests for the resource than can be accommodated at one time.

3.3 area: A component of a batch manufacturing site that is identified by physical, geographical, or logical segmentation within the site.

NOTE – An area may contain process cells, units, equipment modules, and control modules.

3.4 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.

3.5 batch: 1.) The material that is being produced or that has been produced by a single execution of a batch process. 2.) An entity that represents the production of a material at any point in the process.

NOTE – Batch means both the material made by and during the process and also an entity that represents the production of that material. Batch is used as an abstract contraction of the words "the production of a batch."

3.6 batch control: Control activities and control functions that provide a means to process finite quantities of input materials by subjecting them to an ordered set of processing activities over a finite period of time using one or more pieces of equipment.

3.7 batch process: A process that leads to the production of finite quantities of material by subjecting quantities of input materials to an ordered set of processing activities over a finite period of time using one or more pieces of equipment.

3.8 batch schedule: A list of batches to be produced in a specific process cell.

NOTE – The batch schedule typically contains such information as what is to be produced, how much is to be produced, when or in what order the batches are to be produced, and what equipment is to be used.

3.9 common resource: A resource that can provide services to more than one requester.

NOTE – Common resources are identified as either exclusive-use resources or shared-use resources (3.22 and 3.54).

3.10 control module: The lowest level grouping of equipment in the physical model that can carry out basic control.

NOTE – This term applies to both the physical equipment and the equipment entity.

3.11 control recipe: A type of recipe which, through its execution, defines the manufacture of a single batch of a specific product.

3.12 coordination control: A type of control that directs, initiates, and/or modifies the execution of procedural control and the utilization of equipment entities.

3.13 enterprise: An organization that coordinates the operation of one or more sites.

3.14 equipment control: The equipment-specific functionality that provides the actual control capability for an equipment entity, including procedural, basic, and coordination control, and that is not part of the recipe.

3.15 equipment entity: A collection of physical processing and control equipment and equipment control grouped together to perform a certain control function or set of control functions.

3.16 equipment module: A functional group of equipment that can carry out a finite number of specific minor processing activities.

NOTES

1 An equipment module is typically centered around a piece of process equipment (a weigh tank, a process heater, a scrubber, etc.). This term applies to both the physical equipment and the equipment entity.

2 Examples of minor process activities are dosing and weighing.

3.17 equipment operation: An operation that is part of equipment control.

3.18 equipment phase: A phase that is part of equipment control.

3.19 equipment procedure: A procedure that is part of equipment control.

3.20 equipment unit procedure: A unit procedure that is part of equipment control.

3.21 exception handling: Those functions that deal with plant or process contingencies and other events which occur outside the normal or desired behavior of batch control.

3.22 exclusive-use resource: A common resource that only one user can use at any given time.

3.23 formula: A category of recipe information that includes process inputs, process parameters, and process outputs.

3.24 general recipe: A type of recipe that expresses equipment and site independent processing requirements.

3.25 header: Information about the purpose, source and version of the recipe such as recipe and product identification, creator, and issue date.

3.26 ID: A unique identifier for batches, lots, operators, technicians, and raw materials.

3.27 line; train: See definition for train.

3.28 lot: A unique amount of material having a set of common traits.

NOTE – Some examples of common traits are material source, the master recipe used to produce the material, and distinct physical properties.

3.29 master recipe: A type of recipe that accounts for equipment capabilities and may include process cell-specific information.

3.30 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.

3.31 operation: A procedural element defining an independent processing activity consisting of the algorithm necessary for the initiation, organization, and control of phases.

3.32 path; stream: The order of equipment within a process cell that is used, or is expected to be used, in the production of a specific batch.

3.33 personnel and environmental protection: The control activity that

— prevents events from occurring that would cause the process to react in a manner that would jeopardize personnel safety and/or harm the environment; and/or

— takes additional measures, such as starting standby equipment, to prevent an abnormal condition from proceeding to a more undesirable state that would jeopardize personnel safety and/or harm the environment.

3.34 phase: The lowest level of procedural element in the procedural control model.

3.35 procedural control: Control that directs equipment-oriented actions to take place in an ordered sequence in order to carry out some process-oriented task.

3.36 procedural element: A building block for procedural control that is defined by the procedural control model.

3.37 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.

3.38 process: A sequence of chemical, physical, or biological activities for the conversion, transport, or storage of material or energy.

3.39 process action: Minor processing activities that are combined to make up a process operation.

NOTE – Process actions are the lowest level of processing activity within the process model.

3.40 process cell: A logical grouping of equipment that includes the equipment required for production of one or more batches. It defines the span of logical control of one set of process equipment within an area.

NOTE – This term applies to both the physical equipment and the equipment entity.

3.41 process control: The control activity that includes the control functions needed to provide sequential, regulatory, and discrete control and to gather and display data.

3.42 process input: The identification and quantity of a raw material or other resource required to make a product.

3.43 process management: The control activity that includes the control functions needed to manage batch production within a process cell.

3.44 process operation: A major processing activity that usually results in a chemical or physical change in the material being processed and that is defined without consideration of the actual target equipment configuration.

3.45 process output: An identification and quantity of material or energy expected to result from one execution of a control recipe.

3.46 process parameter: Information that is needed to manufacture a material but does not fall into the classification of process input or process output.

NOTE – Examples of process parameter information are temperature, pressure, and time.

3.47 process stage: A part of a process that usually operates independently from other process stages and that usually results in a planned sequence of chemical or physical changes in the material being processed.

3.48 recipe: The necessary set of information that uniquely defines the production requirements for a specific product.

NOTE – There are four types of recipes defined in this standard: general, site, master, and control.

3.49 recipe management: The control activity that includes the control functions needed to create, store, and maintain general, site, and master recipes.

3.50 recipe operation: An operation that is part of a recipe procedure in a master or control recipe.

3.51 recipe phase: A phase that is part of a recipe procedure in a master or control recipe.

3.52 recipe procedure: The part of a recipe that defines the strategy for producing a batch.

3.53 recipe unit procedure: A unit procedure that is part of a recipe procedure in a master or control recipe.

3.54 shared-use resource: A common resource that can be used by more than one user at a time.

3.55 site: A component of a batch manufacturing enterprise that is identified by physical, geographical, or logical segmentation within the enterprise.

NOTE – A site may contain areas, process cells, units, equipment modules, and control modules.

3.56 site recipe: A type of recipe that is site specific.

NOTE – Site recipes may be derived from general recipes recognizing local constraints, such as language and available raw materials.

3.57 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.

3.58 stream; path: See definition for path.

3.59 train; line: A collection of one or more units and associated lower level equipment groupings that has the ability to be used to make a batch of material.

3.60 unit: A collection of associated control modules and/or equipment modules and other process equipment in which one or more major processing activities can be conducted.

NOTES

1 Units are presumed to operate on only one batch at a time. Units operate relatively independently of one another.

2 This term applies to both the physical equipment and the equipment entity.

3 Examples of major processing activities are react, crystallize, and make a solution.

3.61 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.

3.62 unit recipe: The part of a control recipe that uniquely defines the contiguous production requirements for a unit.

NOTE – The unit recipe contains the unit procedure and its related formula, header, equipment requirements, and other information.

3.63 unit supervision: The control activity that includes control functions needed to supervise the unit and the unit's resources.

4 Batch processes and equipment

This section provides an overview of batch processing and the batch manufacturing plant. The models and terminology defined in this section provide a foundation for understanding the application of batch control to the batch manufacturing plant in Sections 5 and 6. Specifically, this section discusses batch processes, a physical model, and process cell classification.

4.1 Processes, batches, and batch processes

A process is a sequence of chemical, physical or biological activities for the conversion, transport or storage of material or energy. Industrial manufacturing processes can generally be classified as continuous, discrete parts manufacturing, or batch. How a process is classified depends on whether the output from the process appears in a continuous flow (continuous), in finite quantities of parts (discrete parts manufacturing), or in finite quantities of material (batches). Although aspects of this standard may apply to discrete parts manufacturing or continuous processes, this standard does not specifically address these types of processes.

4.1.1 Continuous processes

In a continuous process, materials are passed in a continuous flow through processing equipment. Once established in a steady operating state, the nature of the process is not dependent on the length of time of operation. Start-ups, transitions, and shutdowns do not usually contribute to achieving the desired processing.

4.1.2 Discrete parts manufacturing processes

In a discrete parts manufacturing process, products are classified into production lots that are based on common raw materials, production requirements, and production histories. In a discrete parts manufacturing process, a specified quantity of product moves as a unit (group of parts) between workstations, and each part maintains its unique identity.

4.1.3 Batch processes

The batch processes addressed in this standard lead to the production of finite quantities of material (batches) by subjecting quantities of input materials to a defined order of processing actions using one or more pieces of equipment. The product produced by a batch process is called a batch. Batch processes are discontinuous processes. Batch processes are neither discrete nor continuous; however, they have characteristics of both.

The subdivisions of a batch process can be organized in a hierarchical fashion as shown in Figure 1. The example batch process used in this section is the production of polyvinyl chloride by the polymerization of vinyl chloride monomer.

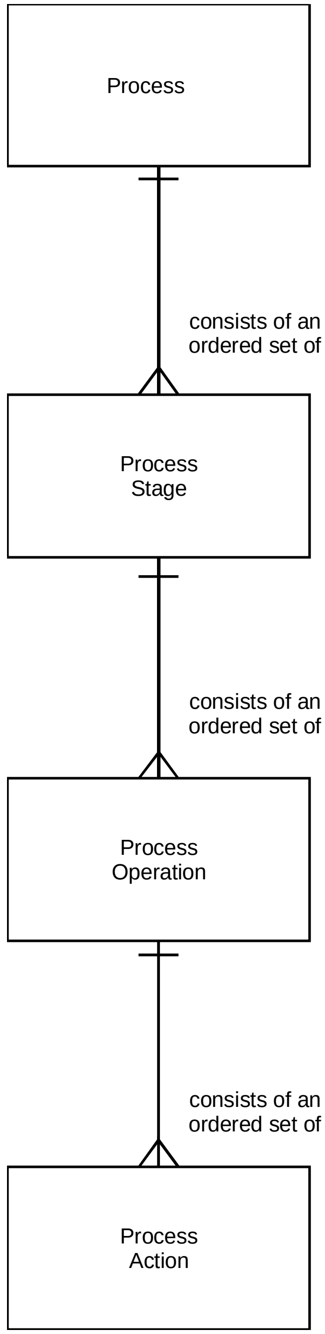


Figure 1 — Process model (Entity - Relationship diagram)

4.1.3.1 Process stages

The process consists of one or more process stages which are organized as an ordered set, which can be serial, parallel, or both. A process stage is a part of a process that usually operates independently from other process stages. It usually results in a planned sequence of chemical or physical changes in the material being processed. Typical process stages in the polyvinyl chloride process might be the following:

— Polymerize: Polymerize vinyl chloride monomer into polyvinyl chloride.

— Recover: Recover residual vinyl chloride monomer.

— Dry: Dry polyvinyl chloride.

4.1.3.2 Process operations

Each process stage consists of an ordered set of one or more process operations. Process operations represent major processing activities. A process operation usually results in a chemical or physical change in the material being processed. Typical process operations for the polymerization of vinyl chloride monomer into polyvinyl chloride process stage might be the following:

— Prepare reactor: Evacuate the reactor to remove oxygen.

— Charge: Add demineralized water and surfactants.

— React: Add vinyl chloride monomer and catalyst, heat to 55 - 60°C, and hold at this temperature until the reactor pressure decreases.

4.1.3.3 Process actions

Each process operation can be subdivided into an ordered set of one or more process actions that carry out the processing required by the process operation. Process actions describe minor processing activities that are combined to make up a process operation. Typical process actions for the react process operation might be the following:

— Add: Add the required amount of catalyst to the reactor.

— Add: Add the required amount of vinyl chloride monomer to the reactor.

— Heat: Heat the reactor contents to 55 - 60°C.

— Hold: Hold the reactor contents at 55 - 60°C until the reactor pressure decreases.

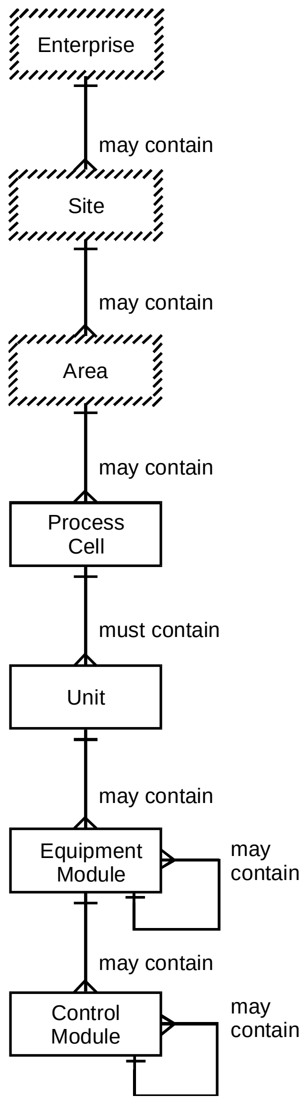
4.2 Physical model

This section discusses a physical model that can be used to describe the physical assets of an enterprise in terms of enterprises, sites, areas, process cells, units, equipment modules, and control modules.

The physical assets of an enterprise involved in batch manufacturing are usually organized in a hierarchical fashion as described in Figure 2. Lower level groupings are combined to form higher levels in the hierarchy. In some cases, a grouping within one level may be incorporated into another grouping at that same level.

The model has seven levels, starting at the top with an enterprise, a site, and an area. These three levels are frequently defined by business considerations and are not modeled further in this document. The three higher levels are part of the model to properly identify the relationship of the lower level equipment to the manufacturing enterprise.

The lower four levels of this model refer to specific equipment types. An equipment type in Figure 2 is a collection of physical processing and control equipment grouped together for a specific purpose. The lower levels in the model are specific to technically defined and bounded groupings of equipment. The four lower equipment levels (process cells, units, equipment modules, and control modules) are defined by engineering activities (see 5.2.3 and 6.1.3). During these engineering activities, the equipment at lower levels is grouped together to form a new higher level equipment grouping. This is done to simplify operation of that equipment by treating it as a single larger piece of equipment. Once created, the equipment cannot be split up except by re-engineering the equipment in that level.



NOTE – The boxes for the top three levels are shown with slashed lines to indicate that the criteria that are used for configuring the boundaries of these three levels is often beyond the scope of batch control and this standard. Therefore, criteria for configuring the boundaries of these three levels of the physical model will not be discussed in this standard.

Figure 2 — Physical model

4.2.1 Enterprise level

An enterprise is a collection of one or more sites. It may contain sites, areas, process cells, units, equipment modules, and control modules.

The enterprise is responsible for determining what products will be manufactured, at which sites they will be manufactured, and in general how they will be manufactured.

There are many factors other than batch control that affect the boundaries of an enterprise. Therefore, the criteria for configuring the boundaries of an enterprise are not covered in this standard.

4.2.2 Site level

A site is a physical, geographical, or logical grouping determined by the enterprise. It may contain areas, process cells, units, equipment modules, and control modules.

The boundaries of a site are usually based on organizational or business criteria as opposed to technical criteria. There are many factors other than batch control that affect these boundaries. Therefore, the criteria for configuring the boundaries of a site are not covered in this standard.

4.2.3 Area level

An area is a physical, geographical, or logical grouping determined by the site. It may contain process cells, units, equipment modules, and control modules.

The boundaries of an area are usually based on organizational or business criteria as opposed to technical criteria. There are many factors other than batch control that affect these boundaries. Therefore, the criteria for configuring the boundaries of an area are not covered in this standard.

4.2.4 Process cell level

A process cell contains all of the units, equipment modules, and control modules required to make one or more batches.

Process control activities must respond to a combination of control requirements using a variety of methods and techniques. Requirements that cause physical control actions may include responses to process conditions or to comply with administrative requirements.

A frequently recognized subdivision of a process cell is the train. A train is composed of all units and other equipment that may be utilized by a specific batch. A batch does not always use all the equipment in a train. Furthermore, more than one batch and more than one product may use a train simultaneously. The order of equipment actually used or expected to be used by a batch is called the path. Although a process cell may contain more than one train, no train may contain equipment outside the boundaries of the process cell.

A process cell is a logical grouping of equipment that includes the equipment required for production of one or more batches. It defines the span of logical control of one set of process equipment within an area. The existence of the process cell allows for production scheduling on a process cell basis, and also allows for process cell-wide control strategies to be designed. These process cell-wide control strategies might be particularly useful in emergency situations.

4.2.5 Unit level

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.

One or more major processing activities — such as react, crystallize, and make a solution — can be conducted in a unit. It combines all necessary physical processing and control equipment required to perform those activities as an independent equipment grouping. It is usually centered on a major piece of processing equipment, such as a mixing tank or reactor. Physically, it includes or can acquire the services of all logically related equipment necessary to complete the major processing task(s) required of it. Units operate relatively independently of each other.

A unit frequently contains or operates on a complete batch of material at some point in the processing sequence of that batch. However, in other circumstances it may contain or operate on only a portion of a batch. This standard presumes that the unit does not operate on more than one batch at the same time.

4.2.6 Equipment module level

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.

An equipment module can carry out a finite number of specific minor processing activities such as dosing and weighing. It combines all necessary physical processing and control equipment required to perform those activities. It is usually centered on a piece of processing equipment, such as a filter. Functionally, the scope of the equipment module is defined by the finite tasks it is designed to carry out.

4.2.7 Control module level

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.

Some examples of control modules are

— a regulating device consisting of a transmitter, a controller, and a control valve that is operated via the set point of the device;

— a state-oriented device that consists of an on/off automatic block valve with position feedback switches, that is operated via the set point of the device; or

— a header that contains several on/off automatic block valves and that coordinates the valves to direct flow to one or several destinations based upon the set point directed to the header control module.

4.3 Process cell classification

This section discusses the classification of process cells by the number of different products manufactured in the process cell and by the physical structure of the equipment used in the manufacturing.

4.3.1 Classification by number of products

A process cell is classified as single-product or multi-product based on the number of products planned for production in that process cell.

A *single product* process cell produces the same product in each batch. Variations in procedures and parameters are possible. For example, variations may occur in order to compensate for differences in equipment, to compensate for substitute raw materials, to compensate for changes in environmental conditions, or to optimize the process.

A *multi-product* process cell produces different products utilizing different methods of production or control. There are two possibilities:

— All products are produced with the same procedure using different formula values (varying materials and/or process parameters).

— The products are produced using different procedures.

4.3.2 Classification by physical structure

The basic types of physical structures discussed here are *single path*, *multiple path*, and *network*.

A single-path structure is a group of units through which a batch passes sequentially (see Figure 3). A single-path structure could be a single unit, such as a reactor, or several units in sequence. Multiple input materials are typically used; multiple finished materials may be generated. Several batches may be in progress at the same time.

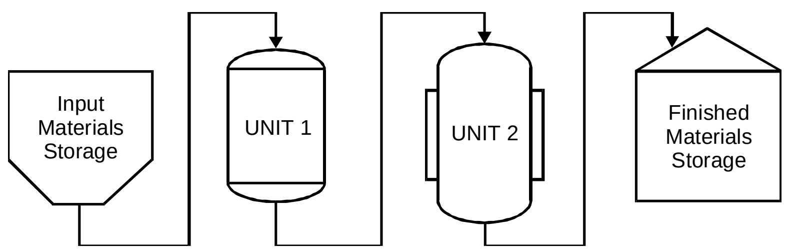


Figure 3 — Single-path structure

A *multiple-path* structure is shown in Figure 4. It consists of multiple single-path structures in parallel with no product transfer between them. The units may share raw material sources and product storage. Several batches may be in progress at the same time. Although units within a multi-path structure may be physically similar, it is possible to have paths and units within a multi-path structure that are of radically different physical design.

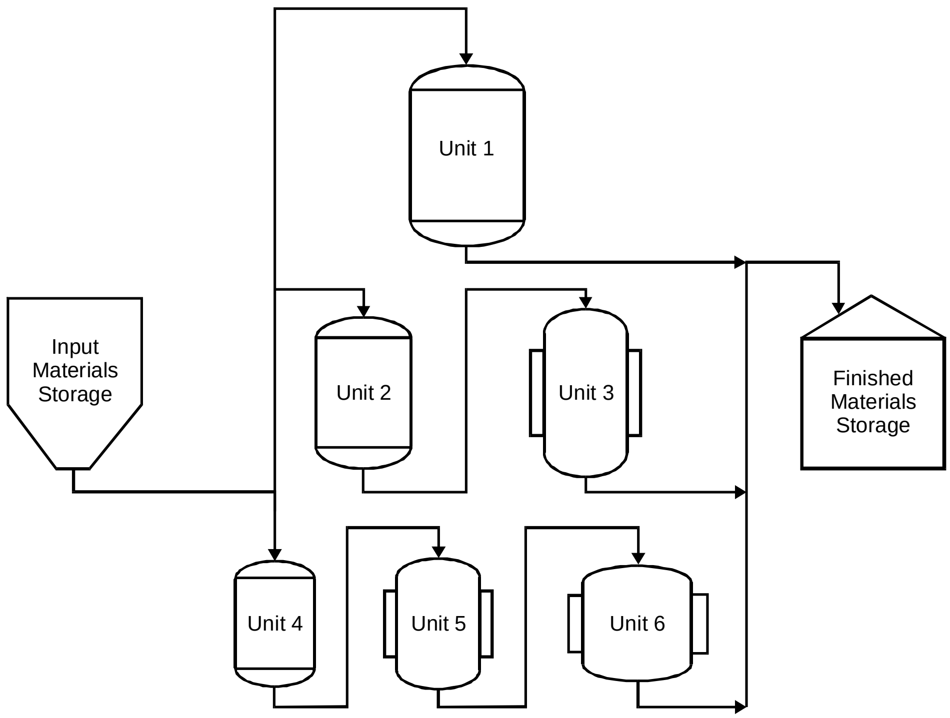


Figure 4 — Multiple-path structure

A *network* structure is shown in Figure 5. The paths may be either fixed or variable. When the paths are fixed, the same units are used in the same sequence. When the path is variable, the sequence may be determined at the beginning of the batch or it may be determined as the batch is being produced. The path could also be totally flexible. For example, a batch would not have to start at either Unit 1 or Unit 3; it could start with any unit and take multiple paths through the process cell. The units themselves may be portable within the process cell. In this case, verification of the process connections may be an important part of the procedures. Note that several batches may be in production at the same time. The units may share raw material sources and product storage.

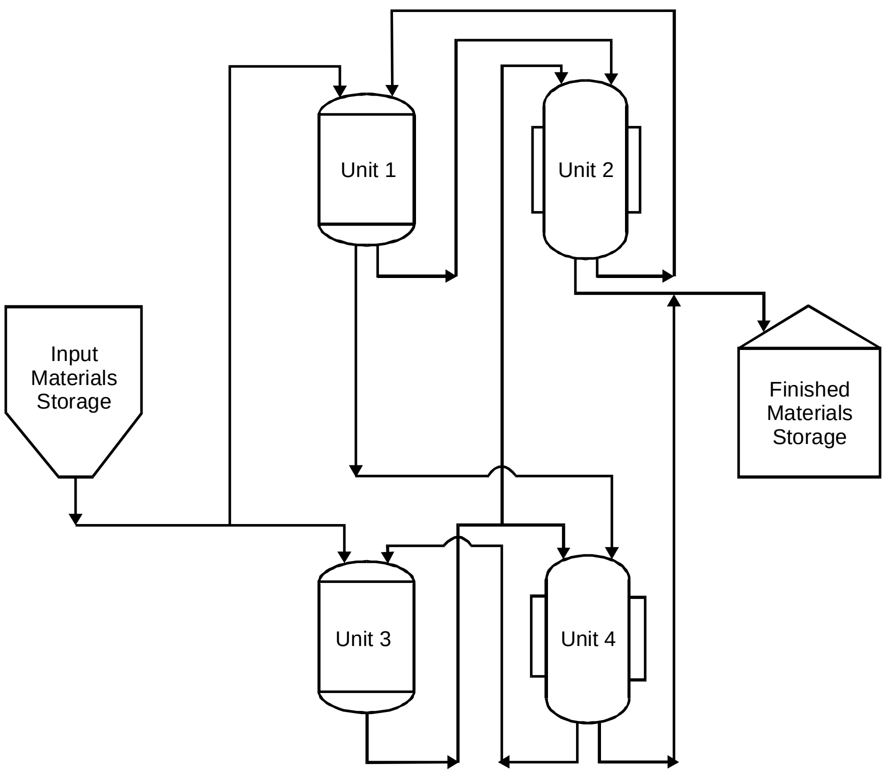


Figure 5 — Network structure

5 Batch control concepts

This section discusses the batch control concepts needed to address the batch processing/batch manufacturing needs presented in the preceding section and to define a consistent way of operating a batch manufacturing plant. A structure for batch control is discussed that introduces three types of control needed for batch manufacturing. When these control types are applied to equipment, the resulting equipment entities provide process functionality and control capability.

The concept of recipes is discussed, including the four types of recipes described in this standard and the contents of these recipes (in terms of the information categories used to describe a recipe). A relationship is established between the procedure in a recipe and the control associated with specific equipment entities (equipment control). The concept of collapsibility of the recipe procedure and of equipment control is discussed. Recipe transportability criteria are introduced for the four types of recipes.

Production plans and schedules, reference information, production information, allocation and arbitration, modes and states, and exception handling are other batch control concepts discussed in this section.

The intent of the models and terminology introduced in this section is to establish the necessary batch control understanding so that the control functions that are needed to address the diverse control requirements of batch manufacturing can be discussed in Section 6.

5.1 Structure for batch control

Section 4 introduced a physical model that defined terms for the hierarchy of equipment typically found in a batch manufacturing environment. This section describes the three types of control (basic control, procedural control, and coordination control) typically needed in batch manufacturing.

5.1.1 Basic control

Basic control comprises the control dedicated to establishing and maintaining a specific state of equipment and process. Basic control

— includes regulatory control, interlocking, monitoring, exception handling, and repetitive discrete or sequential control;

— may respond to process conditions that could influence the control outputs or trigger corrective actions;

— may be activated, deactivated, or modified by operator commands or by procedural or coordination control.

Basic control in a batch environment is in principle no different from the control of continuous processes. However, in the batch environment, there may be higher requirements on the ability for basic control to receive commands and to modify its behavior based on these commands.

5.1.2 Procedural control

Procedural control directs equipment-oriented actions to take place in an ordered sequence in order to carry out a process-oriented task.

Procedural control is a characteristic of batch processes. It is the control that enables equipment to perform a batch process.

Procedural control is made up of procedural elements that are combined in a hierarchical manner to accomplish the task of a complete process as defined by the process model. The hierarchy of identified and named procedural elements is illustrated in Figure 6 and consists of procedures, unit procedures, operations, and phases.

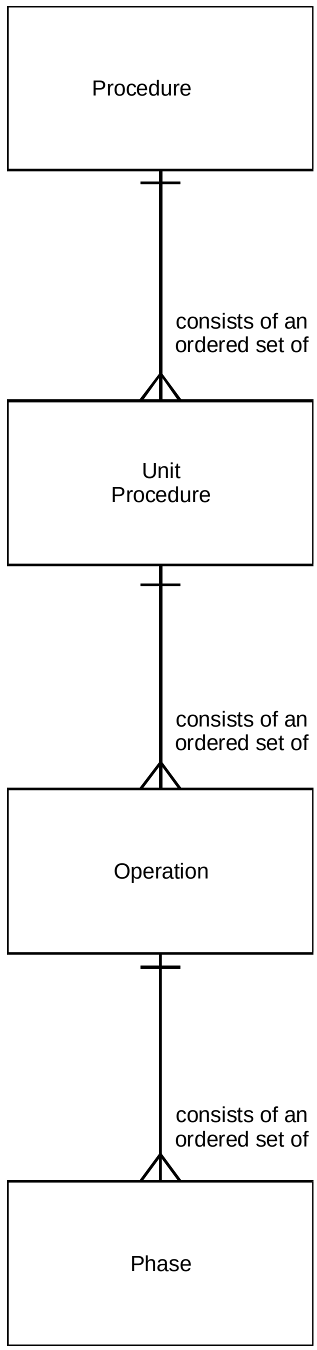


Figure 6 — Procedural control model

5.1.2.1 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."

5.1.2.2 Unit procedure

A unit procedure consists of an ordered set of operations that causes 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.

5.1.2.3 Operation

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.

5.1.2.4 Phase

The smallest element of procedural control that can accomplish a process-oriented task is a phase. A phase may be subdivided into smaller parts. The steps and transitions as described in IEC 848: 1988 document one method of defining subdivisions of a phase.

A phase can issue one or more commands or cause one or more actions, such as

— Enabling and disabling regulating and state-oriented types of basic control and specifying their set points and initial output values

— Setting, clearing, and changing alarm and other limits

— Setting and changing controller constants, controller modes, and types of algorithms

— Reading process variables, such as the gas density, gas temperature, and volumetric flow rate from a flowmeter, and calculating the mass flow rate through the flowmeter

— Conducting operator authorization checks.

The execution of a phase may result in

— commands to basic control;

— commands to other phases (either in the same or another equipment entity); and/or

— the collection of data.

The intent of the phase is to cause or define a process-oriented action, while the logic or set of steps that make up a phase are equipment specific. Examples of phases include the following:

— Add VCM.

— Add catalyst.

— Heat.

5.1.3 Coordination control

Coordination control directs, initiates, and/or modifies the execution of procedural control and the utilization of equipment entities. It is time varying in nature, like procedural control, but it is not structured along a specific process-oriented task.

Examples of coordination control are algorithms for

— supervising availability or capacity of equipment;

— allocating equipment to batches;

— arbitrating requests for allocation;

— coordinating common resource equipment;

— selecting procedural elements to be executed;

— propagating modes.

The control functions that are needed to implement coordination control are discussed in more detail in Section 6 under the topic of control activities.

5.2 Equipment entities

This section discusses equipment entities that are formed from the combination of equipment control and physical equipment. This combination results in four equipment entities: process cells, units, equipment modules, and control modules. Guidelines for structuring these equipment entities are also discussed.

When the terms process cell, unit, equipment module, and control module are used, they generally refer to the equipment and its associated equipment control. Whether equipment control in an equipment entity is implemented manually or by way of automation, it is only through the exercise of equipment control that the equipment can produce a batch.

The notion of equipment control being part of an equipment entity is to be understood logically. It is not a statement of the physical implementation of equipment control. However, it must be possible to identify equipment control for a particular equipment entity.

This interaction of equipment control and physical equipment is described purposely without any reference to language or implementation. The intent is to describe a framework within which equipment control and physical equipment may be defined and discussed.

5.2.1 Procedural control model/physical model/process model relationship

The general relationship between the procedural control model, the physical model, and the process model is illustrated in Figure 7. This mapping of procedural control with individual equipment provides processing functionality described in the process model.

The concept of equipment capabilities and usage of these capabilities to accomplish processing tasks is a major point of this standard. The procedural control capability of equipment entities is the mechanism that enables this. The procedural control may be entirely defined as part of equipment control, or it may be based on procedural information passed on to the equipment entity from the recipe.

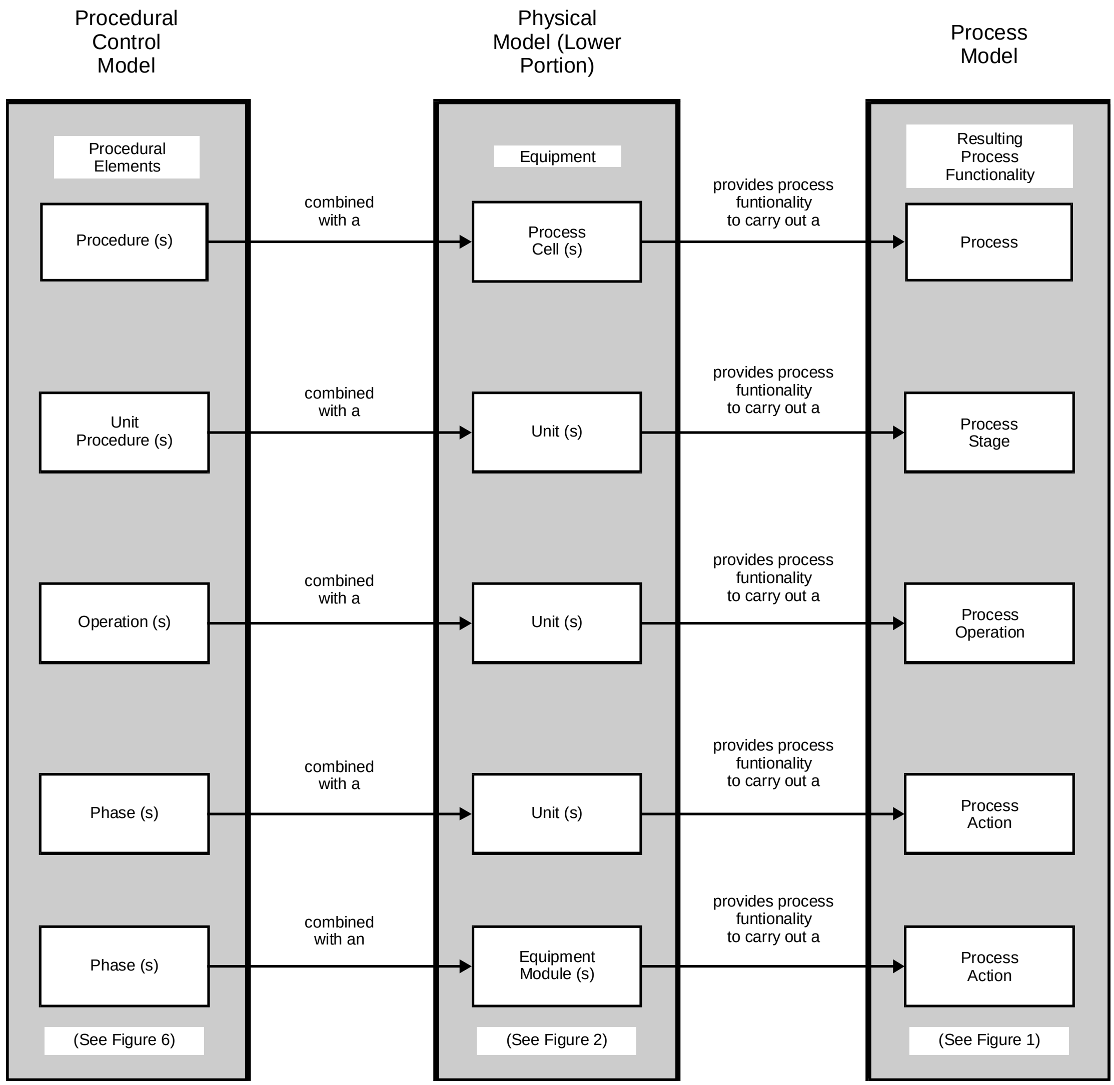


Figure 7 — Procedural control/equipment mapping to achieve process functionality

5.2.2 Equipment control in equipment entities

The control capability possible in the different equipment entities are important characteristics and a main basis for classification of equipment entities. In the following paragraphs equipment control for the individual equipment entities is discussed.

5.2.2.1 Process cell

The process cell is capable of orchestrating all processing activities for one or more batches. It receives recipes containing procedure, parameter, and other information and a schedule containing operational requirements for each batch. It may also need to prepare and monitor equipment or resources not currently involved in batch processing, such as which units are available, what units and piping are going through a clean-in-place (CIP) routine, and what the current inventories of raw materials are.

The complexity of control within the process cell will depend on the equipment available within the process cell, the interconnectivity among this equipment, the degree of freedom of movements of batches through this equipment, and the arbitration of the use of this equipment so that the equipment can be used most effectively.

Equipment control in the process cell may be distributed in the same manner as the physical equipment is subdivided. For example, if the process cell is subdivided into trains, equipment control within the process cell may be distributed among the various trains.

Equipment modules and control modules may exist as separate entities under direct control of the process cell.

5.2.2.1.1 Basic control in process cells

The process cell may include basic control that spans several units. For example, an interlock that shuts one unit down may need to be propagated to the upstream units that are feeding this particular unit.

5.2.2.1.2 Procedural control in process cells

The execution of a procedure and initiation of the individual unit procedures is a process cell responsibility. The execution may or may not be integral to the coordination control involved with the movement of batches as described in 5.2.2.1.3.

5.2.2.1.3 Coordination control in process cells

More coordination control is needed in process cells than in the lower level equipment entities because

- the process cell may contain multiple units and process multiple batches at the same time. This involves coordinating the execution of a number of different procedures;
- the control of the movement of batches may involve a number of choices between alternate paths. Although these choices may be made via links between units, the process cell may also have to determine the routing;
- arbitration may be needed at the process cell level to optimize the use of resources, such as shared resources and resources that must be reserved well in advance of the time actually needed.

Examples of coordination control in a process cell include algorithms that

- manage the initialization and movement of the batches being processed within the process cell; and

— initiate and/or associate unit procedures, parameters and other information in individual units in the proper order to cause them to process the product described by the unique combination of schedules and recipes.

5.2.2.2 Unit

Units coordinate the functions of the lower level entities, such as equipment modules and control modules. The primary purpose of equipment control in a unit is to control the processing of the batch that is currently associated with the unit.

5.2.2.2.1 Basic control in units

Basic control in a unit is generally performed by regulatory control and discrete control in equipment modules and control modules within the unit.

5.2.2.2.2 Procedural control in units

Units may include and execute equipment phases, equipment operations, and equipment unit procedures or they may execute recipe operations and recipe unit procedures passed on to it.

5.2.2.2.3 Coordination control in units

Equipment control in a unit will include a substantially higher level of coordination control than any of the lower level equipment entities. This may include, for example, algorithms that manage unit and acquired resources; arbitrate requests for services from other units or from the process cell; acquire the services of resources from outside the unit; and communicate with other equipment entities outside unit boundaries.

5.2.2.3 Equipment module

The primary purpose of equipment control in an equipment module is to coordinate the functions of other equipment modules and lower level control modules. An equipment module may be commanded by a process cell, a unit, an operator, or, in some cases, another equipment module.

5.2.2.3.1 Basic control in equipment modules

Basic control in an equipment module is generally performed by regulatory control and discrete control in control modules within the equipment module.

5.2.2.3.2 Procedural control in equipment modules

Equipment modules may execute equipment phases, but they do not have the capability of executing higher level procedural elements.

5.2.2.3.3 Coordination control in equipment modules

Coordination control in an equipment module includes coordination of its component parts and may include algorithms for propagating modes and for arbitrating requests from units.

5.2.2.4 Control module

Equipment control normally found at this level directly manipulates actuators and other control modules. A control module can direct commands to other control modules and to actuators if they have been configured as part of the control module. Control of the process is effected through the equipment specific manipulation of control modules and actuators.

Examples of equipment control in control modules include

— opening or closing a valve, with confirmation failure alarms;

— regulating the position of a control valve based on a sensor reading and PID control algorithm;

— setting and maintaining the state of several valves in a material header.

5.2.2.4.1 Basic control in control modules

Control modules contain basic control. Although this control is normally either regulatory or state oriented, in some cases it is both. It may also include conditional logic. For example, open the valve if the temperature is within limits and the downstream valve is open.

Regulatory control is dedicated to maintaining a process variable or variables at or near some desired value. Complex control strategies such as multivariable control, model-based control, and artificial intelligence techniques may also fit into this category of regulatory control.

State-oriented control refers to setting the state of a piece of equipment as opposed to the state of a process variable or variables. A state-oriented device has a finite number of states. It defines a product independent processing sequence.

Control modules may contain exception handling.

5.2.2.4.2 Procedural control in control modules

Control modules do not perform procedural control.

5.2.2.4.3 Coordination control in control modules

Coordination control in a control module may include, for example, algorithms for propagating modes and for arbitrating requests from units for usage.

5.2.3 Structuring of equipment entities

This section discusses the general principles involved in segmenting a process cell into equipment entities that can carry out specified processing activities or equipment-specific actions. Total explanation of process segmentation principles is beyond the scope of this standard.

It is important to note that the physical process cell design can greatly influence the implementation of batch control. Minor differences in the physical system can dramatically affect the organization of equipment entities and procedural elements.

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

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

5.2.3.1 Structuring of process cells

The subdivision of a process cell usually follows the principles listed below :

— The function any equipment entity serves in product processing must be clear and unambiguous.

— The function performed by the equipment entity must be consistent in terms of processing task, and should be usable for that task no matter what product is being manufactured at a given time.

— Subordinate equipment entities should be able to execute their task(s) independently and asynchronously, allowing the highest level equipment entity to orchestrate the activities of its subordinates.

— Interactions between equipment entities should be minimized. While planned interaction is periodically necessary, each equipment entity should perform its functions while influencing the functioning of other equipment entities as little as possible.

— Equipment entities must have clear boundaries.

— A consistent basis is required for the definition of equipment entities. An operator subsequently interacting with similar equipment entities should be able to do so naturally and without confusion.

— Necessary interaction between equipment entities is, insofar as possible, coordinated by equipment entities at the same level or at the next higher level.

5.2.3.2 Structuring of units

The definition of a unit requires knowledge of the major processing activities, as well as the equipment capabilities. The following guidelines apply:

— One or more major processing activities, such as reaction or crystallization, may take place in a unit.

— Units should be defined such that they operate relatively independently of each other.

— A unit is presumed to operate on only one batch at a time.

5.2.3.3 Structuring of equipment modules

The definition of an equipment module requires knowledge of specific minor processing activities and equipment capabilities. Equipment modules can carry out a finite number of minor processing activities, such as dosing and weighing, and are typically centered around a set of process equipment. Collections of control modules can be defined as equipment modules or as control modules. If the collection executes one or more equipment phases, then it is an equipment module.

5.3 Recipes

This section discusses the four types of recipes covered in this standard, the five categories of information contained in a recipe and how this information changes for the different recipe types, and the relationship of the control recipe procedure to the equipment procedure. Some guidelines for recipe transportability are also presented.

5.3.1 Recipe types

This section discusses four types of recipes typically found in an enterprise. A recipe is an entity that contains the minimum set of information that uniquely defines the manufacturing requirements for a specific product. Recipes provide a way to describe products and how those products are produced. Depending on the specific requirements of an enterprise, other recipe types may exist. However, this standard discusses only the general recipe, site recipe, master recipe, and control recipe (see Figure 8).

Fundamental to the practical application of recipes is the concept that different parts of an enterprise may need information about the manufacture of a product in varying degrees of specificity, because different recipients of the information use it for different purposes. Therefore, more than one type of a recipe is needed in an enterprise.

It should be noted that whether a particular recipe type actually exists, who generates it, and where it is generated will vary from case to case and from enterprise to enterprise. For example, an enterprise may choose not to implement one or more of the recipe types.

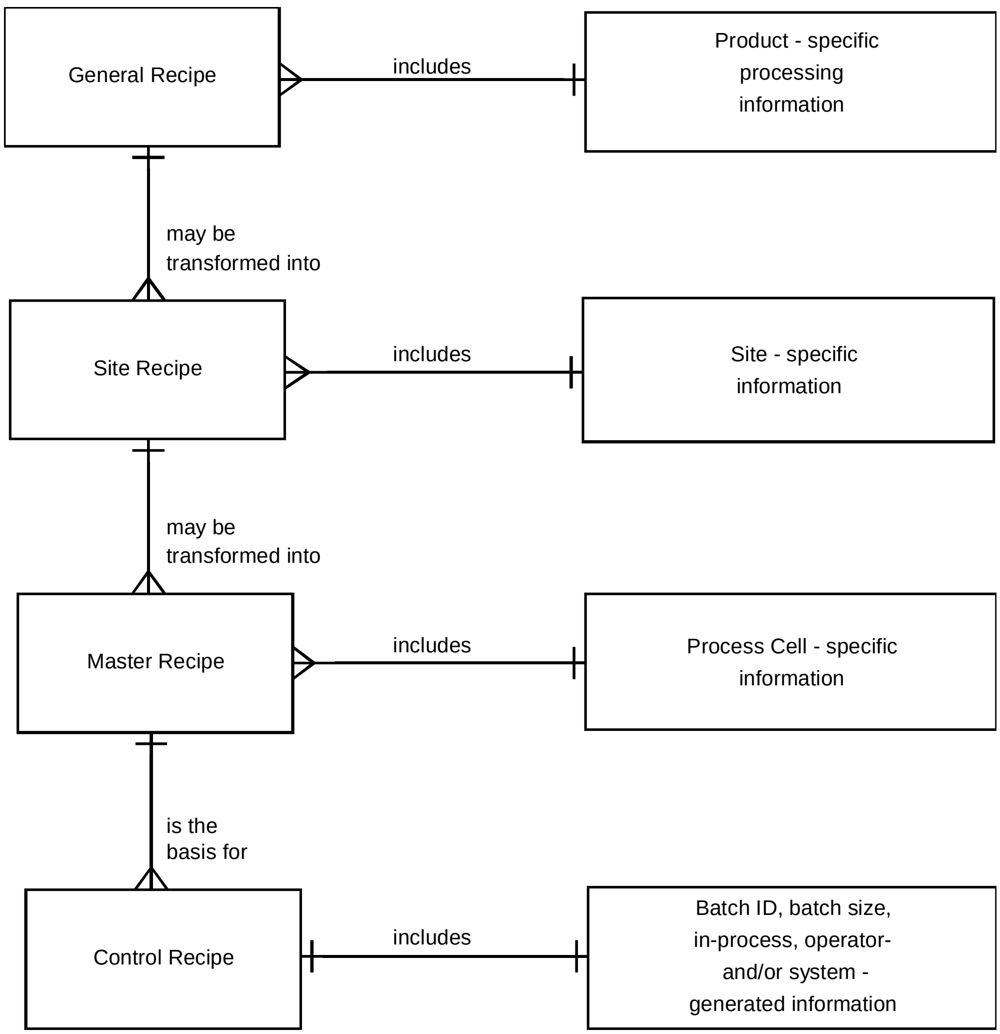


Figure 8 — Recipe types

A product may be made in many different arrangements of equipment at many different sites. Recipes that are appropriate for one site or set of equipment may not be appropriate for another site or set of equipment. This can result in multiple recipes for a single product. There should be sufficient structure in the definition of recipes to allow tracing of the genealogy of any given recipe.

The recipe contains neither scheduling nor equipment control. The recipe contains process-related information for a specific product. This permits batch processing equipment to make many different products without having to redefine equipment control for each product.

There is a substantial difference between general/site recipes and master/control recipes. The general and site recipes describe the technique, that is, how to do it in principle. Master and control recipes describe the task, that is, how to do it with actual resources.

5.3.1.1 General recipe

The general recipe is an enterprise level recipe that serves as the basis for lower-level recipes. The general recipe is created without specific knowledge of the process cell equipment that will be used to manufacture the product. It identifies raw materials, their relative quantities, and required processing, but without specific regard to a particular site or the equipment available at that site. It is created by people with knowledge of both the chemistry and processing requirements peculiar to the product in question, and reflects their interests and concerns.

While the general recipe is not specific to equipment or to a particular site, the technology for manufacturing a product will usually have evolved sufficiently beyond the laboratory so that equipment requirements can be described in enough detail to define the type of equipment needed at a particular site or in a particular set of batch plant equipment. The general recipe provides a means for communicating processing requirements to multiple manufacturing locations.

Quantities may be expressed as fixed or normalized values; equipment requirements are expressed in terms of the attributes needed by the equipment, such as pressure requirements and materials of construction.

The general recipe may be used as a basis for enterprise-wide planning and investment decisions. It may be part of, or referenced by production specifications and, as such, used for production planning and for information to customers and authorities.

5.3.1.2 Site recipe

The site recipe is specific to a particular site. It is the combination of site-specific information and a general recipe. It is usually derived from a general recipe to meet the conditions found at a particular manufacturing location and provides the level of detail necessary for site-level, long-term production scheduling. However, it may also be created directly without the existence of a general recipe. Such things as the language in which it is written or local raw material differences are accommodated as site-specific variances. It is still not specific to a particular set of process cell equipment. Typically, the site recipe is the output of a local "site focused" process development function.

There may be multiple site recipes derived from a general recipe, each covering a part of the general recipe that may be implemented at a specific site.

5.3.1.3 Master recipe

The master recipe is that level of recipe that is targeted to a process cell or a subset of the process cell equipment. A master recipe can be derived from a general recipe or a site recipe. It can also be created as a stand-alone entity if the recipe creator has the necessary process and product knowledge.

Some characteristics of master recipes include the following:

— There may be multiple master recipes derived from a site recipe, each covering a part of the site recipe that may be implemented in a process cell.

— The master recipe has to be sufficiently adapted to the properties of the process cell equipment to ensure the correct processing of the batch. This is done by combining the functionality of the specific set of process cell equipment with the information from the master recipe.

— In a master recipe, the formula data may be specified as normalized values, calculated values, or fixed values.

— The master recipe may contain product-specific information required for detailed scheduling, such as process input information or equipment requirements.

— The master recipe level is a required recipe level, because without it no control recipes can be created and, therefore, no batches can be produced.

— Whether the batch manufacturing equipment is operated manually or fully automatically, the master recipe exists either as an identifiable set of written instructions or as an electronic entity.

5.3.1.4 Control recipe

The control recipe starts as a copy of a specific version of a master recipe and is then modified as necessary with scheduling and operational information to be specific to a single batch. It contains product-specific process information necessary to manufacture a particular batch of product. It provides the level of detail necessary to initiate and monitor equipment procedural entities in a process cell. It may have been modified to account for actual raw material qualities and actual equipment to be utilized. The selection of units and appropriate sizing can be done any time before that information is needed.

Since modifications of a control recipe can be made over a period of time based on scheduling, equipment, and operator information, a control recipe may go through several modifications during the batch processing. Examples include

— defining the equipment that will actually be used for the control recipe at the initiation of the batch or when it becomes known;

— adding or adjusting parameters based on an "as-charged" raw material quality or mid-batch analysis;

— changing the procedure based on some unexpected event.

5.3.2 Recipe contents

Recipes contain the following categories of information: header, formula, equipment requirements, procedure, and other information. The following subparagraphs provide details regarding these categories. Any significant changes from one recipe type to another are noted.

5.3.2.1 Header

The administrative information in the recipe is referred to as the header. Typical header information may include the recipe and product identification, the version number, the originator, the issue date, approvals, status, and other administrative information. For example, a site recipe may contain the name and version of the general recipe from which it was created.

5.3.2.2 Formula

The formula is a category of recipe information that includes process inputs, process parameters, and process outputs.

A process input is the identification and quantity of a raw material or other resource required to make the product. In addition to raw materials which are consumed in the batch process in the manufacture of a product, process inputs may also include energy and other resources such as manpower. Process inputs consists of both the name of the resource and the amount required to make a specific quantity of finished product. Quantities may be specified as absolute values or as equations based upon other formula parameters or the batch or equipment size. Process inputs may specify allowable substitutions, expressed in the same basic form.

A process parameter details information such as temperature, pressure, or time that is pertinent to the product but does not fall into the classification of input or output. Process parameters may be used as set points, comparison values, or in conditional logic.

A process output is the identification and quantity of a material and/or energy expected to result from one execution of the recipe. This data may detail environmental impact and may also contain other information such as specification of the intended outputs in terms of quantity, labeling, and yield.

The types of formula data are distinguished to provide information to different parts of an enterprise and need to be available without the clutter of processing details. For example, the list of process inputs may be presented as a condensed list of ingredients for the recipe or as a set of individual ingredients for each appropriate procedural element in a recipe.

5.3.2.3 Equipment requirements

Equipment requirements constrain the choice of the equipment that will eventually be used to implement a specific part of the procedure.

In the general and site recipes, the equipment requirements are typically described in general terms, such as allowable materials and required processing characteristics. It is the guidance from and constraints imposed by equipment requirements that will allow the general or site recipe to eventually be used to create a master recipe which targets appropriate equipment. At the master recipe level, the equipment requirements may be expressed in any manner that specifies allowable equipment in process cells. If trains have been defined, then it is possible for the master recipe (and the resulting control recipe) to be based on the equipment of the train rather than the full range of equipment in the process cell. At the control recipe level, the equipment requirements are the same as, or a subset of, the allowable equipment in the master recipe. The control recipe may be used to include specific allocations of process cell equipment, such as Reactor R-501, when this becomes known.

5.3.2.4 Recipe procedure

The recipe procedure defines the strategy for carrying out a process. The general and site recipe procedures are structured using the levels described in the process model since these levels allow the process to be described in non-equipment specific terms. The master and control recipe procedures are structured using the procedural elements of the procedural control model, since these procedural elements have a relationship to equipment.

The recipe creator is limited to the use of procedural elements that have been, or will be, configured and made available for use in creating a procedure. He or she may use any combination of these procedural elements to define a procedure. Determination of which of these procedural elements may be part of the procedure is an application specific design decision based on many factors including the capabilities of the controls and the degrees of freedom appropriate for the recipe creator in a given application.

5.3.2.4.1 General recipe procedure

The procedure information in the general recipe is expressed in three levels of breakdown: Process Stages, Process Operations, and Process Actions (see Figure 9). The functionality of these levels corresponds to the functionality of the analogous levels in the Process Model (see 4.1.3).

The process stage, process operation, and process action are not constrained by unit boundaries in any real plant. They describe processing activities that others may choose to execute in one or in many different units as the general and site recipe is transformed to run in one or more real plants.

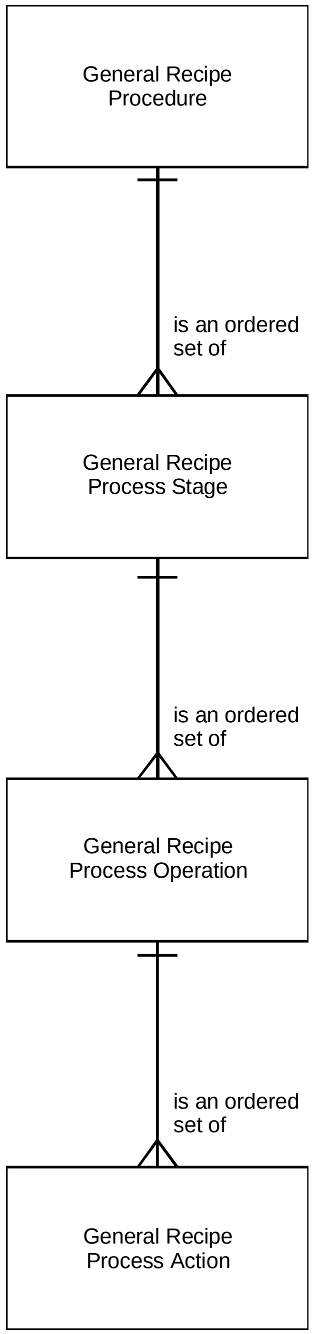


Figure 9 — General recipe procedure

5.3.2.4.2 Site recipe procedure

The procedure information in a site recipe consists of process stages, process operations, and process actions that relate directly to those defined by the general recipe. In general, there is a 1:1 correspondence between the process stages in a general recipe and the process stages in a site recipe, between the process operations in a general recipe and the process operations in a site recipe, and between the process actions in a general recipe and the process actions in a site recipe. As with the other site recipe information, the process stages, process operations, and process actions may be modified to make the recipe site-specific.

5.3.2.4.3 Master recipe procedure

The recipe procedure portion of the master recipe may contain recipe unit procedures, recipe operations, and recipe phases (see Figure 10).

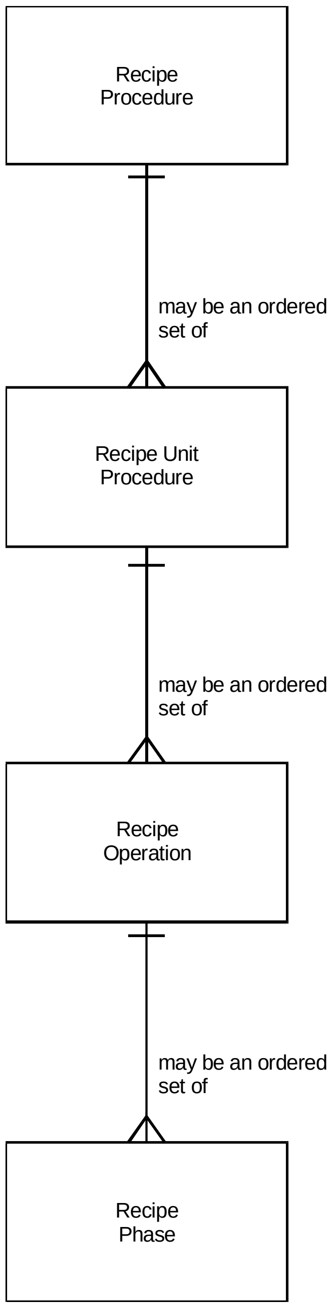


Figure 10 — Master recipe procedure

The creation of a procedure in a master recipe from a procedure in a site recipe may be quite complex. The master recipe must contain sufficiently detailed equipment requirements information so that resources may be determined and allocated to create and initiate a control recipe. It is at this recipe level that the set of recipe phases necessary to carry out the intended process actions, process operations, and process stages can be determined.

There may be a 1:1, 1:n, or n:1 relationship between process actions in the general or site recipe and recipe phases in the master recipe, between process operations in the general or site recipe and recipe operations in the master recipe, and between process stages in the general or site recipe and recipe unit procedures in the master recipe (see Figure 11). The actual relationship may depend on the equipment being used.

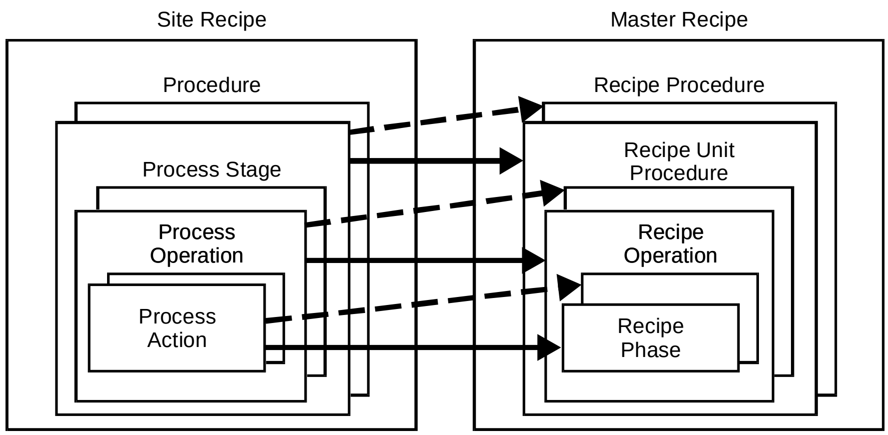


Figure 11 — Procedural element relationships in the site recipe and master recipe

Although there is a general similarity between the processing intent of process actions and the processing function defined by recipe phases, there is not necessarily a one-to-one correspondence between the two. One process action may correspond to several recipe phases, and several process actions may correspond to a single recipe phase.

There is a similar relationship between process operations and operations. There are significant differences also. Operations are carried to completion in a single unit in the target equipment while process operations are not constrained to units in any specific facility. A single process operation might require one or more operations to carry out the processing intent described.

There is a similar relationship between process stages and unit procedures as there is between process operations and operations. Unit procedures are also carried to completion in a single unit in the target equipment while process stages are not constrained by equipment boundaries in any specific facility. A single process stage might require one or more unit procedures to carry out the processing intent described.

5.3.2.4.4 Control recipe procedure

The procedure of a control recipe consists of recipe unit procedures, recipe operations and recipe phases that relate directly to those defined by the master recipe. At the control recipe creation time, there is a 1:1 correspondence between recipe unit procedures in the master recipe and recipe unit procedures in the control recipe, between recipe operations in the master recipe and recipe operations in the control recipe, and between recipe phases in the master recipe and recipe phases in the control recipe. Changes in the control recipe procedure during the execution may cause it to differ from the master recipe procedure. In a control recipe, as in a master recipe, the procedure is divided along unit procedure boundaries to provide the process cell with the processing requirements of the recipe on a unit-by-unit basis.

5.3.2.5 Other information

Other information is a category of recipe information that may contain batch processing support information not contained in other parts of the recipe. Examples include regulatory compliance information, materials and process safety information, process flow diagrams, and packaging/ labeling information.

5.3.3 Control recipe procedure/equipment control relationship

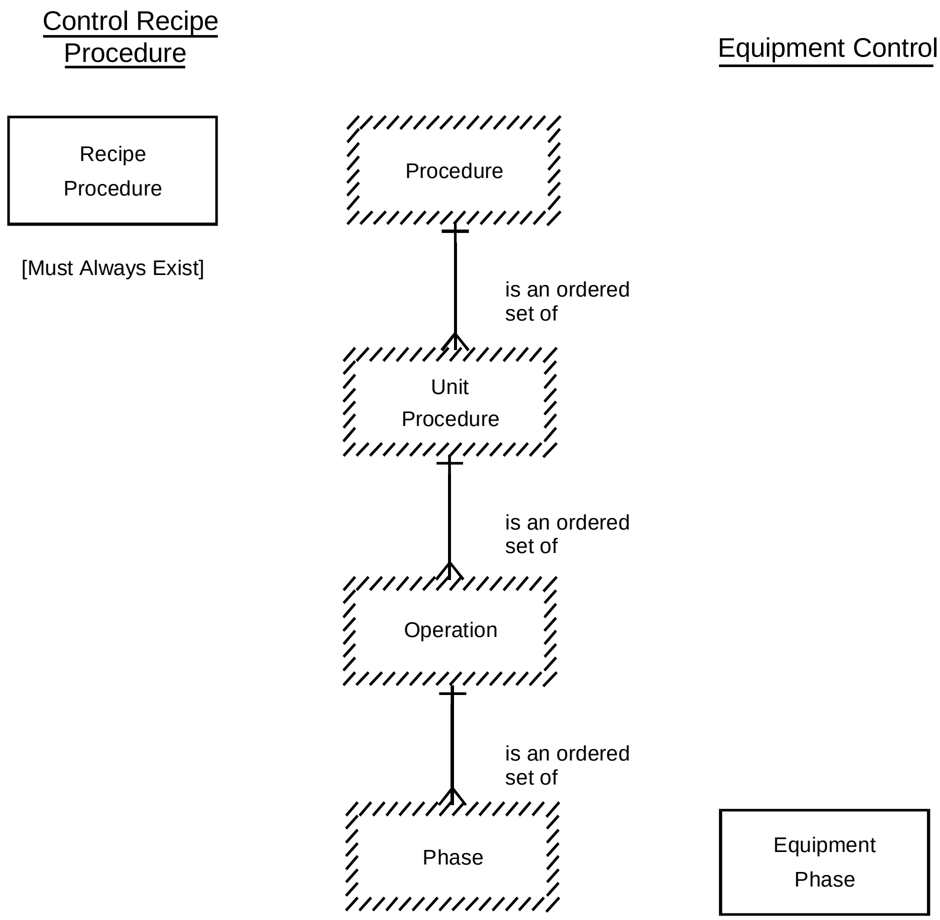
The control recipe itself does not contain enough information to operate the process cell. It must be linked to equipment control that actually causes the equipment to operate and make batches. Equipment control is not considered to be part of the recipe. This section discusses the separation between the control recipe procedure and equipment control, the procedural elements that are used in the control recipe procedure and in equipment control, and the mechanism that is used to link the control recipe procedure with equipment control.

5.3.3.1 Control recipe procedure/equipment control separation

Figure 12 shows the separation between the control recipe procedure and equipment control. The control recipe procedure must contain at least one procedural element, which is the recipe procedure. Equipment control must also contain at least one procedural element that provides the linkage needed to operate the physical equipment. For the example described in Figure 12, this is assumed to be the equipment phase.

The control recipe procedure might not include recipe unit procedures, recipe operations, and recipe phases. Such a recipe procedure must then be linked (by reference) to an equipment procedure in equipment control if batches are to be executed. Whenever a procedural element, such as a recipe procedure, recipe unit procedure, recipe operation, or recipe phase, is linked to equipment control, it must exist as that recipe procedural element (such as a recipe operation) and as that equipment procedural element (such as an equipment operation). Whenever recipe phases are used in the control recipe procedure, recipe phases are linked to equipment phases.

When recipe unit procedures, recipe operations, and recipe phases are not used as part of the control recipe procedure, it may still be helpful to use lower level equipment procedural elements (some or all) as part of equipment control to provide a modular structure to the equipment control.



NOTE – The boxes with slashed lines for borders are highlighted to point out that these procedural elements may be part of either the control recipe procedure or equipment control.

Figure 12 — Control recipe procedure/equipment control separation

5.3.3.2 Control recipe/equipment procedural elements

The following are typically associated with recipe procedural elements:

— A description of the functionality required

— Formula and other parameter information specific to the procedural element

— Equipment requirements specific to the procedural element

In order for a recipe procedural element to be able to reference an equipment procedural element, it must have an identification that enables the element to be correctly linked. In other cases, it must reference or include other recipe procedural elements and a specification of the execution order of those procedural elements.

The equipment procedural element to be linked typically has the following:

— An identification that can be referenced by the recipe procedural element or a higher level equipment procedural element

— A description of the functionality provided

— Variables that can receive the formula and other parameter information from the recipe

— Execution logic

It is possible for the recipe creator to work with a higher level procedural element for defining the procedure and still have the lower level procedural elements as part of the procedure. This could occur when the higher level procedural element has been pre-configured in terms of one or more lower level procedural elements. When the recipe creator invokes the use of a higher level procedural element, the lower level procedural elements are carried along, even though they may be invisible to the recipe creator, and become part of the procedure.

When a procedural element is used more than once in a recipe, there may be a need to uniquely identify each occurrence of the procedural element to the operator and batch history.

5.3.3.3 Control recipe procedure/equipment control linking

There must be some method to link the control recipe procedural elements with the equipment procedural elements. The examples below demonstrate this linkage between the control recipe procedure and equipment control when all procedural elements from the procedural control model are used in the application.

This linking is done by associating the recipe procedural elements with equipment procedural elements. In this way, the call for a certain processing function is separated from equipment control. It also enables the same recipe procedural element to use different equipment procedural elements, depending on what equipment the recipe addresses.

An equipment phase may be initiated by things other than the execution of a control recipe. It may be initiated by the request of another unit or on the request of an operator. The independent execution of a phase may be useful for handling exception conditions, during start-up or maintenance, and/or to prepare a unit for production.

If unit procedures, operations, and phases are part of the control recipe procedure, linking (by reference) of the control recipe procedure to equipment control is done at the phase level (see Figure 13). This drawing applies to one control recipe.

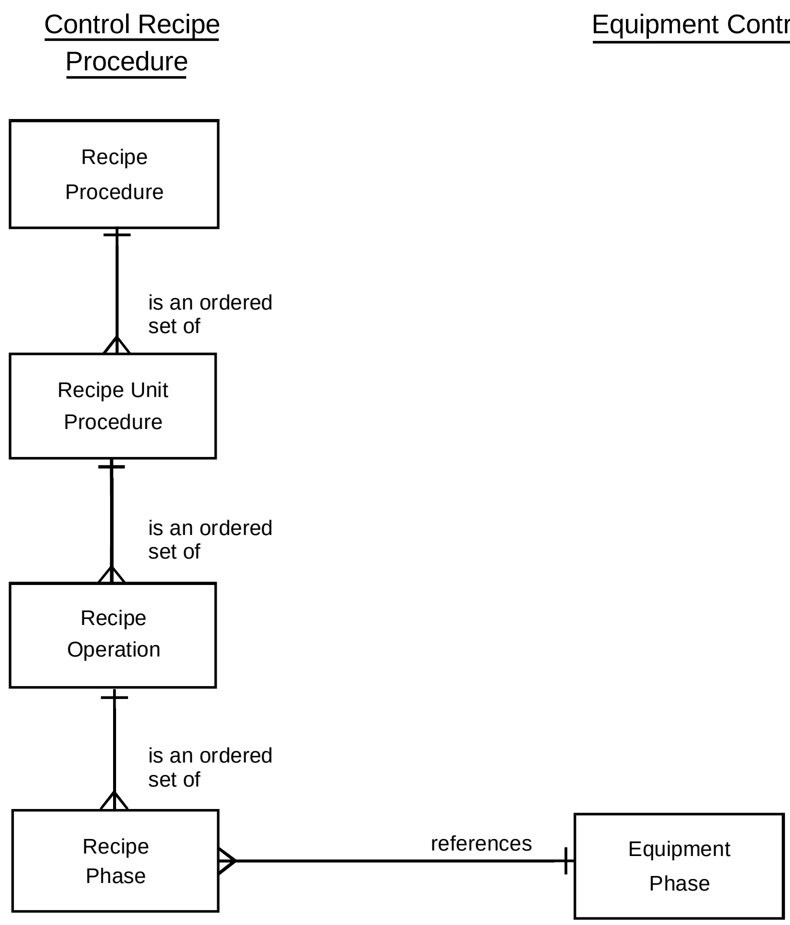


Figure 13 — Control recipe procedure example with unit procedures, operations, and phases

If phases do not exist as part of the control recipe but operations do, the linking would be done at the operation level (see Figure 14). This example applies to one control recipe.

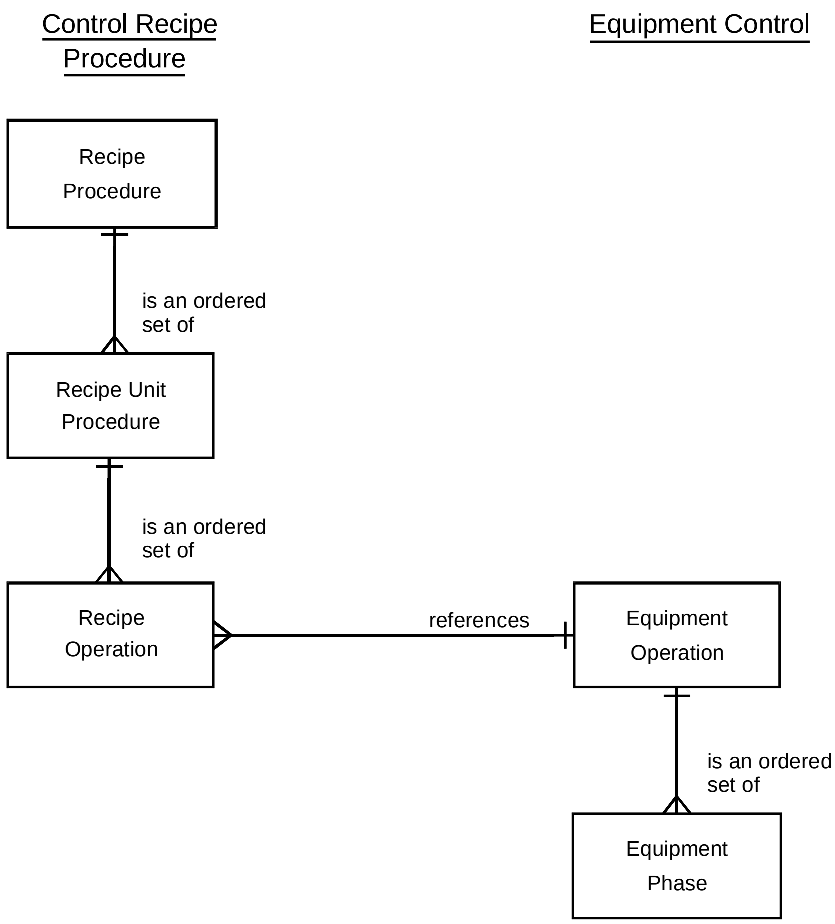


Figure 14 — Control recipe procedure example with unit procedures and operations

If neither phases nor operations exist as part of the control recipe but unit procedures do, the linking would be done at the unit procedure level (see Figure 15). This example applies to one control recipe.

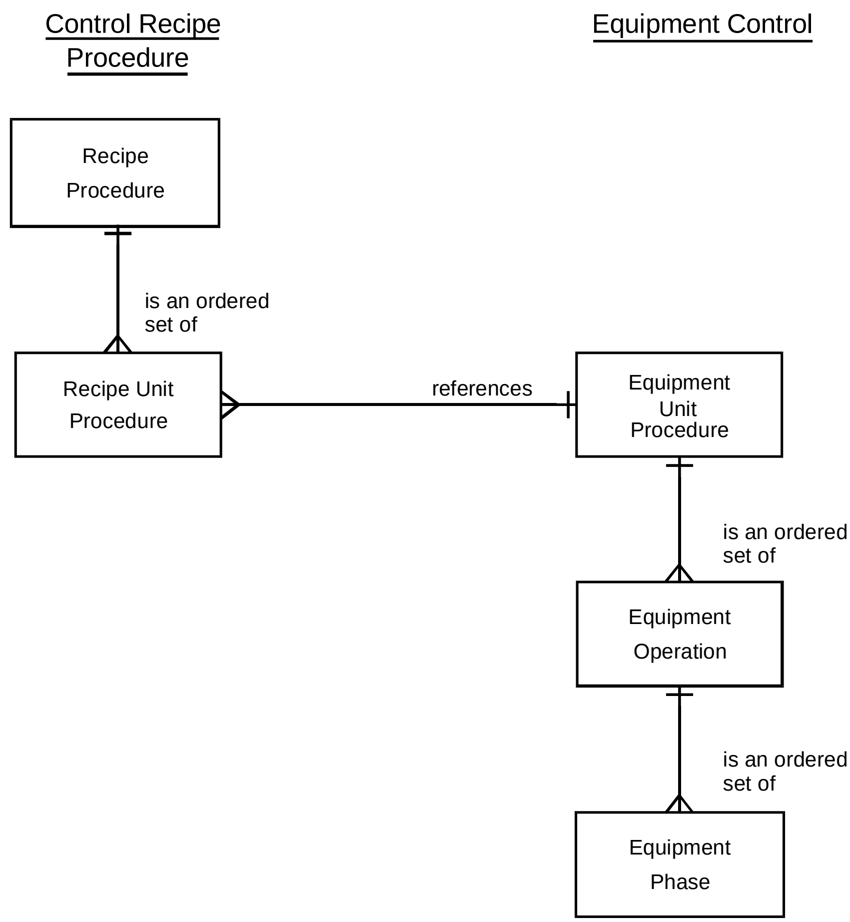


Figure 15 — Control recipe procedure example with unit procedures

If only the procedure exists as part of the control recipe, the linking would be done at the procedure level (see Figure 16). This example applies to one control recipe.

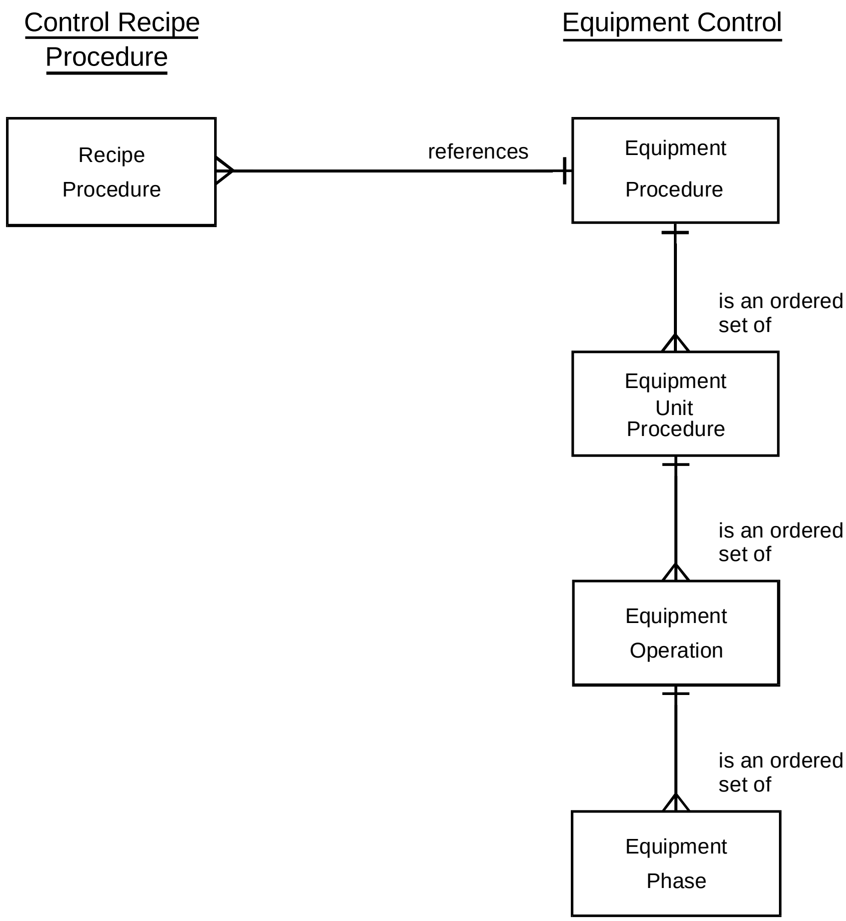


Figure 16 — Control recipe procedure example with only a procedure

5.3.3.4 Control recipe procedure/equipment control collapsibility

The preceding examples have assumed that all levels of the procedure model are being used. As with other models of this standard, the procedural control model is collapsible. Levels in the procedural control model may be left out in a specific application. Some examples are discussed below.

If a procedure addresses a single unit, the procedure itself may take the place of the unit procedure, and the recipe procedure has collapsed (see Figure 17a).

Recipe phases alone might be used to define a recipe procedure that addresses a single unit. Then the recipe procedure consists of the phases needed to accomplish the function of the procedure and the strategy needed to organize and properly sequence the phases. The procedure model is collapsed to eliminate the use of unit procedures and operations as overtly stated subdivisions (see Figure 17b).

The same collapsing may happen with an equipment procedure, as shown in Figure 17c, where no unit procedures, operations, or phases are used in the recipe procedure, such as when the recipe procedure has been collapsed to just the procedure name, and no unit procedures or operations are used in the equipment procedure. So now the equipment procedure is made up of an ordered set of equipment phases.

The phase level may be omitted if a specific application is better described with operations that are not further subdivided. Then the operation interacts directly with basic control (see Figure 17d).

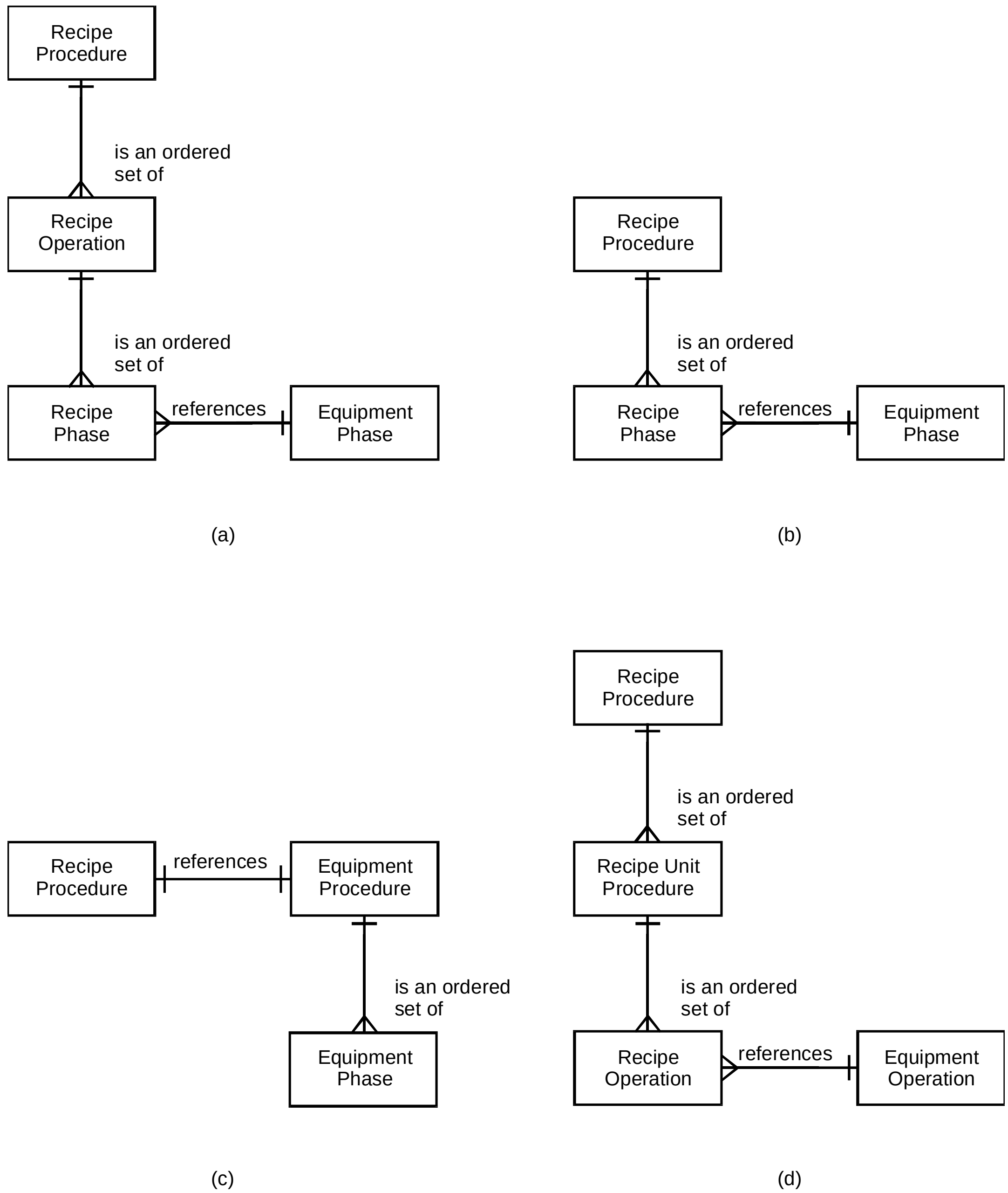


Figure 17 — Control recipe procedure/equipment control collapsibility examples

The following considerations must be taken into account when collapsing:

— When a procedural element level is taken out, the next higher level must take over its functions and contain the ordering logic controlling the next lower level and any other information that would have been stated in the collapsed level, including equipment requirements and other information.

— The lowest level of equipment procedural control must have the functionality to activate equipment through basic control.

5.3.4 Recipe transportability

Recipe transportability ensures that recipe information movement is possible between batch control implementations at the same recipe level. The recipe information must be understood by each implementation.

The general recipe is transportable from where it was created to any site. The site recipe is transportable, but not to the same extent as the general recipe. It is intended to be used within a specific site and is transportable within that site.

The master recipe is transportable to another process cell, recognizing that the master recipe has been customized for a particular set of process cell equipment. When the master recipe is transported to another process cell, some process engineering analysis may be necessary to

— determine that the new set of process cell equipment is configured similarly so that the master recipe can be used unchanged or;

— make the necessary changes so that the modified master recipe will run on the target set of process cell equipment.

The control recipe is not transportable.

5.4 Production plans and schedules

Production plans and schedules state the production requirements for the enterprise, sites, areas, and process cells. Since these levels of the physical model operate on different time horizons, a number of different types of plans and schedules are typically needed within an enterprise. A detailed discussion of the various types of plans and schedules is outside the scope of this standard. Only the scheduling needs at the process cell level, the batch schedule, will be discussed.

The batch schedule typically contains more detailed information than production plans and schedules aimed at higher levels in the enterprise. It contains information such as the products that are to be produced, how much of each product is to be produced, and when they are required for a specific process cell. It identifies which batches are to be made, their order, and the equipment to be used. This schedule also deals with issues such as personnel requirements, raw material options, and packaging requirements.

Time horizons for the batch schedule are dependent on the speed of the processes and might be measured in minutes, hours, shifts, or days. The batch schedule is based on the specific resources and requirements of the process cell. The possible paths and equipment options are determined at this point. For the batch schedule to be totally meaningful, this schedule would need to be redone any time there is significant variance from the time projections, resource assumptions, or other anticipated factors on which the schedule was based. For example, the schedule may have to be updated if an activity is not completed close to the scheduled time. Whether that activity is delayed or whether it is completed ahead of time, the primary concern is whether that activity can affect other schedules in this process cell or other associated process cells.

The following is the typical information found in a batch schedule:

- — Product name
- — Master recipe name
- — Quantity (with engineering units) of product
- — Equipment and materials permitted to be used, such as path and raw material
- — Projected mode of operation
- — Order of initiation and priority
- — Lot ID (if preassigned)
- — Batch ID (if preassigned)
- — Projected start time and end time
- — Disposition of the finished batch
- — Specific customer requirements

A key to efficient batch manufacturing is a comprehensive method that links the various plans and schedules with batch data collection. Batch data collection is the source of timely information that provides feedback so that these plans and schedules can be fine tuned. During the actual manufacturing of a batch, information is needed in real time so that schedules can be updated within a short time horizon. This update information also allows the user to be kept apprised of the status of lots and/or batches in the schedule.

5.5 Production information

This section discusses information that is generated in the course of production. Information needs to be collected and made available to various levels of the enterprise. The type of information needed varies between different parts of the enterprise. At the enterprise level, for example, summary information may be all that is needed. Examples include the amount of production of a particular product that was achieved at a specific site or at all sites, or how much product is available in inventory.

Process development may need detailed processing information on the individual batches in order to perform statistics and comparisons. At the process cell level where the batches are actually executed, there is a need for more detailed information in order to monitor the day-to-day production, to perform adjustments to the schedule, or to adjust processing from batch to batch.

Production information may be batch specific or it may be common to several or all of the batches produced.

5.5.1 Batch-specific information

The batch-specific information may include the following:

- — A copy of the control recipe that was used to make the batch. This may not be identical to the original recipe because of operator changes, equipment problems, etc. It may be desirable to record both the original recipe and the actual recipe.

— Recipe data. This is actual process data that corresponds exactly to the recipe formula, such as the amount and type of material charged. This can then be compared to the original recipe.

— Recipe-specified data. This is data whose collection is specified by the recipe. An example is process control information to be trended.

— Summary batch data. This is data such as utilities consumption, equipment run times, and temperatures for the entire batch.

— Operator comments

— Continuous data. This is process data that is collected independent of specific events within the batch with the purpose of giving an accurate history of that measurement.

— Event data. This is data from predictable and unpredictable events, such as recording start and stop times of procedural elements, or unpredictable process or equipment events.

— Operator data. This includes any operator intervention that may affect the processing of the batch (includes operator's ID).

— Analysis data. This is data that is related to off-line measurements or analyses such as measured variables, operator ID, lab technician ID, time of entry of results, and time of sample.

5.5.2 Common (non-batch specific) batch information

Examples of common (non-batch specific) batch information include:

— Quality control information. This is information related to monitoring raw material qualities and processing quality.

— Utility systems information. This is process information for equipment such as process heating and cooling that do not produce batches themselves but support equipment that does produce batches.

— Equipment history. This is historical information, such as equipment utilization, calibration, and maintenance.

— Operational documentation. This includes documentation such as production volumes, material consumption summaries, and inventory statistics.

— Materials information. This is typically information such as quality information and packaging and labeling information of input and output materials.

5.5.3 Batch history

All recorded information pertaining to a batch is referred to as the batch history. The batch history will typically include the batch-specific information. Common (non-batch specific) batch information may be included in the batch history. Since information of this nature typically applies to all or several batches being processed in a process cell, it may be included in the individual batch histories by reference.

In many regulated industries, the record of the batch history is as important as the product itself. Without reliable and accurate batch record keeping, product quality and traceability cannot be ensured. Complete batch record keeping also provides information that is invaluable in process analysis and continuous improvement efforts.

Batch history must be stored in a way that makes it possible to associate the data with that batch (or batches) to which it relates and the processing that has taken place. This means that, in addition to the specific batch identity, the data must be associated with the actual execution of the appropriate procedural elements, where relevant. The structure of the executed procedure may differ from what is specified in the original recipe because of operator intervention, exception handling, or even planned diversity in the procedure, such as changes caused by varying resource limitations.

5.5.4 Batch reports

The extraction of data related to one or more batches is called a batch report. The extraction and ordering of the data in a report may vary based on the intended recipient of the batch report. Some of the typical recipients of batch reports and the types of information typically included in their reports are

— production management: These batch reports typically provide key economic information on the processing result and resource utilization from multiple batches.

— product development: These batch reports typically include detailed process information for an individual batch or compare similar data between a group of batches.

— plant operations: These batch reports typically include the data collected to the current point of processing.

— quality management: These batch reports typically contain information for documenting batch quality, which may be useful in quality statistics.

— authorities: These batch reports are typically provided as documentation of production complying with regulations.

— customers: These batch reports usually are documentation of product quality and process uniformity.

5.6 Allocation and arbitration

This section discusses mechanisms for allocating resources to a batch or unit and for arbitrating the use of common resources when more than one requester needs to use a common resource at the same time.

Resources such as equipment are assigned to a batch or a unit as they are needed to complete or to continue required processing. *Allocation* is a form of coordination control that makes these assignments. When more than one candidate for allocation exists, a selection algorithm such as "select lowest duty time" might be used as a basis for choosing the resource. When more than one request for a single resource is made, *arbitration* is needed to determine which requester will be granted the resource. An algorithm such as "first come/first served" might be used as a basis for arbitration.

In the following sections, allocation and arbitration are discussed in terms of equipment. The concepts apply equally well to other resources, such as operators.

5.6.1 Allocation

The very nature of batch processing requires that many asynchronous activities take place in relative isolation from each other with periodic points of synchronization. Many factors, both expected and unexpected, can affect the time required by one or more of the asynchronous activities from one point of synchronization to the next. For those reasons, and because of the inherent variation in any manufacturing process, the exact equipment which will be available at the time it is needed is very difficult to predict over a significant period of time. Even though a schedule may have been planned to totally optimize the processing sequence from the standpoint of equipment utilization, it often is desirable to allow alternate equipment to be used if the units planned for a batch are not available when planned. In this case the allocation of units to the batch -- the routing or *path* of the batch -- is a decision which must be made every time there is more than one path the batch can take through the available equipment.

If more than one unit can acquire or request the services of a single resource, the resource is designated as a common resource. Common resources are often present with complex batch processes. Common resources are often implemented as either equipment modules or control modules. A common resource may be either exclusive-use or shared-use.

If the resource is designated as exclusive-use, only one unit may use the resource at a time. A shared weigh tank in a batch plant might be an example of an exclusive-use resource. It can be used by only one reactor at a time. The schedule or some other basis for allocation must take this exclusive-use resource into consideration. If a reactor is waiting for the use of the weigh tank while another is using it, the waiting reactor is idle and is not making product, which has a negative effect on equipment utilization.

If the common resource is designated as shared, several units may use the resource at the same time. Some shared-use resources in a batch plant might be a process heater serving multiple units at the same time or a raw material distribution system which is capable of delivering material to more than one unit at a time. If the capabilities of a shared-use resource are limited, then it is possible that the requests for service might exceed the capacity of the resource. In that case some of the same concerns about allocation which apply to exclusive-use resources also apply to shared-use resources. Care must also be taken so that one unit does not improperly shut off or deactivate a resource while other units are using it.

5.6.2 Arbitration

If there are multiple requesters for a resource, arbitration is required so that proper allocations can be made. Arbitration resolves contention for a resource according to some predetermined algorithm and provides definitive routing or allocation direction. The algorithm may take various forms such as a predetermined schedule with reservations, a batch priority scheme, or it might rely upon operator judgment. Arbitration may bring with it two distinct issues which affect complexity, resource reservation and preemption.

Reservation allows a claim to be placed on a resource prior to actual allocation. Reservation allows arbitration to be based on future needs rather than allowing the first request for allocation of an idle resource to take precedence regardless of priority. Preemption occurs when a higher priority batch is allowed to cancel or interrupt the use of a resource assigned to a lower priority batch. When allowed, it is most often associated with allocation of exclusive-use common resource but can apply to allocation of any resource.

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.
Semi-automatic (Procedural Only)	Transitions within a procedure are carried out on manual commands as appropriate conditions are fulfilled.	Operators may pause the progression or re-direct 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.

5.7.2 States

Equipment entities and procedural elements may have states. Example states are described in this standard in relation to batch control. The state completely specifies the current condition of equipment entities or procedural elements. In the case of a valve, the state may be "percent open," and in the case of a procedural element, it may be "running" or "holding."

This standard uses *as an example* a self-consistent set of procedural states and commands. The number of possible states and commands and their names vary for equipment entities and for procedural elements.

Examples of states for procedural elements include running, holding, paused, stopped, aborted, and complete. Examples of states for equipment entities include on, off, closed, open, failed, travelling, tripped, 35% open, and available. Examples of commands applicable to procedural elements are start, hold, pause, stop, and abort.

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

Equipment entities or procedural elements may change state. 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.

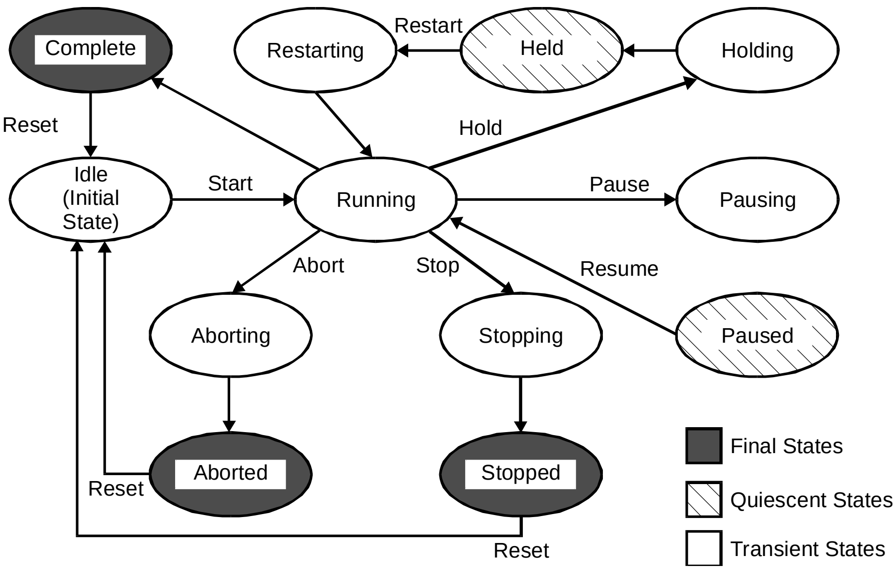
A change of state in one equipment entity type or procedural element type may cause corresponding changes in other types. For example, putting a unit procedure to the *Held* state may cause all procedural elements in that unit to go to the *Held* state, or, a safety interlock trip may cause all procedural elements in that unit to go to the *Aborting* state. 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.

A set of procedural states and commands is provided below as a representative example to illustrate one way to define these procedural states and commands. The list of states and commands is summarized in the state transition matrix (see Table 2). An example state transition diagram is derived from the matrix for the first three lines in the matrix (Idle, Running, Complete) (see Figure 18).

Table 2 — State transition matrix for example states for procedural elements

	Command	**Start**	**Stop**	**Hold**	**Restart**	**Abort**	**Reset**	**Pause**	**Resume**
Initial State	**No Command End State**	**State Transition Matrix**							
Idle		Running							
Running	Complete		Stopping	Holding		Aborting		Pausing	
Complete							Idle		
Pausing	Paused		Stopping	Holding		Aborting			
Paused			Stopping	Holding		Aborting			Running
Holding	Held		Stopping			Aborting			
Held			Stopping		Restarting	Aborting			
Restarting	Running		Stopping	Holding		Aborting			
Stopping	Stopped					Aborting			
Stopped						Aborting	Idle		
Aborting	Aborted								
Aborted							Idle		

NOTE – The states ending with "ING" are transient states. If their logic completes normally, then a state transition to the state listed under NO COMMAND END STATE occurs. For example, if the RUNNING state completes normally, then the state automatically transitions to COMPLETE. Execution of the transient states (ending in -ING) is governed by the mode.



NOTE – This state transition diagram is derived from the first three initial states of the state transition matrix in Table 2 (Idle, Running, Complete)

Figure 18 — State transition diagram for example states for procedural elements

5.7.2.1 Procedural states

For this example, the list of valid procedural states are the following:

— **IDLE:** The procedural element is waiting for a START command that will cause a transition to the RUNNING state.

— **RUNNING :** Normal operation

— **COMPLETE:** Normal operation has run to completion. The procedural element is now waiting for a RESET command that will cause a transition to IDLE.

— **PAUSING:** The procedural element or equipment entity has received a PAUSE command. This will cause the procedural element to stop at the next defined safe or stable stop location in its normal RUNNING logic. Once stopped, the state automatically transitions to PAUSED.

— **PAUSED:** Once the procedural element has paused at the defined stop location, the state changes to PAUSED. This state is usually used for short-term stops. A RESUME command causes transition to the RUNNING state, resuming normal operation immediately following the defined stop location.

— **HOLDING:** The procedural element has received a HOLD command and is executing its HOLDING logic to put the procedural element or equipment entity into a known state. If no sequencing is required, then the procedural element or equipment entity transitions immediately to the HELD state.

— **HELD:** The procedural element has completed its HOLDING logic and has been brought to a known or planned state. This state is usually used for a long-term stop. The procedural element or equipment entity is waiting for a further command to proceed.

— **RESTARTING:** The procedural element has received a RESTART command while in the HELD state. It is executing its restart logic in order to return to the RUNNING state. If no sequencing is required, then the procedural element or equipment entity transitions immediately to the RUNNING state.

— **STOPPING:** The procedural element has received a STOP command and is executing its STOPPING logic, which facilitates a controlled normal stop. If no sequencing is required, then the procedural element or equipment entity transitions immediately to the STOPPED state.

— **STOPPED:** The procedural element has completed its STOPPING logic. The procedural element or equipment entity is waiting for a RESET command to transition to IDLE.

— **ABORTING:** The procedural element has received an ABORT command and is executing its ABORT logic, which is the logic that facilitates a quicker, but not necessarily controlled, abnormal stop. If no sequencing is required, then the procedural element transitions immediately to the ABORTED state.

— **ABORTED:** The procedural element has completed its ABORTING logic. The procedural element is waiting for a RESET command to transition to IDLE.

5.7.2.2 Commands

For this example, the list of valid commands are the following:

— **START:** This command orders the procedural element to begin executing the normal RUNNING logic. This command is only valid when the procedural element is in the IDLE state.

— **STOP:** This command orders the procedural element to execute the STOPPING logic. This command is valid when the procedural element is in the RUNNING, PAUSING, PAUSED, HOLDING, HELD, OR RESTARTING state.

— **HOLD:** This command orders the procedural element to execute the HOLDING logic. This command is valid when the procedural element is in the RUNNING, PAUSING, PAUSED or RESTARTING state.

— **RESTART:** This command orders the procedural element to execute the RESTARTING logic to safely return to the RUNNING state. This command is only valid when the procedural element is in the HELD state.

— **ABORT:** This command orders the procedural element to execute the ABORTING logic. The command is valid in every state except for IDLE, COMPLETED, ABORTING and ABORTED.

— **RESET:** This command causes a transition to the IDLE state. It is valid from the COMPLETE, ABORTED, and STOPPED states.

— **PAUSE:** This command orders the procedural element to pause at the next programmed pause transition within its sequencing logic and await a RESUME command before proceeding. This command is only valid in the RUNNING state.

— **RESUME:** This command orders a procedural element that has PAUSED at a programmed transition as the result of either a PAUSE command or a SINGLE STEP mode to resume execution. This command is only valid when the procedural element is in the PAUSED state.

5.8 Exception handling

An event which occurs outside the normal or desired behavior of batch control is commonly called an exception. Handling of these exceptions can occur at all of the levels in the control activity model and may be part of procedural, basic, and coordination control.

Exception handling is an essential function of batch manufacturing. Exception handling is an integral part of all control and typically constitutes a very large portion of the control definition.

Examples of events that may indicate the need for exception handling are

— unavailability of raw materials, utilities, or plant equipment when needed;

— product or process problems;

— control equipment malfunction;

— hazardous conditions such as fire or chemical spills.

From the standpoint of control, exception handling is no different from desired control strategies in that an event is detected, evaluated, and a response generated.

Exception response functions may affect the modes and states of equipment entities and of procedural elements. For example, high pressure in a reactor could lead to the exception response function transferring the process to a STOPPED state, or an operator could detect some unusual condition and initiate similar action.

6 Batch control activities and functions

This section discusses control functions that are associated with the batch processing, manufacturing, and control tasks described in the previous two sections. The control functions defined in this section elaborate on the control tasks defined in Section 5.1 for the equipment entities discussed in Section 5.2, which are the bottom four levels of the Physical Model described in Section 4.2. Control functions that meet the control needs of the higher levels of the Physical Model are also described. For convenience purposes, these control functions have been grouped into, and are discussed in the context of, control activities. The control activities that are discussed in this section are Recipe Management, Production Planning and Scheduling, Production Information Management, Process Management, Unit Supervision, Process Control, and Personnel and Environmental Protection. The intent of this section is to clearly identify the individual functionality associated with batch control. This will make it easier to define the requirements for batch control for a given application.

6.1 Control activities

Many control functions must be implemented to successfully manage batch production. These control functions define how equipment in the batch manufacturing plant will be controlled. They are needed to support the equipment entities described previously. They are combined into seven control activities, as represented in the Control Activity Model of Figure 19.

6.1.1 Control activity model

The Control Activity Model in Figure 19 provides an overall perspective of batch control and shows the main relationships between the various control activities. It is not intended to show all relationships. These relationships are achieved via information flow between the control activities. The purpose of this drawing is simply to show where there is a relationship and not to define that relationship. The definition of these relationships will take place later in this section as the control functions grouped within each control activity are discussed. A few of the relationships shown in Figure 19 are not discussed further in this standard.

The control activities shown relate to real needs in a batch manufacturing environment. The need to have control functions that can manage general, site, and master recipes implies a need for the Recipe Management control activity. Production of batches must occur within a time domain that is planned and subsequently carried out. Production Planning and Scheduling is the control activity where these control functions are discussed. Various types of production information must be available, and the collection and storage of batch history is a necessity. The Production Information Management control activity in the model covers these control functions.

Control recipes must be generated, batches must be initiated and supervised, unit activities require coordination, and logs and reports must be generated. These control functions fall under the Process Management control activity in the model. There are many control functions needed at the Unit Supervision control activity level. For example, there is a need to allocate resources, to supervise the execution of procedural elements, and to coordinate activities taking place at the Process Control level. In Process Control, control functions are discussed that deal directly with equipment actions such as the need to implement control functions using regulating devices and/or state-oriented devices.

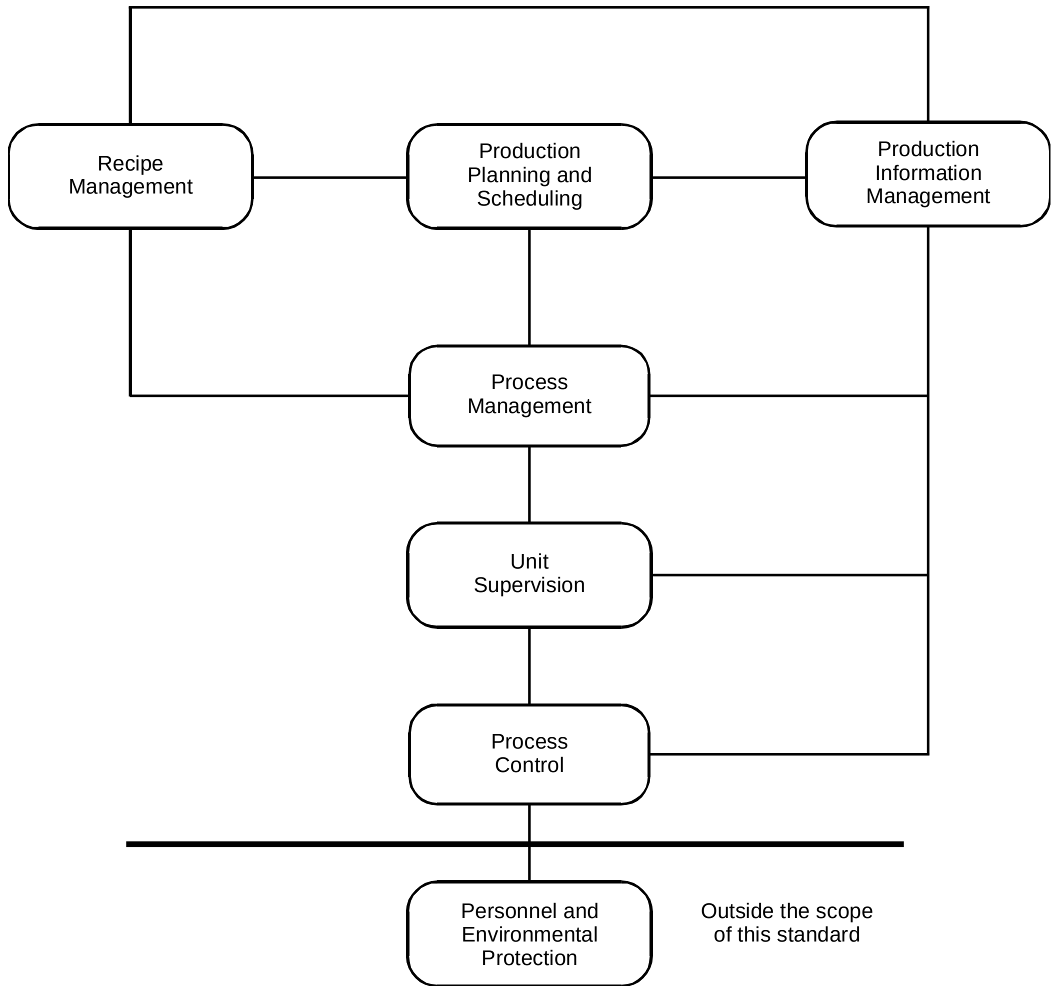


Figure 19 — Control activity model

Finally, the safety of personnel and the surrounding communities must be a prime concern, along with protection of the environment. The Personnel and Environmental Protection control activity covers these control functions.

6.1.2 Information handling

One dimension of the control activity model is its description of information flow throughout the levels. As such, there are a number of information handling functions that can be applied to all categories of data addressed by the control activity model. These are applicable regardless of the combination of manual and computerized systems that are established at a site. Additional information handling aspects that are specific to a particular control activity are described within their respective sections.

6.1.2.1 Reference information

The batch manufacturing enterprise may incorporate activities that fall outside the scope of this standard. Examples include

— material inventory management;

— process and product development;

— customer service support;

— regulatory reporting and process validation; and

— inter-departmental coordination, such as production versus support services.

To provide an interface to these information sources, the control activities discussed in this section need to store information in a way that provides a usable, accessible data source to these external activities. Similarly, each control activity should have the ability to access relevant reference information as needed to fulfill its function.

Examples of reference information include

— sales or marketing data, including customer orders or other statements of product demand;

— raw material vendor data;

— final products specifications;

— costing data;

— research and development data;

— standard consumptions of raw materials and standard yields for the products manufactured;

— rate information for the various process cells;

— equipment capability specifications;

— operational procedures for equipment maintenance and process safety;

— human resource information;

— quality control information such as the procedure used to perform a particular laboratory analysis; and

— regulatory requirements.

Reference information may be enterprise-wide, site-wide, area-wide, or process cell-wide.

6.1.2.2 Security

Within the control environment, information is used to impact the control functions, to communicate between levels and entities, and to provide communication to control functions outside of the control activity model. Access to this information is restricted to ensure that only authorized and/or qualified resources can affect the information.

6.1.2.3 Availability

Control activity information should be stored and retrieved in a way that provides the necessary safeguards to ensure access to critical data. The time necessary to recover access to the data in case of loss at one location should be considered carefully. These considerations will vary based on the different levels of the control activity model, the types of information, and the level of detail required.

6.1.2.4 Archival

Removal of information from the control activity and into a long-term archive is often desirable to improve storage efficiency and recoverability. Once archived, it should be possible to retrieve the archived data in a usable form. For example, once a master recipe is no longer in active use, it would be useful to be able to extract all information (both structural and historical) related to that master recipe from the main repository.

6.1.2.5 Change management

Information that defines control — including configuration of equipment control and recipes — may be subject to formal change management. Means may be provided to support

- — requests for and authorization of changes;
- — version numbering and documentation;
- — validation of changes; and
- — audit tracking.

Change management may also include restrictions and checks necessary to maintain the integrity of the configuration. For example, it may be necessary to prevent a recipe creator from modifying a procedural element in use by an active recipe.

6.1.2.6 Reference tracking

Historical tracking of information references — for example, which definitions are used within which others or which served as the basis for others — can be important in analysis of production performance and in demonstrating compliance with production guidelines. This function can also provide a means to attach written comments about the changes, to assist in subsequent interpretation.

6.1.3 Process and control engineering

In order for required processing functions to be properly carried out in a batch manufacturing environment, the equipment structure needed, the process functionality, and the exception handling for that equipment have to be fully developed. This requires a coordinated engineering effort that continues from initial definition through the life of the batch processing facility. This section describes the process and control engineering needed for the design of the controls needed to support the recipe hierarchy, the definition of equipment capability, and the development of the functionality required in the procedures to produce a batch.

Process and control engineering is needed at the general and site recipe levels to describe procedures, process stages, process operations, and process actions and at the master recipe level to describe recipe procedures, recipe unit procedures, recipe operations, and recipe phases.

The precise definition of appropriate procedural elements and equipment entities is an iterative process. The dual work process is illustrated in Figure 20. Considerations affecting one decision process also affect the other. Processing considerations are the primary input to the definition (or selection) of procedural elements that will characterize functionality for associated equipment entities. Since the functionality defined will be affected by the equipment used, equipment considerations must be a secondary input. In the same way, equipment considerations form the primary input and processing considerations the secondary input when making the definition (or selection) of equipment entities.

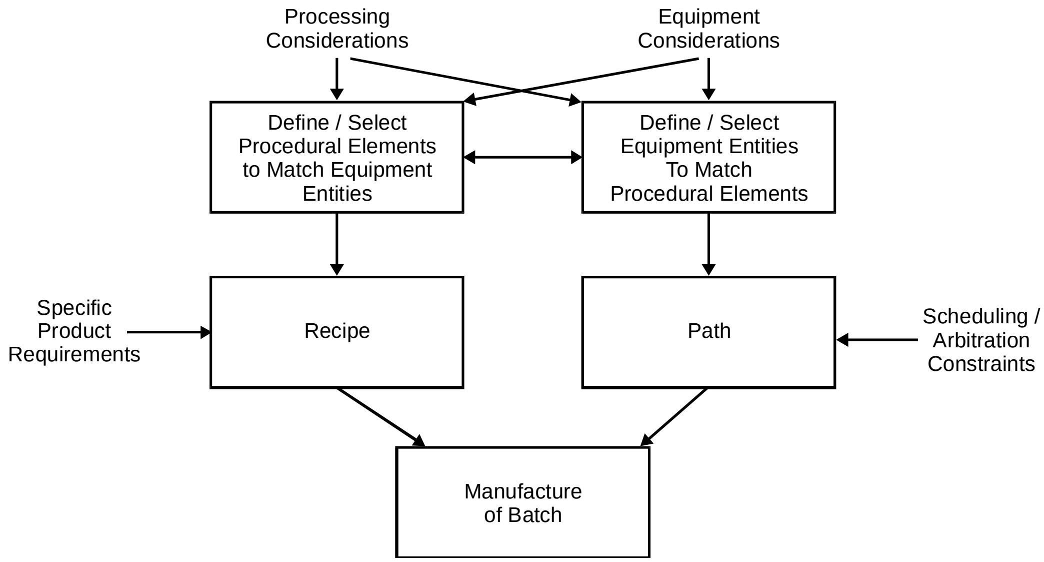


Figure 20 — Simultaneous definition/selection of procedural elements and equipment entities

Recipes can be constructed using these procedural elements and specific product information. The equipment entities are arranged into a path that is determined by scheduling and taking into account arbitration constraints. The combination of the results of these activities provides a framework within which a batch of material can be manufactured.

Process and control engineering also includes the development and revision of the equipment phases corresponding to the recipe phases that are used to define the recipe. As far as possible, recipe and equipment phases should be defined such that any reasonable functionality of a unit can be expressed in terms of these phases. They should generally not be tailored to a set of known recipes. Then, new recipes can in most cases be written by using existing recipe phases that reference existing equipment phases. The development and revision of recipe and equipment phases is an ongoing activity that provides ongoing support to the batch manufacturing facilities. This activity is the result of the ongoing drive for continuous improvement and the periodic addition of new process technology.

6.2 Recipe management

Recipe Management is made up of the control functions that create, store, and maintain general, site, and master recipes. The overall output of this control activity is a master recipe that is made available to Process Management, which uses it to create a control recipe.

Recipe Management will be discussed in terms of managing the three levels of recipes and defining the procedural elements used in the recipe procedures (see Figure 21).

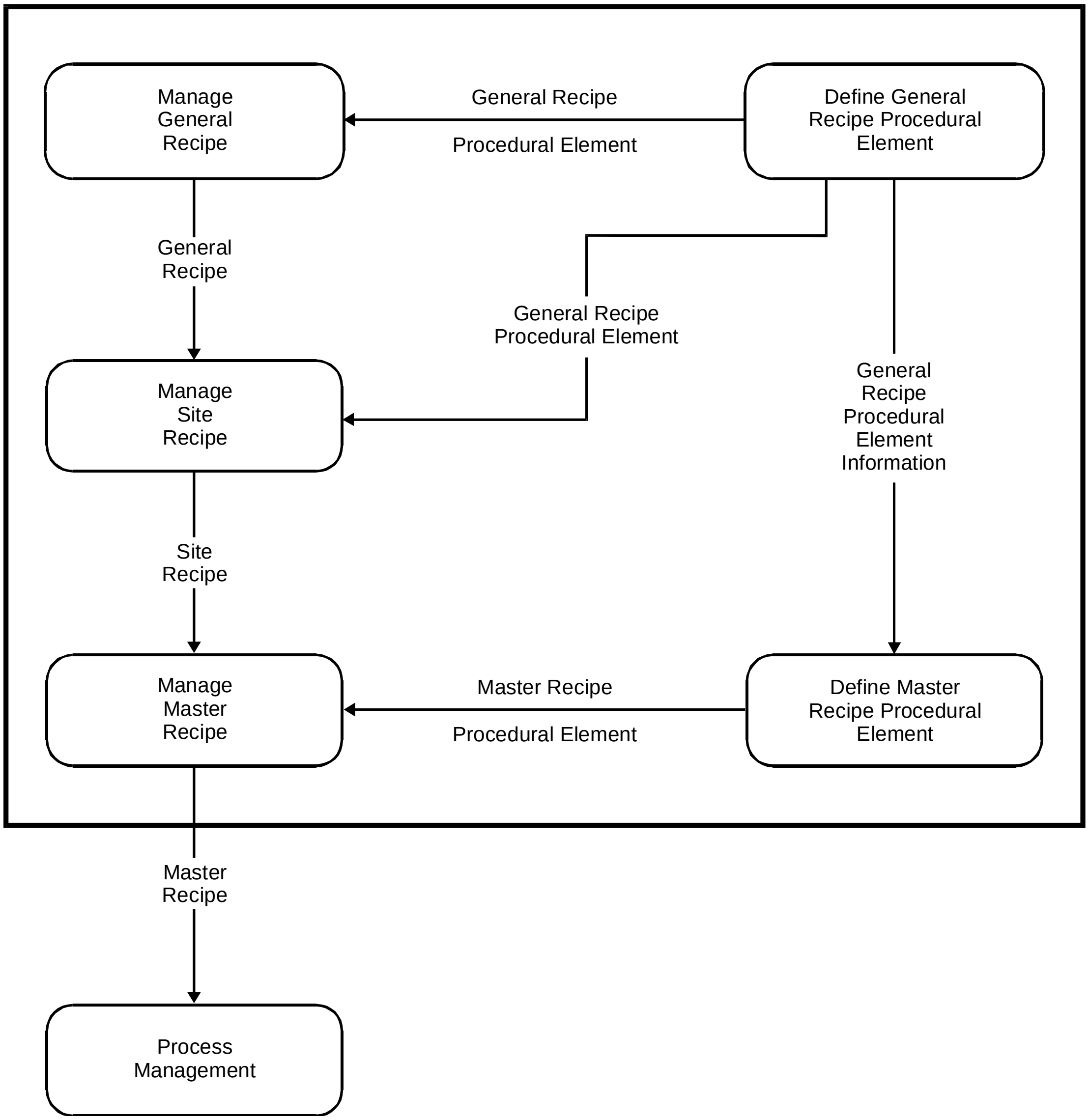


Figure 21 — Recipe management

6.2.1 Manage general recipes

Manage general recipes is the control function by which general recipes are created, maintained and stored. The specific processing requirements furnished by the process development activity for the product being considered serve as the basis for the general recipe.

In connection with the definition of the individual general recipe, the following capabilities may be required:

— Selecting and combining procedural elements to create a general recipe procedure

— Incorporating formula information

— Specifying equipment requirements and other information

— Maintaining the general recipe

— Managing changes to general recipes

6.2.2 Define general recipe procedural elements

The *define general recipe procedural elements* control function creates, maintains and makes available for subsequent use, the procedural elements that are used as building blocks in general recipe and site recipe procedures.

The procedural elements defined by this control function may be process actions, process operations, process stages, and/or an entire general recipe procedure. Not all levels of procedural elements need to be defined.

This control function defines general recipe procedural elements based on the processing strategies required by the different products and described by process development. This information is then made available to the *define master recipe procedural elements* control function. In this way, the process intent of the general recipe procedural elements may be known at the master recipe level. The strategies have to be interpreted and transformed into procedural elements that enable the flexible and modular construction of general recipes. The broader the range of products that can utilize these modular process activity descriptions, the easier it is to create a general recipe. More importantly, modular process actions, process operations, process stages, and/or complete procedures that are frequently reused tend to make recipe transformations at lower levels much easier to accomplish and recipes more consistent.

This procedural element information is then made available to the *define master recipe procedural elements* control function. In this way, the process intent of the general recipe procedural elements may be known at the master recipe level.

In connection with the definition of the individual general recipe procedural elements, the following capabilities may be required:

- Naming the individual procedural elements
- Specifying parameter variables
- Describing the intended processing functionality
- Combining lower level procedural elements and specifying the sequence of execution
- Creating, modifying, and archiving general recipe procedural elements
- Maintaining an inventory of procedural elements available
- Managing changes to procedural elements

6.2.3 Manage site recipes

Manage site recipes is the control function by which site recipes are created, maintained and stored. A site recipe is created by combining the information of the appropriate general recipe with site specific information. If additional or alternate procedural elements are required, only those defined under the *define general recipe procedural elements* control function are used.

6.2.4 Manage master recipes

Manage master recipes is the control function by which master recipes are created, maintained and stored. Master recipes are defined based on the specific processing requirements for the product in question. These specific processing requirements may be expressed in a general or site recipe.

The transformation of the site recipe into a master recipe may be a complex task. The creation of a procedure, based on predefined procedural elements, must match the intent of the site recipe procedure. Transformation (or creation) of the content of the formula follows the same general logic that is used to map process actions to recipe phases. The batch size is fixed, or the range of batch sizes permissible for the recipe is established, if there are constraints on the degree of scaleability. Formula information is adjusted accordingly. The equipment requirements are transformed into requirements that can be verified against the actual target equipment.

In connection with the definition of the individual master recipe, the following capabilities may be required:

— Selecting and combining procedural elements to create a master recipe procedure

— Incorporating formula information

— Specifying equipment requirements and other information

— Creating, modifying, and archiving master recipes and maintaining the recipe headers

— Maintaining an inventory of master recipes

— Managing changes to master recipes

6.2.5 Define master recipe procedural elements

The *define master recipe procedural elements* control function creates, maintains and makes available for subsequent use, the procedural elements used in master recipe procedures. These become the building blocks of the master recipe procedure.

The master recipe procedural elements must reflect the processing capabilities required by master recipes. If these are generated from general and site recipes, then process stages, process operations, and process actions will map into unit procedures, operations, and phases. This control function defines the relationship between process actions and phases, between process operations and operations, and between process stages and unit procedures. It also defines the general scope of procedures, unit procedures, operations, and phases to allow maximum consistent use of pre-defined procedural elements across the range of products to be made in the facility.

The master recipe procedural elements must, at least at the recipe phase level, be able to reference equipment procedural elements when the derived control recipe is executed. A close coordination with the engineering of the equipment procedural elements must therefore take place, ensuring that the recipe procedural elements adequately reflect the control capabilities of the target equipment. If required, any new functionality is made available through creation of new procedural elements, along with associated control and equipment modifications (see 6.1.3).

In addition to providing the building blocks for the master recipe procedure, this control function may also define constraints on the configuration of master recipes, such as rules on the allowable order of recipe phases and limitations in the recipe creator's right to use recipe phases as building blocks. The determination of such constraints must be made based on many factors, such as safety, complexity of the recipe creator's task, required flexibility, and validation of individual procedural elements.

In connection with the definition of the individual procedural elements, the following capabilities may be required:

— Naming of the individual procedural elements

— Specifying parameter variables

— Describing the intended processing functionality

— Combining lower level procedural elements and specification of the sequence of execution

— Creating, modifying, and archiving master recipe procedural elements

— Maintaining an inventory of procedural elements available

— Managing changes to procedural elements

6.3 Production planning and scheduling

Production Planning and Scheduling is a high level control activity on a peer level with Recipe Management and Production Information Management. It is the decision process associated with producing a batch schedule that is provided to Process Management. Although several control functions would need to be collected together to make up this control activity, most of those control functions are outside the scope of this standard. This section will consider only one of these control functions: *Develop batch schedules*.

The *develop batch schedules* control function accepts inputs from sources such as other types of schedules, master recipes, and resource databases, and, based upon a scheduling algorithm (automated or manual), develops a batch schedule (see 5.4 for a list of typical information in a batch schedule).

The following capability is typically included in this control function:

— Developing a batch schedule based on information from the appropriate source and some scheduling algorithm

— Developing a revised batch schedule on demand based on significant changes in *batch progress and process cell status information* provided by Process Management

— Allowing for manual intervention into the scheduling process

— Determining the availability of resources as an input into the scheduling process

— Providing a procedure or method for batch sizing along with a means to organize the production of batches

— Determining the feasibility of the schedule based on the target equipment

6.4 Production information management

Production Information Management is a high level control activity on a peer level with Recipe Management and Production Planning and Scheduling. It is the control activity that is involved in collecting, storing, processing, and reporting production information.

The non-batch-related use of production information is not dealt with in this section, but in actual applications the management of batch-related information and non-batch-related information may very well be integral. Both batch-related and non-batch-related information may be used as input to higher-level control functions such as the generation of production reports to management. These activities will not be modeled in this standard.

Although several control functions would need to be collected together to make up this control activity, most of those control functions are outside the scope of this standard. This section will consider only one of these control functions: *Manage batch history*.

Batch history is a collection of data related to one batch. It may be organized in one or more files or tables per batch, or it may be present as a part of a database and retrievable via key fields, etc.

Batch history is built up of entries. An entry is a portion of information on the batch representing one value or a set of values describing one event, logged into the batch history in one action.

Manage batch history is the control function that typically includes the following capabilities:

— Receiving and storing information from other parts of the overall batch control application on batches

— Manipulating historical data

— Producing batch reports

The *manage batch history* control function is performed regardless of the equipment used or when a batch is produced. For example, lab data often may be added after the execution of the batch.

6.4.1 Receiving and storing batch history information

The entering of data from the outside into batch history is initiated from Process Management, Unit Supervision, and Process Control.

6.4.1.1 General collection and storage guidelines

All of the data for the batch history should be collected and stored in a way that includes or gives simple access to

— batch identification;

— absolute time stamp (Real time);

— identification of procedural elements with which the data is associated;

— time relative to the start or end of a batch or of the execution of a procedural element;

— equipment-independent entry identification;

— equipment utilized.

Adequate storage capacity is needed for the required number of batch histories. This should include sufficient capacity to store the batch histories of all running batches, and for finalized batches until appropriate actions have been taken (reports printed, long term backup or whatever action is specified).

To the extent that the storage time requirement exceeds the storage capacity of *manage batch history*, the capability must exist to export the batch histories onto long-term storage media or external systems. It must be possible to retrieve these batch histories for further extraction of data.

Reports or displays on the batch archive (number of batches in archive, amount of data, status [finalized, printed, archived in long term archive, etc.]).

6.4.1.2 Reliability of batch history entries

The requirements for reliability will vary from application to application and between the different entry types. In the following, a number of issues of reliability are described. For each type of entry, the appropriate level of reliability must be selected to match the needs of the individual application. Reliability issues include

— access control: control of access to the data-gathering system, including the configuration and the actual data collected;

— audit trail: identification of all manipulation that happened with each individual piece of information — including identification of the person or controls involved, the time and, in some cases, an explanation;

— logging reliability: specification of the required reliability of logging. Three levels may be distinguished:

a) Nice to have — no specific action in case of failure. Examples include data for optimization, equipment reliability statistics, etc.;

b) Limited holes acceptable if the failure is indicated in the batch history (logging absent from . . . to . . .);

c) Critical — data must be available. If it is missing, then backup procedures must be possible (electronic or manual backup, possibility of reconstruction, etc.).

The importance of exact logging of the latter type of information may be equivalent to the achieved product quality, either for financial reasons (accounting) or for product safety/responsibility reasons. Therefore the receiving function must be capable of providing feedback information on the general status of the receiving function (as well as specific confirmation feedback for each entry to the control activity that performs the logging) enabling them to perform buffering, redundancy or reintegration activities or, if required and allowed, to hold up the process.

— level of detail: This level should be well defined in the recipe, or it should be related to the process cell or parts of the process cell. It must be possible to see if an entry is absent because the corresponding event did not occur or because it is below the selected level of detail.

— logging of actual historic information: Batch history entries should, to the largest possible extent, reflect the actual physical/chemical events that influence the batch, not only what was anticipated in the recipe. That means that the character and amount of data logged will vary due to the variations in batch production.

— long-term consistency: The extent to which the interpretation of batch data relies on information outside of the batch history, such as cross reference lists between actual tags and batch entry tags or names of variables, should be well described. Such information should be stable in the long term. If changes or modifications do occur, then the versions that were relevant at the time of processing should be stored for use in data retrieval.

— speed of collection: Speed of collection should be considered a critical factor. In order to analyze the reasons for any abnormal conditions, it is important that the system be capable of recording the events and actions in the precise order in which they occurred.

6.4.1.3 Batch and material tracing

The collection of batch histories can support batch and material tracing if it has a complete overview of the batches, including the equipment utilized and the identification of raw materials.

Batch history provides backwards tracing if a certain end product batch history can be traced back to all involved processes, equipment, and ingredients (and to the involved processes, equipment, and ingredients of these ingredients). Forward tracing is available if the consequences of a certain event or the usage of a certain raw material can be traced to all end products affected.

6.4.1.4 Logging from process management

Process Management logging should include information associated with initiating and routing the batch, and the equipment-independent information associated with the batch. This includes

— master recipe: the master recipe from which the control recipe was derived — either in copy or by reference. In case of reference, the master recipe should be maintained unchanged as long as the reference may be called.

— Process Management events and control recipe information: information on any changes to and the execution of the control recipe. This includes information such as equipment allocation and start times for batches and unit procedures.

— operator comments: narrative descriptions or comments based on the operators' observations of the batch processing. This information entry should be capable of being recorded with the operator's identification.

6.4.1.5 Logging from unit supervision and process control

This data could be dedicated to a single batch or to several batches, such as data from shared resources, utility systems, etc. In the latter case the data should be available to all the required batch histories. This includes:

— continuous data: Continuous data is defined as process data that is collected independent of specific events within the batch, with the purpose of giving an accurate history of that measurement.

— pre-specified batch data: data that is specified to be logged during execution of the control recipe. The specification of this data may come from the recipe or be pre-configured.

This would include such things as total feed to a reactor or mixing time.

— predictable events: events that are expected to occur, such as start and stop times of procedural elements.

— unpredictable events: Unpredictable event data is defined as a single point entry based on an unpredictable process or physical condition within the batch. This includes such items as process alarms, equipment failures or other upset conditions. In the case of process alarms, the historical data may include the following:

a) Time of activation

b) Time of acknowledgment

c) Time of disappearance of the alarm condition

d) Alarm limit

e) Maximum deviation while the alarm is active

f) Trending information while the alarm is active

— operator interventions: any operator intervention that may affect the processing of the batch. The operator intervention typically is logged with the following information:

a) Intervention type

b) Operator ID

6.4.1.6 Late entries

Late entry data is data entered after execution of the part of the control recipe procedure to which it is related, or after production of the batch. This is typically data that is related to off-line measurements or analyses. *Manage batch history* includes the logging of such entries, including establishing the link to the associated batch events (like sampling). The following data may be associated with late entries:

— Measured value(s)

— Operator ID

— Lab technician ID

— Time of entry

— Time of sample

6.4.2 Manipulating historical data

The following functions are typical:

— Data manipulation: altering (if legal) or supplementing archived batch data.

— Calculations: perform calculations on batch data creating new batch data related to one batch.

— Data reduction: data reduction on batch history information that is especially relevant with trend information. Loss of data in connection with data reduction should be well defined and related to the dynamics of the data, as well as the requirements of information based on this data.

— Batch tracking information: establishing or maintaining links between batch histories corresponding to the physical movements of the batches, ranging from the use of one batch as raw material to another, to the splitting or combining of batch histories due to splitting or combining of batches.

6.4.3 Producing batch reports

In this section any export of data — electronically or on paper — is designated a report.

A batch report is, in general, made on a specific request. Such a request must be possible without knowledge of equipment and time of production. This is the case when

— the batch ID is used as entry key to access the data, not a piece of equipment;

— timing is relative to identified batch events (start of batch, start of operation, etc.);

— entries are identified in generic, batch-related terms and not in equipment-specific tags.

6.4.3.1 Recipients of batch reports

Batch history data may be retrieved on request for a number of reasons:

— Production management: production overview summaries, consumption of raw materials and other resources, lot and batch tracking information

— Recipe management: recipe optimization information, comparison between recipe data and actual values, analysis of correlation across several batches, and comparison of trend information

— Process management: history of current batches and comparisons with old batches for operator display and process control optimization

— External systems:

a) quality control: statistical process control, compliance with product specifications, GMP (Good Manufacturing Practice) documentation

b) maintenance: alarms, equipment usage documentation

c) financial: raw material consumption, yields, produced quantities, etc.

d) customer support: product documentation

— Internally within *manage batch history*: Process Management may include functions to perform the queries mentioned above and the ability to export or print them on request, at regular intervals, or after each batch.

6.4.3.2 Elements of batch reports

Some of the possible elements of a batch report include

— report header: This header contains information on the report type, batch or batches displayed in the report, descriptive text, etc.

— single elements: These data elements are displayed somewhere on the paper/screen.

— event lists: These are chronological lists of event-type entries with associated data. For example, this might include a list of alarms or a list of operator interventions.

— merging of entries in event lists: Entries with different tags and of different types may be merged into the same list.

— selection of entries into lists: Entries may be selected according to different criteria before entering lists. For example, the entries may include only high priority alarms.

— trends: These displays show one or more values on the same time axis.

a) single batch trend: These are trends that display data from one batch or a portion of a batch . They may display several values with individual time axis. The display may be in relative or absolute time.

b) multi-batch trend: These are trends that compare values from several batches in one trend display. They must be with relative time-axis. Some variables may be normalized to a standard amount.

c) event-marking in trends: Events may be introduced in the trend display by "ticks" on trends or other indications. The tick should refer to a specific event-type entry.

— time-series: These are displays of a time-series of one or more entries in a table-like fashion. The time-deadband, which is how close in time entries with different tags have to be in order to be displayed on the same line, must be specified in time series displays.

— interpolation: Rules for interpolation of data have to be established if data with different entry-times have to be displayed on one line or if the data are used in calculations.

6.5 Process management

Process Management is the collection of control functions that manages all batches and resources within a process cell. Within this control activity, control recipes are created from master recipes, each batch is defined as an entity, individual batches are initiated and supervised, resources within the process cell are managed to resolve conflicts for their use and process cell and batch data are collected. Process Management interfaces with Unit Supervision, Recipe Management, Production Planning and Scheduling, and Production Information Management (see Figure 22).

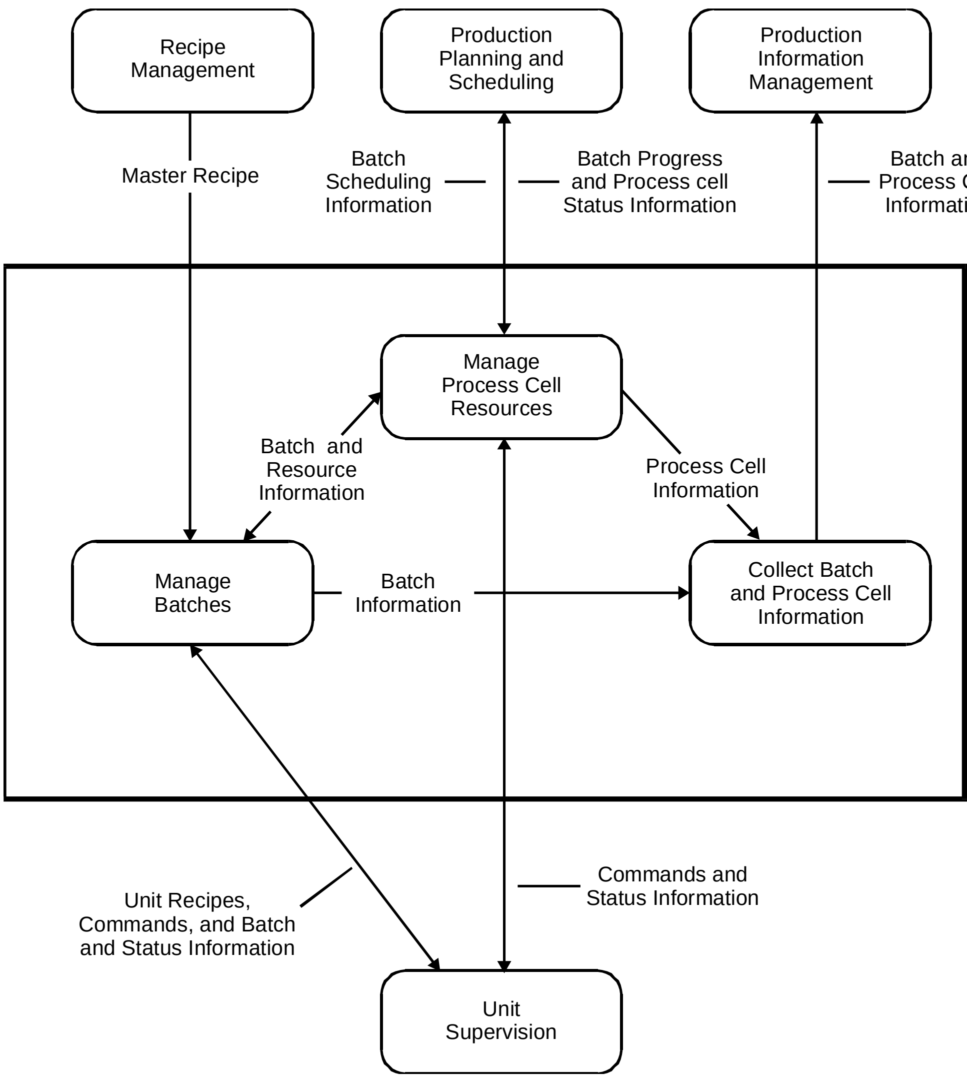


Figure 22 — Process management

At the process cell level, there are often multiple batches and multiple units, and each unit may be carrying out a unit procedure for a different batch. The progression of the procedure for each batch and the utilization of the individual pieces of equipment has to be coordinated based on information derived from the control recipe, scheduling information, and status of equipment and other common resources.

The domain of Process Management is the process cell. The successful execution of a control recipe makes a batch, and Process Management is finished with the batch when the control recipe procedure is complete. The batch that has been produced does not have to be a final product. It may take several control recipes running in the same process cell or in different process cells and/or sites to make the finished product(s). When a batch leaves the process cell, it is no longer the responsibility of Process Management associated with that process cell in terms of identification, batch tracking, etc.

Process Management can be discussed in terms of the following three control functions (see Figure 22):

— Manage batches

— Manage process cell resources

— Collect batch and process cell information

6.5.1 Manage batches

This is the control function in which a control recipe is created from a copy of a master recipe, a batch is initiated based on the scheduling information and operator input, and the execution of the batch is supervised.

The following capability is typically included in this control function:

— Creating a control recipe from the master recipe, scheduling information, and input received from the operator. This may happen with widely varying lead times, such as at the instant needed in some situations and well in advance of the scheduled execution time in others. The control recipe may be created initially in its entirety, or it may be created incrementally as the information is needed.

— Assigning a unique batch identification (batch ID) to each batch and to the associated control recipe. A batch may be identified or named in many different ways, but at least one identification, referred to here as the batch ID, must be verified to be totally unique within the process cell at any given time. The batch ID may be provided by the operator, in the scheduling information, or from within Process Management, but uniqueness is typically verified before it is associated with a batch.

— Verifying the control recipe as it is created. Verifying consists of ensuring that the control recipe is complete and is executable on the selected set of units. This includes verifying that all procedural elements are available, that formula information is valid, and that necessary resources can be expected to be available when needed.

— Sizing the control recipe to meet the batch quantity needed based on the sizing rules in the master recipe and the quantity specified in the batch scheduling information. The recipe may include the range over which it may be scaled.

— Maintaining all the current control recipes within Process Management until the batches are completed.

— Assigning the start conditions as specified in the scheduling information and/or provided by the operator. Some batch start conditions that may be used, either individually or in some combination, include the following:

a) Start batch as soon as a unit becomes available

b) Start based on operator direction

c) Start when specific units are available

d) Start based on the scheduled priority of the batch

— Modifying any part of a control recipe that has not been executed. This may include the ability to modify the procedure, such as adding and deleting unit procedures, operations, and/or phases, or looping back to repeat unit procedures, operations, and/or phases that have previously been executed.

— Requesting and releasing units and other equipment, changing their status to indicate use, and updating the *manage process cell resources* control function on the status of the batch.

— Monitoring and controlling the executing control recipe(s) including the current status of the batch, such as what unit procedures have been executed, and what unit procedure is next.

— Processing requests for state and mode changes to procedures, unit procedures, operations, and phases.

— Allowing a control recipe to span multiple units in the same process cell, including distributing unit recipes to Unit Supervision in a timely manner.

— Allowing a batch to be suspended, removed from the processing equipment (packaged for temporary storage), and therefore out of the control of Process Management, and later recalled to complete the batch processing.

— Maintaining batch status information. The control recipe, including all modifications, should be logged as part of the batch history as it is executed or at least when the batch leaves the process cell.

— Updating information on batches to the *collect batch and process cell information* control function.

6.5.2 Manage process cell resources

This is the control function in which process cell resources are managed by allocating and reserving units and other equipment, by arbitrating multiple requests for the same equipment, and by providing a mechanism for controlling unallocated equipment. Process cell resources also include the materials within the process cell. Process cell resource management must know which materials are in the process cell, their location, and their disposition.

An assignment of resources at the process cell or unit level (resource allocation) needs to be provided in order for Process Management to be able to assign the equipment or equipment options from the batch schedule. Some limited equipment reassignment and generation of a new resource allocation at the process cell or unit level may also be needed by the operator. This new resource allocation may be necessary because of such variables as a malfunction in equipment or availability of raw materials. Production Planning and Scheduling may require notification of this new resource allocation to allow for assessment of impact.

The following capabilities are typically included in this control function:

— Obtaining scheduling information from Production Planning and Scheduling and providing this information to the *manage batches* control function

— Allocating or reserving equipment as requested by the *manage batches* control function. Within a process cell, batches may move from unit to unit. In each unit a portion of the control recipe, corresponding to the unit procedure, is executed. The control of what equipment to allocate to the different batches, and when transfers can take place may require control at the process cell level. Some examples of how this allocation may be done are

a) according to a batch schedule designating each individual unit allocation; or

b) according to a strategy defined at the process cell level combining the equipment requirements of the control recipe and the availability and capabilities of equipment.

— Arbitrating, as required, multiple requests for reservation or allocation of the same equipment. The rules for arbitration may be simple or complex, depending on the application. Examples of arbitration rule sets include the following:

a) Order of request (FIFO)

b) Timed requests (such as by reserving the equipment)

c) Priority of batch

d) Maximizing equipment utilization (such as by minimizing cleaning requirements, minimizing energy consumption, or maximizing throughput)

e) Operator judgment

— Managing unallocated equipment within the process cell

— Receiving status information sent by Unit Supervision and/or status information sent by Process Control related to unallocated equipment within the process cell

— Updating information on all process cell resources to the collect batch and process cell information control function

— Updating Production Planning and Scheduling with batch progress information, such as

a) batch ID;

b) batch state change events;

c) actual quantities of raw materials, products, and utilities;

d) equipment assignments; and

e) projected and actual allocation and de-allocation times of process cell resources.

6.5.3 Collect batch and process cell information

This is the control function in which information is collected about Process Management events, both batch and equipment oriented, from the *manage batches* and *manage process cell resources* control functions. This information is made available to Production Information Management.

Examples of the types of information collected include the following:

— Mode and state changes

— Incremental copies of control recipes as each portion is finished

— Time that commands were sent to Unit Supervision and Process Control

— Time that unit recipes were sent to Unit Supervision

— Delays encountered due to lack of equipment availability

— Time of allocation, reservation and release of each process cell resource

— Requests and result of requests for equipment allocation or reservation which required arbitration

— Status changes in unallocated equipment

— Operator intervention

6.6 Unit supervision

Unit Supervision is the control activity that ties the recipe to equipment control via Process Control (see Figure 23). This control activity interfaces with Process Management, Process Control, and Production Information Management. There are three main control functions within this control activity that are discussed in this section. They include acquiring and executing procedural elements, managing unit resources, and collecting batch and unit information.

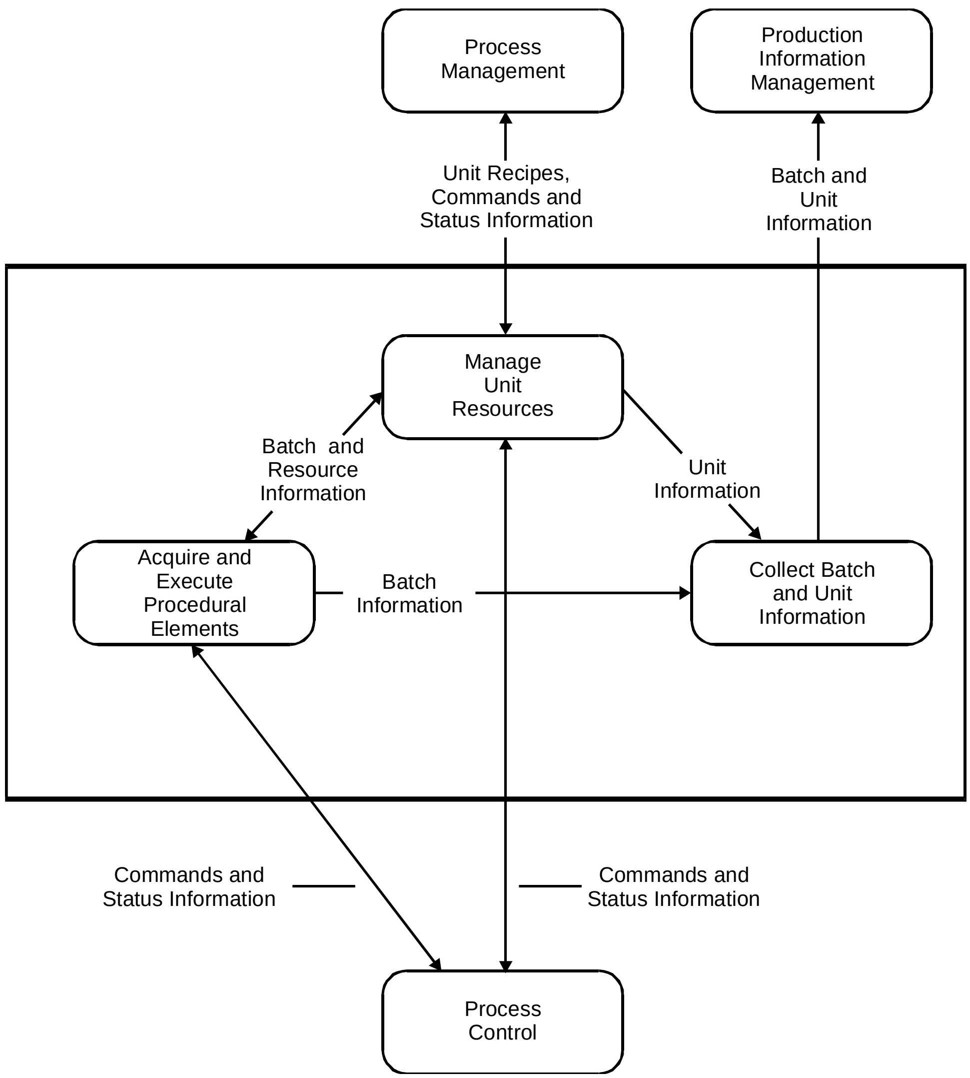


Figure 23 — Unit supervision

6.6.1 Acquire and execute procedural elements

Process Management supplies the unit recipe that will be executed within the unit and also supplies other batch information required to manufacture the batch.

Unit Supervision has to be able to determine from the unit recipe the procedural logic to be run, the appropriate parameters, the equipment entities to be utilized, and other pertinent information, such as the name of the product, equipment restrictions, and the batch number.

Acquire and execute procedural elements includes the execution of unit procedures. If the unit procedure is part of equipment control in the unit, this control function associates the recipe unit procedure, including the parameters, with the equipment unit procedure.

Acquire and execute procedural elements includes the execution of operations. If the operation is part of equipment control in the unit, this control function associates the recipe operation, including the parameters, with the equipment operation. The initiation and parameterization of phases is part of the execution of an operation.

Acquire and execute procedural elements includes the initiation and/or execution of phases. If the phase is part of equipment control in the unit, this control function associates the recipe phase, including the parameters, with the equipment phase. If the phase is part of equipment control in an equipment module, this control function must initiate and parameterize the equipment phase.

The following capabilities are typically included in this control function:

— Determining which procedural elements are to be executed

— Verifying that the procedural elements exist

— Executing unit procedures, operations, and phases

— Associating recipe procedural elements with equipment procedural elements

— Initiating and parameterizing equipment phases

6.6.2 Manage unit resources

This control function includes the management of resources that are part of the unit, management of resources that might have been acquired and have not yet been released, initiation of requests for resources that are not currently part of the unit, requests for services from other units, and providing services to another unit.

During the execution of a recipe, it may be necessary to acquire shared-use and/or exclusive-use resources that will subsequently be released. Although units cannot acquire other units, they can request services from or provide services to another unit as long as the recipe has specified compatible procedural logic for both units. The phases or operations in the units can communicate to perform a coordinated function.

Unit-to-unit coordination may be used to enable functions such as material transfers between units.

The following capabilities are typically included in this control function:

— Issuing requests to, reacting on feedback from, and interfacing with arbitration functions related to the equipment in question

— Ensuring appropriate propagation of unit and procedural element modes and states

— Enabling collection of production information relevant to the batch from external equipment

6.6.3 Collect batch and unit information

The *Collect Batch and Unit Information* control function makes information available to Production Information Management about Unit Supervision events, both batch and equipment oriented.

Data collection may be conditional. That is, certain data might not always be collected or might be sampled at a different time interval, depending upon information received from another control function, such as from parameters passed to the equipment phase.

Examples of the types of information collected include the following:

— Mode and state changes

— Timing of commands sent to Process Control

— Timing of the execution of the unit recipe procedure events

— Timing and sequence of allocation, reservation, and release of equipment entities acquired by the unit

— Status changes in unit equipment

— Values derived during execution of the unit recipe

6.7 Process control

This control activity encompasses procedural and basic control, including sequential, regulatory, and discrete control, in addition to gathering and displaying data. This control activity will be distributed among several equipment entities, including units, equipment modules and control modules. It interfaces with Production Information Management, Unit Supervision, and Personnel and Environmental Protection.

Process Control can be discussed in terms of three control functions: execute equipment phases, execute basic control, and collect data (see Figure 24).

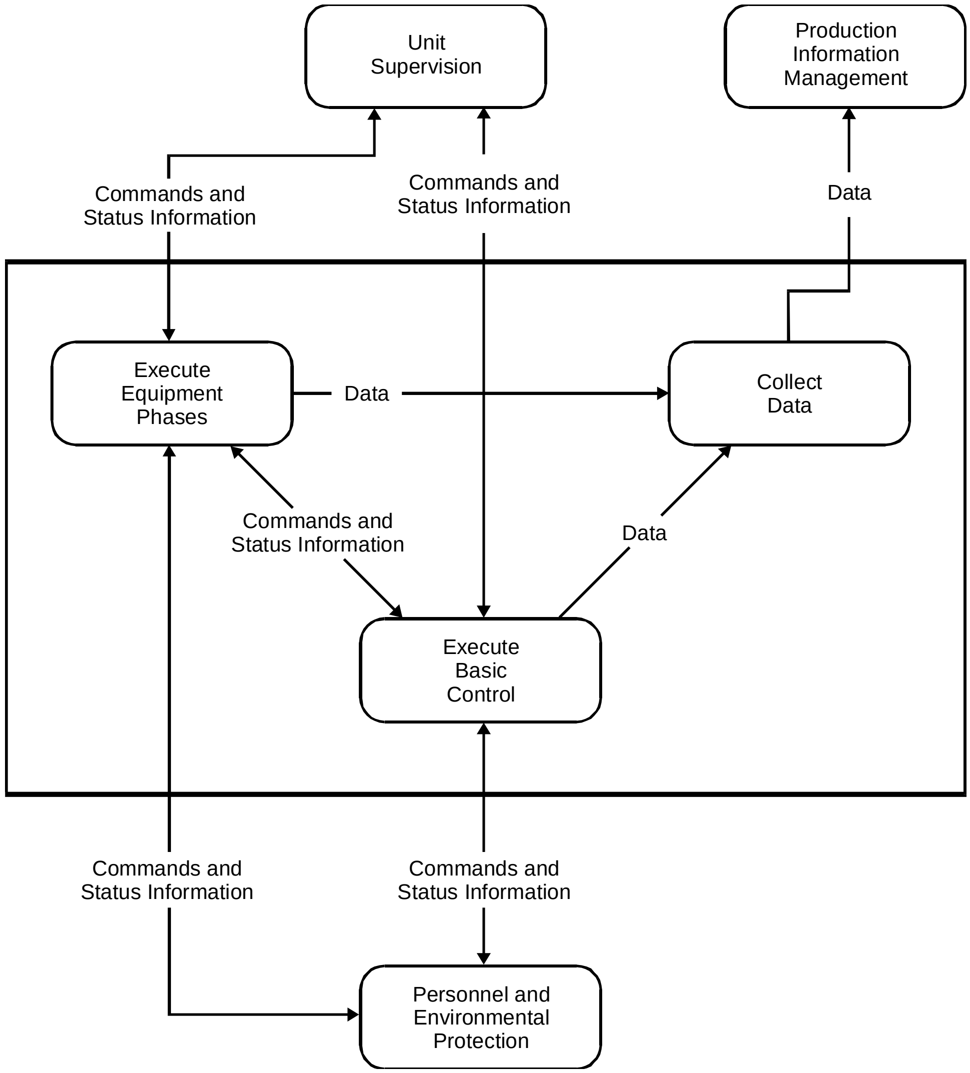


Figure 24 — Process control

6.7.1 Execute equipment phases

This is the control function in which equipment entities receive commands to perform procedural control described by an equipment phase. This control function is initialized by the *Acquire and execute procedural element* control function in Unit Supervision (6.6.1). By definition, the equipment phase is configured as part of an equipment entity. However, parameter values may be necessary in order to execute the equipment phase. The *Execute equipment phase* control function interprets the phase initialization command and associates the necessary parameters with the equipment phase. Equipment phases may be commanded and parameterized before or during execution. Equipment entities capable of performing this control function are equipment modules and units.

This control function does not act directly on physical equipment. It influences the process only through the basic control in a control module.

This control function also includes the supervision of equipment phase modes and states. This includes

— the propagation of modes and states from/to any procedural element and/or equipment entity;

— the propagation of modes and states from the unit or equipment module executing the equipment phase; and

— manual intervention into the execution of the equipment phase.

6.7.2 Execute basic control

Executing basic control is a control function that causes changes in equipment and process states by sending commands to actuators and other control modules. Commands to basic control may come from the execution of an equipment phase or from another control function, such as a manual command from an operator. Basic control uses input from sensors and other control functions in order to execute its function. The execution of this control function may also result in process, equipment, and other status information being provided to high level control functions. Some other basic control functions that may be included are exception handling, calculations, and treatment of operator-entered information, etc.

However, this control function does not contain procedural control and is always configured as part of the equipment entity. This control function also includes the association of the necessary parameters with the appropriate basic control function. Equipment entities capable of performing this control function are control modules, equipment modules, and units.

This control function also includes the supervision of equipment entity modes and states. This includes

— the propagation of modes and states from/to any equipment entities and/or procedural elements; and

— manual intervention.

Where the equipment entity is a common resource, this control function may also be involved in the arbitration of conflicting requests and commands.

6.7.3 Collect Data

In the *Collect Data* control function, data from sensors, derived values, and events that occur within the domain of Process Control are collected and stored in batch history. Data collection may be conditional. That is, certain data might not always be collected or might be sampled at a different time interval, depending upon information received from another control function, such as from parameters passed to the equipment phase.

6.8 Personnel and environmental protection

The Personnel and Environmental Protection control activity provides safety for people and the environment. It is shown in the Control Activity Model in Figure 19 (see Section 6.1.1) below Process Control because no other control activity should intervene between Personnel and Environmental Protection, and the field hardware it is designed to operate with. Personnel and Environmental Protection is, by definition, separate from higher level control activities. It may map to more than one level of equipment entity if that level of organization or sophistication is required to provide adequate safety protection.

Personnel and environmental protection is included in the control activity model to emphasize the importance of these types of protection systems and to indicate the point in the model appropriate for insertion of a separate protection system of this type. A complete discussion of personnel and environmental protection, the classification of these types of systems, and the segregation of levels of interlocks within these systems is a topic of its own and beyond the scope of this standard. More information on this topic can be obtained from some of the standards and guidelines that are under development (see References 1, 2, 3, 4, and 5 in Annex B).

Annex A — (normative) Model philosophy

A number of drawing formats have been used in this standard. Each of these drawing formats is discussed below.

The modeling formats discussed in this section provide a non-rigorous method of portraying information and relationships. They are not intended to recommend or imply an analysis methodology or to have the figures supersede the information described in the text.

— All Entity-Relationship (E-R) diagrams are shown using the format described in Figures A.1 and A.2. The description of the relationship will be shown in one direction only. Figure A.3 is an example.

— Entities are shown as rectangles in all drawings.

— Activities or functions are shown as rounded rectangles in all drawings. These drawings only show the explosion of one control activity per diagram. Lines between activities and between functions show information exchange. An example is Figure A.4.

— States are shown as ellipses in all drawings. Lines between states identify commands that cause the state changes. Figure A.5 is an example of a state transition diagram.

— Physical drawings use the ISA symbol standards, where applicable. Figure A.6 is an example.

— Nested drawings are only used where it is desirable to show a relationship between two different types of recipes. Figure A.7 is an example.

Basic Associations:

For each occurence of A, there is one and only one occurrence of B.

For each occurence of A, there is zero or one occurrence of B.

For each occurence of A, there is one or more occurrences of B.

For each occurence of A, there is zero, one, or more occurrences of B.

Looped Associations:

Any of the associations above may be used in a loop. Here, an occurrence of an entity is associated with one or more occurrences of entities of the same type.

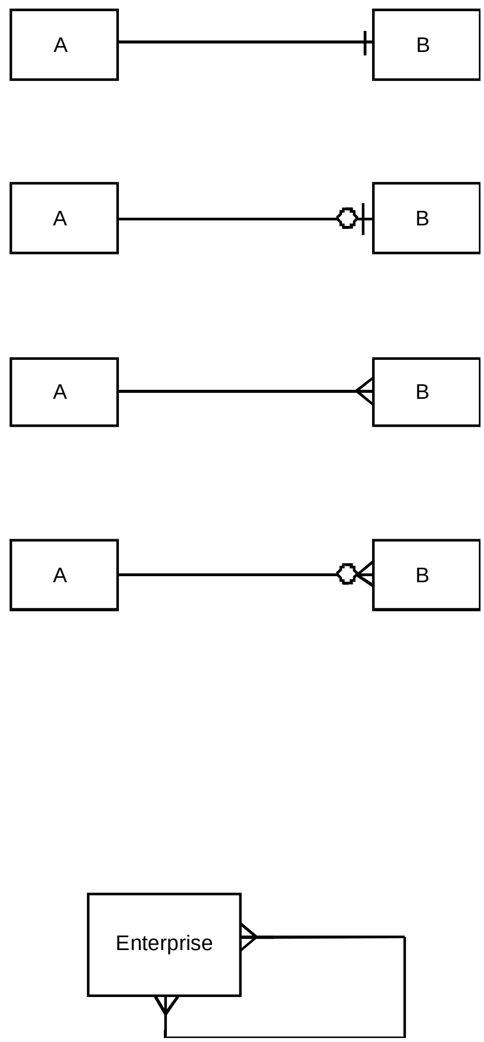


Figure A.1 — Basic and looped associations in Entity-Relationship diagrams

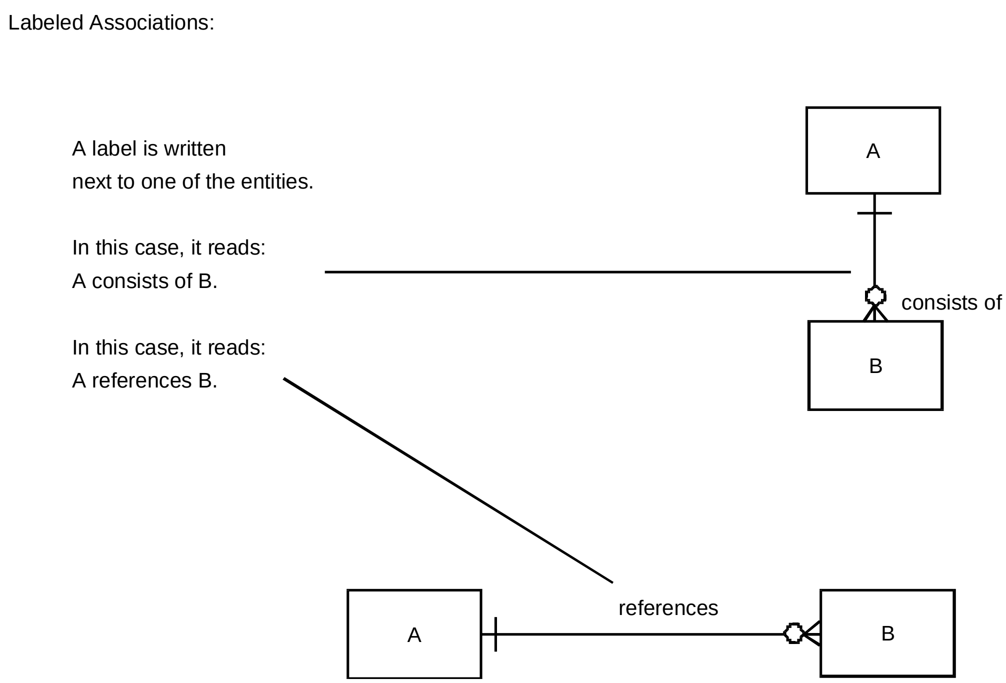


Figure A.2 — Labeled associations in Entity-Relationship diagrams

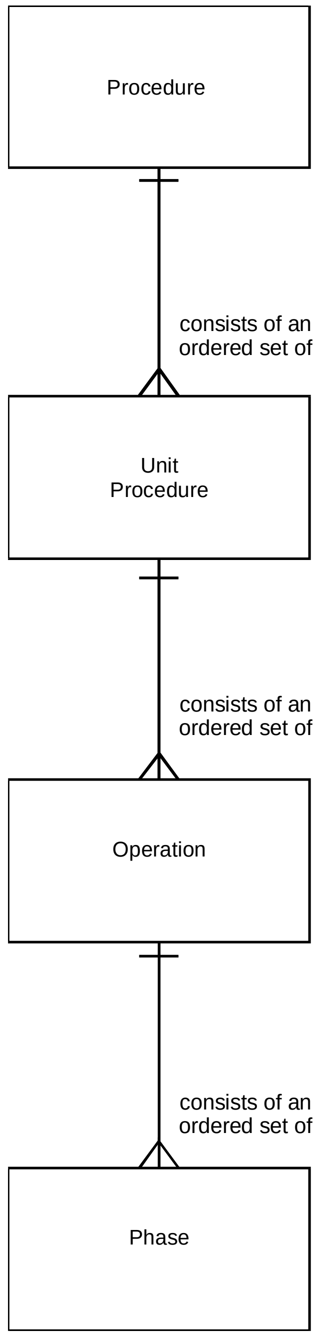


Figure A.3 — Process model (Entity-Relationship diagram)

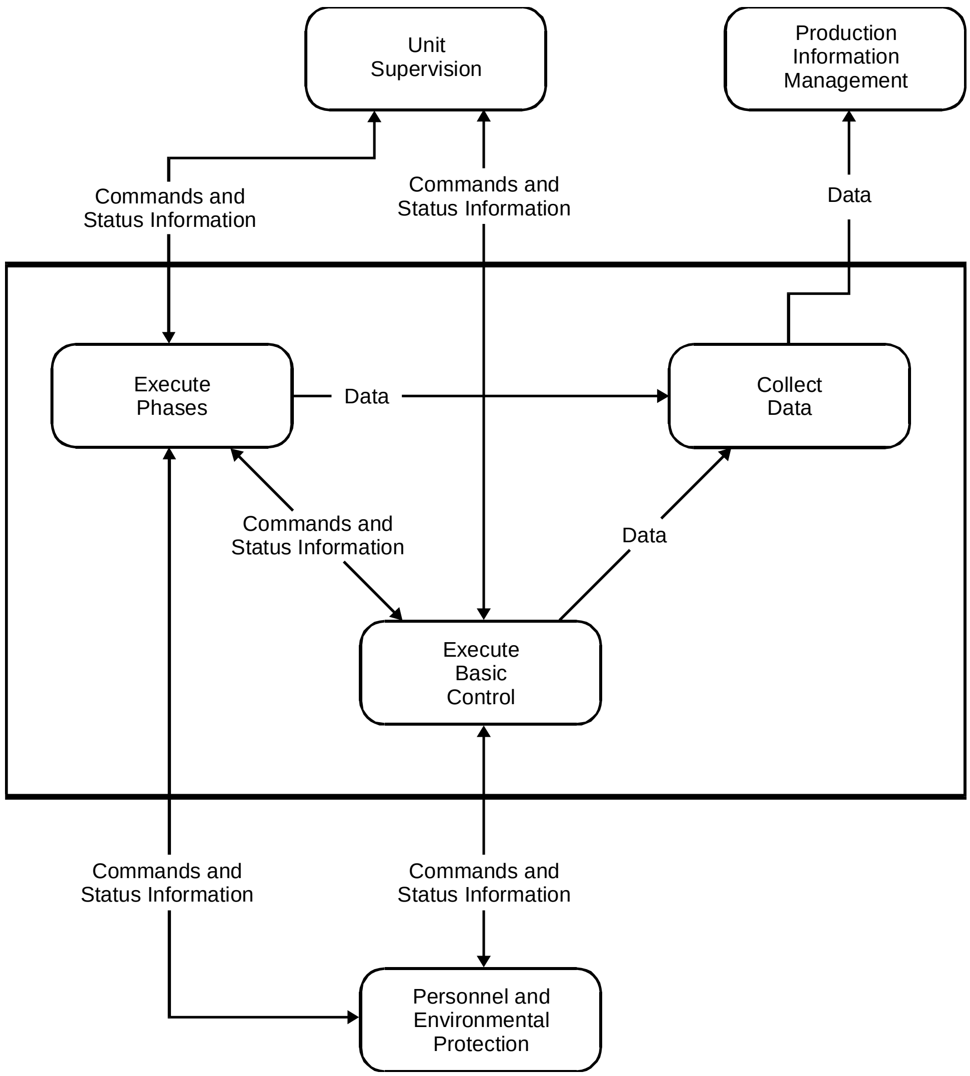


Figure A.4 — Process control (control activity with breakdown into control functions)

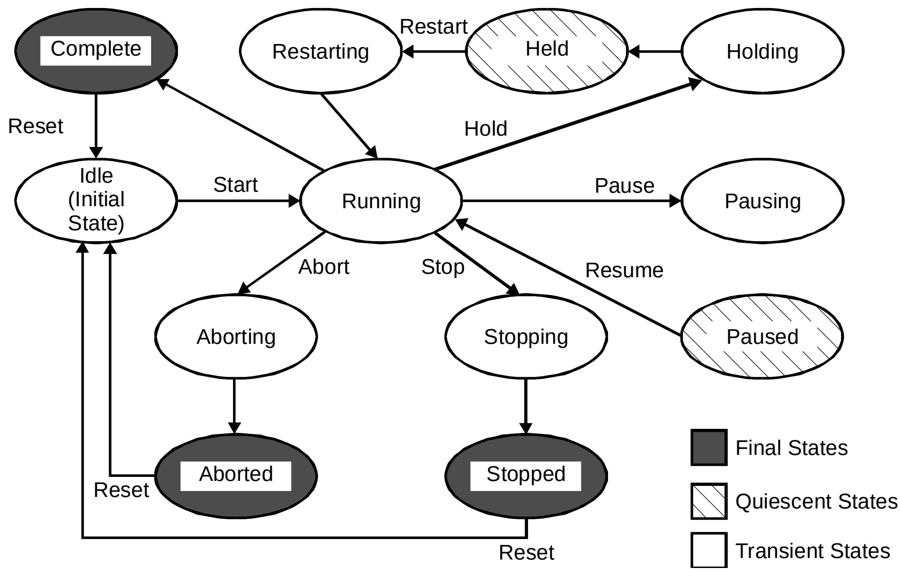


Figure A.5 — State transition diagram

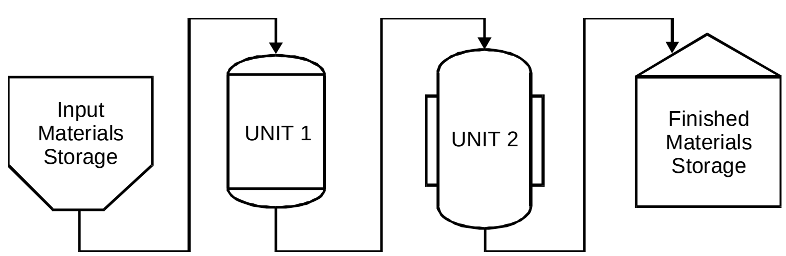


Figure A.6 — Single-path structure (physical drawing)

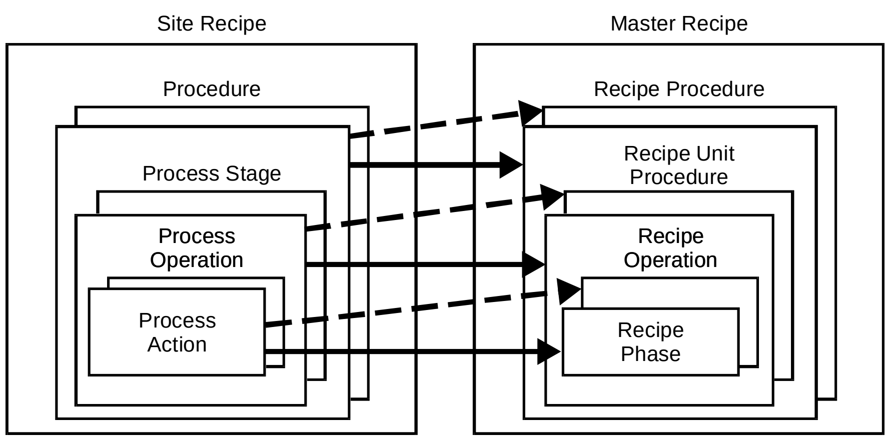


Figure A.7 — Site recipe procedure to master recipe procedure relationship (nesting model)

Annex B — (informative) Bibliography

1. ISA-dS84.01: Applications of Safety Instrumented Systems for the Process Industries, Instrument Society of America.

2. IEC SC65A/WG10, 65A (Secretariat) 122: Functional safety: safety-related systems. Part 1: General requirements, International Electrotechnical Commission.

3. IEC SC65A/WG9, 65A (Secretariat) 122: Functional safety: safety-related systems. Part 2: Requirements for Electrical/Electronic/Programmable electronic systems, International Electrotechnical Commission.

4. IEC SC65A/WG9, 65A (Secretariat) 122: Functional safety: safety-related systems. Part 3: Software requirements, International Electrotechnical Commission.

5. Guidelines for Safe Automation of Chemical Processes, Center for Chemical Process Safety, American Institute of Chemical Engineers, New York 1993.

Developing and promulgating technically sound consensus standards, recommended practices, and technical reports is one of ISA's primary goals. To achieve this goal the Standards and Practices Department relies on the technical expertise and efforts of volunteer committee members, chairmen, and reviewers.

ISA is an American National Standards Institute (ANSI) accredited organization. ISA administers United States Technical Advisory Groups (USTAGs) and provides secretariat support for International Electrotechnical Commission (IEC) and International Organization for Standardization (ISO) committees that develop process measurement and control standards. To obtain additional information on the Society's standards program, please write:

ISA
Attn: Standards Department
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, NC 27709

ISBN: 1-55617-562-0

AMERICAN NATIONAL STANDARD

ANSI/ISA–88.00.02–2001

Batch Control Part 2: Data Structures and Guidelines for Languages

Approved 7 February 2001

ISA–The Instrumentation, Systems, and Automation Society

ANSI/ISA–88.00.02–2001
Batch Control Part 2: Data Structures and Guidelines for Languages

ISBN: 1-55617-745-3

ISA
67 Alexander Drive
PO Box 12277
Research Triangle Park, NC 27709
USA

Preface

This preface, as well as all footnotes and annexes, is included for information purposes and is not part of ANSI/ISA-88.00.02-2001.

The standards referenced within this document may contain provisions that, through reference in this text, constitute requirements of this document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated within this document. Members of IEC and ISO maintain registers of currently valid International Standards. ANSI maintain registers of currently valid U.S. National Standards.

This document has been prepared as part of the service of ISA–The Instrumentation, Systems, and Automation Society, toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static, but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms and asks that they be addressed to the Secretary, Standards and Practices Board; ISA; 67 Alexander Drive; P. O. Box 12277; Research Triangle Park, NC 27709; Telephone (919) 549-8411; Fax (919) 549-8288; E-mail: standards@isa.org.

The ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards. The Department is further aware of the benefits to USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, this Department will endeavor to introduce SI-acceptable metric units in all new and revised standards, recommended practices, and technical reports to the greatest extent possible. *Standard for Use of the International System of Units (SI): The Modern Metric System*, published by the American Society for Testing & Materials as IEEE/ASTM SI 10-97, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

It is the policy of ISA to encourage and welcome the participation of all concerned individuals and interests in the development of ISA standards, recommended practices, and technical reports. Participation in the ISA standards-making process by an individual in no way constitutes endorsement by the employer of that individual, of ISA, or of any of the standards, recommended practices, and technical reports that ISA develops.

CAUTION — ISA ADHERES TO THE POLICY OF THE AMERICAN NATIONAL STANDARDS INSTITUTE WITH REGARD TO PATENTS. IF ISA IS INFORMED OF AN EXISTING PATENT THAT IS REQUIRED FOR USE OF THE STANDARD, IT WILL REQUIRE THE OWNER OF THE PATENT TO EITHER GRANT A ROYALTY-FREE LICENSE FOR USE OF THE PATENT BY USERS COMPLYING WITH THE STANDARD OR A LICENSE ON REASONABLE TERMS AND CONDITIONS THAT ARE FREE FROM UNFAIR DISCRIMINATION.

EVEN IF ISA IS UNAWARE OF ANY PATENT COVERING THIS STANDARD, THE USER IS CAUTIONED THAT IMPLEMENTATION OF THE STANDARD MAY REQUIRE USE OF TECHNIQUES, PROCESSES, OR MATERIALS COVERED BY PATENT RIGHTS. ISA TAKES NO POSITION ON THE EXISTENCE OR VALIDITY OF ANY PATENT RIGHTS THAT MAY BE INVOLVED IN IMPLEMENTING THE STANDARD. ISA IS NOT RESPONSIBLE FOR IDENTIFYING ALL PATENTS THAT MAY REQUIRE A LICENSE BEFORE IMPLEMENTATION OF THE STANDARD OR FOR INVESTIGATING THE VALIDITY OR SCOPE OF ANY PATENTS BROUGHT TO ITS ATTENTION. THE USER SHOULD CAREFULLY INVESTIGATE RELEVANT PATENTS BEFORE USING THE STANDARD FOR THE USER'S INTENDED APPLICATION.

HOWEVER, ISA ASKS THAT ANYONE REVIEWING THIS STANDARD WHO IS AWARE OF ANY PATENTS THAT MAY IMPACT IMPLEMENTATION OF THE STANDARD NOTIFY THE ISA STANDARDS AND PRACTICES DEPARTMENT OF THE PATENT AND ITS OWNER.

ADDITIONALLY, THE USE OF THIS STANDARD MAY INVOLVE HAZARDOUS MATERIALS, OPERATIONS OR EQUIPMENT. THE STANDARD CANNOT ANTICIPATE ALL POSSIBLE APPLICATIONS OR ADDRESS ALL POSSIBLE SAFETY ISSUES ASSOCIATED WITH USE IN HAZARDOUS CONDITIONS. THE USER OF THIS STANDARD MUST EXERCISE SOUND PROFESSIONAL JUDGMENT CONCERNING ITS USE AND APPLICABILITY UNDER THE USER'S PARTICULAR CIRCUMSTANCES. THE USER MUST ALSO CONSIDER THE APPLICABILITY OF ANY GOVERNMENTAL REGULATORY LIMITATIONS AND ESTABLISHED SAFETY AND HEALTH PRACTICES BEFORE IMPLEMENTING THIS STANDARD.

This standard is structured to follow IEC (International Electrotechnical Commission) guidelines. Therefore, the first three clauses discuss the *Scope* of the standard, *Normative References*, and *Definitions*, in that order.

Clause 4 is entitled *Data Model*. The intent of this clause is to describe a data structure for batch control systems using an object model approach.

Clause 5 is entitled *Relational Tables for Information Exchange*. The intent of this clause is to discuss a data format that can be used to share recipes and other batch information across different systems.

Clause 6 is entitled *Procedure Function Charts*. The intent of this clause is to describe a symbolic language for recipe depiction.

This standard is intended for people who are

a) involved in designing and/or operating batch manufacturing plants;

b) responsible for specifying controls and the associated application programs for batch manufacturing plants; or

c) involved in the design and marketing of products in the area of batch control.

The following people served as active members of ISA CommitteeSP88:

NAME	**COMPANY**
L. Craig, Chairman	Rohm and Haas Co.
T. Fisher, Editor	The Lubrizol Corp.
M. Albano	Honeywell, Inc.
J. Barrault	Siemens AG
D. Brandl	Sequencia Corp.
B. Braunstein	Exxon Chemical Co.
E. Bristol	The Foxboro Co.
L. Charpentier	GSE Systems Inc.
T. Crowl	Siemens Moore Process Automation
C. Eaves	Intellution, Inc.
D. Emerson	Yokogawa Corp. of America
L. Falkenau	E.I. du Pont de Nemours and Co.
M. Fersky	ABB Automation Inc.
D. Fleming	Rockwell Automation
A. Ghosh	ARC

R. Hall	Sequencia Corp.
W. Hawkins	HLQ Ltd.
N. Haxthausen	Novo Nordisk Engineering
B. Jensen	Yokogawa Corp. of America
B. Korkmaz	Automation Vision Inc.
D. Kraska	Geon Co.
T. McFarlane	Neles Automation Inc.
R. Mergen	The Lubrizol Corp.
D. Moffatt	Wonderware Corp.
P. Moylan	Rockwell Automation
T. Müller-Heinzerling	Siemens AG
L. Natiello	Merck
P. Nelson	Dow Corning Corp.
L. Noble	ABB Automation, Inc.
L. Pillai	Pharmacia & Upjohn, Inc.
S. Prichard	GE Fanuc Automation
T. Province	Eastman Chemical Co.
R. Salisbury	ABB Industrial Systems Inc.
M. Saucier	Sequencia Corp.
D. Sweeny	PharmComm Inc.
B. Sykes	Yokogowa Corp. of America
T. Tom	Siemens Moore Process Automation
K. Unger	Real Enterprise Solutions
M. Van Epps	Pharmacia Corp.
J. Vieille	Jean Vieille Conseil
A. Webster	E.I. du Pont de Nemours and Co.
A. Weidenbach	Eastman Chemical Co.
R. Winslow	Sterling Electronics
B. Winters	Honeywell IAC
D. Wolin	Siemens Moore Process Automation

This standard was approved for publication by the ISA Standards and Practices Board on 03 January 2001.

NAME	**COMPANY**
M. Zielinski	Fisher-Rosemount Systems, Inc.
D. Bishop	Consultant
M. Cohen	Senior Flexonics, Inc.
M. Coppler	Ametek, Inc.
B. Dumortier	Schneider Electric SA
W. Holland	Southern Company
E. Icayan	Advanced Control & Engineering Solutions
A. Iverson	Ivy Optiks
R. Jones	Dow Chemical Co.
V. Maggioli	Feltronics Corp.
T. McAvinew	Bateman Engineering, Inc.
A. McCauley, Jr.	Chagrin Valley Controls, Inc.
G. McFarland	Westinghouse Process Control Inc.
D. Rapley	Rapley Consulting Inc.
R. Reimer	Rockwell Automation
J. Rennie	Factory Mutual Research Corp.
H. Sasajima	Yamatake Corp.
I. Verhappen	Syncrude Canada Ltd.

R. Webb	Altran Corp.
W. Weidman	Parsons Energy & Chemicals Group
J. Weiss	EPRI
M. Widmeyer	EG&G Defense Materials
R. Wiegle	CANUS Corp.
C. Williams	Eastman Kodak Co.
G. Wood	Graeme Wood Consulting

Contents

Foreword

1) The formal decisions or agreements of the IEC on technical matters, prepared by technical committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.

2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.

3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules insofar as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rules should, as far as possible, be clearly indicated in the latter.

4) The IEC has not laid down any procedure concerning marking as an indication of approval and has no responsibility when an item of equipment is declared to comply with one of its recommendations.

This standard has been prepared by IEC SC65A WG11 and ISA SP88.

It forms Part 2 of a series, the other part being ANSI/ISA-88.01-1995, Batch Control Part 1: Models and Terminology.

Annexes A and B form an integral part of this standard. Refer to annex A for an explanation of the UML notation that is used in this standard. Refer to annex B for a summation of all of the SQL definitions from clause 5. Annexes C and D are for information only.

Introduction

ANSI/ISA-88.01-1995, Batch Control Part 1: Models and Terminology (referred to as Part 1 throughout this standard), provides models and terminology applicable to batch control. This standard (referred to as Part 2 throughout the standard) addresses data structures and guidelines for languages. Data structures are addressed by the data model that is defined in clause 4 that more precisely identifies objects and relationships that are addressed by models and concepts of Part 1. Data structures are also addressed by relational tables for information exchange that are defined in clause 5. Languages are addressed by a recipe depiction methodology that is defined in clause 6.

The intended use of the data model is to provide a starting point for developing interface specifications for software components that address any subset of the Part 1 standard. The data model addresses all of the Part 1 standard as an integrated object model, but it does not presume or preclude any specific system architecture or information exchange. The model does not assume any specific division of functionality between systems.

A specific method for the exchange of selected data is defined in clause 5. Relational tables are used as the information exchange method because, within the bounds of the information treated, they

a) utilize broadly available technologies;

b) are amenable to translation to other technologies;

c) are adequate; and

d) are consistent with other sections of the standard.

Multiple methods of information transfer have not been defined, nor has there been an attempt to identify all information that might be exchanged. In the future, additional methods may be defined to provide alternate ways to exchange data.

Clause 6 defines the symbols and rules for a graphical language that can be used to depict recipes. Recipes are the central feature of batch control, and they can address a wide range of complexity, but there is no one depiction that is ideal for all circumstances. A simple table, for example, might be the most appropriate recipe form for simple cases. This standard specifies a method for depiction of master and control recipe procedures that can be applied over a broader range of complexity.

Although this standard is intended primarily for batch processes, there may be considerable value for other types of processes.

1 Scope

This Part 2 standard on Batch Control defines data models that describe batch control as applied in the process industries, data structures for facilitating communications within and between batch control implementations, and language guidelines for representing recipes.

2 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions this standard. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid normative documents.

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

IEC 60902:1987, *Industrial-process measurement and control: Terms and definitions*

NOTE — IEC 60902:1987 has been replaced by IEC 60050-351:1998, International electrotechnical vocabulary-Part 351: Automatic control

IEC 61131-3:1993, *Programmable controllers – Part 3: Programming languages*

IEC 61512-1:1997, *Batch control – Part 1: Models and terminology*

ANSI/ISA-88.01-1995, *Batch Control – Part 1: Models and Terminology*

ISO/IEC 9075:1992, *Information processing systems – Database language SQL with integrity enhancement*

3 Definitions

For the purposes of this standard, the following definitions apply. Definitions and concepts expressed in ANSI/ISA-88.01-1995 apply, except where differences are explicitly stated in this Part 2 standard. Definitions in IEC 60902:1987 were also used as a basis.

3.1 allocation symbol:
a graphical symbol that is used to represent the encapsulation of the resource allocation and de-allocation rules for a recipe procedural element.

3.2 building block:
a recipe entity that exists in a library.

3.3 enumeration set:
a list of predefined strings and their associated numerical values.

3.4 exchange table:
a database table that is used to exchange batch-related information between systems.

3.5 link:
an object that specifies the connection between two other objects (e.g., the connection between recipe entities or between recipe entities and transitions).

3.6 procedure function chart:
a graphical representation of a recipe procedure that specifies the processing order for recipe procedural elements.

3.7 recipe element:
a structural entity that is used to represent recipe entities and symbols, except transitions and directed links, that are used in procedure function charts.

3.8 recipe entity:
the combination of a procedural element with associated recipe information (e.g., header, formula, equipment requirements, other information). General, site, master, and control recipes are also recipe entities.

4 Data model

4.1 Introduction

This clause defines data models that specify a set of objects, attributes, and their basic relationships that cover the concepts of Part 1 at a high level of abstraction. The models apply to interfaces to batch control systems in a technology-independent manner. The models are not intended to address the internal system architecture of batch control systems.

The intended use of these models is to provide a starting point for developing interface specifications for software components that address any subset of Part 1.

The models address all of Part 1 as an integrated object model, but they do not presume or preclude any specific system architecture or information exchange. The models do not assume any specific division of functionality between systems.

In the cases where the objects and relationships defined in this clause are presented through an interface, then that interface shall use the object names, the attribute names, and the relationships of this clause commensurate with the interface technology chosen and the capabilities offered. An example of such an interface is the SQL relational table interface that is defined in clause 5.

An exchange format or interface specification may implement only some objects or parts of objects (e.g., not all properties are defined). An exchange format or interface specification may also provide additional objects or properties (e.g., phase duration information), including the expansion of any data model attribute into multiple attributes. Any such implementation shall be consistent with the data model that is presented here and the concepts of Part 1.

4.1.1 Modeling techniques

The models that are described in this clause are based on the Unified Modeling Language (UML) (see clause A.1).

The tables describe only the class attributes of the objects. The relationships between objects are described in the figures.

4.2 Overview model

This model (see figure 1) gives a high level overview of the main classes that are defined and the relations between these classes for the batch domain that is described by the Control Activity Model of Part 1. The individual object classes are more thoroughly described in the subsequent models in this subclause.

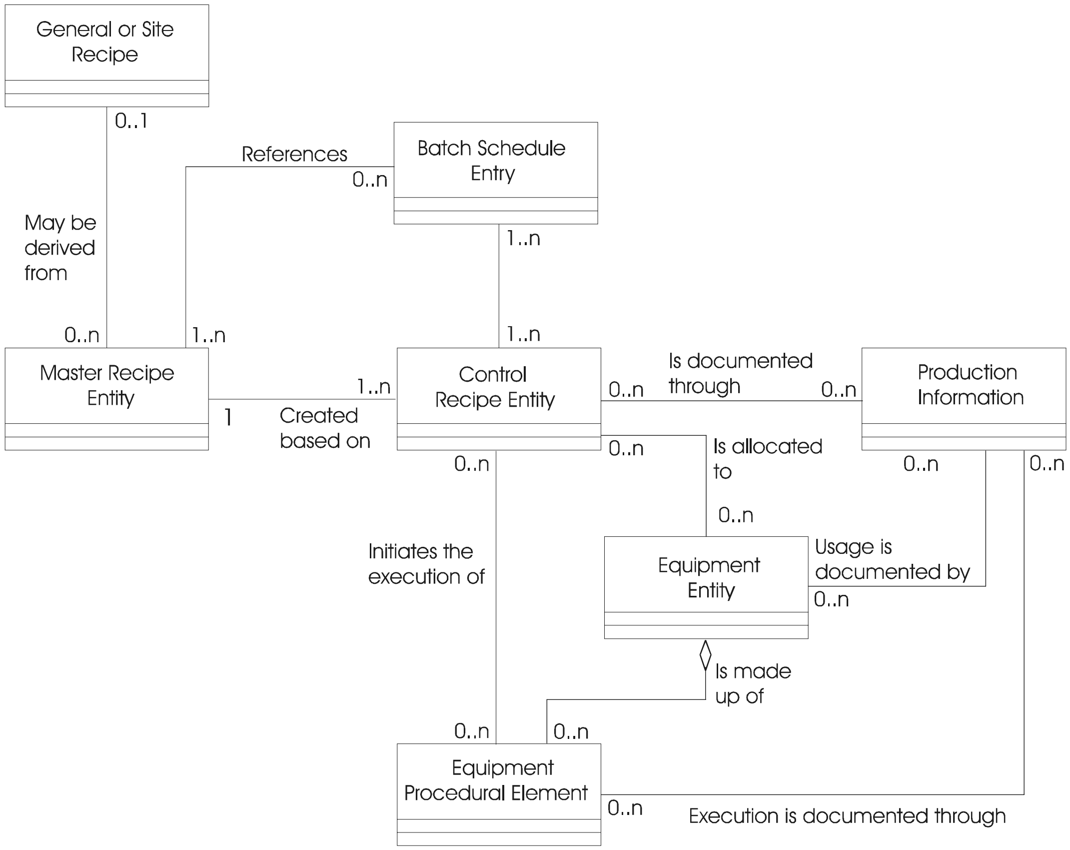


Figure 1 — Overview model

A general or site recipe is made up of a hierarchy of *general recipe entities* that correspond to the procedural entities (process stages, process operations, and process actions).

A master recipe may be derived from a general or site recipe. The master recipe itself may be seen as a top-level master recipe entity. A master recipe is made up of a hierarchy of *master recipe entities* that correspond to the procedural entities (i.e., procedures, unit procedures, operations, phases).

A *batch schedule entry* specifies production of a specific batch through the execution of a recipe. A batch schedule is, in principle, a list that specifies the production of batches, and it includes information about timing. The product-specific information necessary for this purpose is derived from a related *master recipe entity*.

Based on a *Batch Schedule Entry*, a control recipe starts as a copy of a specific version of a master recipe, and it is then modified to create the recipe that will produce the batch. A control recipe includes the information necessary for equipment control.

Control recipe entities are created based on *master recipe entities*. A control recipe may be enhanced with additional information (e.g., scaling, equipment assignment), and it may be modified (including creating or removing control recipe entities).

Equipment entities are selected and allocated to the *control recipe entities.*

A *control recipe entity* may be linked to an *equipment procedural entity* within the *equipment entity* (normally the unit). The *equipment procedural entity* may be initiated, and its parameters may be assigned recipe values.

Production information is generated during the production of the batch. This information may be related to recipe entities, equipment entities, and/or equipment procedural elements.

4.3 Recipe model

4.3.1 Recipe entity

Recipes are organized hierarchically, with various categories of information at each level. The recipe entity is the construct that is used to represent the tight coupling of the data at a particular level.

The recipe entity is the fundamental structure in all kinds of recipes (see figure 2). The recipe entity structurally takes the place of the recipe procedural element as defined in Part 1, but it may include any or all of the recipe components: procedural definitions, parameters with values, equipment requirements and other information.

The class specifications are shown in table 1.

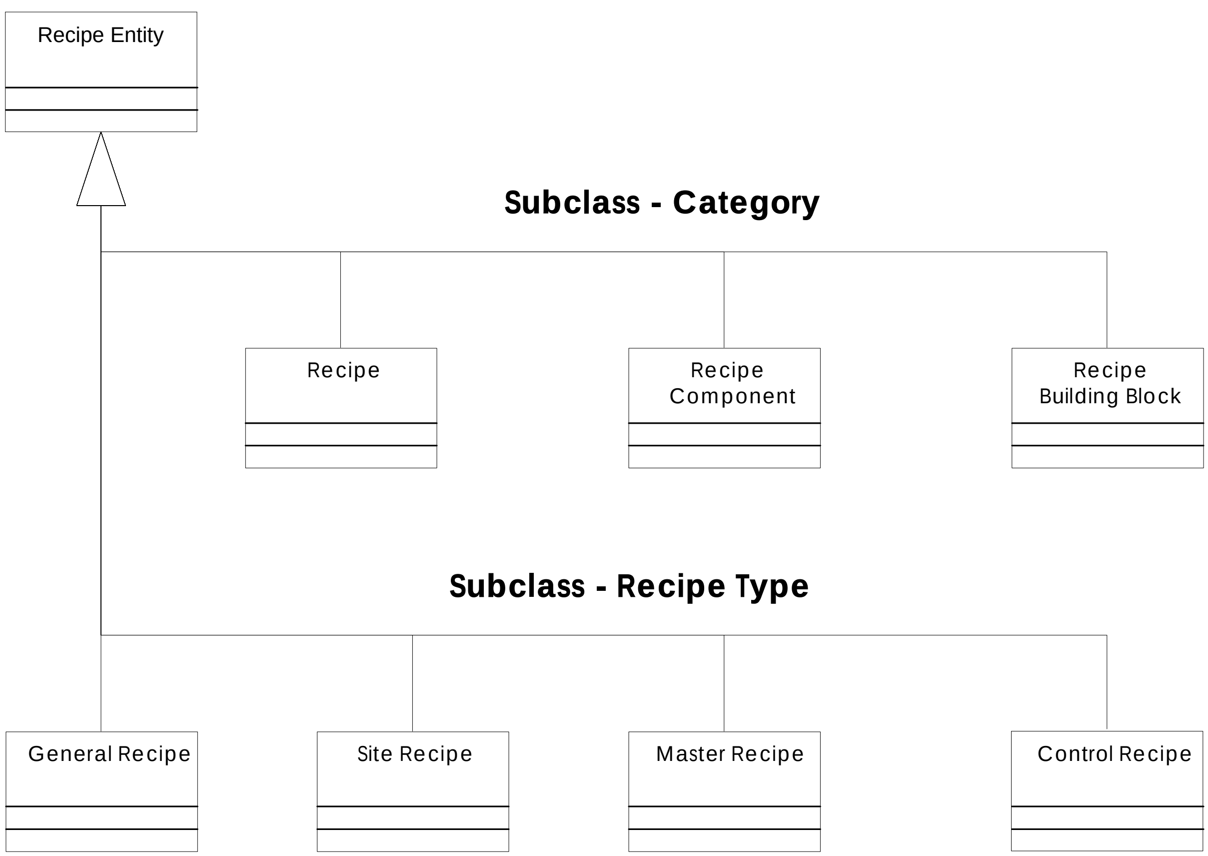


Figure 2 — Recipe entities

Table 1 — Recipe entity

NAME	RECIPE ENTITY
FunctionalDescription	A recipe entity is the combination of a procedural element with associated recipe information (e.g., header, formula, equipment requirements and other information). General, site, master, and control recipes are recipe entities. One example is a unit recipe (ANSI/ISA-88.01-1995, 5.3.2). "unit recipe: the part of a control recipe that uniquely defines the contiguous production requirements for a unit." (ANSI/ISA-88.01-1995, 3.62).
ATTRIBUTES	
RecipeEntityID	Provides unique Identification.

A recipe itself is considered to be a recipe entity (category: recipe). The recipe is built up of lower-level recipe entities (e.g., unit recipes) (category: component). When constructing a specific recipe, the components may be created from library elements (category: building block).

The recipe entity concept is applied to all recipe types: general, site, master and control. When a recipe is executed, the representation in the batch history of the executed recipe entities will have much the same structure, and it is therefore shown as a subclass. An overview of the subclasses is shown in table 2. Subclasses by category are shown in tables 3 to 5. Subclasses by type are shown in tables 6 to 9. General and site recipes are not discussed further in this subclause.

Table 2 — Subclasses – overview

	GENERAL RECIPE ENTITY	SITE RECIPE ENTITY	MASTER RECIPE ENTITY	CONTROL RECIPE ENTITY
RECIPE	A complete and self-contained general recipe.	A complete and self-contained site recipe.	A complete and self-contained master recipe.	A complete and self-contained control recipe.
RECIPE BUILDING BLOCK	A generic general recipe entity type that can be instantiated in a specific recipe or in another building block.	*Building blocks for site recipes may not exist. Site recipes are normally modified using general recipe building blocks.*	A generic master recipe entity type that can be instantiated in a specific recipe or in another building block.	*Building blocks for control recipes do not exist. Control recipes are modified using master recipe building blocks.*
RECIPE COMPONENT	A component of a general recipe or library element - may be an instantiation of a building block.	A component of a site recipe or library element - may be an instantiation of a general recipe building block.	A component of a master recipe or library element - may be an instantiation of a building block.	A component of a control recipe - may be an instantiation of a master recipe building block.

Table 3 — Recipe

NAME	RECIPE
FunctionalDescription	The top level recipe entity.
ATTRIBUTES	
RecipeID	Identifies the recipe. When combined with the "RecipeVersion," defines a unique instance of a recipe.
RecipeVersion	Identifies the version of a recipe. When combined with a "RecipeID," defines a unique instance of a recipe (e.g., Red Oak - A10.3).
VersionDate*	Identifies the date and time that this version of the recipe was created or modified.
ApprovalDate*	Identifies the date and time that this version of the recipe was approved.
EffectiveDate*	Identifies the earliest date and time that this version of the recipe may be used.
ExpirationDate*	Identifies the date and time that this version of the recipe expires.
ProductID*	Identifies the product or product family that would be created by execution of this version of the recipe (e.g., Premium Beer).
Author*	Identifies the person or system that authored this version of the recipe. (e.g., J. Smith).
ApprovedBy*	Identifies the person or system that approved this version of the recipe.
Description	Describes this version of the recipe and/or product (e.g., North Carolina’s Finest Premium Beer).
Status*	Defines the Status of the information (e.g., "Approved for Production", "Approved for Test", "Not Approved", "Inactive", "Obsolete").
*Not required for control recipe (available by reference to master recipe).	

Table 4 — Recipe component

NAME	RECIPE COMPONENT
FunctionalDescription	A recipe entity that is part of a recipe or building block (i.e., an instance of a building block in a particular recipe or containing building block recipe entity).
ATTRIBUTES	
Level	Indicates the procedural hierarchy level (i.e., process stage, process operation, or process action for general and site recipes and unit procedure, operation, or phase for master and control recipes).
RE_Use	Defines if the recipe component is a copy of a building block or a reference to it.

Table 5 — Recipe building block

NAME	RECEIPE BUILDING BLOCK
FunctionalDescription	A recipe entity that exists in a library. A building block can be parameterized and used when building recipes.
ATTRIBUTES	
RecipeVersion	Identifies the version of the recipe entity.
VersionDate	Identifies the date and time that this version of the recipe was created or modified.
ApprovalDate	Identifies the date and time that this version of the recipe was approved.
Author	Identifies the person or system that authored this version of the recipe (e.g., J. Smith).
ApprovedBy	Identifies the person or system that approved this version of the recipe.
Description	Describes the function that is achieved through execution of this version of the recipe entity.
Level	Indicates the level of the recipe entity.
UsageConstraint	Defines other rules that determine the usage (e.g., “always succeeded by...” or “never runs in parallel with...”).
Status	Defines the Status of the recipe entity (e.g., “Approved for Production”, "Approved for Test", "Not Approved", "Inactive", "Obsolete").
Function	Defines how the recipe entity is executed (e.g., by referencing an equipment procedural element, through execution of contained logic).

Table 6 — General recipe entity

NAME	GENERAL RECIPE ENTITY
FunctionalDescription	All of a general or site recipe, a component of a general or site recipe, or a building block for creation of a general or site recipe.
ATTRIBUTES	
ScaleReference	Defines a reference scale for the parameter values.

Table 7 — Site recipe entity

NAME	SITE RECIPE ENTITY
FunctionalDescription	All of a site recipe, a component of a site recipe, or a building block for creation of a site recipe.
ATTRIBUTES	
ScaleReference	Defines a reference scale for the parameter values.

Table 8 — Master recipe entity

NAME	MASTER RECIPE ENTITY
FunctionalDescription	All of a master recipe, a component of a master recipe, or a building block for creation of a master recipe.
ATTRIBUTES	
ScaleReference	Defines a reference scale for the parameter values.
ProcessCellID	Identifies the equipment category for which the recipe entity was defined (e.g., process cell or process cells for which this master recipe was defined).

Table 9 — Control recipe entity

NAME	CONTROL RECIPE ENTITY
FunctionalDescription	A recipe entity that is all of or a part of a control recipe.
ATTRIBUTES	
BatchID	Specifies the actual Batch ID.
BatchSize	Defines the requested size or scale factor for the batch, based on the scale factor for the batch as defined in the master recipe.
Status	Describes execution status (e.g., not yet activated, active, or completed).

4.3.2 Parts of recipe entities

The model (see figure 3 and tables 10 through 13) shows the categories of information of a recipe as specified in Part 1. The model indicates that these components may exist at any level of the decomposition of the recipe (e.g., a unit recipe may have its own equipment requirements).

a) The Header category of information is contained in attributes to the recipe entity itself, instead of being a distinct object class in this model.

b) The Formula category of Part 1 is modeled as a set of parameter objects. All levels of the recipe decomposition may have parameters, including the recipe itself. See 4.3.6 on parameters.

c) The modeling of Equipment Requirements is discussed in 4.3.5.

d) The Other Information category, as defined in Part 1, is represented as a single object class, even though other information may have multiple elements and different structures.

e) The Procedure category of Part 1 is modeled as a set of procedural structural elements.

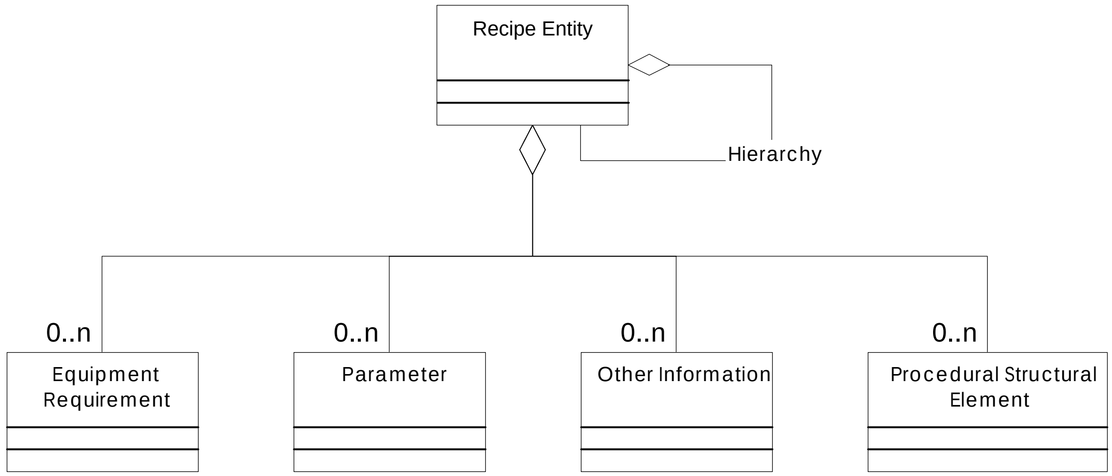


Figure 3 — Parts of recipe entities

Table 10 — Parameter

NAME	PARAMETERS
FunctionalDescription	Formula values or placeholders for values that are to be communicated to and from recipe entities during execution.
ATTRIBUTES	
ParameterID	Provides unique identification.
ParameterType	Specifies how the parameter value is interpreted (e.g., constant, reference equation).
Description	Describes the parameter or use of the parameter.
EngineeringUnits	Identifies the engineering units of measure for the Value (e.g., kg, pounds).
Value	Contains the parameter value. If Value is a relation, it contains the equation, form, deferral rule or whatever ties the related parameters together. If Value is a building block parameter, this attribute will hold the default value.
Scaled	Specifies the scaling rule. Simplest case: scaled or not scaled with batch reference size.
Usage	Specifies the parameter as a process input, process output, or process parameter.

Table 11 — Equipment requirement

NAME	EQUIPMENT REQUIREMENT
FunctionalDescription	Represents an equipment requirement as specified in the recipe entity.
ATTRIBUTES	

Table 12 — Other information

NAME	OTHER INFORMATION
FunctionalDescription	A category of recipe information that may contain batch processing support information that is not contained in other parts of the recipe (e.g., regulatory compliance information, materials and process safety information, process flow diagrams, packaging/labeling information).
ATTRIBUTES	

Table 13 — Procedural structural element

NAME	PROCEDURAL STRUCTURAL ELEMENT
FunctionalDescription	The recipe procedural elements and the ordering information for their execution.
ATTRIBUTES	

4.3.3 Recipe entity relation (procedural structure)

Recipe entities are decomposed hierarchically along the structures for procedural entities as defined in Part 1 (i.e., a recipe procedure contains unit procedures that contain operations that contain phases). This hierarchy is modeled using recursive containment. Higher-level objects may contain lower-level objects.

Procedural structural elements include the recipe procedural elements and the connections (e.g., link, transition) that are used to order them (e.g., a unit recipe's procedural structural elements are the operations and the ordering of those operations contained within it). The procedural structural elements may be related to other procedural structural elements.

4.3.4 Recipe building blocks

Recipe building blocks are an important concept in the data model (see figure 4). This figure represents the relations at a single level of the procedure hierarchy.

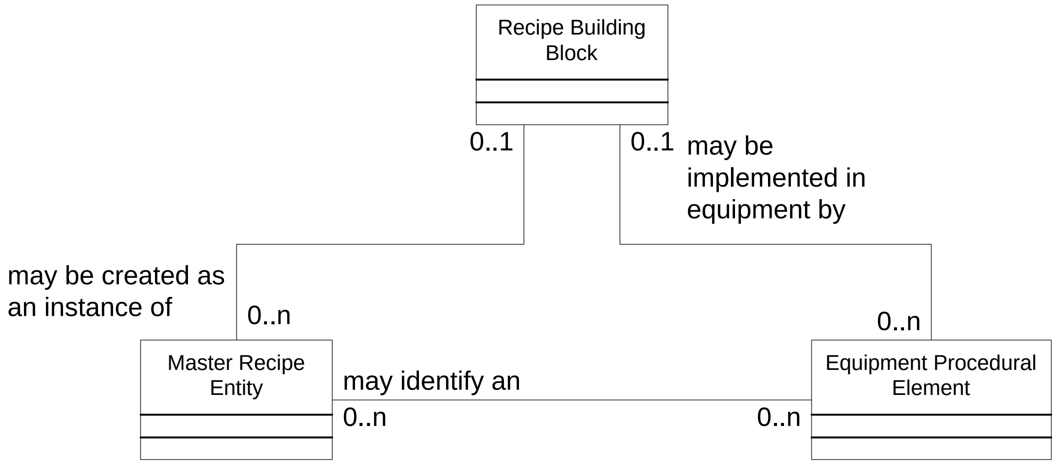


Figure 4 — Recipe building block

Recipe building blocks are the building blocks from which master recipes are created. When a *recipe building block* is instantiated in a master recipe as a m*aster recipe entity,* it may carry parameters, equipment requirements, and other information that may be assigned master recipe specific values. Lower-level contents of the *recipe building block* (e.g., subordinate recipe entities) may be copied into *master recipe entities.* These same lower-level contents may also be accessible by reference to the *recipe building block.*

A r*ecipe building block's* functionality may be implemented in equipment through e*quipment procedural elements* (see table 14), which is necessary for the execution of the lowest-level recipe entities (i.e., the recipe entities that are intended to be linked to equipment procedural elements).

Table 14 — Equipment procedural element

NAME	EQUIPMENT PROCEDURAL ELEMENT
FunctionalDescription	A procedural element that is associated with a piece of equipment (e.g., an equipment phase or equipment operation).
ATTRIBUTES	
EquipmentProcedural ElementID	Provides unique identification.
Version	Identifies the version of the procedural element.
VersionDate	Identifies the date and time that this version was created or modified.
ApprovalDate	Identifies the date and time that this version was approved.
Author	Identifies the person or system that authored this version (e.g., J. Smith).
ApprovedBy	Identifies the person or system that approved this version.
Description	Describes the function that is achieved through execution of the recipe entity.
Level	Indicates the level of the equipment entity. The equipment entity may only be used at this level.
Mode	Indicates the current mode of the procedural element.
State	Indicates the current state of the procedural element.

The mechanisms may be illustrated by the following example (see figure 5) that could represent an object model of a part of an actual application. The generic concept of a building block is instantiated as a specific building block "heat." The "is based on" relation between building blocks and components is replaced by a subclass relation (this is one possible implementation, and it indicates that if the "heat" building block is changed, then the change will propagate to all recipes that use "heat"). Another implementation could be that "heat" is just copied when instantiated.

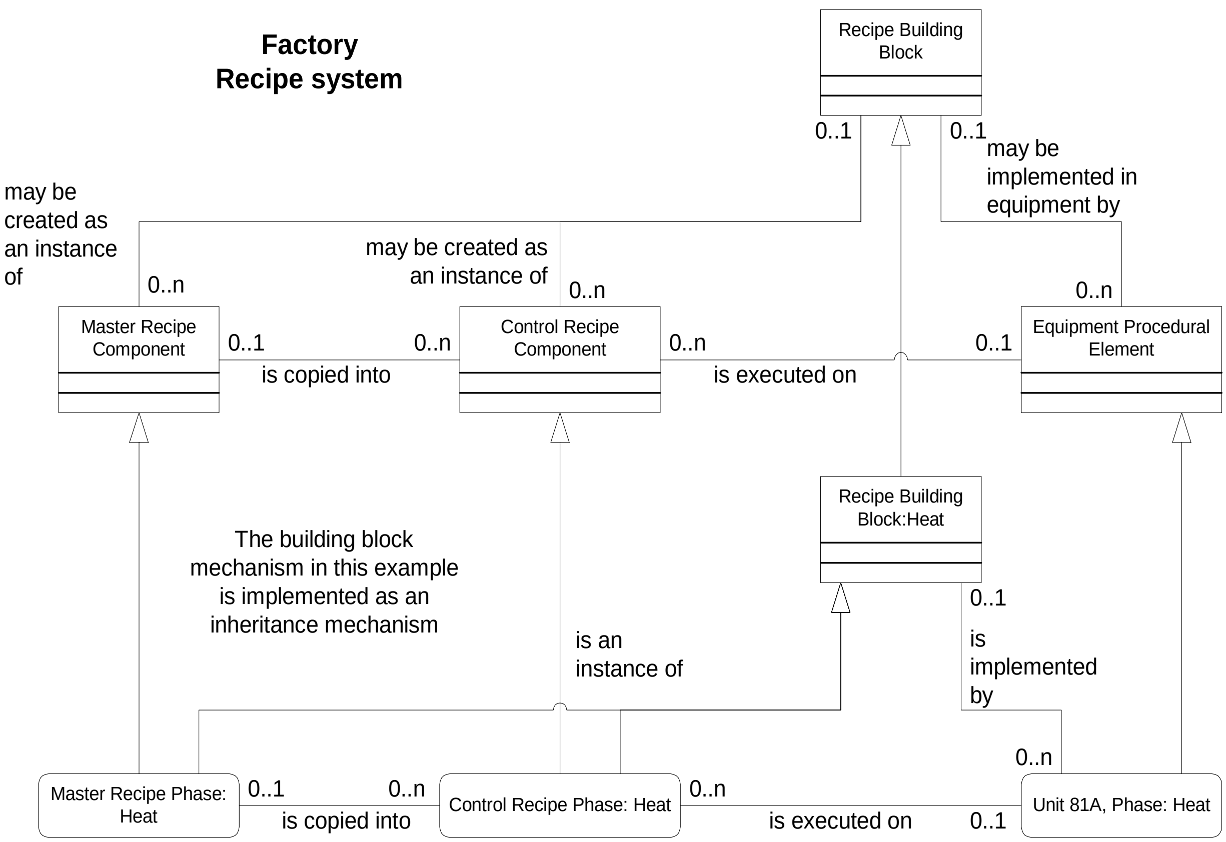


Figure 5 — Building block concept

4.3.5 Equipment requirements

Recipe entities may contain *equipment requirements* (see figure 6 and tables 15, 16, and 17). Equipment requirements reference a specific *equipment property type* (e.g., an *equipment property type* may be the 'size of vessel' or the 'lining of vessel'). A specific equipment requirement could then specify a minimum value for the size of a vessel.

This requirement may then be met by a piece of equipment with a certain equipment property that references the same equipment property type. For example, a specific unit, UNIT12, would have a value for the property type 'size of vessel'.

An equipment entity is a specific piece of equipment, and it may be replaced by a class of equipment (equipment class). See 4.4.1.

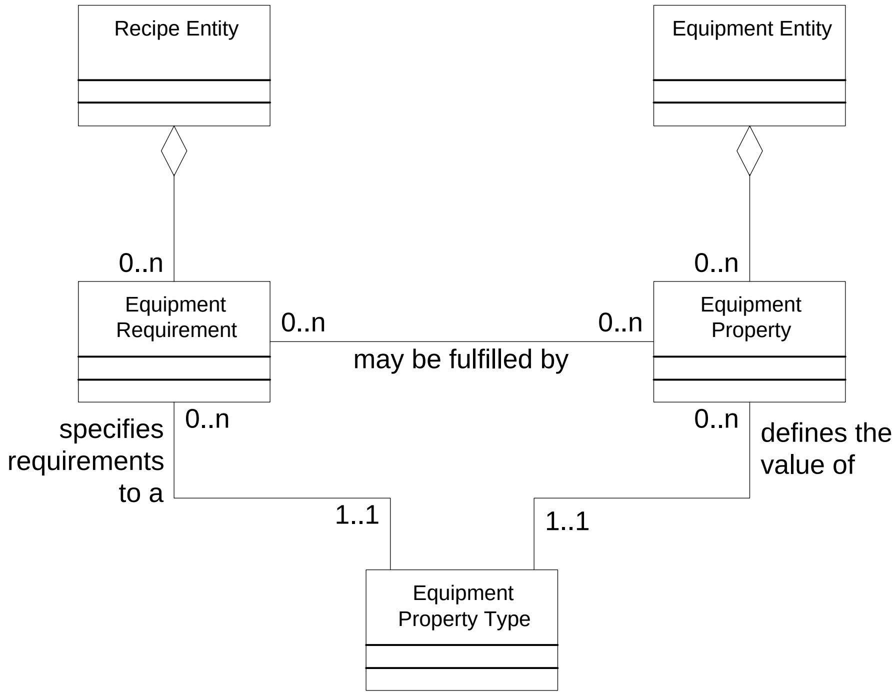


Figure 6 — Recipe entity equipment requirements

Table 15 — Equipment entity

NAME	EQUIPMENT ENTITY
FunctionalDescription	A collection of physical processing and control equipment and equipment control that is grouped together to perform a certain control function or set of control functions.
ATTRIBUTES	
EquipmentEntityID	Provides unique identification.
EquipmentLevel	Specifies the physical hierarchy level (e.g., process cell, unit, equipment module, control module).
Mode	Indicates the current mode of the equipment entity.
State	Indicates the current state of the equipment entity.

Table 16 — Equipment property

NAME	EQUIPMENT PROPERTY
FunctionalDescription	Identifies a property that the equipment entity or class supplies. These properties are application specific (e.g., lining type, size, heat capability, steam temperature).
ATTRIBUTES	
PropertyID	Provides unique identification.
Value	Identifies the value of the property (e.g., glass, 50000, 650).
ValueRange	Defines limits or constraints that are related to Value.
EngineeringUnits	Defines the engineering units of the property.
Description	Describes the type of the equipment property.

Table 17 — Equipment property type

NAME	EQUIPMENT PROPERTY TYPE
FunctionalDescription	The general class of equipment attributes (e.g., lining type, size, heat capability, steam temperature).
ATTRIBUTES	

Control recipe entities will initially contain the equipment requirements that are copied from the *master recipe entity*, and these need to be fulfilled by the corresponding property of one or more equipment entities in order for specific equipment to be allocated. The initial *equipment requirement* may be replaced by specific equipment allocations. These allocations are also modeled as equipment requirements.

4.3.6 Recipe parameters

Parameters are variables associated with recipe entities. These variables may be used by equipment procedural elements, they may be used by other activities (e.g., scheduling), or they may be referenced by other parts of the recipe (e.g., transition criteria) (see figure 7).

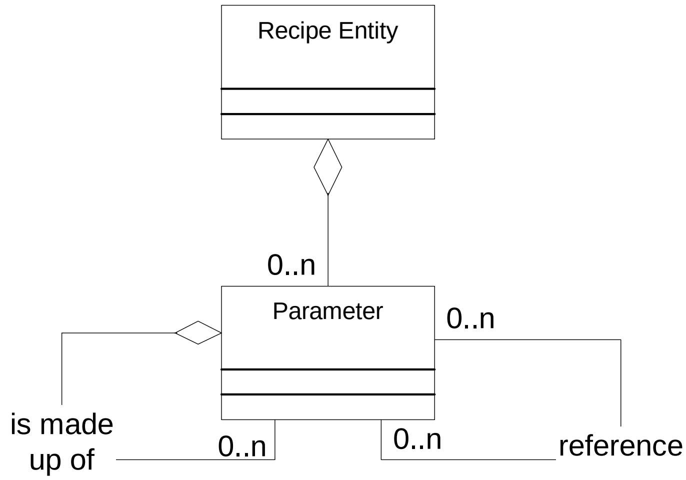


Figure 7 — Parameter model

Parameters may be categorized as process inputs, process outputs, or process parameters.

Parameters may be made up of a collection of other parameters. The model supports the concept of structured parameters. Therefore, the model allows the possibility of including parameters of different types (process parameters, process inputs, process outputs) in the same structure as well as defining single type data structures.

Parameter value attributes may be organized by defining parameter value types. Parameter value types could include

a) IEC 61131-3 basic data types;

b) compatibility matrix information that is used for determining clean-in-place (CIP) or sterilize-in-place (SIP) requirements;

c) data sets that define material transactions (transfer, consumption, generation of material); or

d) data series (e.g., a temperature profile that will be tracked).

Parameter values may be simple values, expressions, or references to parameters that are defined at the same level or higher levels in the procedural hierarchy. Values that are expressions may include references to other parameters.

Parameters may be related in a number of ways, including the following:

a) Algebraic or Boolean equations

b) Product specific entry forms that work on one or more parameters

c) Standard operating procedures (SOPs) that display or otherwise utilize parameters (e.g., dynamic values, recipe values)

d) Deferral of parameters to different recipe entities (at the same or another level)

e) External applications that use parameters

The formula is represented in the data model as recipe parameters (see table 10). A recipe's formula is a collection of selected parameters to the recipe procedure, and it may also include parameters that are defined at lower levels of the procedural hierarchy.

Parameters are often scaled, based on batch size or other key attributes. Scaling may be more complex than a simple linear relationship. More complex scaling methods can be accommodated with user-defined algorithms and relationships.

4.4 Equipment model

The physical structure of the plant needs to be taken into account in the evaluation of equipment selection during recipe execution (see figure 8 and table 18). In particular, the transfer capabilities between equipment or the ability to allocate shared equipment are important to routing a batch.

Equipment entities are defined as in the hierarchy that is specified in Part 1. This hierarchy is modeled through the recursive nature of the objects. This construct allows for the configuration of expandability and collapsibility.

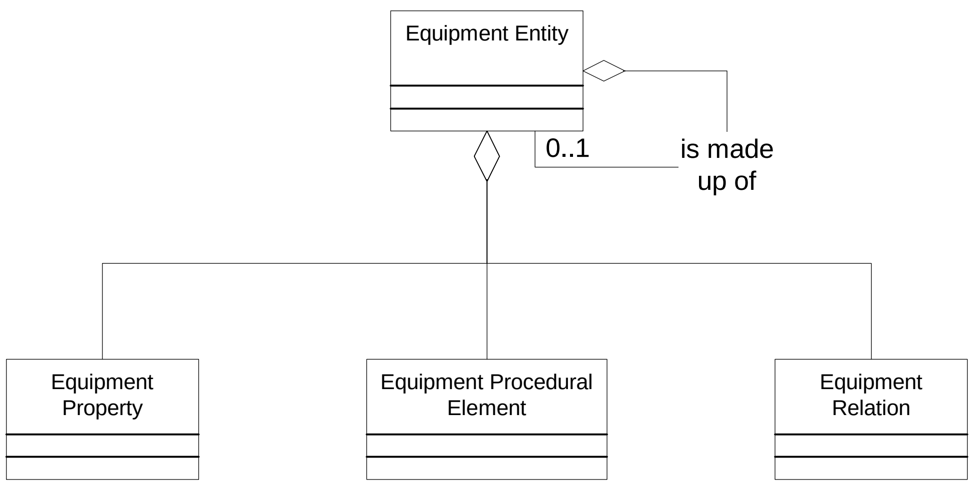


Figure 8 — Equipment structure

Table 18 — Equipment relation

NAME	EQUIPMENT RELATION
FunctionalDescription	A representation of the connections between equipment (e.g., pipes, conveyors, flexible connections), but may also be used to represent other kinds of relations between equipment (e.g., relation to shared-use equipment).
ATTRIBUTES	
RelationID	Provides unique identification.

The equipment in, for example, a process cell (i.e., units, equipment modules, control modules) is normally related to each other by pipes or other connections. The connections may be modeled as equipment relations (see figure 9), optionally with a direction (e.g., a flow direction). The relations (e.g., pipes) are part of the higher-level equipment entity. The connections may conveniently be categorized in relation classes and this ensures a consistent evaluation. Equipment relations include

a) permanent connection;

b) temporary connection;

c) may be used as resource for; and

d) always runs the same product as.

Note that relations other than connections may be possible.

Equipment may have properties. The properties are specific to each user implementation. Equipment properties may be used for examining equipment characteristics and matching recipe equipment requirements. See 4.3.5.

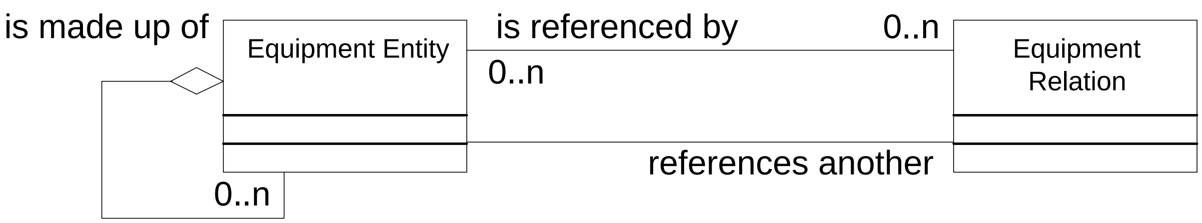


Figure 9 — Equipment entity relations

4.4.1 Equipment classes

Equipment classes (see figure 10 and table 19) provide a means to group equipment entities by common characteristics. Equipment entities may be a member of one or more equipment classes, or they may not belong to a class at all. Equipment classes may be used to specify groups of units and they may be used as alternatives during equipment selection. For example, a recipe may require a reactor for a unit procedure, so its equipment requirements may specify a specific reactor (e.g., R-101), a set of reactor units (e.g., R-101, R-103) or the reactor class (e.g., the class “Reactor” that contains reactors R-101, R-102 and R-103).

Equipment entities may be members of an *equipment class*, and the class determines some of the properties of the class members. As an example, certain equipment properties (e.g., glass lining) are shared with the class.

Equipment entities may belong to zero or more *equipment classes* (e.g., vessel BV1 may be both a reactor and a holding tank).

Equipment classes may determine some or all *equipment properties*, *equipment procedural elements* and *equipment relations* of the referencing equipment entities.

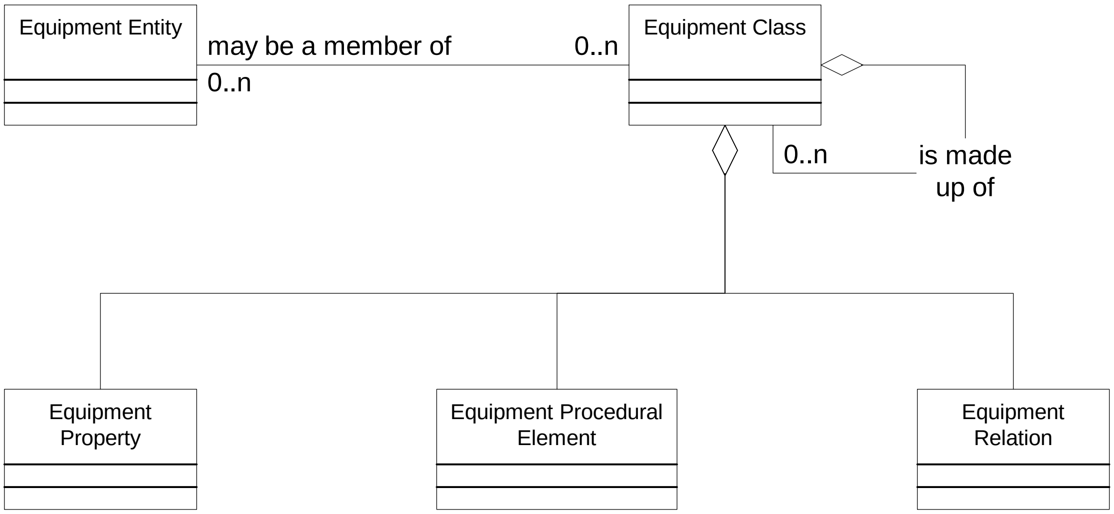


Figure 10 — Equipment classes

Table 19 — Equipment class

NAME	EQUIPMENT CLASS
FunctionalDescription	An equipment entity class.
ATTRIBUTES	
EquipmentClass	Provides unique identification.
EquipmentLevel	Specifies the physical hierarchy level (e.g., process cell, unit, equipment module, control module).

4.5 Production planning and scheduling

The central entity in a schedule (see figure 11) is the batch schedule entry. This object defines an intended execution of one or more batches/control recipes, or other control recipe entities (typically unit procedures)(see table 20). The batch schedule entry may also be used to schedule other activities (e.g., equipment downtime). A batch schedule entry may include formula/parameter values that are to be used in a control recipe (see table 21).

The batch schedule entry can be used to represent higher-level schedule entities (e.g., a production campaign or a production order).

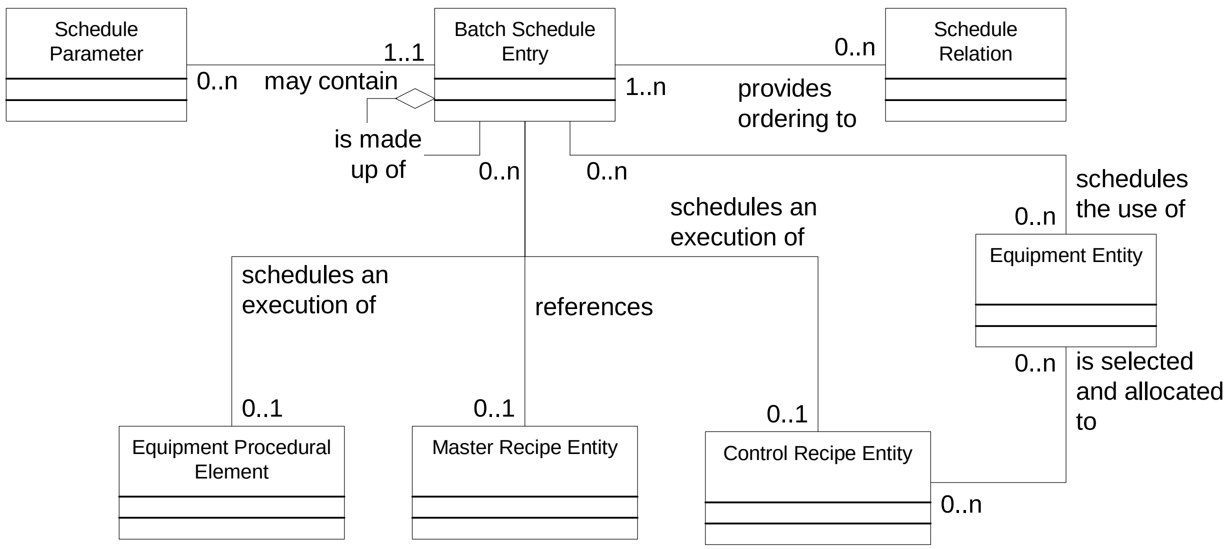


Figure 11 — Batch schedule

Table 20 — Batch schedule entry

NAME	BATCH SCHEDULE ENTRY
FunctionalDescription	A scheduled item that represents a unit procedure in a batch, a complete batch, or a set of batches (e.g., a campaign).
ATTRIBUTES	
ID	Provides unique identification (e.g., actual campaign, lot, batch ID, procedural entity ID).
Level	Specifies the hierarchy level (e.g., campaign, batch, unit procedure).
BatchSize	Defines the requested size or scale factor for the batch, based on the scale factor for the batch as defined in the master recipe.
Schedule	Defines scheduled execution times (start/stop).
ResourceUsage	Specifies resource usage for this schedule entry.
Status	Specifies schedule status (e.g., proposed for evaluation (such as what-if analysis), planned, committed, started, completed).

Table 21 — Schedule parameter

NAME	SCHEDULE PARAMETER
FunctionalDescription	Formula values that are to be communicated to and from batch schedule entries.
ATTRIBUTES	
ParameterID	Provides unique identification.
ParameterType	Includes how the Value is interpreted (e.g., constant, reference, or equation).
Description	Describes the parameter or use of the parameter.
EngineeringUnits	Identifies the engineering units of measure for the Value (e.g., kg, pounds).
Value	Contains the parameter value. If it is a relation, Value contains the equation, form, deferral rule or whatever ties the related parameters together. If Value is a building block parameter, this attribute will hold the default value.
Scaled	Defines the scaling rule. Simplest case: scaled or not scaled with batch reference size.
Usage	Identifies the parameter as a process input, process output, or process parameter.

The schedule relations can be used to represent the scheduling-relevant subset of recipe relations (e.g., the relations related to batch transfers)(see table 22). At higher levels, they can be used to represent required or desirable relations between schedule entries (e.g., batch xx should follow batch yy or a specific cleaning should take place between the two batches). The specific subclasses and characteristics of schedule entry relations are not modeled here.

A higher-level batch schedule entry will include the lower-level schedule entries and relations (e.g., a scheduled campaign may include scheduled batches and relations between these batches).

In the simplest case, the batch schedule entry represents a batch in a batch execution list or queue that is expected to be initiated in sequence. Intended starting times and projected duration/ending times may be added. Further, equipment assignments and the use of other resources may be specified by the batch schedule entry. The scheduling may happen at more detailed levels (e.g., scheduling of individual unit recipes and their assignment to equipment, and potentially scheduling individual operations or phases, their projected duration, and their consumption of resources, including shared or exclusive-use resources that constrain the schedule). Units and equipment modules are allocated or de-allocated as required by the specific control recipe.

The collection of schedule entries may be viewed in the following different ways:

a) As a list or chart of batches in a process cell or part of a process cell, which provides an overview of process cell utilization

b) As a list or chart of resource utilizations, which provides a resource or equipment schedule

Table 22 — Schedule relation

NAME	SCHEDULE RELATION
FunctionalDescription	A representation of relations between schedule entries (e.g., required line clearance, cleaning between scheduled items, specifications of sequential relations within a procedure).
ATTRIBUTES	
ExecuteOrder	Specifies in what sequence multiple schedule entries should be processed relative to each other.

4.6 Production information management

This subclause describes models that define the collection of production information.

Production information, including timely information of how things have progressed, may include both batch-specific information and selected non-batch-specific, or common, information (see figure 12 and tables 23 through 28).

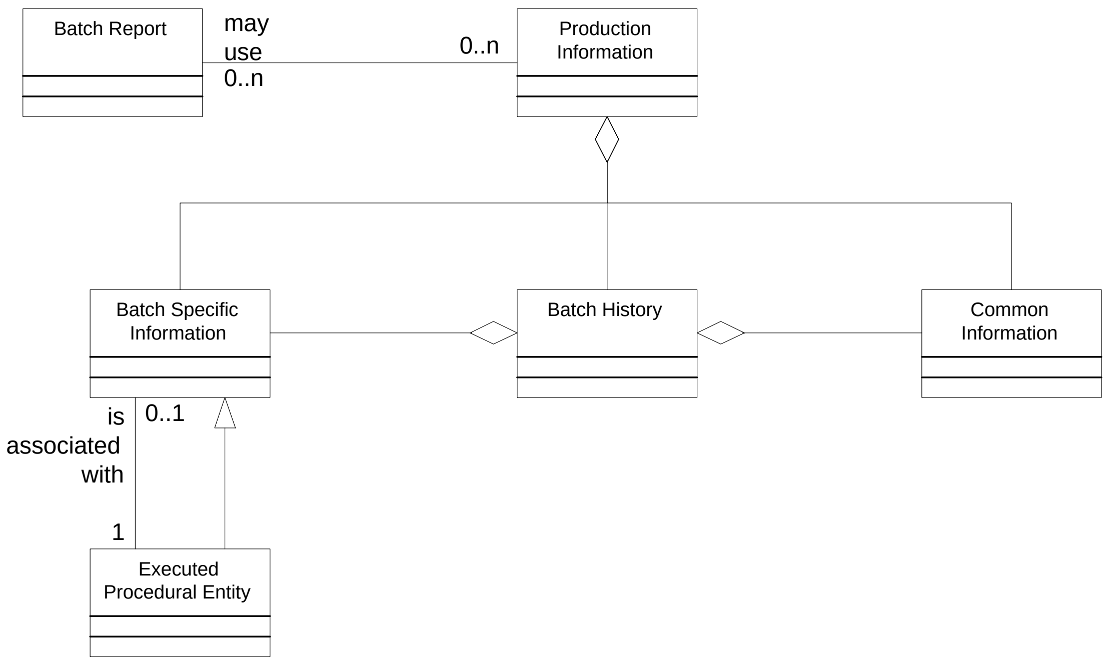


Figure 12 — Production information

Table 23 — Production information

NAME	PRODUCTION INFORMATION
FunctionalDescription	Information generated during the production of a batch.
ATTRIBUTES	

Table 24 — Batch specific information

NAME	BATCH SPECIFIC INFORMATION
FunctionalDescription	Data that relates to one batch history entry.
ATTRIBUTES	
BatchID	Defines the actual Batch ID.
EntryID	Provides unique identification.
NewValue	Indicates the current actual value.
EngineeringUnits	Defines the engineering units, if any, that are appropriate for the NewValue.
EquipmentID	Identifies an equipment element that may be associated with the entry.
UTC	Identifies the Universal Coordinated Time (UTC) and date of the recorded entry.
UserID	Identifies the user, if any, who is associated with the change.

Table 25 — Batch history

NAME	BATCH HISTORY
FunctionalDescription	An item of information that documents batch production.
ATTRIBUTES	
BatchID	Defines the actual Batch ID.

Table 26 — Common information

NAME	COMMON INFORMATION
FunctionalDescription	Data that relates to more than one batch history entry (e.g., cooling water temperature, atmospheric pressure, steam capacity).
ATTRIBUTES	
EntryID	Provides unique Identification.
NewValue	Indicates the current actual value.
EngineeringUnits	Defines the engineering units, if any, that are appropriate for the NewValue.
EquipmentID	Identifies an equipment element that may be associated with the entry.
UTC	Identifies the Universal Coordinated Time (UTC) and date of the recorded entry.
UserID	Identifies the user, if any, who is associated with the change.

Production information may include some of the following:

a) A copy of the control recipe

b) A copy of the master recipe

c) Information about materials that are used and produced

d) Trend information

e) Alarms and messages

f) Operator interactions with the batch (e.g., overwrites, acknowledgements)

g) Late records and asynchronous records (e.g., lab sample measurements)

h) Additional information (e.g., allocation, start/stop)

An executed procedural entity is a recording of an instance of execution of a recipe entity or an equipment procedural entity (see table 25). Data is associated with the execution instance and that data is maintained in the related batch history records.

The executed procedural entities that result from the execution of a control recipe will frequently have the same structure as the control recipe entities. The structure of executed procedural elements may, however, deviate from the control recipe in some cases. The following are some typical examples:

a) Repeated instances of a recipe entity created through looping in the procedural logic

b) Not executed instances due to branches or GoTos in the procedural logic

c) Recipe entities that are inserted or repeated manually

d) Recipe entities that are activated on the spot in units that are already associated with a batch or that are manually associated with the batch

The recipe entity relation and the rest of the recipe entity design pattern is not repeated here, because the history is about logging the facts as they were and not the intended structure.

Table 27 — Executed procedural entity

NAME	EXECUTED PROCEDURAL ENTITY
FunctionalDescription	A representation of a recipe entity (e.g., equipment procedural element) that has been executed.
ATTRIBUTES	
ExecutedProceduralEntityID	Provides a unique identification.
ProceduralEntityCounter	Uniquely identifies repeated execution of the same procedural entity.

Batch reports (see table 28) are understood to be any extraction of batch data from production information for display on screen or on paper, or for transfer to other systems.

Table 28 — Batch report

NAME	BATCH REPORT
FunctionalDescription	A component of a batch report.
ATTRIBUTES	
ReportID	Provides unique identification.

5 Relational tables for information exchange

5.1 Introduction

This clause defines the structure of SQL relational tables for the exchange of selected batch control related information between systems. It defines an interface specification, meeting the requirements in clause 4, for the exchange of batch information in the following selected categories:

a) Master and control recipe information

b) Process cell equipment information

c) Schedule information

d) Production information

Exchange tables shall be built using the table names, field names and relations that are defined in this clause. Not all tables need to be implemented, but all fields in an implemented table shall be included. Any such implementation shall be consistent with the table definitions presented here and the concepts of Part 1.

The format for the exchange tables only defines standard information that can be exchanged. The table definitions can be extended through the inclusion of additional attributes, additional related tables and additional enumerations. The extended structures may be used to exchange information among tools that understand the structures, but this information is outside the scope of this standard.

Examples of additional information are

a) the addition of icons that represent different elements of the equipment hierarchy, so that the equipment may be represented consistently by different tools; and

b) the addition of control system addresses for equipment element procedures, procedure parameters and data elements.

5.1.1 Method

The relational table structures are defined using SQL, as specified in ISO/IEC 9075:1992, *Information processing systems – Database language SQL with integrity enhancement*.

The exchange mechanism is based on a common structure for database tables. These tables are defined as a data base schema that can be defined in SQL tables. Annex B contains the SQL table definitions of the exchange tables.

Figure 13 illustrates how the exchange tables would be used to exchange batch information between different tools. Each tool will generally have its own local data stores for batch information and each tool is responsible for importing from, and exporting to, the exchange tables.

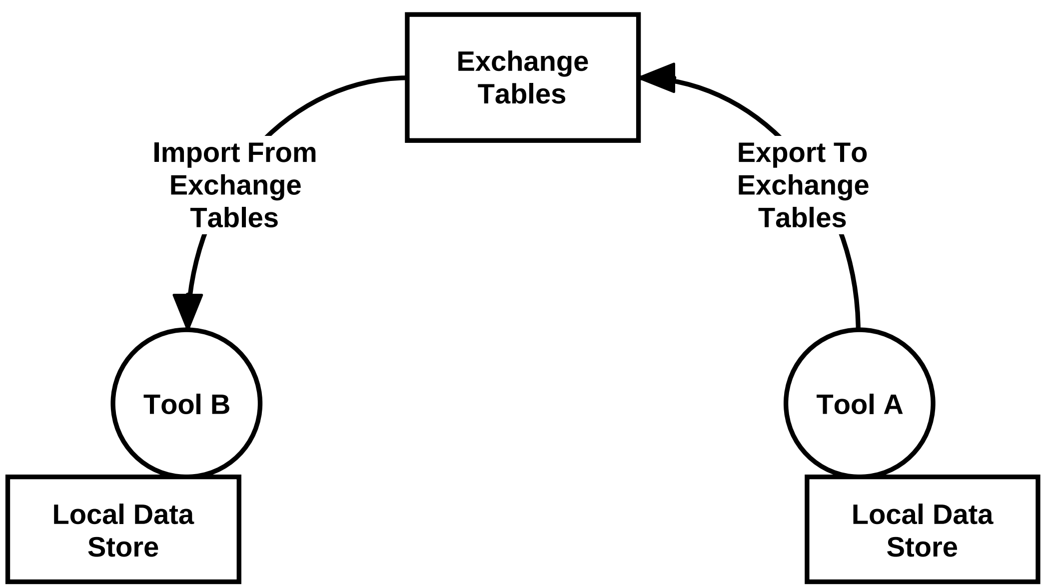


Figure 13 — Data transfer using exchange tables

Most of the remaining models in this clause are represented using entity relationship diagrams (ERDs) (see clause A.3).

The type and use of the tools that are using the batch information are not specified in this standard. Possible tools include, but are not limited to,

a) recipe authoring systems;

b) recipe execution systems;

c) documentation systems;

d) configuration management systems;

e) simulation systems;

f) batch control systems;

g) planning and scheduling systems; and

h) information management systems.

5.1.2 Exchange tables

This standard does not specify which tools create and maintain the exchange tables. This standard only defines the structure of the tables. Tools could be designed such that they can only read from pre-existing exchange tables, only write to pre-existing exchange tables, read and write to pre-existing tables, or create, read, and write to exchange tables.

The exchange table structure is designed to allow multiple recipes to be exchanged in the same set of tables and it allows multiple versions of the same recipe to be exchanged. The exchange table structure also allows the exchange of the equipment capability and specification information about process cells and process cell equipment.

The exchange table structure also allows the transfer of either complete or incomplete subsets of a master recipe, equipment descriptions, schedule information, or batch history.

The syntax of data strings in the exchange tables (e.g., computed formula items, transition conditions) is not defined in this standard. Tools that read and write the SQL exchange tables should resolve syntax differences.

The field lengths that are defined in the SQL definition of the exchange tables represent default values and they are not intended to enforce a standard, minimum, or maximum field length. Tools that read and write the SQL exchange tables shall resolve field length differences.

5.1.3 Common exchange information

A set of tables is defined to contain information that describes the exchange format and that can be commonly used in the different exchange table sets. Figure 14 shows the tables that are involved in the common exchange information.

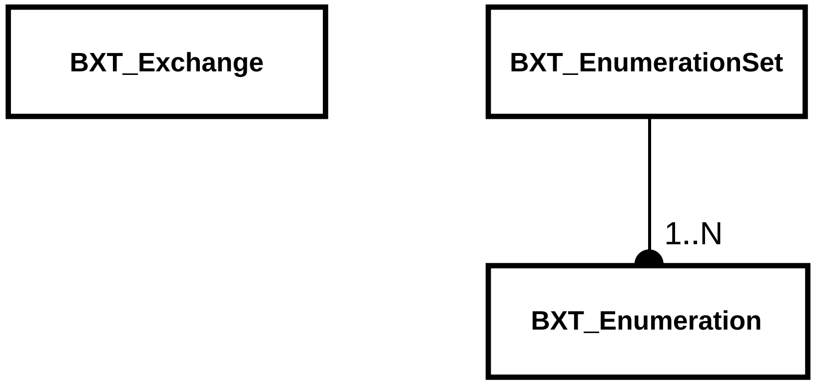


Figure 14 — Common information exchange tables

5.1.3.1 Exchange information

The common exchange information table, BXT_Exchange, contains all of the information that is needed only once in the exchange of data (see table 29).

The table, BXT_Exchange table contents (see table 30), contains one record for each of the defined items (e.g., SCHEMA and the DELIMITER characters). It may also contain other user-defined information.

Table 29 — BXT_Exchange

ATTRIBUTE	DESCRIPTION
ExchangeID	Identifies the exchange element.
ExchangeValue	Identifies the value for the exchanged information.

Table 30 — BXT_Exchange table contents

EXCHANGEID	DESCRIPTION
Schema	The version of the standard that the database schema was defined against. The initial version value shall be the title of this standard (e.g., "ISA-88.00.02-2001").
Delimiter	The character that is used to differentiate between the element names in the recipe element hierarchical name.
ToolID	The identification of the tool that created the exchange tables.
ToolVersion	The version of the tool that created the exchange tables.
ToolSchema	The customized schema version created by the tool.

5.1.3.2 Enumeration sets

Many of the tables contain fields that can contain standard or user-defined enumerated items. These enumerated items are passed as numbers in the exchange tables, with the strings contained in an enumeration set table. The enumeration set tables (see table 31) provides a single location for translation of the strings between different languages. The BXT_EnumerationSet table defines the enumeration set. The BXT_Enumeration table defines the members of the set and the associated numerical value.

The enumeration tables contain standard enumerations and values. These may be extended by the user. Additional user enumerations of the standard enumeration sets may take on values of 100 and above. Enumeration values of 0-99 are either used or are reserved by this standard. In addition, user defined enumeration sets and their respective values may also be defined in the enumeration tables. For example, the enumeration set "Blend Oil" may be defined for phase parameters with members of "Virgin Oil," "Blended Oil," and "Recovered Oil" having values of 101, 102, and 103, respectively.

Table 31 — BXT_EnumerationSet

ATTRIBUTE	DESCRIPTION
EnumSet	Identifies the standard enumeration set.
Description	Contains the use of the enumeration set. (Provided to assist in the translation of the TextString.)

Table 32 defines the standard enumeration sets that are defined in this standard.

Table 32 — Standard enumeration sets

ENUMSET	DESCRIPTION
Boolean	Defines a set of Boolean values.
DirectionType	Defines how a parameter is intended to be handled.
EquipmentLevel	Defines the equipment hierarchical level for equipment elements.
EquipmentType	Defines the type of equipment record for equipment elements.
EvaluationRule	Defines the evaluation rules for equipment properties.
FormulaType	Defines the recipe formula types.
FormulaSubType	Specifies user-supplied formula sub type definitions.
LinkDepiction	Defines how links between recipe elements are to be depicted.
LinkToType	Defines if a link is referencing a step or transition.
LinkType	Defines the type of link.
RE_Type	Defines the recipe element (RE), either recipe procedure level or allocation symbol.
RE_Use	Defines how a recipe element (RE) is used in a recipe.
RecipeStatus	Definition of the possible status of a recipe.
RecordSet	Defines the enumeration set that is used to classify a record into a category of batch history information.
RecordSetControlRecipe	Provides further history record classification under the category of ControlRecipe.
RecordSetMasterRecipe	Provides further history record classification under the category of MasterRecipe.
RecordSetExecutionInfo	Provides further history record classification under the category of ExecutionInfo.
RecordSetMaterialInfo	Provides further history record classification under the category of MaterialInfo.
RecordSetContinuousData	Provides further history record classification under the category of ContinuousData.
RecordSetEvents	Provides further history record classification under the category of Events.
RecordSetOperatorChange	Provides further history record classification under the category of OperatorChange.
RecordSetOperatorComment	Provides further history record classification under the category of OperatorComment.
RecordSetAnalysisData	Provides further history record classification under the category of AnalysisData.
RecordSetLateRecord	Provides further history record classification under the category of LateRecord.
RecordSetRecipeData	Provides further history record classification under the category of RecipeData.
RecordSetRecipeSpecified	Provides further history record classification under the category of RecipeSpecified.
RecordSetSummaryData	Provides further history record classification under the category of SummaryData.
ScheduleAction	Defines the intended action of the schedule record.
ScheduleMode	Defines the mode in which the schedule record begins execution.
ScheduleStatus	Defines the possible status of a schedule.
SE_Type	Defines the type of entity in a schedule record.
ValueDataType	Defines the data type of an associated data value.
ValueType	Defines how a value string is interpreted.

Table 33 shows how enumerations are defined by this standard.

Table 33 — BXT_Enumeration

ATTRIBUTE	DESCRIPTION
EnumSet	Identifies the name of the enumeration set.
EnumValue	Specifies the numerical value associated with the enumeration member.
EnumString	Defines the associated text for the enumeration member.
Description	Contains the use of the enumeration member. (Provided to assist in the translation of the TextString.)

Table 34 contains the list of standard enumeration members that are defined by this standard.

Table 34 — Standard enumerations

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRIPTION
Boolean	0	FALSE	Definition of a Boolean value.
	1	TRUE	
DirectionType	0	Invalid	Entry not valid
	1	Internal	Identifies how a parameter is handled. Internal = only available within the Recipe Element. Defined at creation or created as an intermediate value.
	2	Input	The Recipe Element receives the value from an external source.
	3	Output	The Recipe Element creates the value and makes it available for external use.
	4	Input/Output	The Recipe Element and external element exchange the value, and may change its value.
	5-99		Reserved
	100+		User defined
EquipmentLevel	0	Invalid	Entry not valid
	1	Enterprise	Identifies the equipment hierarchical level for BXT_EquipElement.
	2	Site	
	3	Area	
	4	Process Cell	
	5	Unit	
	6	Equipment Module	
	7	Control Module	
	8-99		Reserved
	100+		User defined
EquipmentType	0	Invalid	Entry not valid
	1	Class	Identifies the record type for BXT_EquipElement
	2	Element	
	3-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRPTION
EvaluationRule	0	Invalid	Entry not valid
	1	=	Equals comparison operator for equipment properties.
	2	<>	Not equals comparison operator for equipment properties.
	3	<	Less than comparison operator for equipment properties.
	4	>	Greater than comparison operator for equipment properties.
	5	<=	Less than or equals comparison operator for equipment properties.
	6	>=	Greater than or equals comparison operator for equipment properties.
	7	Member	Is a member of comparison operator for equipment properties.
	8	Notmember	Is not a member of comparison operator for equipment properties.
	9	Not	Not comparison operator for equipment properties.
	10-99		Reserved
	100+		User defined
FormulaSubType	0	Invalid	Entry not valid
	1-99		Reserved
	100+		User defined. Allows further user classification of a formula type.
FormulaType	0	Invalid	Entry not valid
	1	Process Input	Recipe formula type
	2	Process Output	
	3	Process Parameter	
	4-99		Reserved
	100+		User defined
LinkDepiction	0	Invalid	Entry not valid
	1	None	No link depiction
	2	Line	Link shown with line only.
	3	ID	Link shown with identifier only.
	4	Line & ID	Link shown with line and identification.
	5	Line & Arrow	Link shown with line and material flow arrow.
	6	Line, Arrow, & ID	Link shown with line, material flow arrow and identification.
	7-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRPTION
LinkToType	0	Invalid	Entry not valid
	1	Recipe Element	Link is referencing a record in the BXT_MRecipeElement table.
	2	Transition	Link is referencing a record in the BXT_MRecipeTransition table.
	3-99		Reserved
	100+		User defined
LinkType	0	Invalid	Entry not valid
	1	ControlLink	Defines a link between recipe elements that indicates a flow of procedural control.
	2	TransferLink	Defines a link between recipe elements that indicates a material transfer.
	3	SynchronizationLink	Defines a link between recipe elements where there is some form of synchronization.
	4-99		Reserved
	100+		User defined
RE_Type	0	Invalid	Entry not valid
	1	Master Recipe	Specifies the type of recipe element.
	2	Procedure	
	3	Unit Procedure	
	4	Operation	
	5	Phase	
	6	Allocation	
	7	Begin	
	8	End	
	9	Start Parallel	
	10	End Parallel	
	11	Start Branch	
	12	End Branch	
	13-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRIPTION
RE_Use	0	Invalid	Entry not valid
	1	Linked	A recipe element (RE) may have several referencing RE Steps.
	2	Embedded	A RE has only one referencing RE; one RE is defined for each use of the RE.
	3	Copied	The same as Embedded, but the specific RE was modified from its original definition.
	4-99		Reserved
	100+		User defined
RecipeStatus	0	Invalid	Entry not valid
	1	Approved for Production	Recipe is approved for production.
	2	Approved for Test	Recipe is only approved for test.
	3	Not Approved	Recipe is not approved for production or test.
	4	Inactive	Recipe is not active.
	5	Obsolete	Recipe is obsolete.
	6-99		Reserved
	100+		User defined
RecordSet	0	Invalid	Entry not valid
	1	RecordSetControlRecipe	Defines that a batch history information record is part of the ControlRecipe category.
	2	RecordSetMasterRecipe	
	3	RecordSetExecutionInfo	
	4	RecordSetMaterialInfo	
	5	RecordSetContinuousData	
	6	RecordSetEvents	
	7	RecordSetOperatorChange	
	8	RecordSetOperatorComment	
	9	RecordSetAnalysisData	
	10	RecordSetLateRecord	
	11	RecordSetRecipeData	
	12	RecordSetRecipeSpecified	
	13	RecordSetSummaryData	
	14-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRIPTION
RecordSet ControlRecipe	0	Invalid	Entry not valid
	1	Entire Control Recipe	History record is related to the entire control recipe.
	2-99		Reserved
	100+		User defined
RecordSet MasterRecipe	0	Invalid	Entry not valid
	1	Entire Master Recipe	History record is related to the entire master recipe.
	2-99		Reserved
	100+		User defined
RecordSet ExecutionInfo	0	Invalid	Entry not valid
	1	Allocation	
	2	De-allocation	
	3	State Change	
	4	State Command	
	5	Mode Change	
	6	Mode Command	
	7	Procedural Entity Message	
	8	Procedural Entity Alarm	
	9	Procedural Entity Version	
	10	Procedural Entity Prompt	
	11	Procedural Entity Prompt Resp	
	12-99		Reserved
	100+		User defined
RecordSet MaterialInfo	0	Invalid	Entry not valid
	1	Material Consumption	
	2	Material Production	
	3	Material Allocation	
	4	Material De-allocation	
	5-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRIPTION
RecordSet ContinuousData	0	Invalid	Entry not valid
	1	Continuous Data Value	
	2	Trend Association	
	3	Trend Disassociation	
	4-99		Reserved
	100+		User defined
RecordSet Events	0	Invalid	Entry not valid
	1	General Event	
	2-99		Reserved
	100+		User defined
RecordSet OperatorChange	0	Invalid	Entry not valid
	1	General Operator Intervention	
	2-99		Reserved
	100+		User defined
RecordSet OperatorComment	0	Invalid	Entry not valid
	1	General Operator Comment	
	2-99		Reserved
	100+		User defined
RecordSetAnalysis Data	0	Invalid	Entry not valid
	1	General Analysis Message	
	2-99		Reserved
	100+		User defined
RecordSet LateRecord	0	Invalid	Entry not valid
	1	General Late Record	
	2-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRIPTION
RecordSetRecipe Data	0	Invalid	Entry not valid
	1	Generic Recipe Data	
	2	Recipe Parameter Value Change	
	3	Recipe Result Data	
	4-99		Reserved
	100+		User defined
RecordSet RecipeSpecified	0	Invalid	Entry not valid
	1	Generic Recipe Specified Data	
	2-99		Reserved
	100+		User defined
RecordSet SummaryData	0	Invalid	Entry not valid
	1	Generic Summary Data	
	2	Utilities Consumption	
	3	Equipment Run Time	
	4-99		Reserved
	100+		User defined
ScheduleAction	0	Invalid	Entry not valid
	1	New	Schedule record change action
	2	Update	
	3	Delete	
	4-99		Reserved
	100+		User defined
ScheduleMode	0	Invalid	Entry not valid
	1	Automatic	Schedule record mode
	2	Semi-automatic	
	3	Manual	
	4	Not Specified	
	5-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRIPTION
ScheduleStatus	0	Invalid	Entry not valid
	1	Complete	Batch schedule record status
	2	In-progress	
	3	Scheduled	
	4	Schedule Hold	
	5	Not Specified	
	6-99		Reserved
	100+		User defined
SE_Type	0	Invalid	Entry not valid
	1	Campaign	Defines the type of Scheduled Entry.
	2	Batch	
	3	Unit procedure	
	4	Operation	
	5	Phase	
	6-99		Reserved
	100+		User defined
ValueDataType	0	Invalid	Entry not valid
	1	Boolean	Defines the data type that is expected for an associated value.
	2	8-Bit string	
	3	16-Bit string	
	4	32-Bit string	
	5	8-Bit unsigned integer	
	6	16-Bit unsigned integer	
	7	32-Bit unsigned integer	
	8	8-Bit signed integer	
	9	16-Bit signed integer	
	10	32-Bit signed integer	
	11	32-Bit float	
	12	Double float	
	13	Octet string	
	14	DateTime	
	15-99		Reserved
	100+		User defined

Table 34 — Standard enumerations (continued)

ENUMSET	ENUM VALUE	ENUMSTRING	DESCRIPTION
ValueType	0	Invalid	Entry not valid
	1	Constant	Defines how a value string is interpreted. It contains a fixed value as a string.
	2	Reference	Defines how a value string is interpreted. It points to the source of the value.
	3	Equation	Defines that a value string represents an expression to be evaluated in order to determine the value.
	4	External	Value is supplied by some external means, and it is not contained in the recipe (i.e., value may be supplied by an operator or by a scheduling system).
	5-99		Reserved
	100+		User defined

5.2 Master recipe information

This subclause only deals with *Master Recipes*. The information that is exchanged is what is needed in order to exchange a master recipe, as defined in Part 1. This information contains the procedural control definitions, the formula value definitions, the equipment requirements of the recipe, header information, process cell specific information, other information, and coordination control requirements.

The information that is exchanged is what is needed to **exchange** a master recipe, but this information does not specify the following:

a) How the information was created

b) How the master recipe could be used in a system

5.2.1 Recipe definitions

Creation of a master recipe, as specified in Part 1, may use information from the site recipe and the definition of the process cell's processing capability. A master recipe is tied to the process cell's processing capability, but the definition of the processing capability is not part of the recipe-exchanged information.

5.2.2 Recipe structure

A recipe is made up of the following information categories: header, formula, procedure, equipment requirements, and other information.

The recipe exchange schema is recursive by nature. The basic structure of that schema, as shown in figure 15, is built around these five information categories. Every level of the definition contains all of these information categories until the procedural definition references an equipment procedural entity.

The term Recipe Element (RE) is used to define some of the structural entities in a recipe. A recipe element may be the Master Recipe itself, Recipe Procedure, Recipe Unit Procedure, Recipe Operation, Recipe Phase, Allocation symbol, or other graphical symbols, as defined in table 34.

In the recursive model of the schema, recipe elements contain either lower-level recipe elements or references to Equipment Procedural Elements. While the definition of the recipe element is contained in the recipe element table, each use of a recipe element is called a Step.

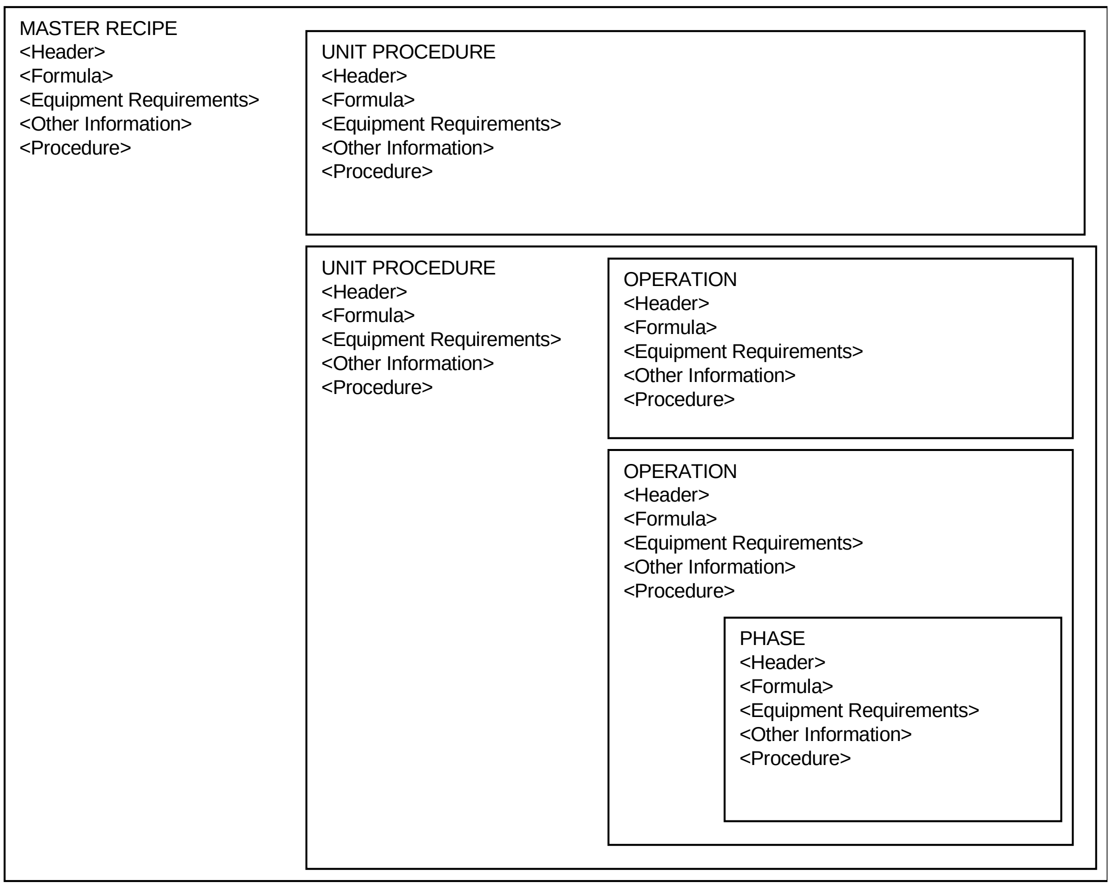


Figure 15 — Nested recipe elements make up a recipe

5.2.3 Table overview and integrity constraints

Figure 16 illustrates the tables that are used to exchange recipes and their relationships and it defines the associated integrity constraints between the tables.

Non-direct relationships, such as between the LINK and RE, are not depicted; however, they are defined through the set of common key fields in the table. "NOT NULL" entries in the associated SQL tables are used only to enforce the integrity constraints of the Entity Relationship diagram.

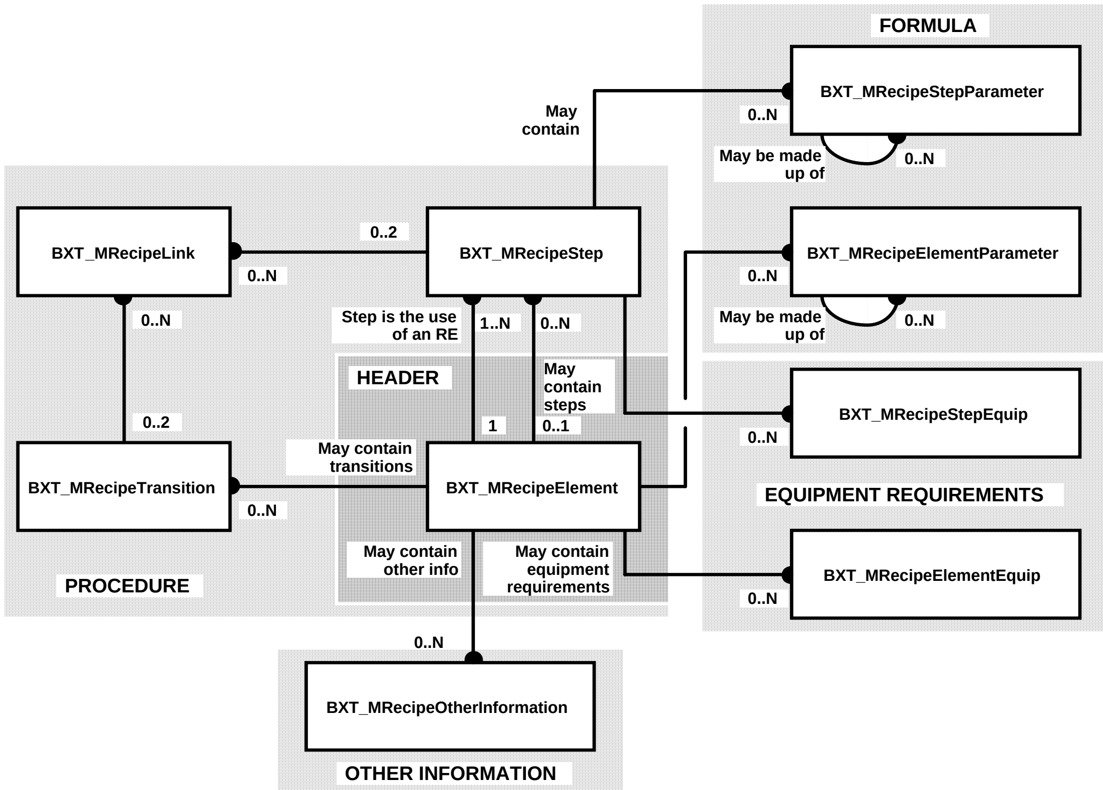


Figure 16 — Exchange table relationships

The recursive definition of the Recipe Elements (REs) is through the two associations between the BXT_MRecipeStep and BXT_MRecipeElement entities. Each defined use of a recipe element is recorded in the BXT_MRecipeStep table. Each definition of a recipe element is recorded in the BXT_MRecipeElement table. One association shows what steps are contained in a recipe element. The other association shows what element is referenced by a step. Each RE that is referenced by the owning RE has a record in the Step table. The Step table then references the actual RE definition. The tables support a single RE definition per Step, and they support multiple Steps that reference the same RE.

The relationship between Steps and Transitions is maintained in the BXT_MRecipeLink table.

The master recipe formula definition is the collection of all master recipe parameter records and it describes the process input, process output, and process parameters within the recipe. The formula values are contained in the BXT_MRecipeStepParameter table and they have their definition in the BXT_MRecipeElementParameter table.

The recipe's equipment requirements are contained in the BXT_MRecipeStepEquip and BXT_MRecipeElementEquip tables.

Figure 17 shows how entries in each table are related to each other for the BXT_MRecipeStep, BXT_MRecipeElement, BXT_MRecipeElementParameter, and BXT_MRecipeStepParameter tables. One BXT_MRecipeElement record exists for each version of an exchanged recipe. A relationship between this BXT_MRecipeElement record and a single BXT_MRecipeStep table record exists, and this relationship contains the specific recipe information, including formula values in the BXT_MRecipeStepParameter table.

The BXT_MRecipeStep table contains a key into a single record in the BXT_MRecipeElement table. This BXT_MRecipeElement table record contains the definition of the recipe's procedure, including the definition of the formula values in the BXT_MRecipeElementParameter table.

The BXT_MRecipeElement table for this procedure contains a key into multiple records in the BXT_MRecipeStep table, one for each Unit Procedure. (Other procedure level recipe elements are omitted from this example for simplicity.) The BXT_MRecipeStep record for the Unit Procedures contains a key to the BXT_MRecipeElement record that defines the Unit Procedure. The BXT_MRecipeElement record for each Unit Procedure contains a key into the BXT_MRecipeStep table for each Operation. This structure continues down to the Phase definitions.

This table format uses the BXT_MRecipeStep and BXT_MRecipeElement tables to contain the procedural hierarchy of a recipe's procedure.

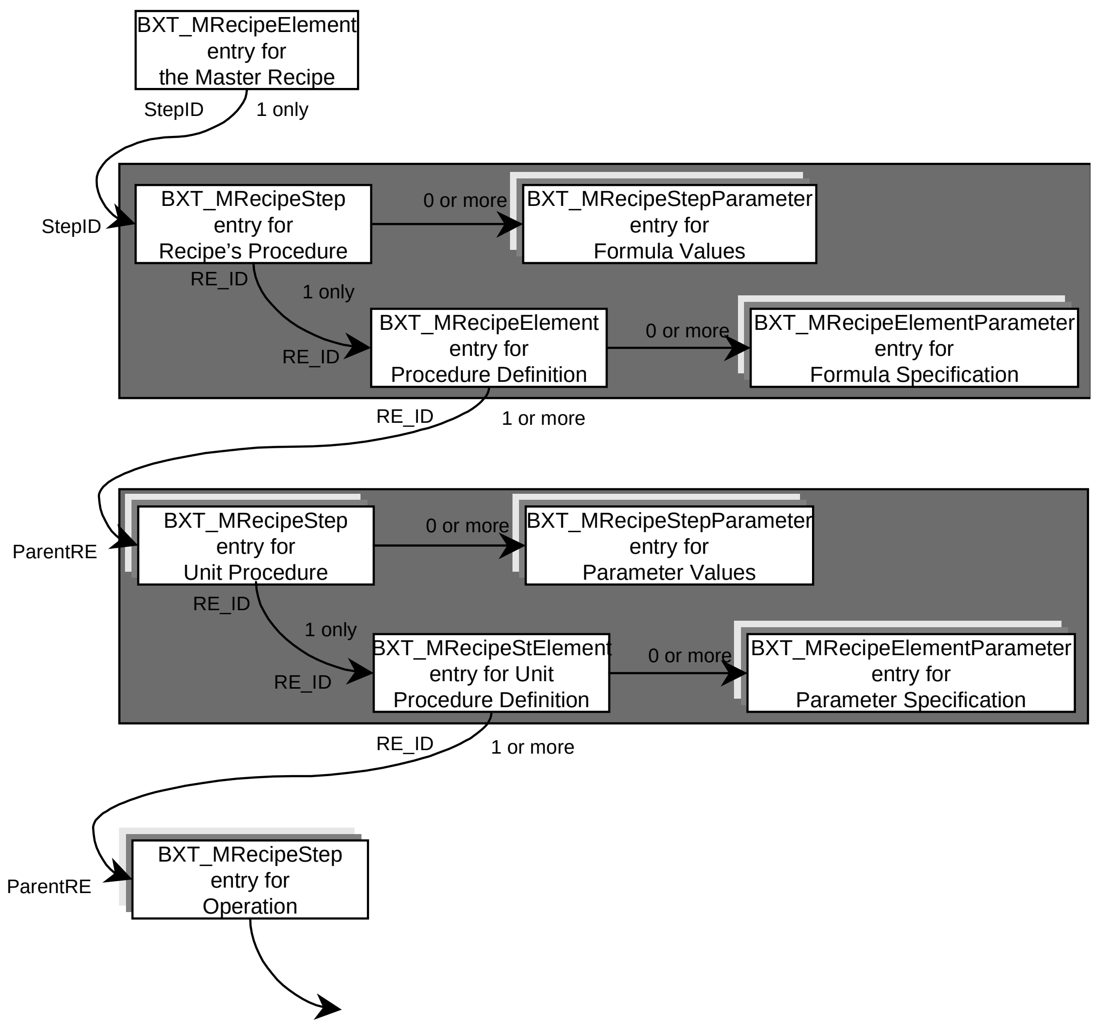


Figure 17 — How entries relate in the tables

5.2.4 Recipe table summary

The tables that are defined for recipe exchange are shown in table 35.

Table 35 — Recipe exchange tables

TABLE NAME	DESCRIPTION
BXT_MRecipeStep	One record for each use of a RE within a RE.
BXT_MRecipeElement	One record for each recipe element that is exchanged.
BXT_MRecipeTransition	One record for each transition that is used within a RE.
BXT_MRecipeLink	One record for each link between Steps and Transitions.
BXT_MRecipeElementParameter	One record for each parameter for each RE.
BXT_MRecipeStepParameter	One record for each parameter for each Step.
BXT_MRecipeOtherInformation	One record for each element of other information.
BXT_MRecipeElementEquip	One record for each property requirement for a RE.
BXT_MRecipeStepEquip	One record for each value for an equipment property defined in a step.

5.2.5 Recipe table definitions

5.2.5.1 Header information

Master recipe header information is transferred as fields of a record in the BXT_MRecipeElement table. The BXT_MRecipeElement table (see table 36) contains one element for each master recipe that will be exchanged. The combination of RE_ID and REVersion defines the exchanged recipe master recipe.

Table 36 — BXT_MRecipeElement

ATTRIBUTE	DESCRIPTION
RE_ID	Identifies the Recipe Element that will be exchanged (e.g., *Red Oak*). When combined with the "version," this field defines a unique instance of a RE. When the record represents the master recipe, this field contains the master recipe ID.
REVersion	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a RE (e.g., V10.3).
VersionDate	Identifies the date and time that this version of the RE was last modified.
ApprovalDate	Identifies the date and time that this version of the recipe was approved.
EffectiveDate	Identifies the date and time that this version of the recipe is effective.
ExpirationDate	Identifies the date and time that this version of the recipe expires.
Author	Identifies the person or system that authored this version (e.g., J. Smith).

Table 36 — BXT_MRecipeElement (continued)

ATTRIBUTE	DESCRIPTION
ApprovedBy	Identifies the person or system that approved this version of the recipe.
ProcessCellID	Identifies the process cell or class of process cells for which this version of the master recipe was defined.
ProductID	Identifies the product or product family that would be created by execution of this version of the recipe (e.g., Premium Beer).
UsageConstraint	Defines other rules that determine the usage (e.g., “must be succeeded by...” or “must not run in parallel with...”).
Description	Describes the recipe element.
Status	Defines the status of the information that is being exchanged as an enumeration from the enumeration set "Recipe Status.”
RE_Type	Identifies the type of the RE from the enumeration set "RE_Type.”
RE_Function	Contains an optional reference to the equipment information exchange tables. The format for this information is not defined in this clause. An example is a reference to an equipment procedural element (see BXT_EquipInterface table). If this entry is NULL, then the function of this RE is defined by BXT_MRecipeStep and BXT_MRecipeTransition entries that have a ParentRE that matches this table’s RE_ID.
RE_Use	Identifies the relationship between the RE and the BXT_MRecipeStep, from the enumeration "RE Use.” Linked specifies that there may be multiple BXT_MRecipeStep uses of the RE definition. Linked would be used when the BXT_MRecipeElement is a library building block. Embedded specifies that the RE has only one referencing RE and one RE is defined for each use of the RE. In this case, the RE is "embedded" in the recipe definition of a single recipe step and it is not used elsewhere. Copied specifies the same as Embedded, but the RE was modified from some original definition. Copied would be used when the RE was a library building block that was fully reproduced in the recipe, and its linkage to the library removed.
DerivedRE	Identifies the recipe element from which this recipe element was derived.
DerivedVersion	Identifies the version of the recipe element from which this recipe element was derived.

5.2.5.2 Procedure information

The procedural parts of a master recipe are included or contained in a combination or collection of tables. These records define

a) the procedural control element steps;

b) the procedural control elements;

c) allocation symbols and other graphical representation symbols;

d) the parameterization of procedural elements with limits;

e) the linkage between the elements; and

f) the transition definitions between elements.

The Step, Transition, and Link tables contain the definition of REs that contain lower-level REs. The BXT_MRecipeStep table (see table 37) contains each instance of use of a RE. Each BXT_MRecipeStep contains the parameters used when a RE is used. Because a RE may be used multiple times in a recipe, there is only one record in the BXT_MRecipeElement table, but there are multiple records in the BXT_MRecipeStep table, one record for each use of the RE.

Table 37 — BXT_MRecipeStep

ATTRIBUTE	DESCRIPTION
ParentRE	Identifies the RE or master recipe with which this Step is associated.
ParentVersion	Identifies the version of the RE or master recipe with which this Step is associated. When combined with a "ParentRE," this field defines a unique instance of an RE.
StepID	Identifies the unique execution instance of the referenced RE, with a name that is unique to the scope of its parent RE. (A simple example might just be the step number in the RE.)
RE_ID	Identifies the name of the RE that this step is an instance of, with a name that is unique to the scope of its parent RE.
REVersion	Identifies the version of the RE this step is an instance of.
VerticalStart	Specifies the vertical starting position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
VerticalStop	Specifies the vertical stopping position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
HorizontalStart	Specifies the horizontal starting position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
HorizontalStop	Specifies the horizontal stopping position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
ScaleReference	Specifies the reference size for recipe elements; all formula values are based on this reference size (e.g., 1234.5 kg).
ScaleEngrUnits	Specifies the units of ScaleReference.
MaximumScale	Specifies the maximum scale factor, or size, of the recipe element.
MinimumScale	Specifies the minimum scale factor, or size, of the recipe element.

5.2.5.2.1 Recipe element

The BXT_MRecipeElement table (see table 36) contains one record for each Procedural Element that is referenced in the exchanged master recipe. This table contains the definition of the Element, not the use of the element. In the table, one record exists for the procedure, for each unit procedure, for each operation, and for each recipe phase that is exchanged. The BXT_MRecipeElement and BXT_MRecipeElementParameter tables contain the specifications for the use of the RE, the number and types of parameters that are passed, and the default values for the parameters.

REs shall be unique to a parent RE. The RE_ID is a fully qualified name of the RE under its parent REs; therefore, the RE_ID is enough to contain all of the parent's RE_IDs.

5.2.5.2.2 Transitions

The BXT_MRecipeTransition table contains one record for each transition connection that is defined by the REs (see table 38). These records correspond to the transitions in Procedure Function Charts (see clause 6).

Table 38 — BXT_MRecipeTransition

ATTRIBUTE	DESCRIPTION
RE_ID	Identifies the RE the transition is contained in.
REVersion	Identifies the version of the RE. When combined with a "RE_ID" defines a unique instance of a master recipe.
TransitionID	Identifies the unique execution instance of this transition element. The ID contains the full hierarchy of parent RE instance names of which this transition is a member element.
Condition	Contains the expression or condition that, if TRUE, causes or allows the transition.
VerticalStart	Specifies the vertical starting position in the presentation of this element in the procedural view of the RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
VerticalStop	Specifies the vertical stopping position in the presentation of this element in the procedural view of the RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
HorizontalStart	Specifies the horizontal starting position in the presentation of this element in the procedural view of the RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
HorizontalStop	Specifies the horizontal stopping position in the presentation of this element in the procedural view of the RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.

5.2.5.2.3 Links

The BXT_MRecipeLink table contains one record for each connection that is defined in the REs and/or transitions (see table 39). These records correspond to the lines that connect elements in the Procedure Function Charts that are described in clause 6.

Table 39 — BXT_MRecipeLink

ATTRIBUTE	DESCRIPTION
RE_ID	Identifies the RE with which the Step and/or Transition is associated.
REVersion	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a RE.
LinkID	Specifies a unique ID for the link, which simplifies access to the table.
FromType	Kept as an enumeration, defines if the FromElement specifies a StepID for a step or a TransitionID for a transition, from the enumeration set "LinkToType."
FromElement	Specifies the Step name or TransitionID. The ID contains the full hierarchy of parent RE instance names of which this element is contained. This shall match the step or transition name in the step or transition table.
ToType	Kept as an enumeration, defines if the ToElement specifies a StepID for a step or a TransitionID for a transition, from the enumeration set "LinkToType."
ToElement	Specifies the Step name or TransitionID. The ID contains the full hierarchy of parent RE instance names of which this element is contained. This shall match the step or transition name in the step or transition table.
LinkType	Specifies if the link is a procedural control flow or a material transfer association. Kept as an enumeration, as ControlLink or TransferLink, from the enumeration set "LinkType."
VerticalStart	Specifies the vertical starting position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
VerticalStop	Specifies the vertical stopping position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
HorizontalStart	Specifies the horizontal starting position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
HorizontalStop	Specifies the horizontal stopping position in the presentation of this element in the procedural view of the parent RE, in normalized coordinates of (0,0) upper left to (1,1) lower right.
Depiction	Defines how the link is presented, from the enumeration set "LinkDepiction."
EvaluationOrder	Defines the specified order of evaluation of the link (if required) to meet the left-to-right evaluation of PFC transition checks that are specified in clause 6. All links from the same step to multiple transitions are assumed to be evaluated in the order that is specified by the order field. Lower numbers are evaluated first.

5.2.5.2.4 Parameters

The BXT_MRecipeElementParameter table contains one record for each parameter for each RE that is defined (see table 40). For example, a RE called *CHARGE* may be defined with two parameters: the type of material to charge and the amount to charge. One record would exist in the BXT_MRecipeElement table and two records would exist in the BXT_MRecipeElementParameter table for the RE *CHARGE*.

Table 40 — BXT_MRecipeElementParameter

ATTRIBUTE	DESCRIPTION
RE_ID	Identifies the RE with which the parameter is associated.
REVersion	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a master recipe.
ParameterID	Identifies the Procedural Element parameter. Note that if the parameter is part of a set, the ParentParamID field is used and the parameter set becomes part of the name of the parameter. For example, for a parameter set of MINOR_CHARGES and a parameter of BLUE_DYE, the ParameterID would be MINOR_CHARGES.BLUE_DYE if the delimiter character is a period.
ParentParamID	Identifies the parent parameter set of which this parameter is a member. This field will be NULL if there is no parameter set.
DataInterpretation	Defines how the default parameter value is interpreted, as an enumeration (i.e., constant, reference, or equation), from the enumeration set "ValueType."
DataDirection	Defines how the parameter value is intended to be handled, as an enumeration (i.e., input, output, input/output, neither), from the enumeration set "DirectionType."
DefaultValue	Contains the default parameter value that is used if the instance of execution does not specify a value and it may be a member of a user enumeration set.
ValueType	Defines the data type of the value, from the enumeration set "ValueDataTypes."
Description	Describes the parameter or the use of the parameter in the RE.
EngrUnits	Identifies the engineering units of measure for the Value. (e.g., kg, pounds).
EnumSet	Identifies the enumeration that this element is a member of (if not NULL).
DefaultScaling	Specifies the default selection when defined as an instance in the BXT_MRecipeStepParameter table and it is kept as an enumeration from enumeration set "Boolean." If TRUE, the formula value will be scaled when the size of the batch is scaled. If FALSE, the formula value will not be scaled when the size of the batch is scaled. Non-linear scaling may be accomplished through the use of expressions in the formula values.
ParamType	Specifies the default selection when defined as an instance in the BXT_MRecipeStepParameter table. This field identifies the use of the formula value as an enumeration from the enumeration set "FormulaType" (i.e., Process Input, Process Output, or a Process Parameter). The enumeration set is user extensible.
ParamSubType	Specifies the default selection when defined as an instance in the BXT_MRecipeStepParameter table. This field identifies the use of the formula value as an enumeration from the FormulaSubType enumeration set. Elements in this set are all user defined.

5.2.5.2.5 Standard sub-parameters

This standard recognizes a set of sub-parameters that may be related to a parameter value, to further qualify its definition and use. For example, defining and transferring the high and low limits that a parameter is allowed to take may be useful. This type of information is transferred in the exchange tables by defining sub-parameters for the affected parameter. Sub-parameters are defined for a given parameter, by creating a new table record with the affected ParameterID used as the ParentParamID. The set of standard sub-parameters and their defined use is given in table 41. Other sub-parameters may also be defined by the user. This set of sub-parameters should be supported for use with the

BXT_MRecipeElementParameter table, the BXT_MRecipeStepParameter table (see 5.2.5.3.1), the BXT_EquipInterfaceParameter table (see 5.3.5.8), and BXT_ScheduleParameter table (see 5.4.3.4).

Table 41 — Standard sub-parameters

PARAMETERID	DESCRIPTION
HighValueLimit	Specifies the largest value that the associated parameter is allowed to take on.
LowValueLimit	Specifies the smallest value that the associated parameter is allowed to take on.
HighTolerance	Specifies the largest upward deviation from the value of the associated parameter that is allowed.
LowTolerance	Specifies the largest downward deviation from the value of the associated parameter that is allowed.

5.2.5.3 Formula

Values of components that are to be used in the production or execution of a batch are passed as step parameters and their limits**.** The BXT_MRecipeStepParameter table contains one record for each parameter that is used for each Step that is defined (see table 42). Because REs can evaluate default parameter values, not all parameters of a RE need to be defined in the Step.

Table 42 — BXT_MRecipeStepParameter

ATTRIBUTE	DESCRIPTION
ParentRE	Part of the KEY to the BXT_MRecipeStep table for this parameter.
ParentVersion	Part of the KEY to the BXT_MRecipeStep table for this parameter.
StepID	Identifies the unique execution instance of the RE, with a name unique to the scope of its parent RE.
ParameterID	Identifies the RE parameter.
ParentParameterID	Identifies the parent parameter ID if defined. (Allows for parameter sets.)
ParameterValue	Contains the parameter value (numbers are transferred as ASCII representations of the number). If the parameter type is a user enumeration set, then this may be a member of the set.
DataInterpretation	Defines how the parameter value is interpreted, as an enumeration (constant, reference, external, or equation), from the set "ValueType."
Scaled	Kept as an enumeration, from the enumeration set "Boolean." If TRUE, then the parameter value will be scaled when the batch size is scaled. If FALSE, then the formula value will not be scaled when the batch size is scaled.

5.2.5.4 Other information

Other information typically has to be transferred with the recipe (see table 43). The meaning of the information is not specified in the exchange definition, but it is an agreement between the sender and receiver. This other information is usually extra documentation or descriptive information that may be needed in order to exchange a valid master recipe, but it is not information that is needed in order to execute the recipe. Examples of other information may include compliance documentation, molecular structure diagrams or even pictures of the expected product. The RE Other Information table, named BXT_MRecipeOtherInformation, contains this data.

Because the structure and form of this data are so variable, the recipe exchange language provides a means to reference, or point to, this information. The actual information may be exchanged in the DataValue field, if it is text information, or it may reference other files whose names are in the DataValue fields. Exchange of the other files is outside the scope of this standard.

Table 43 — BXT_MRecipeOtherInformation

ATTRIBUTE	DESCRIPTION
RE_ID	Identifies the RE with which the "other" data is associated.
REVersion	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of the RE.
StepID	Identifies the unique execution instance of the RE if the other information is associated with the step. If null, then the other information is associated with the RE.
DataID	Defines the identification of a data element with other information.
DataType	Identifies the type of the data value from the enumeration set ValueDataType.
DataValue	Specifies the value of the other data information. This field may be the name of a file that contains the actual data. The importing tool should have access to this file.
Description	Describes the data element type.

5.2.5.5 Equipment requirements

RE equipment requirements contain the equipment constraints and conditions of the recipe, and they are defined using similar tables and relations as those that are used to define the equipment capabilities in 5.3.

The equipment constraints and conditions may be associated with any Recipe Element in the Recipe Element hierarchy. RE equipment requirements define the equipment constraints and conditions that are applicable at that level (e.g., at the Unit Procedure or Operation level).

The equipment requirements are contained in the BXT_MRecipeElementEquip table and the BXT_MRecipeStepEquip table. The BXT_MRecipeElementEquip table contains the definition and default value of the property, and the BXT_MRecipeStepEquip table contains the specific property value that is required for a step.

5.2.5.5.1 RE equipment requirement

A RE may have equipment requirements (e.g., "Reactor name" or "Reactor lining"), and these specify the required properties for the RE to execute. Specific allowed values for the property for a specific step are defined in the BXT_MRecipeStepEquip table.

The equipment requirement may define data elements that the process cell supplies for use in the recipe. These elements could be used in the recipe’s transition conditions or elsewhere in expressions. Examples of data elements are "VesselPressure" available from a unit and "SteamPressure" from an equipment module.

Table 44, named BXT_MRecipeElementEquip, defines this set of related equipment requirements.

Table 44 — BXT_MRecipeElementEquip

ATTRIBUTE	DESCRIPTION
RE_ID	Identifies the RE for which the equipment requirements are needed. When combined with the "version," this field defines a unique instance of the RE.
REVersion	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a RE.
PropertyID	Defines the equipment property needed for the RE to execute (e.g., "Reactor Class" or "Vessel Pressure Tag").
DefaultValue	Defines the default value for the property, if none is specified in the steps (e.g., "Exothermic Reactor" or "VesPressure").
DataInterpretation	Specifies how the value is interpreted, and it is defined as an enumeration (i.e., constant, reference, external, or equation), from the enumeration set “ValueType.”
EvaluationRule	Specifies how the value is compared against the equipment property value, and it is defined as an enumeration, from the enumeration set "EvaluationRule.”
EngrUnits	Identifies the engineering units of measure for the value (e.g., kg, pounds).
Description	Describes what the property is and it may include why it is needed for the recipe.

5.2.5.5.2 RE step equipment requirement

A BXT_MRecipeStep may have a specific value for an equipment property (e.g., "Reactor Class" of "Exothermic" or "Reactor Lining" = "Glass"). Table 45, named BXT_MRecipeStepEquip, defines this set of related equipment requirements.

Table 45 — BXT_MRecipeStepEquip

ATTRIBUTE	DESCRIPTION
ParentRE	Part of the KEY to the BXT_MRecipeStep table for this parameter.
ParentVersion	Part of the KEY to the BXT_MRecipeStep table for this parameter.
StepID	Identifies the unique execution instance of the RE, with a name unique to the scope of its parent RE.
PropertyID	Identifies the RE property.
PropertyValue	Contains the property value (numbers are transferred as ASCII representations of the number). If the parameter type is a user enumeration set, then this may be a member of the set.

The "meaning" of the constraints is not important to the language definition; however, it is an agreement on the naming of equipment sets or equipment entities.

5.3 Process cell equipment model exchange

A set of tables is defined that describes the capabilities of the equipment in a process cell. This information reflects the actual capabilities of a process cell. The exchange of process cell capabilities can be useful as an information exchange, even if it is not used in conjunction with recipes.

5.3.1 Equipment description

The process cell capabilities tables are organized around a hierarchy of equipment descriptions. The elements in the hierarchy correspond to the elements in the Part 1 equipment hierarchy.

The equipment information exchange tables include the following information:

a) Equipment element - This defines a specific equipment element or a class of equipment.

b) Equipment procedural element interface - This defines the interface to a procedural element that is available in the equipment element.

c) Equipment element property - This defines the properties of the equipment, in a manner that mirrors the property specification of the RE Equipment Properties.

d) Equipment element link - This defines the linkage between equipment, in a manner that mirrors the property specification of the Master Recipe RE Equipment Links.

5.3.2 Table overview and integrity constraints

Figure 18 shows the structure of the equipment information tables.

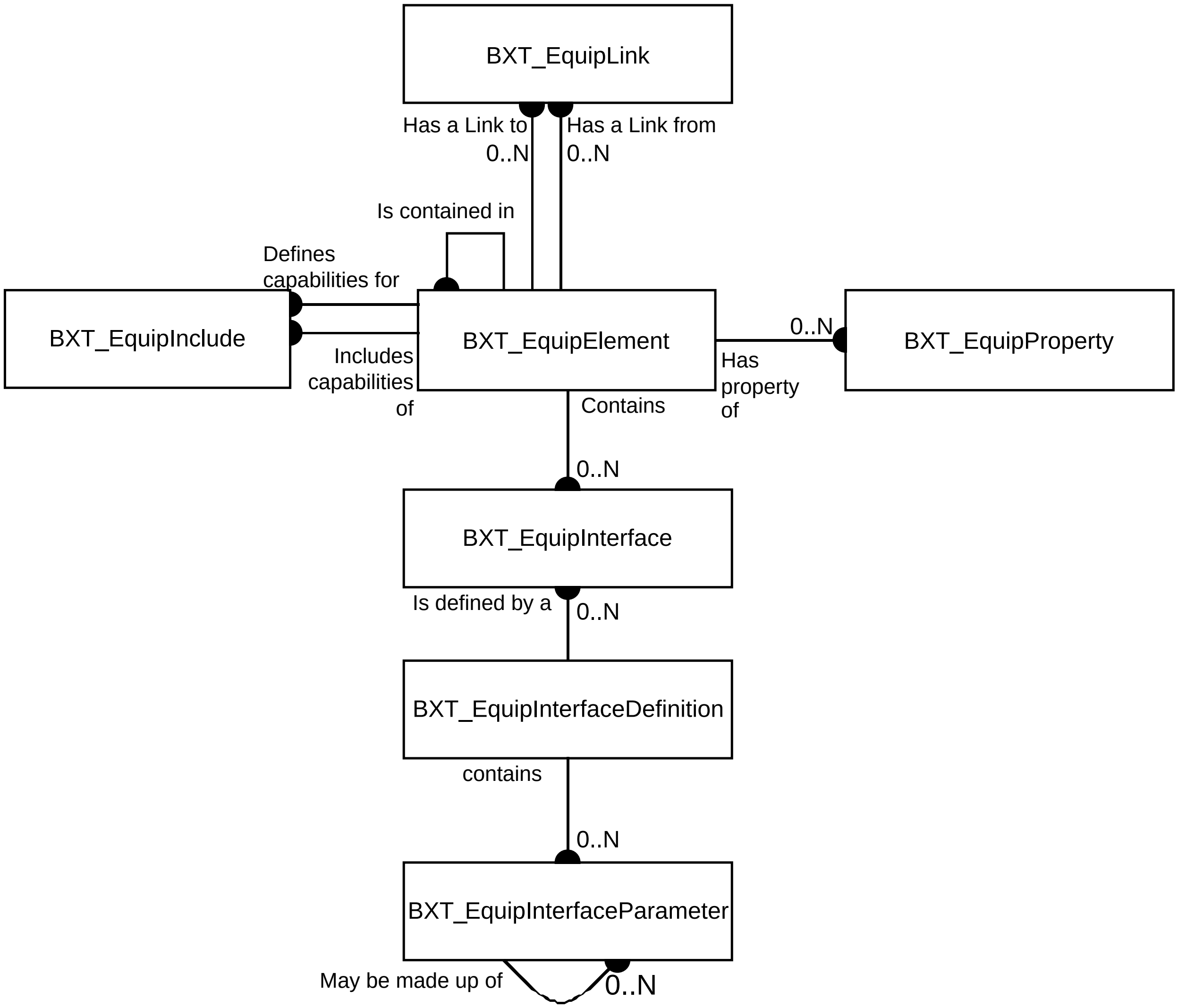


Figure 18 — Equipment information exchange tables

5.3.3 Equipment description table overview

5.3.3.1 Equipment hierarchy

The equipment information tables contain the hierarchical definition of the equipment, through the "ContainedIn" attribute of the BXT_EquipElement table. This table describes which equipment is contained within another piece of equipment. For example, a process cell equipment element may contain equipment elements that are composed of units and/or equipment modules.

Each level may have associated property specifications, data specifications, and link specifications. Each level may also have associated procedural elements in order to support the full Part 1 model, but these will typically be associated with units or equipment modules.

5.3.3.2 Equipment classes

The equipment information tables allow the specification of equipment classes through the "include the capability of" association between elements. An equipment element may also include the capability that is defined in one or more equipment elements.

For example, a unit may be a member of a unit class because it implements a set of procedures that the class defines, because it contains a set of properties of the class, or any combination of this information. One record exists in the BXT_EquipElement table for each equipment instance definition and one record exists in the BXT_EquipElement table for each class definition.

5.3.4 Equipment information table summary

The tables that are used for equipment information exchange are listed in table 46.

Table 46 — Equipment information exchange tables

TABLE NAME	FULL NAME	DESCRIPTION
BXT_EquipElement	Equipment Element	One record for each element of equipment or class of equipment.
BXT_EquipLink	Equipment Element Link Specification	One record for each link between equipment elements.
BXT_EquipInclude	Equipment Element Includes	One record for each equipment element that belongs to a class of equipment elements.
BXT_EquipProperty	Equipment Element Property Specification	One record for each property specification and its value for an equipment element.
BXT_EquipInterface	Equipment Procedural Element Interface	One record for each equipment procedural element that is defined within an equipment element.
BXT_EquipInterfaceDefinition	Equipment Procedural Element Interface Definition	One record for each equipment procedural element interface class that is defined within an equipment element.
BXT_EquipInterfaceParameter	Equipment Procedural Element Interface Parameter	One record for each data element input to, or output from, the equipment procedural element that is defined within an equipment element.

5.3.5 Equipment table definitions

The following subclause contains the detailed specifications of the equipment information tables.

5.3.5.1 Equipment element

The BXT_EquipElement table contains one record for each equipment entity (i.e., process cell, unit, equipment module, control module) (see table 47). The BXT_EquipElement table contains the definitions of entities and classes of entities (e.g., "Reactor101", "Filter20", "Reactor", "Filter"). The BXT_EquipInclude table that is defined below is used to contain the class relationships.

Table 47 — BXT_EquipElement

ATTRIBUTE	DESCRIPTION
EquipmentID	Identifies an equipment element or class of equipment.
EE_Type	Identifies the type of record as an enumeration (i.e., Class definition or Element definition), from the enumeration "EquipmentType.”
EE_Level	Identifies the equipment level as an enumeration (i.e., Area, Unit, Equipment Module, Control Module), from the enumeration "EquipmentLevel.”
ContainedIn	Identifies the equipment element that this equipment is contained in (e.g., the process cell that a unit is contained in). This field may be NULL if the equipment is not contained in any equipment or if the containing equipment is not defined.
Description	Describes the equipment element.

5.3.5.2 Equipment element links

An equipment element may have an equipment linkage specification (e.g., that equipment may feed into other equipment). These links typically define material transfers that the process cell (or units) supports. For example, a recipe may specify that a MIXER links to a REACTOR. The BXT_EquipLink table contains the equipment linkage (see table 48).

Table 48 — BXT_EquipLink

ATTRIBUTE	DESCRIPTION
EquipmentID	Identifies an equipment element or class of equipment.
ToEquipmentID	Identifies an equipment element or class of equipment that the "EquipmentID" links to.
Description	Describes the type of the equipment element link.

5.3.5.3 Equipment element includes capability

An equipment element may include the capabilities of one or more equipment element classes. For example, an equipment element may exist that represents a class of reactors with a specific set of procedural elements that represent the processing capability of reactors. A specific unit would include the capability of the reactor class and it may add additional capabilities or specifications. The BXT_EquipInclude table contains the relationship between the equipment class and the equipment instance (see table 49).

Table 49 — BXT_EquipInclude

ATTRIBUTE	DESCRIPTION
EquipmentID	Identifies an equipment element or class of equipment.
ClassEquipmentID	Identifies a class of equipment within which the EquipmentID is contained.
Description	Describes the association.

5.3.5.4 Equipment element properties

Equipment element properties define the capability that is available within an equipment element. This is done by specifying what equipment is available, in the same form as is specified in the recipe requirements, as a specification definition and the value for the specification.

The specifications may be associated with any Equipment Element in the equipment hierarchy and they define the specifications that are applicable at that level (e.g., at the process cell, unit, or equipment module level).

The BXT_EquipProperty table (see table 50) contains one record for each property that the equipment element has available (e.g., "lining type" and "glass lined unit", "size" and "50,000 gallon"). Specifications may also include equipment data elements that are available from the equipment (e.g., "SteamTemperature" from a heating equipment module, "VesselPressure" from a unit). The data elements could be available for use in recipe transitions and expressions.

Table 50 — BXT_EquipProperty

Attribute	Description
EquipmentID	Identifies an equipment element or class of equipment.
PropertyID	Identifies a property that this equipment supplies (e.g., "Lining Type," "Size," "Heat Capability," "Steam Temperature").
PropertyValue	Identifies the value for the property (e.g., "Glass," "50000," "650").
EngrUnits	Specifies engineering units of the property (e.g., "gallons," "BTU/hr").
Description	Describes the type of the equipment element property.

5.3.5.5 Equipment procedural element interface

The BXT_EquipInterface table (see table 51) contains one record for each equipment element procedure that is defined within an equipment element. Each BXT_EquipInterface record provides a mapping to the interface definition for the associated equipment element procedure. Because multiple BXT_EquipInterfaces (e.g., equipment phases) may have exactly the same external interface definition, they may all reference a single BXT_EquipInterface interface definition. This structure allows the definition of functionally equivalent BXT_EquipInterfaces, in order to allow for class-based recipes.

Table 51 — BXT_EquipInterface

ATTRIBUTE	DESCRIPTION
EquipmentID	Identifies an equipment element or class of equipment.
EPI_ID	Identifies the equipment procedural element interface of an equipment procedural element.
EPI_Definition	Identifies the definition of the BXT_EquipInterface.
Description	Describes the type of the equipment procedural element.

5.3.5.6 Equipment procedural element interface definition

The BXT_EquipInterfaceDefinition table (see table 52) contains one record for each equipment procedural element interface that is defined. The BXT_EquipInterfaceDefinition table contains the definition of an BXT_EquipInterface's input and output parameters.

Table 52 — BXT_EquipInterfaceDefinition

ATTRIBUTE	DESCRIPTION
EPI_Definition	Identifies the equipment procedural element interface definition.
Description	Describes the expected behavior of the BXT_EquipInterface.

5.3.5.7 Equipment procedural element interface parameter

The BXT_EquipInterfaceParameter table (see table 53) contains one record for each data element that is required by, generated by, or modified by an execution of the equipment procedural element that is defined within an equipment element. The BXT_EquipInterfaceParameter table contains the definition of the type and units of the required data element, and an optional reference to an enumeration set.

The BXT_EquipInterfaceParameter table also indicates if the input value is scaleable and it provides a default value that would be used if no actual value is passed to the equipment procedural element.

Table 53 — BXT_EquipInterfaceParameter

ATTRIBUTE	DESCRIPTION
EPI_Definition	Identifies the equipment procedural element interface class.
ParameterID	Identifies the name of the parameter that is used by the equipment procedural element.
ParentParamID	Identifies the parent parameter set of which this parameter is a member, and this field is NULL if there is no parameter set.
Type	Identifies the type of the data that is used by the equipment procedural element, from the enumeration set ValueDataType.
EngrUnits	Identifies the engineering units of the data that is used by the equipment procedural element.
EnumSet	Identifies the enumeration set of which this element is a member (i.e., if not NULL).
Scaled	Specifies if the parameter may be scaled before it is passed to the equipment procedural element. If this field is TRUE, the parameter may be scaled, from the enumeration set "Boolean.”
DefaultValue	Identifies the default value that is used if no value is passed to the equipment procedural element.
Description	Describes the equipment procedural element parameter.

5.4 Schedule information exchange

The schedule tables define one or more scheduled batches for a given process cell. Each scheduled batch contains additional batch specific information that may be used along with the master recipe information, in order to create a control recipe. Each scheduled batch may define a set of parameter values that are needed for recipe creation and a set of equipment requirements. Not all information that is needed to execute a control recipe needs to be provided in the schedule information exchange tables; additional information may be supplied by the control system or operator.

Scheduling activities may require information in addition to that provided by these tables. Scheduling may need to use other means in order to obtain some of the data (e.g., the status of equipment, material inventories, utility conditions). This information is best exchanged by other methods.

5.4.1 Schedule table overview

The schedule information exchange tables allow for information on multiple scheduled batches to be included in a single set of tables.

The schedule information exchange tables do not specify how the information is created or how it is used. Tools that use the information may include scheduling packages, batch automation packages, operational display packages, and work-order tracking packages. The importing and exporting tools shall determine the correct use of schedule information in the tables.

Figure 19 shows the five tables that make up the schedule information exchange tables.

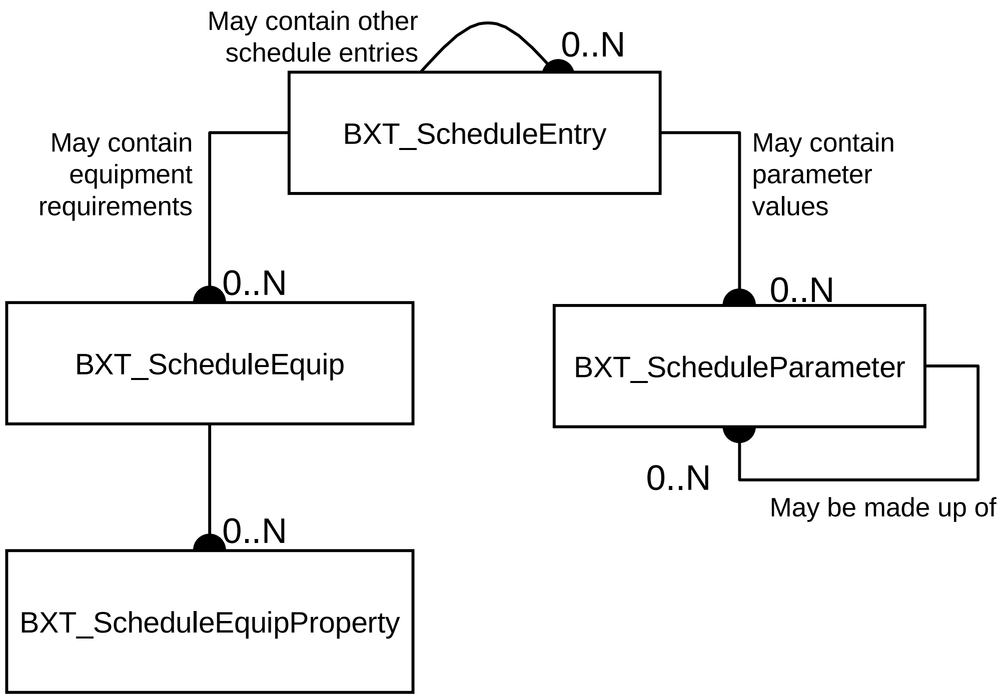


Figure 19 — Schedule structure

5.4.2 Schedule table summary

The tables that are used for schedule information exchange are listed in table 54.

Table 54 — Schedule information exchange tables

TABLE NAME	DESCRIPTION
BXT_ScheduleEntry	One record for each scheduled item (i.e., a batch, unit recipe, or cleaning event).
BXT_ScheduleEquip	One record for each equipment selection requirement. Permissible equipment selections may be defined.
BXT_ScheduleProperty	One record for each property specification for each equipment requirement.
BXT_ScheduleParameter	One record for each parameter item of the schedule record.

5.4.3 Schedule table definitions

5.4.3.1 Schedule entry

The BXT_ScheduleEntry table (see table 55) contains one element for each scheduled event. Schedule entries can represent a batch or some other processing activity (e.g., a unit recipe). If the scheduled event is a batch, the table contains the batch identification that is associated with the scheduled batch (i.e., if assigned at this time) in the ScheduleEntryID field and the master recipe that is associated with the scheduled batch is identified in the RE_ID field. The ScheduleEntryString field is used to uniquely identify scheduled entries when their "real" ID has not yet been assigned.

Table 55 — BXT_ScheduleEnTRY

ATTRIBUTE	DESCRIPTION
ScheduleEntryID	Identifies the unique ID (within this table) of the schedule entry. This may be a campaign ID, batch ID, unit procedure ID, or unique string with no external meaning.
ParentSchedID	Identifies the parent scheduled item record to which this record is related by using the parent's ScheduleEntryString.
ExternalID	Defines the identifier that is used by the business to identify this schedule entry.
RE_ID	Identifies the Recipe Element (e.g., *Red Oak*). When combined with the "version," this field defines a unique instance of a RE. When the record represents the master recipe, this field contains the master recipe ID. (Identifies the Recipe Element that is referenced by this schedule entry.)
REVersion	Identifies the version of the master recipe.
SE_Type	Defines the type of entity that this schedule record represents, from the enumeration set "SE_Type". This definition allows a batch record to have more detailed schedule information at lower levels in the procedure hierarchy (i.e., property and equipment requirements can be identified for each unit procedure in the recipe), as well as handle the scheduling of campaigns and groups of batches.
BatchID	Identifies the ID of the batch for which this schedule item is a part.
LotID	Identifies the ID of the lot for which this schedule item is a part.
CampaignID	Identifies the ID of the production campaign for which this schedule item is a part.
ProductID	Identifies the product to be made.
OrderID	Identifies the production order or customer order(s) with which this schedule record is related.
SE_Action	Defines the expected action by the receiving tool, as an enumeration (i.e., New, Update, Delete, User defined), from the enumeration set "ScheduleAction."
SchedStatus	Defines the status of the schedule record (i.e., Complete, In-process, Scheduled, Schedule Hold), from the enumeration set "ScheduleStatus."
StartCondition	Specifies the expected starting condition of the schedule record, if known (e.g., "Starts before...", "Follow...").
InitialMode	Defines the mode in which the schedule record begins execution as an enumeration (i.e., Automatic, Semi-automatic, Manual), from the enumeration set "ScheduleMode."
SchedStartTime	Specifies the expected starting time of the schedule record, if known.
SchedEndTime	Specifies the expected end time of the schedule record, if known.
BatchPriority	Specifies a priority that is placed on the schedule record, if known. Lower numbers have higher priority (e.g., Priority 1 is more important than Priority 7).
BatchSize	Defines the requested size, or scale factor, for the batch, based on the scale factor for the batch, as defined in the master recipe.
EngrUnits	Optionally identifies the engineering units of measure for the BatchSize.
SENote	Provides information or instructions to operations.
Description	Describes the scheduled item and/or product (e.g., Premium Beer).

5.4.3.2 Schedule record equipment requirement

The BXT_ScheduleEquip table (see table 56) contains one element for each equipment requirement for a schedule record. The related BXT_ScheduleEquipProperty table contains a specific property definition that this equipment needs to satisfy.

Typical requirements for a batch may be equipment selections. A schedule record's equipment requirements and properties will generally correspond to a master recipe's equipment requirements and properties. For example, a scheduling package may specify a specific unit to be used in production. The BXT_ScheduleEquip would specify the equipment identity as defined in a RE and the BXT_ScheduleProperty would specify the name of the selected unit.

Table 56 — BXT_ScheduleEquip

ATTRIBUTE	DESCRIPTION
ScheduleEntryID	Identifies the scheduled item that is exchanged.
RequirementID	Provides a unique name for the equipment requirement of a scheduled item. This name may refer to an individual piece of equipment, a class of equipment, a list of allowable equipment, or other groups of equipment (e.g., a train).
Description	Describes the requirement (e.g., First reactor unit in the batch).

5.4.3.3 Schedule record equipment property requirement

The BXT_ScheduleProperty table (see table 57) contains one element for each property specification for each equipment requirement for a schedule record.

Because a single equipment requirement for a given scheduled item can have multiple property criteria (e.g., material of construction and volume), a separate table is used to list these requirements.

Table 57 — BXT_ScheduleProperty

ATTRIBUTE	DESCRIPTION
ScheduleEntryID	Identifies the scheduled item exchanged.
RequirementID	Identifies the requirement set associated with the batch, typically an equipment or material class.
PropertyName	Identifies the name of the property for the schedule batch.
PropertyValue	Specifies the value for the property for the schedule batch.
EngrUnits	Optionally identifies the engineering units of measure for the PropertyValue.
Description	A description of the property (e.g., Use UNIT345 as the first reactor unit in the batch).

5.4.3.4 Schedule parameter

The BXT_ScheduleParameter table (see table 58) contains one element for each parameter for a scheduled item. Parameters in a schedule item are typically parameters in the master recipe, but they may also be information for the operator or for other users of the schedule information.

Limits for schedule parameter items may be defined in the same way that sub-parameters further define the parameters in recipes.

Table 58 — BXT_ScheduleParameter

ATTRIBUTE	DESCRIPTION
ScheduleEntryID	Identifies the scheduled item that is exchanged.
ParameterID	Identifies the parameter for the scheduled batch.
ParentParameterID	Identifies the parent parameter set of which this parameter item is a member (i.e., NULL if there is none).
ParameterValue	Specifies the value for the parameter for the scheduled batch.
EngrUnits	Optionally identifies the engineering units of measure for the ParameterValue.
ItemLocation	Defines where in the recipe structure this parameter item is applied. This record is used when the ParameterID is a simple alias, and the ParameterID does not provide sufficient information to identify where it is applied.
EnumSet	Identifies the enumeration this element is a member of (i.e., if not NULL).
Description	Describes the parameter for the related scheduled item and/or product. (e.g., Premium Beer).

5.5 Production information exchange

Production information exchange provides a structure in order to exchange all of the information about the execution of a batch.

Many tools (e.g., batch automation systems, laboratory information management systems, batch reporting systems, batch analysis systems, scheduling systems, simulation systems) could use this information.

The structure of the exchange tables allows for data on multiple batches to be exchanged in the same tables.

Production information is contained in three areas:

a) the control recipe;

b) the equipment that the recipe ran against; and

c) a log of all events that occurred during the execution of the recipe.

5.5.1 Control recipe information

The control recipe information may be exchanged using the MR tables, with the ProductID as the representation of the batch identification.

Because the control recipe starts as a copy of a master recipe, the MR tables may contain the master recipe used for production and the event table may contain any changes to the control recipe, or the MR tables may contain the control recipe after modification. In either case, the RecipeID and REVersion information serve to identify the specific master recipe that was used to create the control recipe.

5.5.2 Equipment information

The equipment that is used in the production of the batch may be exchanged using the BXT_EquipElement (Equipment Entity) tables. The tables may contain the definition of the entire process cell that was used in production, or they may only contain the subset of equipment actually that was used in the production of the batch.

5.5.3 Batch history

The batch history is contained in two tables, the BXT_HistoryElement table and the BXT_HistoryLog table.

The BXT_HistoryElement table is the record of each execution of a recipe procedural element and/or the equivalent equipment procedural element. This table contains one element for each RE or EPE execution. Figure 20 shows the elements of batch history and the relationship to the previously defined equipment element tables. The BXT_HistoryElement table (see table 59) contains records that may be used to reference the associated equipment and equipment procedural element.

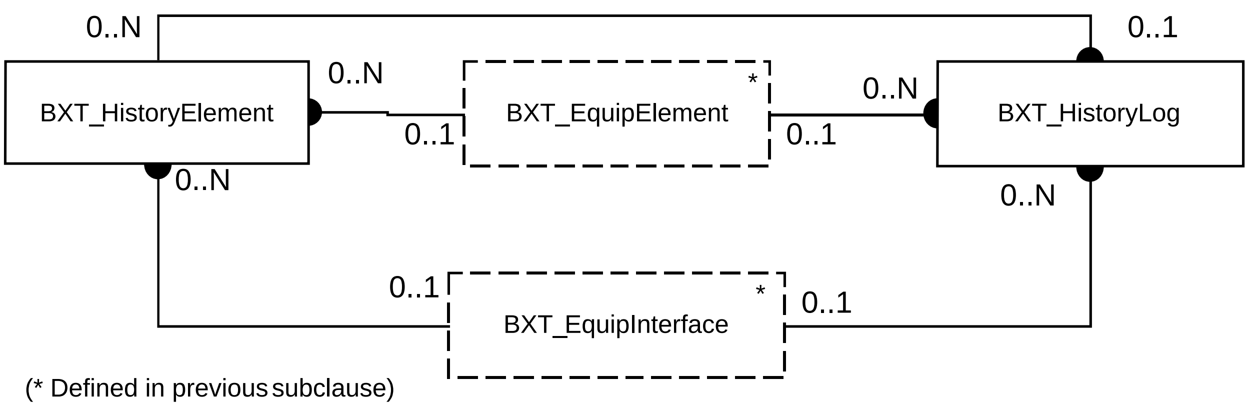


Figure 20 — Batch history

The BXT_HistoryLog and BXT_HistoryElement tables contain a list of time-defined records that occurred during the production of the batch, one record for each recorded event in the batch. The BXT_HistoryElement tables contains each instance of use of a recipe procedural element or equipment procedural element. The BXT_HistoryLog table contains one record for each event that occurred for the procedural element (e.g., the start of execution of the procedural element, a mode change, a state change, a value that is reported from the element).

The batch history tables are designed with several objectives:

a) Information that would be duplicated in multiple records is moved to the BXT_HistoryElement table and this provides a significant amount of reduction in the size of the BXT_HistoryLog table.

b) Information that can be described in the equipment tables is only referenced by key value. The BXT_HistoryElement record contains an equipment element identification to the associated equipment and an identification of the associated equipment procedural element.

c) The BXT_HistoryLog table also includes a reference to the equipment element and equipment procedural element, in order to simplify the use of the BXT_HistoryLog table, even though this information is duplicated in the BXT_HistoryElement table.

5.5.3.1 History element

The BXT_HistoryElement table contains one record for each recipe element (see table 59).

Table 59 — BXT_HistoryElement

ATTRIBUTE	DESCRIPTION
HistoryElementID	Provides a generated identification number that is required for relational integrity.
BatchID	Specifies a unique identification of the batch that is associated with the BXT_HistoryElement record. This is duplicated information with the BXT_HistoryLog table record.
MasterRecipeID	Specifies the identification of the master recipe that is associated with the batch.
MasterRecipeVersion	Specifies version identification of the related master recipe.
ControlRecipeID	Identifies the control recipe ID. In some cases, this identification may be different from the batch ID.
ReferenceEquipProcedure	Identifies whether the procedural control hierarchy is defined to be referring to recipe or equipment, as a Boolean enumeration, with True identifying a reference to equipment.
RecipeProcedure	Identifies the Procedure that is associated with the BXT_HistoryElement record.
UnitProcedure	Identifies the Unit Procedure that is associated with the BXT_HistoryElement record.
UnitProcedureCounter	Specifies an instance of execution counter, which is the number of times that the unit procedure has been executed. This counter is needed because the recipe may execute the same unit procedure multiple times due to manual intervention or looping.
Operation	Identifies the Operation that is associated with the BXT_HistoryElement record.
OperationCounter	Specifies an instance of execution counter, which is the number of times that the unit procedure has been executed. This counter is needed because the recipe may execute the same operation multiple times due to manual intervention or looping.
Phase	Identifies the Recipe Phase that is associated with the BXT_HistoryElement record.
PhaseCounter	Specifies an instance of execution counter, which is the number of times that the unit procedure has been executed. This counter is needed because the recipe may execute the same phase multiple times due to manual intervention or looping.
EquipmentID	Identifies the equipment element that may be associated with the BXT_HistoryElement record. This is duplicated information with the BXT_HistoryLog table record.
EPI_ID	Identifies the equipment procedural element that may be associated with the record. This is duplicated information with the BXT_HistoryLog table record.

5.5.3.2 History Log

The BXT_HistoryLog table (see table 60) contains five sets of information about the event that should be logged:

a) the time of the event;

b) the batch and recipe information that is associated with the event;

c) the equipment that is associated with the event;

d) the operator that is associated with the event; and

e) the event information.

The BXT_HistoryLog contains production information events. The production information events are categorized using enumerations in the RecordSet and RecordSubSet fields to aid information handling (e.g., filtering and sorting).

Table 60 — BXT_HistoryLog

ATTRIBUTE	DESCRIPTION
RecordID	Specifies a generated identification number that is required for relational integrity.
UTC	Identifies the Universal Coordinated Time (UTC) and date of the record.
LocalTime	Identifies the local time and date of the record.
BatchID	Provides a unique identification of the batch that is associated with the record.
HistoryElementID	Provides a unique identification of the instance of execution or the associated recipe element or equipment procedural element that may be associated with the record. This field is a key into the BXT_HistoryElement table.
EquipmentID	Identifies an equipment element that may be associated with the record.
EPI_ID	Identifies the equipment procedural element that may be associated with the record.
UserID	Specifies the name of the user, if any, who is associated with the record.
RecordSet	Specifies the type of the record, from the RecordSet enumerations.
RecordSubSet	Specifies the subtype of the record, from the enumeration set that is specified by the RecordSet record.
RecordAlias	Defines an equipment-independent record specification (e.g., "vessel top temperature").
NewValue	Specifies the data value that is associated with the record type and subtype.
OldValue	Defines a field that may contain the previous data value.
EngrUnits	Specifies the engineering units, if any, that are appropriate for the NewValue and OldValue.

5.6 Exchange table domains

The same information is contained in multiple tables for key fields and many of the fields in the tables have the same domain (i.e., specific data type and range). The ISO/IEC standard for SQL does not define domains, so table 61 contains the definition of the domains for selected table attributes in the exchange tables.

Table 61 — Exchange table domains

DOMAIN NAME	TYPE	DESCRIPTION
BXT_MRecipeElement-Author	CHAR (32)	Name or ID of the author.
BXT_MRecipeStep-StepID BXT_MRecipeStepParameter-StepID BXT_MRecipeStepEquip-StepID BXT_ScheduleEntry-StepID	CHAR (128)	Identifies a step within a recipe procedural element.
BXT_MRecipeElementParameter-EngrUnits BXT_MRecipeStep-ScaleEngrUnits BXT_EquipProperty-EngrUnits BXT_EquipInterfaceParameter-EngrUnits BXT_ScheduleEntry-EngrUnits BXT_ScheduleProperty-EngrUnits BXT_ScheduleParameter-EngrUnits BXT_HistoryLog-EngrUnits	CHAR (32)	Engineering Units Specification.
BXT_MRecipeElement-Status BXT_ScheduleEntry-SchedStatus	INTEGER	Status of the recipe or procedural element as an enumeration.
BXT_MRecipeElement-RE_ID BXT_MRecipeElementParameter-RE_ID BXT_MRecipeStep-ParentRE BXT_MRecipeStep-RE_ID BXT_MRecipeStepParameter-ParentRE BXT_MRecipeTransition-RE_ID BXT_MRecipeLink-RE_ID BXT_MRecipeElementEquip-RE_ID BXT_MRecipeOtherInformation-RE_ID BXT_MRecipeStepEquip-ParentRE	CHAR (128)	Identifies a recipe procedural element.
BXT_MRecipeElement-REVersion BXT_MRecipeElementParameter-REVersion BXT_MRecipeStep-ParentVersion BXT_MRecipeStep-REVersion BXT_MRecipeStepParameter-ParentVersion BXT_MRecipeTransition-REVersion BXT_MRecipeLink-REVersion BXT_MRecipeElementEquip-REVersion BXT_MRecipeStepEquip-ParentVersion BXT_MRecipeOtherInformation-REVersion BXT_ScheduleEntry-Version BXT_HistoryElement-MasterRecipeVersion	CHAR (16)	Version identifier for all elements with versions.
BXT_MRecipeElementParameter-ParameterID BXT_MRecipeElementParameter-ParentParamID BXT_MRecipeStepParameter-ParameterID BXT_MRecipeStepParameter-ParentParamID BXT_EquipInterfaceParameter-ParameterID BXT_EquipInterface-ParentParamID BXT_ScheduleParameter-ParameterID	CHAR (32)	Identifies a parameter of an RE, definition of use.

Table 61 — Exchange table domains (continued)

DOMAIN NAME	TYPE	COMMENTS
BXT_MRecipeElement-ProcessCellID BXT_EquipElement-EquipmentID BXT_EquipLink-EquipmentID BXT_EquipLink-ToEquipmentID BXT_EquipInclude-EquipmentID BXT_EquipInclude-ClassEquipmentID BXT_EquipProperty-EquipmentID BXT_EquipInterface-EquipmentID BXT_EquipInterfaceParameter-EPI_Definition BXT_ScheduleProperty-RequirementID BXT_ScheduleEquip-RequirementID BXT_HistoryLog-EquipmentID BXT_HistoryElement-EquipmentID BXT_ScheduleEquip-RequirementID BXT_HistoryLog-EquipmentID BXT_HistoryElement-EquipmentID	CHAR(32)	Identifies a process cell or other equipment entity.

6 Procedure function charts

This clause defines a method for graphical representation of Master and Control recipes. The representation of the procedure is called a Procedure Function Chart (PFC). This clause also addresses requirements for representation of formula, equipment requirements, header, and other information.

The PFC language, as defined in this standard, is designed to support recipes with complex procedures (e.g., parallel steps, selections) that vary from one product to another.

Procedure Function Charts build upon Function Charts, as defined in IEC 60848. In Procedure Function Charts, it is presumed that steps follow transitions and transitions follow steps, as described in IEC 60848. However, there are significant differences between that standard and this one that are necessary in order to meet the requirements of procedural control, as opposed to documentation. In batch control, equipment procedural elements contain the procedural logic that regulates when and how they complete. In addition, many equipment phases in batch control are designed to complete a task and then terminate. The PFC approach supports the separation of recipe procedural elements from equipment procedural elements, by recognizing that equipment procedural elements, once started, execute independently. Another difference that must be addressed in a procedure function chart is the multiple level structure of recipe procedural elements; therefore, it must be clear whether a symbol represents a recipe phase, a recipe operation, a recipe unit procedure, or an entire recipe procedure.

6.1 Procedure function chart notation

Procedure function charts depict procedural logic, using a series of symbols that are interconnected by directed links, in order to define the execution sequence of procedural elements. The execution of procedural elements may occur in series or parallel and the execution may be dependent upon conditional logic. Activities that are depicted include the intended execution of unit procedures, operations, phases, and the evaluation of transitions. In general, the flow of execution is top to bottom and left to right. Procedure function charts are used to depict the procedural logic for all levels of the recipe: recipe procedure, recipe unit procedure, and recipe operation.

6.1.1 Symbols

A procedure function chart is defined by a set of symbols for

a) elements (i.e., recipe procedural elements);

b) begin and end points;

c) resource allocation;

d) element synchronization;

e) recipe transitions; and

f) basic structures (i.e., directed links, sequence selection, simultaneous sequences).

Only the global representation of the symbols is imposed; dimensions and details (e.g., thickness of lines and font of characters) are left to each implementation.

6.1.1.1 Elements

Symbols shall be used to represent a recipe phase, a recipe operation, a recipe unit procedure, or a recipe procedure. A graphical indication within the symbol shall be used to identify the symbol as representing a recipe phase, a recipe operation, a recipe unit procedure, or a recipe procedure. One example for identifying procedural elements is shown in figure 21.

Part 1 defines only four levels in the procedural hierarchy and only those four levels are described in this document. However, additional levels are possible and they may be defined for various purposes. While this document only addresses the four levels of procedural elements that are defined in Part 1, any additional levels that are defined separately may be identified either graphically, following the same general principles described herein, or textually by a string that starts in the upper left-hand corner of the rectangle.

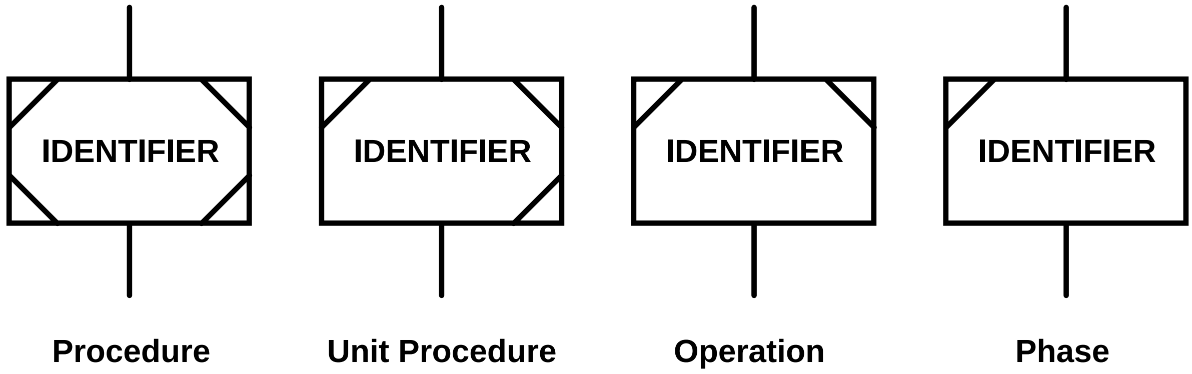


Figure 21 — Recipe procedural element symbols

A procedural element above the level of a phase may represent an encapsulation of other procedural elements at the next lower level in the procedural control hierarchy. A procedural element that represents an encapsulation, where the lower level recipe procedural elements are not shown, shall be identified by a plus sign (+) in the upper right hand corner of the rectangle, representing the procedural element (see figure 22). Procedural element symbols that expose the encapsulation shall be identified by a minus sign (-) (see figure 38). Procedural element symbols that reference an equipment procedural element shall have no indication.

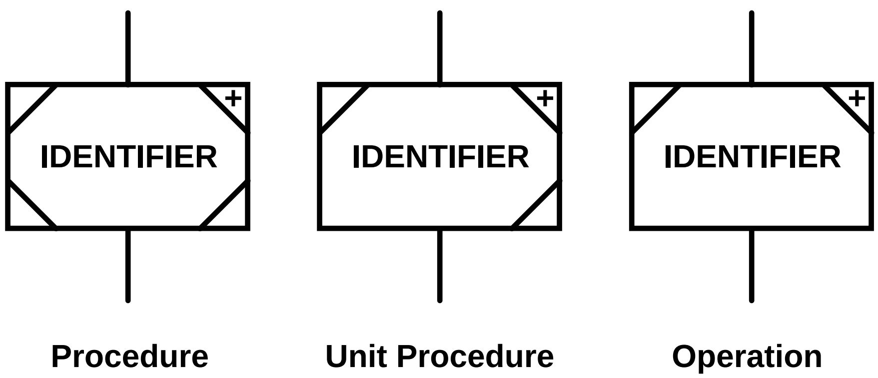


Figure 22 — Procedural elements that encapsulate lower-level recipe procedural elements

When a procedural element represents an encapsulation of subordinate procedural elements, a separate, lower-level procedure function chart that specifies the subordinate procedural elements and associated ordering symbols may be used to define it. The icon representing the encapsulating recipe procedural element may also be expanded to allow the lower level procedure function chart to be depicted within the boundaries of the encapsulating icon. The process of expansion of single symbols may continue until there is no subordinate level. An equipment procedure may possibly be shown as an expansion of the recipe procedural element that refers to it.

6.1.1.2 Begin and end points

Procedure function charts shall have at least one begin point and at least one end point, in contrast to a sequential function chart that may continuously cycle.

6.1.1.2.1 Begin

At least one begin symbol (see figure 23) shall be used to designate the beginning of each procedure function chart and/or each subordinate procedure function chart.

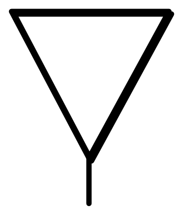


Figure 23 — Begin symbol

6.1.1.2.2 End

At least one end symbol (see figure 24) shall be used to designate the end of each procedure function chart and/or each subordinate procedure function chart.

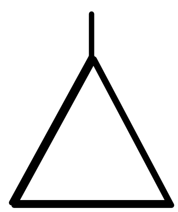


Figure 24 — End symbol

6.1.1.3 Resource allocation

Some resources are allocated to a batch before the batch can make use of them. Controlling the timing of this allocation may be important to the recipe or for scheduling; therefore, resource allocation rules and start conditions are needed. Allocation, if shown, shall be depicted by an oval icon (see figure 25) that represents the encapsulation of the resource allocation requirements for a recipe entity. The allocation symbol is a recipe procedural element appropriate to the level of the procedure function chart and it may be used at any level of the procedural hierarchy.

The allocation symbol represents the data and/or logic that determines what will be allocated to the batch (e.g., which specific unit or criteria for unit selection, equipment modules, materials, personnel) and when it will be allocated to the batch (e.g., two hours after the start of another unit procedure). The allocation symbol may contain logic that is to be executed by process management or through equipment procedural elements that are not necessarily associated with actual physical equipment. An explicit transition that follows the allocation symbol may be used to specify the starting condition for the following recipe entity.

The allocation symbol may also be used to specify explicit de-allocation. In this case, an appropriate text annotation should be used to indicate its use for de-allocation.

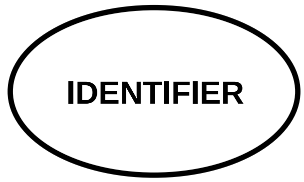


Figure 25 — Allocation symbol

6.1.1.4 Element synchronization

Synchronization between recipe elements may need to be depicted (see figure 26). Synchronization, if shown, shall be depicted by a rectangle that extends out of either side of the symbol for any of the recipe elements that are involved in the synchronization. Corresponding synchronization symbols may be connected with a dashed or other line that is distinguishable from a directed link, when the location of the symbols allows such notation without confusion. If the synchronization represents a material transfer, an arrowhead shall be added to indicate the direction of material movement intended. If no line is used, a unique identifier that identifies the specific interaction shall be provided at each recipe element that is involved in the synchronization.

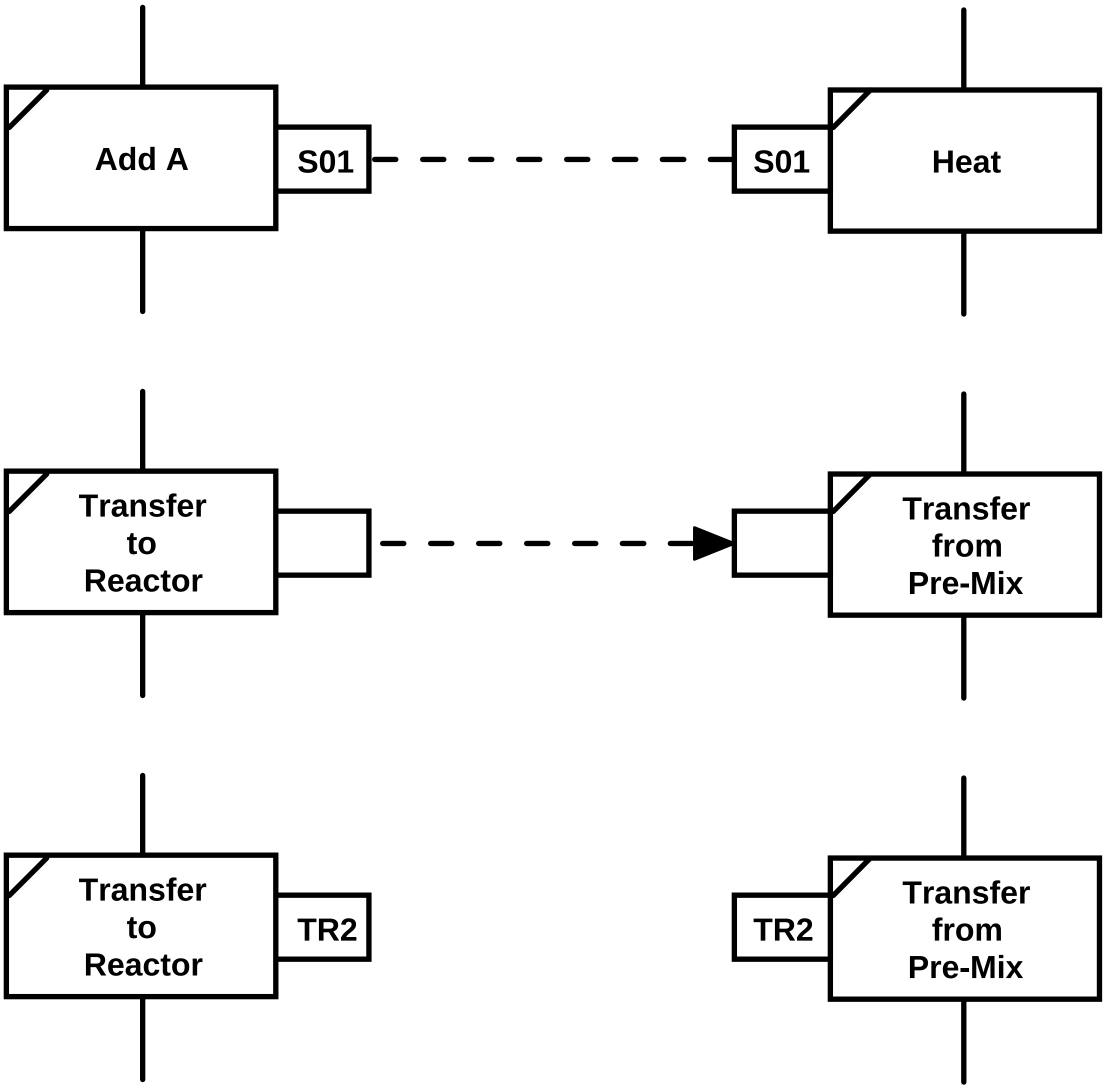


Figure 26 — Element synchronization examples

6.1.1.5 Recipe transitions

Procedure functions charts depict two types of transitions: an implicit and an explicit transition.

6.1.1.5.1 Implicit transitions

A directed link that consists of a single line between recipe entities (see figure 27) shall indicate a transition whose only condition shall be that the entities directly preceding the transition have finished their execution. No logic conditions shall be entered for this type of transition.

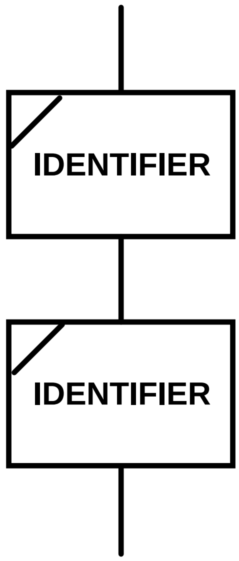


Figure 27 — Implicit transition

6.1.1.5.2 Explicit transition

A directed link that consists of a single line between recipe entities, with two short, tightly spaced double bars perpendicular to the link line (see figure 28), shall be used to indicate the explicit recipe transition.

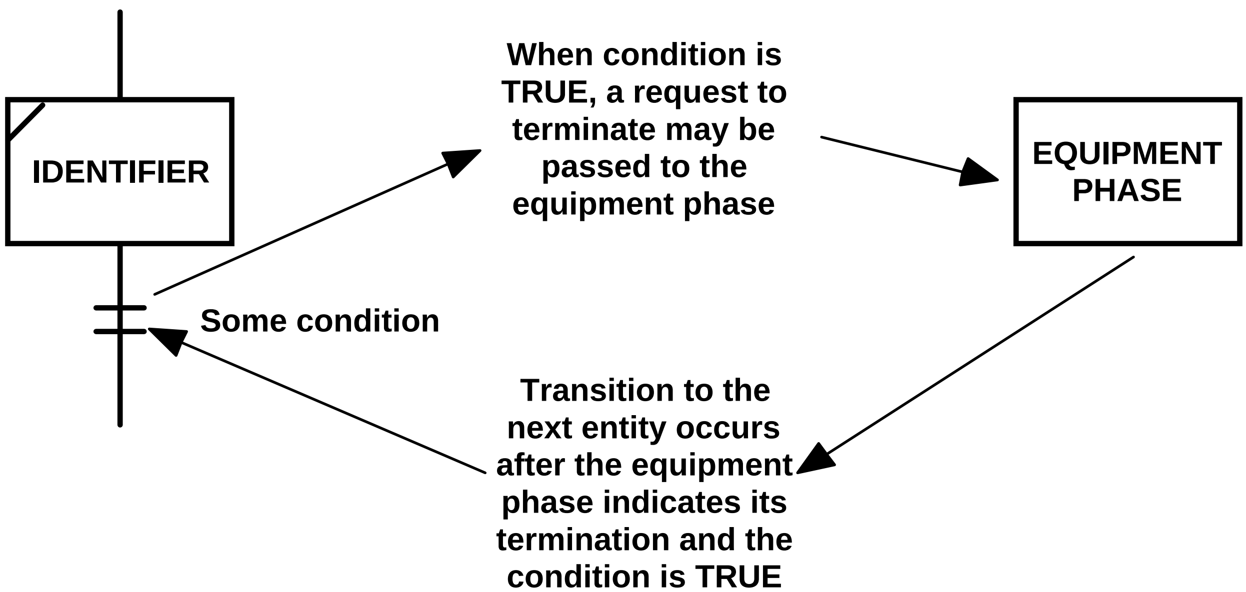


Figure 28 — Explicit transition

This transition is defined by an expression that evaluates to either true or false. The transition is continually evaluated, once the immediately preceding entity becomes active.

The transition is used to perform two functions:

a) to interrupt the execution of a branch of the recipe procedural logic; and

b) to request the termination of all immediately preceding procedural elements (e.g., unit procedures, operations, phases).

The termination of the immediately preceding procedural element may be a condition of the expression. However, a conditional language for the expression is not defined in this standard.

When a transition becomes true, the active procedural elements immediately preceding the transition shall be requested to terminate. When immediately preceding procedural elements terminate before the transition evaluates true, the transition shall continue to evaluate its logic until it is true.

An entity that immediately follows a transition shall be activated only after the transition condition evaluates true and the procedural elements preceding the transition terminate.

6.1.1.6 Basic structures

Structures define the intended thread of execution of the recipe elements. The simplest case is a series of recipe procedural elements that will be activated one after another. More complex structures include sequence selection and simultaneous sequences.

6.1.1.6.1 Beginning of sequence selection

The beginning of a sequence selection is shown in figure 29. Each branch of a sequence selection shall start with a transition. A selection of one path, out of several possibilities, is represented by as many transitions (i.e., under the horizontal line) as there are possibilities. Only one sequence shall be selected from the set of sequences below the line. The transitions shall be evaluated in left to right priority. The sequence below the transition that becomes true first, when evaluated in this manner, shall become the selected sequence.

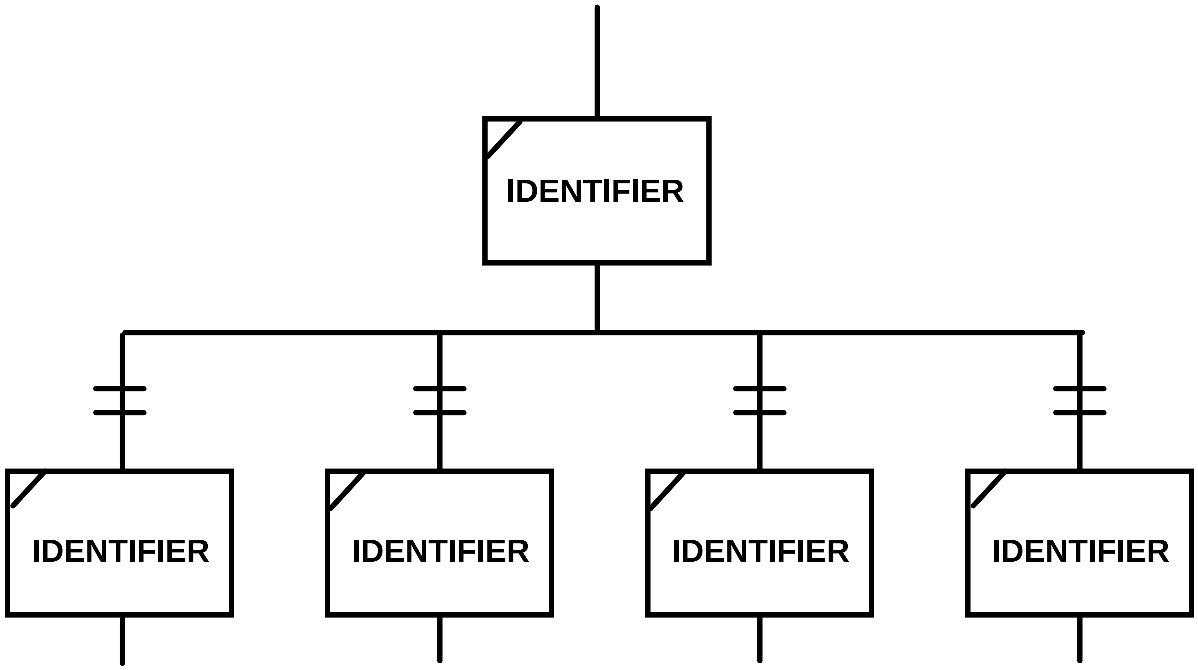


Figure 29 — Beginning of sequence selection

6.1.1.6.2 End of sequence selection

The end of sequence selection shows the joining of possible threads of execution from a sequence selection (see figure 30).

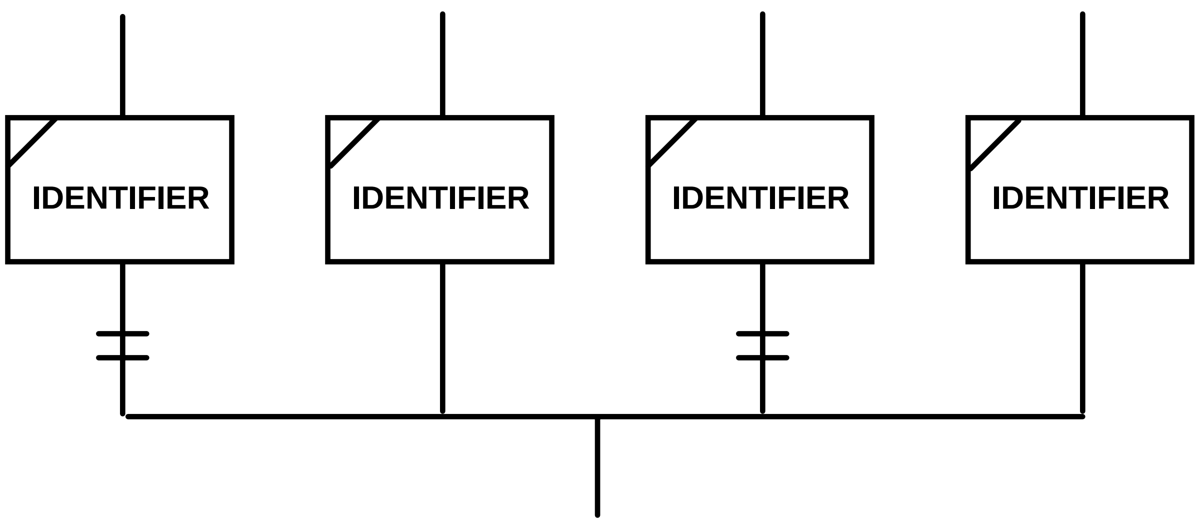


Figure 30 — End of sequence selection

6.1.1.6.3 Beginning of simultaneous sequences

The beginning of simultaneous sequences (see figure 31) shows the start of independent threads of execution of the recipe elements and there is one thread of execution for each path under a start of selection. All threads of execution shall be joined back to a single thread of execution in the recipe entity. The beginning and ending of threads of execution do not have to be matched. If an explicit transition is needed, then it shall be above the parallel lines.

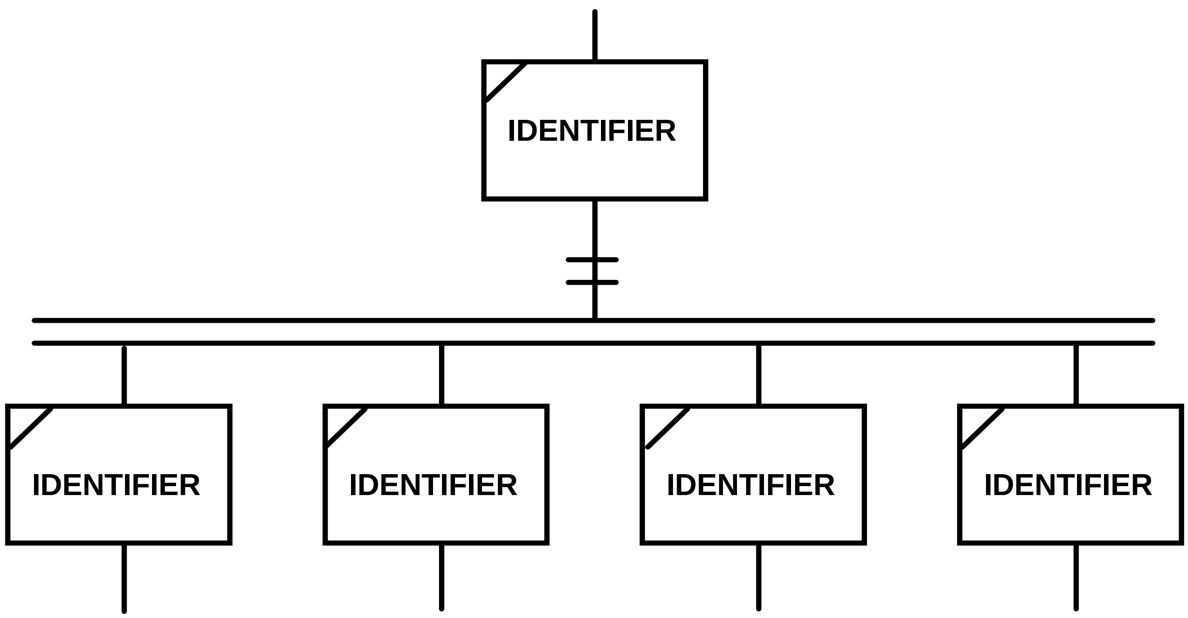


Figure 31 — Beginning of simultaneous sequences

6.1.1.6.4 End of simultaneous sequences

The end of simultaneous sequences shows the joining of independent threads of execution of the recipe element (see figure 32). The transition that immediately follows the parallel lines is evaluated only when all of the entities that immediately precede the parallel lines are either active or have completed.

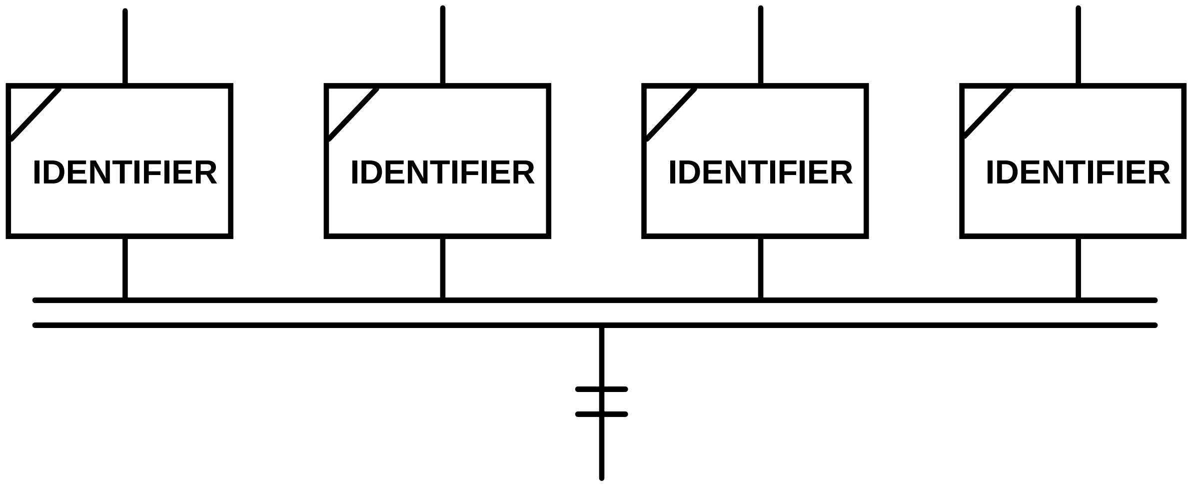


Figure 32 — End of simultaneous sequences

6.1.1.6.5 Rules for valid diagrams

Valid diagrams shall follow consistent rules for threads of execution. Independent simultaneous threads of execution shall be joined. The end of sequence selection cannot be used to join simultaneous threads of execution. Figure 33 shows an example of a valid diagram segment with sequence selection and end of sequence.

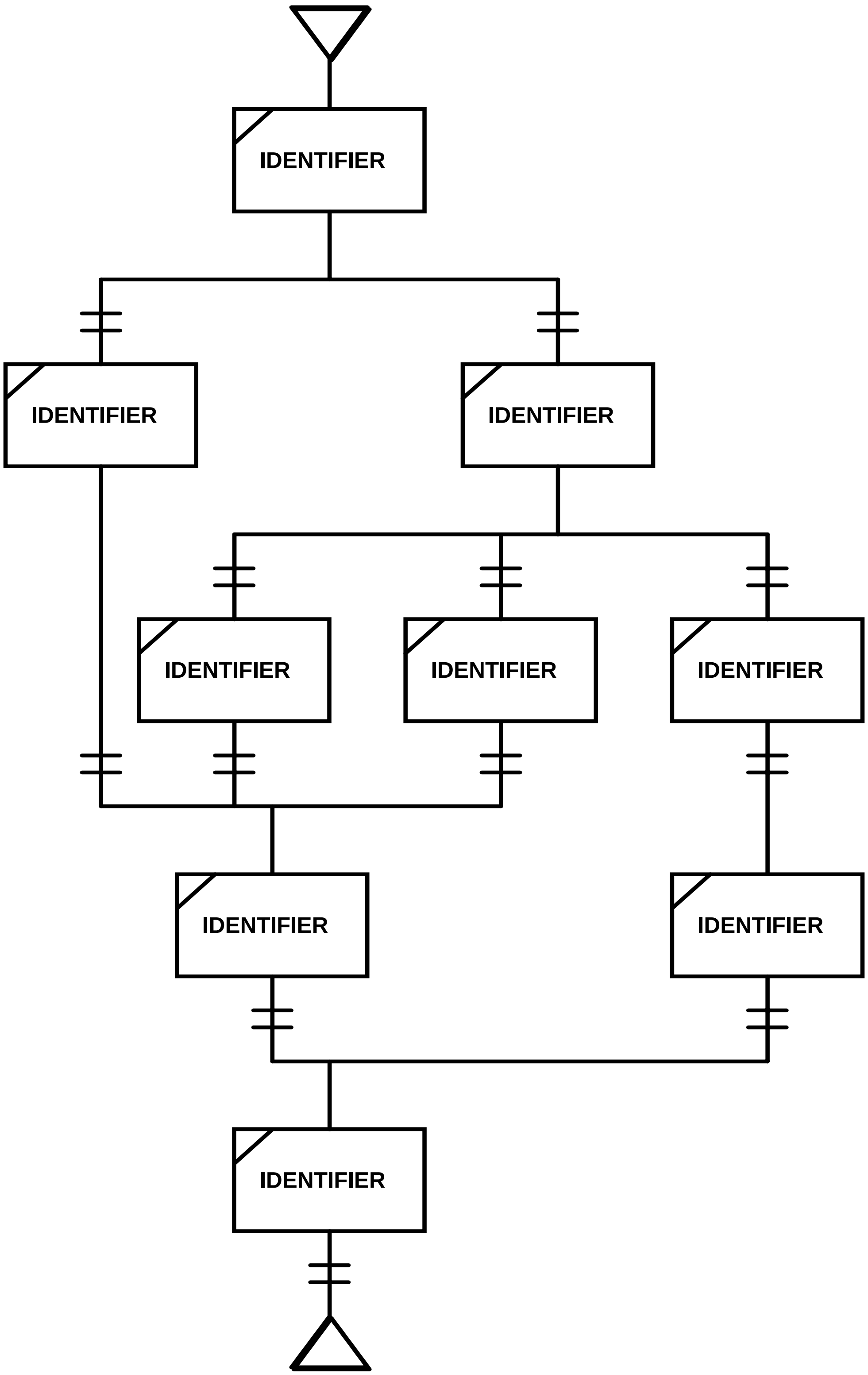


Figure 33 — Valid sequence selection diagram

Figure 34 shows an example of a valid diagram segment that shows start and end of simultaneous sequences.

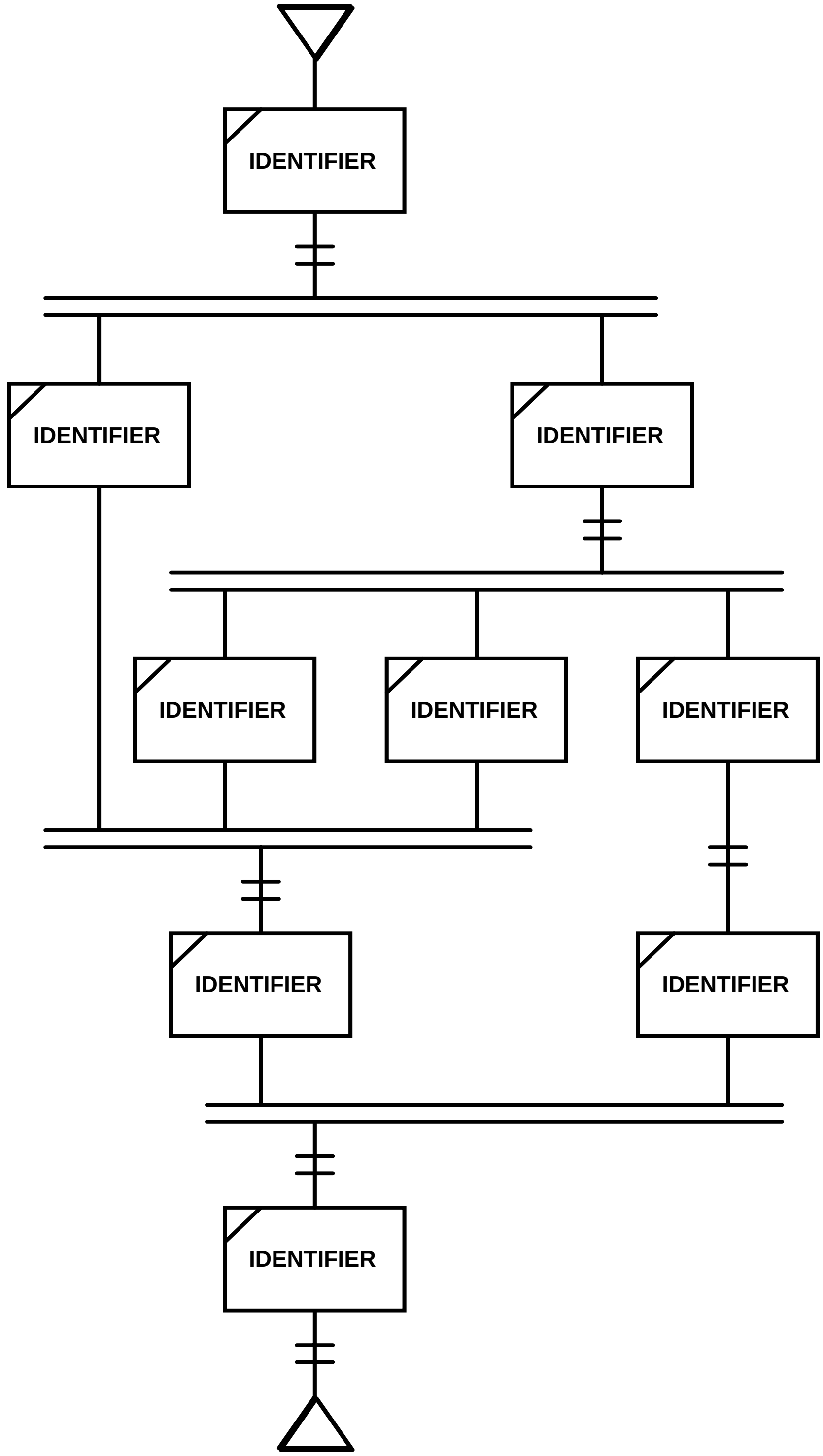


Figure 34 — Valid simultaneous sequence diagram

Looping provides for the re-execution of entities based upon transition conditions (see figure 35). This allows dynamic execution of entities based upon differing conditions.

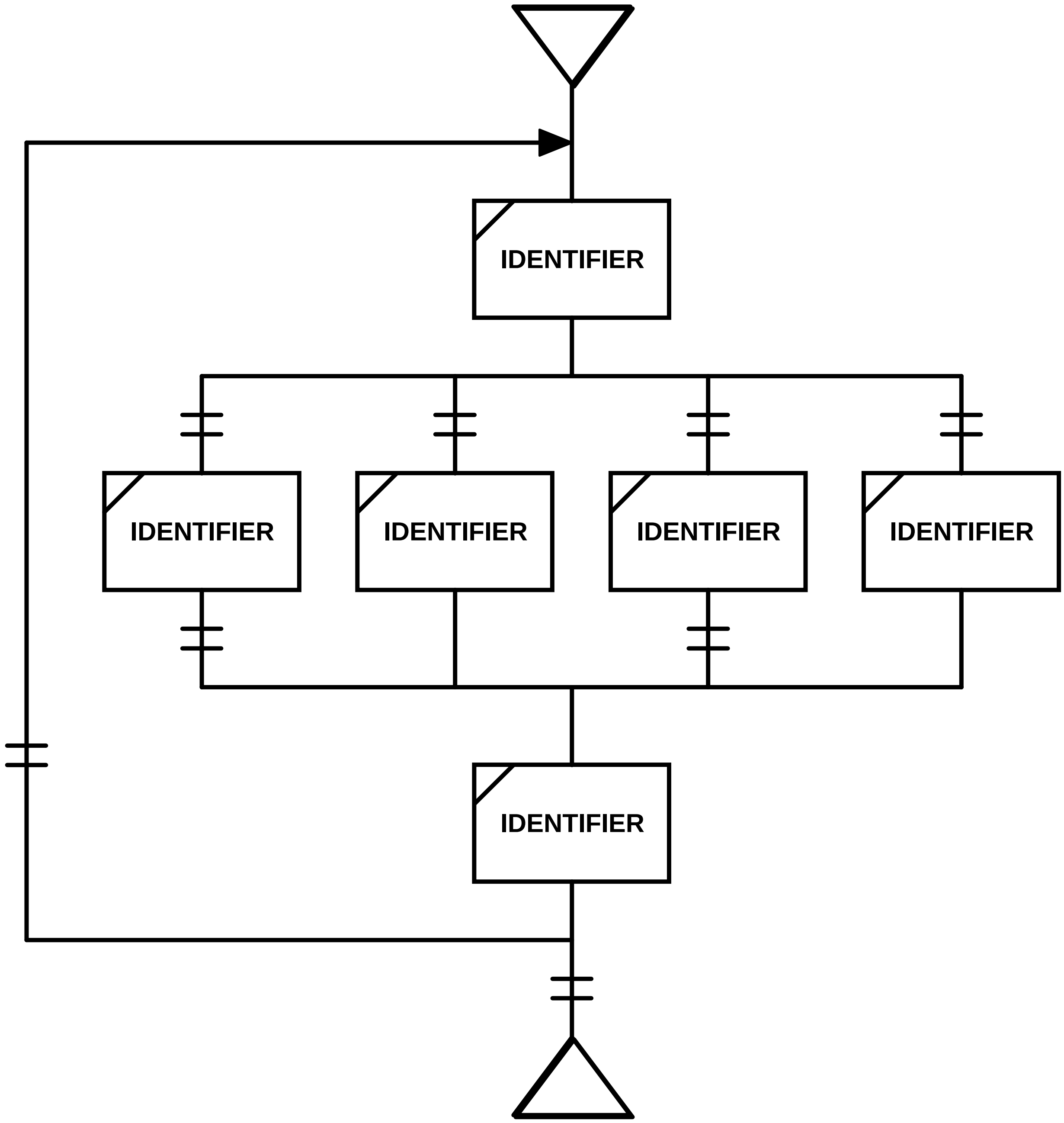


Figure 35 — Looping with explicit recipe procedural elements

This standard cannot define all valid and invalid procedure function charts. PFCs can be constructed that have unreachable procedural entities or that have an invalid execution path (e.g., in figure 36, the thread through "Phase 1" may never complete if the thread through "Phase 5" is executed).

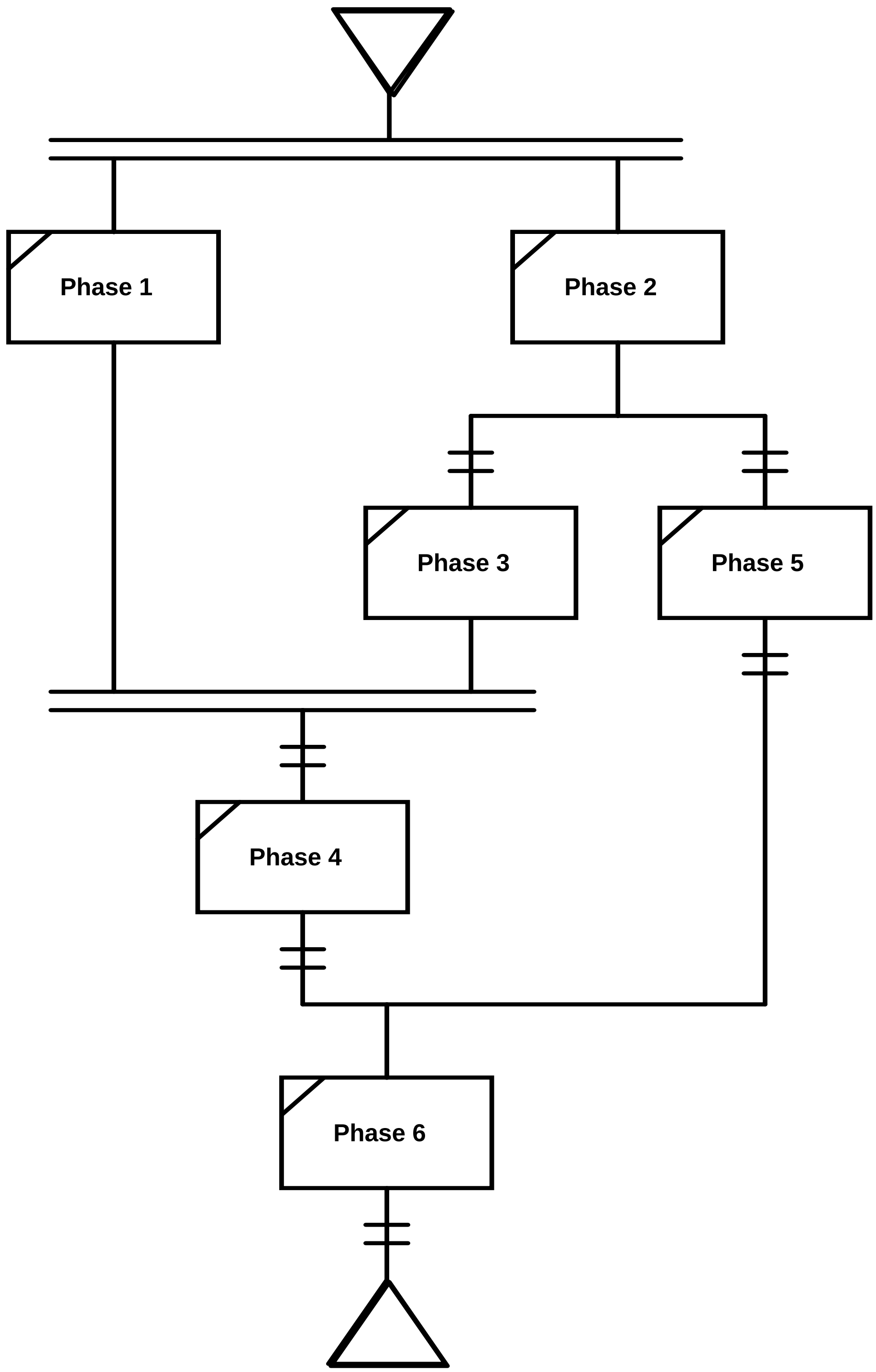


Figure 36 — Invalid procedure function chart

6.1.2 Procedure and unit procedure initiation

A depiction of the beginning of a recipe procedure is shown in figure 37. At every level below the unit procedure, directed links clearly indicate the order in which the recipe procedural elements become active or are initiated. The initiation of a recipe procedure is most likely to be related to scheduling requirements; therefore, it has a need for starting rules, some of which can be based on the schedule. A unit procedure becomes active after the transition following the allocation symbol is true.

6.1.2.1 Procedure, unit procedure, and operation completion

When the end symbol in a procedure function chart is reached, the encapsulating procedural element is complete.

6.1.2.2 Relative relationship between procedural entities

Figures 38 and 39 show two methods for depicting the relative relationship between procedural elements in procedure function charts. This relationship can be accomplished by organizing the procedural elements vertically in relation to each other. The vertical size of the procedural elements can also be varied in order to show relative relationships between depicted elements. The horizontal dashed lines show the synchronization that occurs between the operations within each unit procedure.

A compliant system shall implement at least one of the methodologies that are depicted in figures 38 and 39.

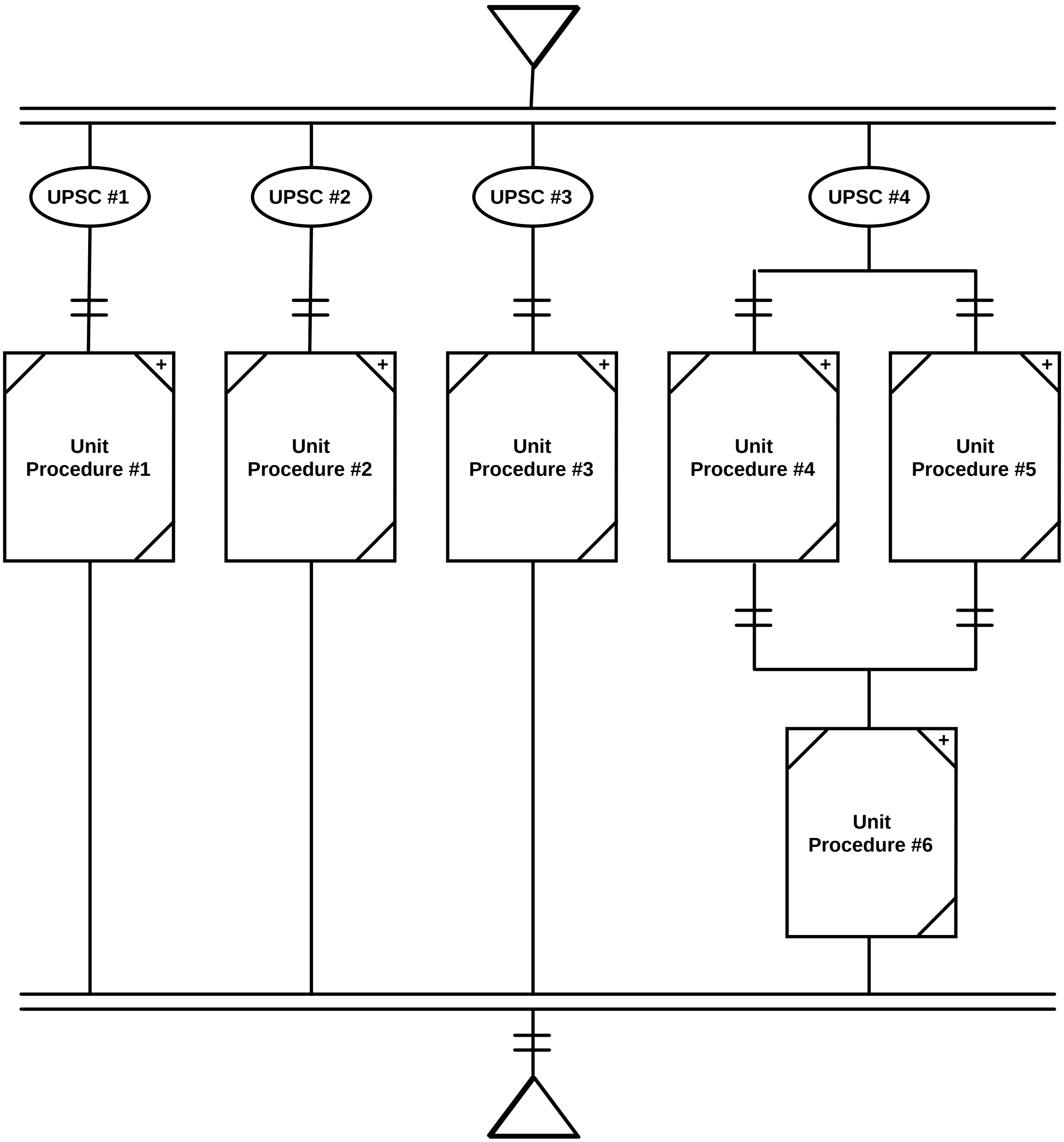


Figure 37 — Depiction of procedure and unit procedure initiation

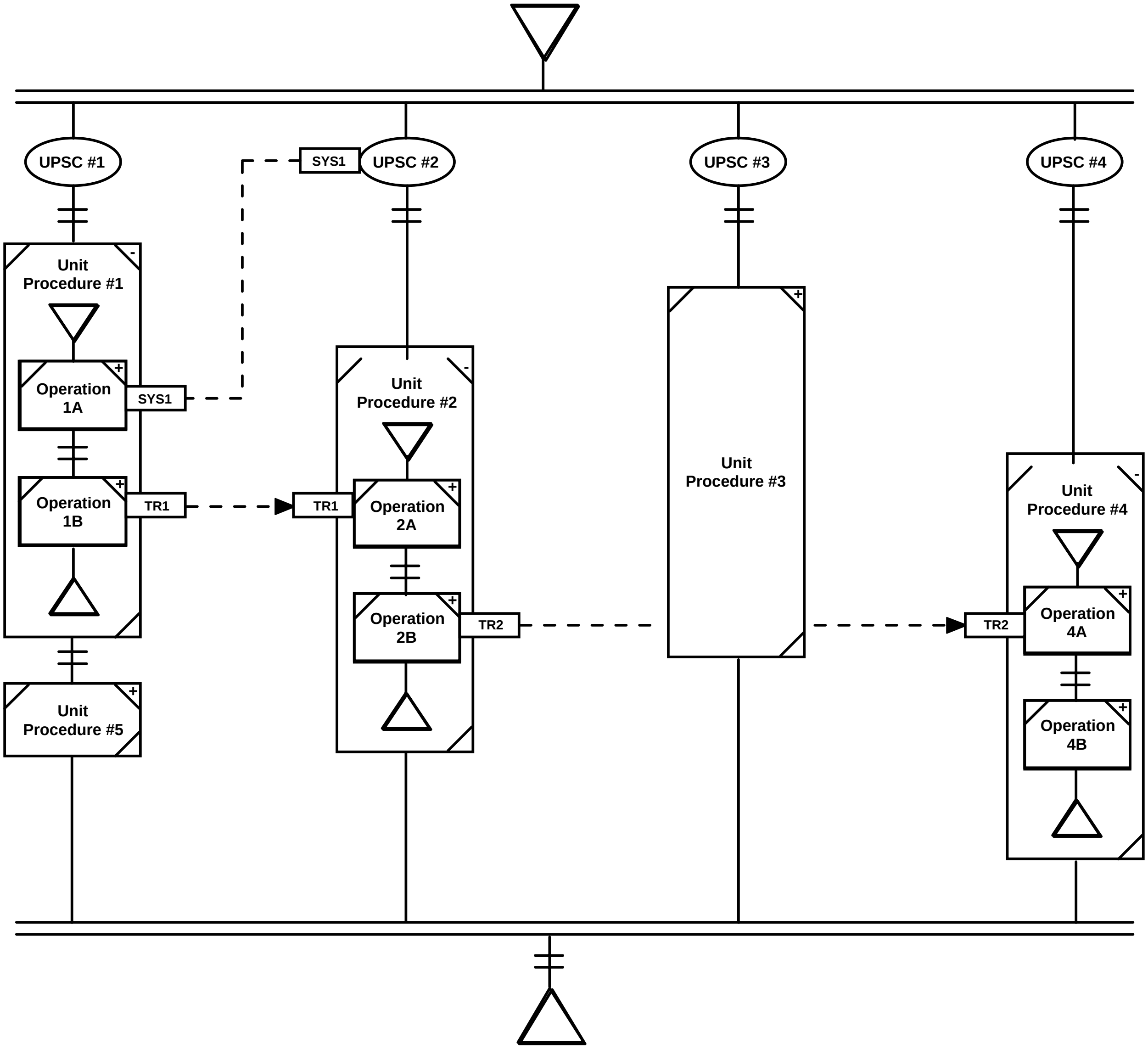


Figure 38 — Relative relationship of procedural entities

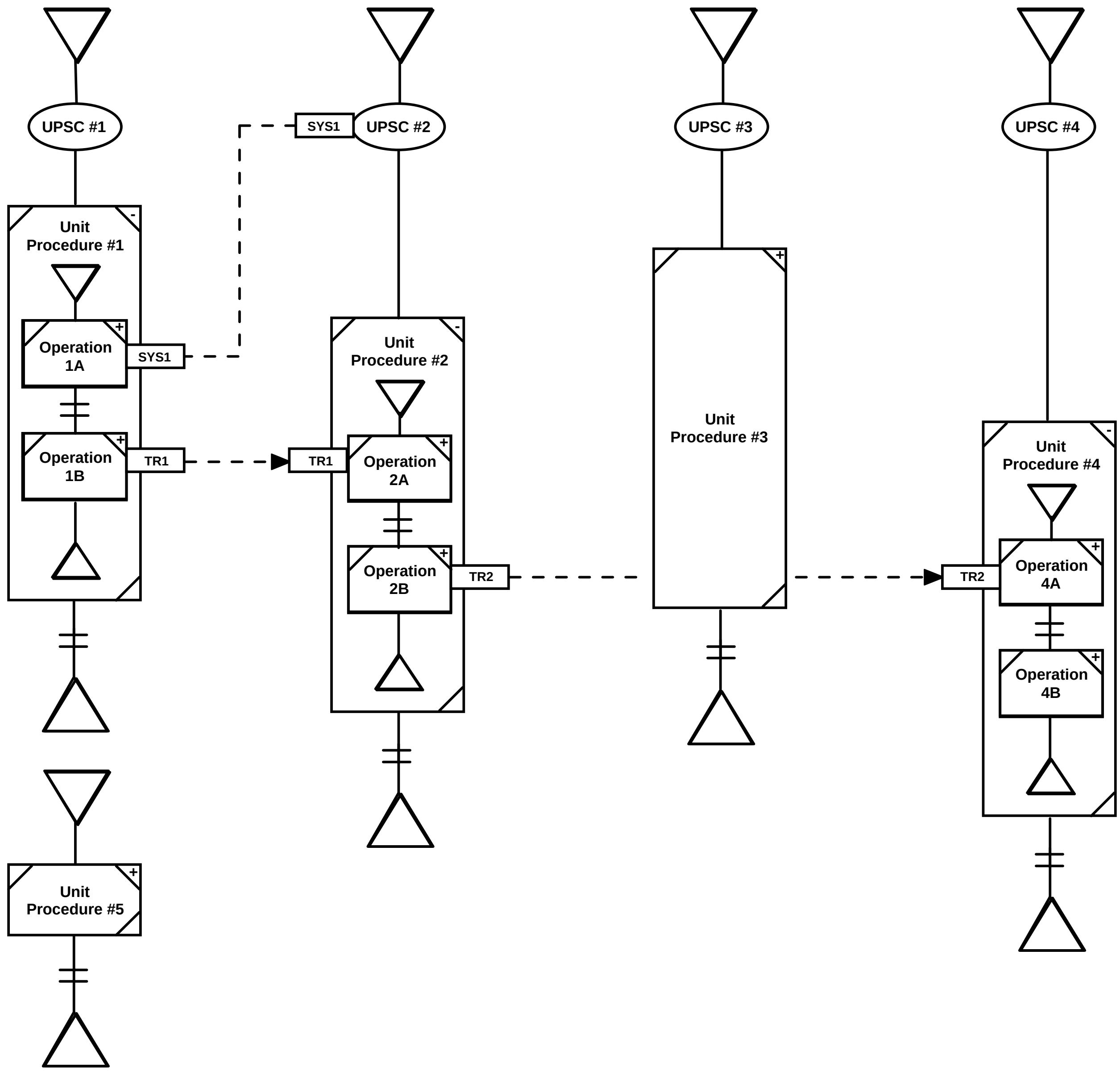


Figure 39 — Relative relationship of procedural entities – Alternate 1

When procedural elements at two different levels (e.g., unit procedures and operations) are shown on a drawing of this type, a box that shows the same grouping as the higher-level procedural element (i.e., unit procedures) shall enclose the lower-level procedural elements (i.e., operations). Unit procedure #1 in figure 38 illustrates a procedural element that is shown with a lower-level PFC that it encapsulates.

6.1.3 Non-procedural master recipe information

All other information that is part of a master recipe is related to a specific element or symbol in the master recipe procedure. This standard purposely does not specify how that relationship or reference is implemented. In a pencil and paper implementation, for example, the reference might be accomplished with something similar to a footnote or the information might be written alongside the procedural element in question. In an electronic implementation, pop-up boxes or some other mechanism that is not yet invented might be the implementation of choice. However, the relationship shall be clearly indicated and it shall be consistent within each application.

6.1.3.1 Master recipe formula

Formula information consists of process inputs, process parameters and process outputs. The formula information shall be able to be represented in its entirety (e.g., associated with a recipe procedure), in parts (e.g., process inputs only or for a specific unit procedure) or as a summary of lower-level formula, as appropriate for the context and intended use. When depicted, the formula shall be associated with a recipe procedural element.

For example, the amount of product that a recipe produces may be associated with the recipe procedure, while the amount of material to add to a reactor may be associated with a specific phase. The use of formula permits a summation of all recipe procedural elements parameters that have been identified as process inputs, so that a list of the process inputs for an entire recipe, unit procedure, or operation can be provided.

6.1.3.2 Master recipe equipment requirements

Equipment requirements are specific to the execution of recipe procedural elements. The representation shall provide a method for the user to view the equipment requirements that are associated with each procedural element individually or for all elements in aggregate.

6.1.3.3 Header and other information

Header information and the “other information” category of recipe information may be related to the recipe in general (e.g., recipe ID, regulatory status) or to specific recipe procedural entities (e.g., protective equipment requirements, hazards of chemicals information). All header and ‘other information’ shall be able to be represented in its entirety or associated with the procedural entity to which the information is related.

6.2 Control recipe depiction

When procedure function charts are used for the depiction of a control recipe, they shall follow the same principles that are defined for the use of procedure function charts for the depiction of a master recipe. In this case, however, the depiction of the control recipe will also, usually, become an active display in automated systems. Because the relationship between recipe procedural elements and equipment procedural elements is known during the execution of the control recipe, colors and/or other means of demarcation may be used to indicate the status of the procedural elements.

6.3 Exception handling

Special processing may be included in the recipe in order to handle product-specific exceptions, in addition to any equipment logic that is used for handling equipment-related exceptions. These product-specific exceptions are generally related to the method of manufacturing a product, often its quality, and they are not specific to equipment. They are interwoven into the normal procedure; therefore, they are not easily distinguished from normal processing and they only become active in the event of an exception.

The extent of the effect of exception handling in the recipe procedure is normally confined to unit procedures because the units operate independently. In some cases, however, commands, states, and/or modes are propagated to another unit when there is a common concern (e.g., a transfer or concurrent processing with common time constraints). In some circumstances, an entire recipe may have to be commanded to a specified state and/or mode.

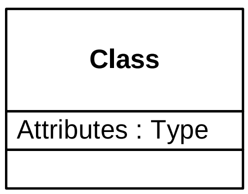
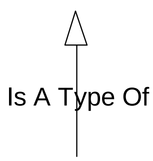
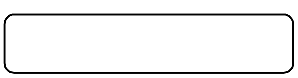
This page intentionally left blank.

Annex A (normative) — Data modeling technique

A.1 UML notation

Table A.1 defines the UML notation that is used in this standard (see J. Rumbaugh/I. Jacobson, The Unified Modeling Language Reference Manual, 1999, Addison Wesley Longman, Inc.).

Table A.1 — UML notation

SYMBOL	DEFINITION
	Defines a class of objects, each with the same types of attributes. Each object is uniquely identifiable or enumerable. No operations or methods are listed for the classes. Attributes with a '-' before their name indicate attributes that are generally optional in any use of the class.
	An association between elements of a class and elements of another or the same class. Each association is identified. Can have the expected number or range of members of the subclass, where 'n' indicates an indeterminate number (e.g., 0,n means that zero or more members of the subclass may exist).
	Generalization (arrow points to the super class) shows that an element of the class is a specialized type of the super class.
	Dependence (i.e., tightly bound relationship between the items) shows that an element of the class depends on an element of another class.
	Aggregation (i.e., made up of) shows that an element of the class is made up of elements of other classes.
	A class of object that is an instance of another class of object.

A.2 Definitions

A.2.1 class:
a description of a set of objects that share the same attributes, behaviors, relationships, and semantics.

A.2.2 encapsulation:
a technique that separates the external aspects of an object from the internal, implementation details of the object (also called information hiding).

A.2.3 instance:
a term that is used to refer to an object that belongs to a particular class but that is not itself a class or a subclass. For example, "reactor401" is an instance of the class "reactor".

A.2.4 model:
a formal abstract representation of a system. A model is usually presented as a collection of diagrams and a data dictionary.

A.2.5 object:
an entity that is composed of state and behavior. State is the value of all attributes at a given time. An attribute is a piece of information that qualifies the object. The behavior of an object is the functionality that is contained in the object that is necessary to manipulate the attributes.

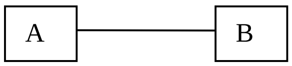
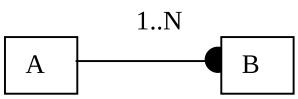
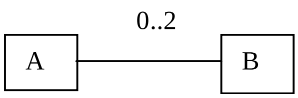
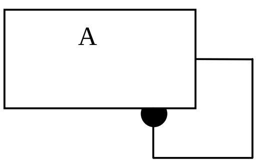
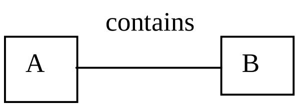
A.2.6 subclass:
a class that is a special case of a more general class (e.g., glass-lined reactor is a subclass of reactor class).

A.2.7 unified modeling language (UML):
a language that is used for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.

A.3 ERD notation

Table A.2 defines the ERD notation that is used in this standard.

Table A.2 — ERD notation

SYMBOL	DEFINITION
	Defines an entity.
	For each occurrence of A, there is one and only one occurrence of B. The association at B may also be marked with the number 1.
	Numerically specified association. In this example, for each occurrence of A, there may be one or more occurrences of B. Another example is 0..N. If no numeric association is given, 0..N is assumed.
	Numerically specified association: 0 to some positive number. In this example, for each occurrence of A, there may be 0 to 2 occurrences of B.
	Looped association. An occurrence of A may be made up of zero or more occurrences of entities of the same type. An optional association is that an occurrence of entity A may contain zero or more occurrences of entities of the same type. Another usage is that an occurrence of entity A may contain zero or more occurrences of entities of the same type.
	The association between entities is labeled in order to specify the nature of the relationship. The label applies to the entity that it is closest to. In this example, the association is read as follows: Each occurrence of A *contains* one and only one occurrence of B.

Annex B (normative) — SQL definition listing

This annex contains the ISO/IEC9075:1992 standard definition of all of the tables that are defined in clause 5.

```
CREATE TABLE BXT_Exchange (
    ExchangeID          CHAR (32)       NOT NULL,
    ExchangeValue       CHAR (128)      NOT NULL,
    PRIMARY KEY (ExchangeID))

INSERT INTO BXT_Exchange (ExchangeID, ExchangeValue)
    VALUES ('Schema','ANSI/ISA-88.00.02-2001')
    VALUES ('Delimiter', '/')
    VALUES ('ToolID','ToolName')
    VALUES ('ToolVersion','4.0')
    VALUES ('ToolSchema','1.2')

CREATE TABLE BXT_EnumerationSet (
    EnumSet             CHAR (32)       NOT NULL,
    Description         CHAR (255),
    PRIMARY KEY (EnumSet))

CREATE TABLE BXT_Enumeration (
    EnumSet             CHAR (32)       NOT NULL,
    EnumValue           INTEGER         NOT NULL,
    EnumString          CHAR (32),
    Description         CHAR (255),
    PRIMARY KEY (EnumSet, EnumValue))
```

```
INSERT INTO BXT_EnumerationSet (EnumSet, Description)
    VALUES ('Boolean',
        'Defines a set of Boolean values')
    VALUES ('DirectionType',
        'Defines how a parameter is intended to be handled')
    VALUES ('EquipmentLevel',
        'Defines the equipment hierarchical level for equipment elements')
    VALUES ('EquipmentType',
        'Defines the type of equipment record for equipment elements')
    VALUES ('EvaluationRule',
        'Defines the evaluation rules for equipment properties')
    VALUES ('FormulaSubType',
        'Defines the recipe formula types')
    VALUES ('FormulaType',
        'User supplied formula sub type definitions')
    VALUES ('LinkDepiction',
        'Defines how links between recipe elements are to be depicted')
    VALUES ('LinkToType',
        'Defines if a link references  a step or a transition')
    VALUES ('LinkType',
        'Defines the type of link')
    VALUES ('RE_Type',
        'Defines the recipe element, either recipe procedure level or allocation entity')
    VALUES ('RE_Use',
        'Defines how a recipe element is used in a recipe')
    VALUES ('RecipeStatus',
        'Defines the possible status of a recipe')
    VALUES ('RecordSet',
        'Defines the enumeration set used to classify a record into a category of batch history
        information.')
    VALUES ('RecordSetControlRecipe',
        'Provides further history record classification under the category of ControlRecipe.')
    VALUES ('RecordSetMasterRecipe',
        'Provides further history record classification under the category of MasterRecipe.')
    VALUES ('RecordSetExecutionInfo',
        'Provides further history record classification under the category of ExecutionInfo.')
    VALUES ('RecordSetMaterialInfo',
        'Provides further history record classification under the category of MaterialInfo.')
    VALUES ('RecordSetContinuousData',
        'Provides further history record classification under the category of ContinuousData.')
    VALUES ('RecordSetEvents',
        'Provides further history record classification under the category of Events.')
    VALUES ('RecordSetOperatorChange',
        'Provides further history record classification under the category of OperatorChange.')
    VALUES ('RecordSetOperatorComment',
        'Provides further history record classification under the category of OperatorComment.')
    VALUES ('RecordSetAnalysisData',
        'Provides further history record classification under the category of AnalysisData.')
    VALUES ('RecordSetLateRecord',
        'Provides further history record classification under the category of LateRecord.')
    VALUES ('RecordSetRecipeData',
        'Provides further history record classification under the category of RecipeData.')
    VALUES ('RecordSetRecipeSpecified',
        'Provides further history record classification under the category of RecipeSpecified.')
```

```
    VALUES (’RecordSetSummaryData’,
        ’Provides further history record classification under the category of SummaryData.’)
    VALUES (’ScheduleAction’,
        ’Defines the intended action of the schedule entry ’)
    VALUES (’ScheduleMode’,
        ’Defines the mode which the schedule entry begins execution in ’)
    VALUES (’ScheduleStatus’,
        ’Defines the possible status of a schedule’)
    VALUES (’SE_Type’,
        ’Defines the type of entity in a schedule record’)
    VALUES (’ValueDataType’,
        ’Defines how a value is represented (e.g. Boolean, float, etc.) ’)
    VALUES (’ValueType’,
        ’Defines how a value string is interpreted’)

INSERT INTO BXT_Enumeration (EnumSet, EnumValue, EnumString, Description)
    VALUES (’Boolean’, 0, ’FALSE’,
        ’Defines a Boolean value’)
    VALUES (’Boolean’, 1, ’TRUE’, ’’)
    VALUES (’DirectionType’, 0, ’Invalid’, ’Entry not valid’)
    VALUES (’DirectionType’, 1, ’Internal’,
        ’Identifies how a parameter is handled.’)
    VALUES (’DirectionType’, 2, ’Input’,
        ’The Recipe Element receives the value from an external source.’)
    VALUES (’DirectionType’, 3, ’Output’,
        ’The Recipe Element creates the value and makes it available for external
        use.’)
    VALUES (’DirectionType’, 4, ’Input/Output’,
        ’The Recipe Element and external element exchange the value, and may
        change its value.’)
    VALUES (’EquipmentLevel’, 0, ’Invalid’, ’Entry not valid’)
    VALUES (’EquipmentLevel’, 1, ’Enterprise’,
        ’Identifies the equipment hierarchical level for BXT_EquipElement’)
    VALUES (’EquipmentLevel’, 2, ’Site’, ’’)
    VALUES (’EquipmentLevel’, 3, ’Area’, ’’)
    VALUES (’EquipmentLevel’, 4, ’Process Cell’, ’’)
    VALUES (’EquipmentLevel’, 5, ’Unit’, ’’)
    VALUES (’EquipmentLevel’, 6, ’Equipment Module’, ’’)
    VALUES (’EquipmentLevel’, 7, ’Control Module’, ’’)
    VALUES (’EquipmentType’, 0, ’ Invalid’, ’Entry not valid’)
    VALUES (’EquipmentType’, 1, ’Class’,
        ’Identifies the record type for BXT_EquipElement’)
    VALUES (’EquipmentType’, 2, ’Element’, ’’)
    VALUES (’EvaluationRule’, 0, ’Invalid’, ’Entry not valid’)
    VALUES (’EvaluationRule’, 1, ’=’,
        ’Equals comparison operator for equipment properties’)
    VALUES (’EvaluationRule’, 2, ’<>’,
        ’Not equals comparison operator for equipment properties’)
    VALUES (’EvaluationRule’, 3, ’<’,
        ’Less than comparison operator for equipment properties’)
    VALUES (’EvaluationRule’, 4, ’>’,
        ’Greater than comparison operator for equipment properties’)
    VALUES (’EvaluationRule’, 5, ’<=’,
        ’Less than or equals comparison operator for equipment properties’)
```

```
VALUES (’EvaluationRule’, 6, ’>=’,
    ’Greater than or equals comparison operator for equipment properties’)
VALUES (’EvaluationRule’, 7, ’Member’,
    ’Is a member of comparison operator for equipment properties’)
VALUES (’EvaluationRule’, 8, ’Not member’,
    ’Is not a member of comparison operator for equipment properties’)
VALUES (’EvaluationRule’, 9, ’Not’,
    ’Not comparison operator for equipment properties’)
VALUES (’FormulaType’, 0, ’Invalid’, ’Entry not valid’)
VALUES (’FormulaType’, 1, ’Process Input’, ’Recipe Formula type’)
VALUES (’FormulaType’, 2, ’Process Output’, ’’)
VALUES (’FormulaType’, 3, ’Process Parameter’, ’’)
VALUES (’FormulaSubType’, 0, ’Invalid’, ’Entry not valid’)
VALUES (’LinkDepiction’, 0, ’Invalid’, ’Entry not valid’)
VALUES (’LinkDepiction’, 1, ’None’, ’No link depiction’)
VALUES (’LinkDepiction’, 2, ’Line’, ’Link shown with line only ’)
VALUES (’LinkDepiction’, 3, ’ID’, ’Link shown with identifier only’)
VALUES (’LinkDepiction’, 4, ’Line & ID’,
    ’Link shown with line and identification ’)
VALUES (’LinkDepiction’, 5, ’Line & Arrow’,
    ’Link shown with line and material flow arrow ’)
VALUES (’LinkDepiction’, 6, ’Line, Arrow, & ID’,
    ’Link shown with line, material flow arrow and identification ’)
VALUES (’LinkToType’, 0, ’Invalid’, ’Entry not valid’)
VALUES (’LinkToType’, 1, ’Recipe Element’,
    ’Link is referencing an entry in the BXT_MRecipeElement table’)
VALUES (’LinkToType’, 2, ’Transition’,
    ’Link is referencing an entry in the BXT_MRecipeTransition table’)
VALUES (’LinkType’, 0, ’Invalid’, ’Entry not valid’)
VALUES (’LinkType’, 1, ’ControlLink’,
    ’Defines a link between recipe elements that indicates a flow of procedural control. ’)
VALUES (’LinkType’, 2, ’TransferLink’,
    ’Defines a link between recipe elements that indicates a material transfer.’)
VALUES (’LinkType’, 3, ’SynchronizationLink’,
    ’Defines a link between recipe elements where there is some form of synchronization.’)
VALUES (’RE_Type’, 0, ’Invalid’, ’Entry not valid’)
VALUES (’RE_Type’, 1, ’Master Recipe’,
    ’Specifies the type of recipe element. ’)
VALUES (’RE_Type’, 2, ’Procedure’, ’’)
VALUES (’RE_Type’, 3, ’Unit Procedure’, ’’)
VALUES (’RE_Type’, 4, ’Operation’, ’’)
VALUES (’RE_Type’, 5, ’Phase’, ’’)
VALUES (’RE_Type’, 6, ’Allocation’, ’’)
VALUES (’RE_Type’, 7, ’Begin’, ’’)
VALUES (’RE_Type’, 8, ’End’, ’’)
VALUES (’RE_Type’, 9, ’Start Parallel’, ’’)
VALUES (’RE_Type’, 10, ’End Parallel’, ’’)
VALUES (’RE_Type’, 11, ’Start Branch’, ’’)
VALUES (’RE_Type’, 12, ’End Branch’, ’’)
VALUES (’RE_Use’, 0, ’Invalid’, ’Entry not valid’)
VALUES (’RE_Use’, 1, ’Linked’,
    ’A recipe element (RE) may have several referencing RE Steps’)
VALUES (’RE_Use’, 2, ’Embedded’,
    ’An RE has only one referencing RE, one RE is defined for each use of the RE. ’)
```

```
VALUES ('RE_Use', 3, 'Copied',
    'The same as Embedded, but the specific RE was modified from its original definition. ')
VALUES ('RecipeStatus', 0, 'Invalid', 'Entry not valid')
VALUES ('RecipeStatus', 1, 'Approved for Production',
    'Recipe was approved for production. ')
VALUES ('RecipeStatus', 2, 'Approved for Test',
    'Recipe was only approved for test. ')
VALUES ('RecipeStatus', 3, 'Not Approved',
    'Recipe was not approved for production or test. ')
VALUES ('RecipeStatus', 4, 'Inactive', 'Recipe was not active. ')
VALUES ('RecipeStatus', 5, 'Obsolete', 'Recipe was obsolete.')
VALUES ('RecordSet', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSet', 1, 'RecordSetControlRecipe',
    'Defines that a batch history information record is part of the ControlRecipe category.')
VALUES ('RecordSet', 2, 'RecordSetMasterRecipe', '')
VALUES ('RecordSet', 3, 'RecordSetExecutionInfo', '')
VALUES ('RecordSet', 4, 'RecordSetMaterialInfo', '')
VALUES ('RecordSet', 5, 'RecordSetContinuousData', '')
VALUES ('RecordSet', 6, 'RecordSetEvents', '')
VALUES ('RecordSet', 7, 'RecordSetOperatorChange', '')
VALUES ('RecordSet', 8, 'RecordSetOperatorComment', '')
VALUES ('RecordSet', 9, 'RecordSetAnalysisData', '')
VALUES ('RecordSet', 10, 'RecordSetLateRecord', '')
VALUES ('RecordSet', 11, 'RecordSetRecipeData', '')
VALUES ('RecordSet', 12, 'RecordSetRecipeSpecified', '')
VALUES ('RecordSet', 13, 'RecordSetSummaryData', '')
VALUES ('RecordSetControlRecipe', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetControlRecipe', 1, 'Entire Control Recipe',
    'History record is related to the entire control recipe.')
VALUES ('RecordSetMasterRecipe', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetMasterRecipe', 1, 'Entire Master Recipe',
    'History record is related to the entire master recipe.')
VALUES ('RecordSetExecutionInfo', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetExecutionInfo', 1, 'Allocation', '')
VALUES ('RecordSetExecutionInfo', 2, 'De-allocation', '')
VALUES ('RecordSetExecutionInfo', 3, 'State Change', '')
VALUES ('RecordSetExecutionInfo', 4, 'State Command', '')
VALUES ('RecordSetExecutionInfo', 5, 'Mode Change', '')
VALUES ('RecordSetExecutionInfo', 6, 'Mode Command', '')
VALUES ('RecordSetExecutionInfo', 7, 'Procedural Entity Message', '')
VALUES ('RecordSetExecutionInfo', 8, 'Procedural Entity Alarm', '')
VALUES ('RecordSetExecutionInfo', 9, 'Procedural Entity Version', '')
VALUES ('RecordSetExecutionInfo', 10, 'Procedural Entity Prompt', '')
VALUES ('RecordSetExecutionInfo', 11, 'Procedural Entity Prompt Resp', '')
VALUES ('RecordSetMaterialInfo', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetMaterialInfo', 1, 'Material Consumption', '')
VALUES ('RecordSetMaterialInfo', 2, 'Material Production', '')
VALUES ('RecordSetMaterialInfo', 3, 'Material Allocation', '')
VALUES ('RecordSetMaterialInfo', 4, 'Material De-allocation', '')
VALUES ('RecordSetContinuousData', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetContinuousData', 1, 'Continuous Data Value', '')
VALUES ('RecordSetContinuousData', 2, 'Trend Association', '')
VALUES ('RecordSetContinuousData', 3, 'Trend Disassociation', '')
VALUES ('RecordSetEvents', 0, 'Invalid', 'Entry not valid')
```

```
VALUES ('RecordSetEvents', 1, 'General Event', '')
VALUES ('RecordSetOperatorChange', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetOperatorChange', 1, 'General Operator Intervention', '')
VALUES ('RecordSetOperatorComment', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetOperatorComment', 1, 'General Operator Comment', '')
VALUES ('RecordSetAnalysisData', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetAnalysisData', 1, 'General Analysis Message', '')
VALUES ('RecordSetLateRecord', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetLateRecord', 1, 'General Late Record', '')
VALUES ('RecordSetRecipeData', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetRecipeData', 1, 'Generic Recipe Data', '')
VALUES ('RecordSetRecipeData', 2, 'Recipe Parameter Value Change', '')
VALUES ('RecordSetRecipeData', 3, 'Recipe Result Data', '')
VALUES ('RecordSetRecipeSpecified', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetRecipeSpecified', 1, 'Generic Recipe Specified Data', '')
VALUES ('RecordSetSummaryData', 0, 'Invalid', 'Entry not valid')
VALUES ('RecordSetSummaryData', 1, 'Generic Summary Data', '')
VALUES ('RecordSetSummaryData', 2, 'Utilities Consumption', '')
VALUES ('RecordSetSummaryData', 3, 'Equipment Run Time', '')
VALUES ('ScheduleChange', 0, 'Invalid', 'Entry not valid')
VALUES ('ScheduleChange', 1, 'New', 'Schedule record change action')
VALUES ('ScheduleChange', 2, 'Update', '')
VALUES ('ScheduleChange', 3, 'Delete', '')
VALUES ('ScheduleMode', 0, 'Invalid', 'Entry not valid')
VALUES ('ScheduleMode', 1, 'Automatic', 'Schedule record mode')
VALUES ('ScheduleMode', 2, 'Semi-Automatic', '')
VALUES ('ScheduleMode', 3, 'Manual', '')
VALUES ('ScheduleMode', 4, 'Not Specified', '')
VALUES ('ScheduleStatus', 0, 'Invalid', 'Entry not valid')
VALUES ('ScheduleStatus', 1, 'Complete', 'Batch schedule record status')
VALUES ('ScheduleStatus', 2, 'In-progress', '')
VALUES ('ScheduleStatus', 3, 'Scheduled', '')
VALUES ('ScheduleStatus', 4, 'Schedule Hold', '')
VALUES ('ScheduleStatus', 5, 'Not Specified', '')
VALUES ('SE_Type', 0, 'Invalid', 'Entry not valid')
VALUES ('SE_Type ', 1, 'Campaign',
    'Defines the type of Scheduled Entry')
VALUES ('SE_Type ', 2, 'Batch', '')
VALUES ('SE_Type ', 3, 'Unit Procedure', '')
VALUES ('SE_Type ', 4, 'Operation', '')
VALUES ('SE_Type ', 5, 'Phase', '')
VALUES ('ValueDataType', 0, 'Invalid', 'Entry not valid')
VALUES ('ValueDataType', 1, 'Boolean',
    'Defines the data type that is expected for an associated value. ')
VALUES ('ValueDataType', 2, '8-Bit string', '')
VALUES ('ValueDataType', 3, '16-Bit string', '')
VALUES ('ValueDataType', 4, '32-Bit string', '')
VALUES ('ValueDataType', 5, '8-Bit unsigned integer', '')
VALUES ('ValueDataType', 6, '16-Bit unsigned integer', '')
VALUES ('ValueDataType', 7, '32-Bit unsigned integer', '')
VALUES ('ValueDataType', 8, '8-Bit signed integer', '')
VALUES ('ValueDataType', 9, '16-Bit signed integer', '')
VALUES ('ValueDataType', 10, '32-Bit signed integer', '')
VALUES ('ValueDataType', 11, '32-Bit float', '')
```

```
    VALUES ('ValueDataType', 12, 'Double float', '')
    VALUES ('ValueDataType', 13, 'Octet string', '')
    VALUES ('ValueDataType', 14, 'DateTime', '')
    VALUES ('ValueType', 0, 'Invalid', 'Entry not valid')
    VALUES ('ValueType', 1, 'Constant',
        'Defines how a value string is interpreted.  It contains a fixed value as a string. ')
    VALUES ('ValueType', 2, 'Reference',
        'Defines how a value string is interpreted.  It points to the source of the value. ')
    VALUES ('ValueType', 3, 'Equation',
        'Defines that a value string is interpreted as an expression to be evaluated in order to determine
        the value. ')
    VALUES ('ValueType', 4, 'External',
        'Value is supplied by some external means, and it is not contained in the recipe (i.e., value may
        be supplied by an operator entry or by a scheduling system).')

CREATE TABLE BXT_MRecipeElement (
    RE_ID               CHAR (128)      NOT NULL,
    REVersion           CHAR (16)       NOT NULL,
    VersionDate         DATETIME,
    ApprovalDate        DATETIME,
    EffectiveDate       DATETIME,
    ExpirationDate      DATETIME,
    Author              CHAR (32),
    ApprovedBy          CHAR (32),
    ProcessCellID       CHAR (32),
    ProductID           CHAR (32),
    UsageConstraint     CHAR (255),
    Description         CHAR (255),
    Status              INTEGER,
    RE_Type             INTEGER,
    RE_Function         CHAR (255),
    RE_Use              INTEGER,
    DerivedRE           CHAR (128),
    DerivedVersion      CHAR (16),
    PRIMARY KEY (RE_ID, REVersion))

CREATE TABLE BXT_MRecipeStep (
    ParentRE            CHAR (128)      NOT NULL,
    ParentVersion       CHAR (16)       NOT NULL,
    StepID              CHAR (128)      NOT NULL,
    RE_ID               CHAR (128)      NOT NULL,
    REVersion           CHAR (16)       NOT NULL,
    VerticalStart       FLOAT,
    VerticalStop        FLOAT,
    HorizontalStart     FLOAT,
    HorizontalStop      FLOAT,
    ScaleReference      FLOAT,
    ScaleEngrUnits      CHAR (32),
    MaximumScale        FLOAT,
    MinimumScale        FLOAT,
    PRIMARY KEY (ParentRE, ParentVersion, StepID),
    FOREIGN KEY (RE_ID, REVersion)
        REFERENCES BXT_MRecipeElement (RE_ID, REVersion))
```

```
CREATE TABLE BXT_MRecipeTransition (
    RE_ID                CHAR (128)   NOT NULL,
    REVersion            CHAR (16)    NOT NULL,
    TransitionID         CHAR (128)   NOT NULL,
    Condition            CHAR (255),
    VerticalStart        FLOAT,
    VerticalStop         FLOAT,
    HorizontalStart      FLOAT,
    HorizontalStop       FLOAT,
    PRIMARY KEY (RE_ID, REVersion, TransitionID),
    FOREIGN KEY (RE_ID, REVersion)
        REFERENCES BXT_MRecipeElement (RE_ID, REVersion))

CREATE TABLE BXT_MRecipeLink (
    RE_ID                CHAR (128)   NOT NULL,
    REVersion            CHAR (16)    NOT NULL,
    LinkID               CHAR (32)    NOT NULL,
    FromType             INTEGER,
    FromElement          CHAR (128),
    ToType               INTEGER,
    ToElement            CHAR (128),
    LinkType             INTEGER,
    VerticalStart        FLOAT,
    VerticalStop         FLOAT,
    HorizontalStart      FLOAT,
    HorizontalStop       FLOAT,
    Depiction            INTEGER,
    EvaluationOrder      INTEGER,
    PRIMARY KEY (RE_ID, REVersion, LinkID),
    FOREIGN KEY (RE_ID, REVersion)
        REFERENCES BXT_MRecipeElement (RE_ID, REVersion) )

CREATE TABLE BXT_MRecipeElementParameter (
    RE_ID                CHAR (128)   NOT NULL,
    REVersion            CHAR (16)    NOT NULL,
    ParameterID          CHAR(32)     NOT NULL,
    ParentParamID        CHAR(32),
    DataInterpretation   INTEGER,
    DataDirection        INTEGER,
    DefaultValue         CHAR(128),
    Description          CHAR(255),
    EngrUnits            CHAR(32),
    EnumSet              CHAR (32),
    DefaultScaling       INTEGER,
    ParamType            INTEGER,
    ParamSubType         INTEGER,
    PRIMARY KEY (RE_ID, REVersion, ParameterID),
    FOREIGN KEY (RE_ID, REVersion)
        REFERENCES BXT_MRecipeElement (RE_ID, REVersion))
```

```
CREATE TABLE BXT_MRecipeStepParameter (
    ParentRE            CHAR (128)      NOT NULL,
    ParentVersion       CHAR (16)       NOT NULL,
    StepID              CHAR (128)      NOT NULL,
    ParameterID         CHAR (32)       NOT NULL,
    ParentParamID       CHAR (32),
    ParameterValue      CHAR (128),
    DataInterpretation  INTEGER,
    Scaled              INTEGER,
    PRIMARY KEY (ParentRE, ParentVersion, StepID, ParameterID),
    FOREIGN KEY (ParentRE, ParentVersion, StepID)
        REFERENCES BXT_MRecipeStep (ParentRE, ParentVersion, StepID))

CREATE TABLE BXT_MRecipeOtherInformation (
    RE_ID               CHAR (128)      NOT NULL,
    REVersion           CHAR (16)       NOT NULL,
    StepID              CHAR (128)      NOT NULL,
    DataID              CHAR (32)       NOT NULL,
    DataType            CHAR (32),
    DataValue           CHAR (255),
    Description         CHAR (255),
    PRIMARY KEY (RE_ID, REVersion, DataID),
    FOREIGN KEY (RE_ID, REVersion)
        REFERENCES BXT_MRecipeElement (RE_ID, REVersion))

CREATE TABLE BXT_MRecipeElementEquip (
    RE_ID               CHAR (128)      NOT NULL,
    REVersion           CHAR (16)       NOT NULL,
    PropertyID          CHAR (32)       NOT NULL,
    DefaultValue        CHAR (128),
    DataInterpretation  INTEGER,
    EvaluationRule      INTEGER,
    EngrUnits           CHAR (32),
    Description         CHAR (255),
    PRIMARY KEY (RE_ID, REVersion, PropertyID),
    FOREIGN KEY (RE_ID, REVersion)
        REFERENCES BXT_MRecipeElement )

CREATE TABLE BXT_MRecipeStepEquip (
    ParentRE            CHAR (128)      NOT NULL,
    ParentVersion       CHAR (16)       NOT NULL,
    StepID              CHAR (128)      NOT NULL,
    PropertyID          CHAR (32)       NOT NULL,
    PropertyValue       CHAR (128),
    PRIMARY KEY (ParentRE, ParentVersion, StepID, PropertyID),
    FOREIGN KEY (ParentRE, ParentVersion, StepID)
        REFERENCES BXT_MRecipeStep (ParentRE, ParentVersion, StepID))
```

```
CREATE TABLE BXT_EquipElement (
    EquipmentID          CHAR (32)        NOT NULL,
    EE_Type              INTEGER,
    EE_Level             INTEGER,
    ContainedIn          CHAR (32),
    Description          CHAR (255),
    PRIMARY KEY (EquipmentID))

CREATE TABLE BXT_EquipLink (
    EquipmentID          CHAR (32)        NOT NULL,
    ToEquipmentID        CHAR (32)        NOT NULL,
    Description          CHAR (255),
    PRIMARY KEY (EquipmentID , ToEquipmentID),
    FOREIGN KEY (EquipmentID)
        REFERENCES BXT_EquipElement,
    FOREIGN KEY (ToEquipmentID)
        REFERENCES BXT_EquipElement )

CREATE TABLE BXT_EquipInclude (
    EquipmentID          CHAR (32)        NOT NULL,
    ClassEquipmentID     CHAR (32)        NOT NULL,
    Description          CHAR (255),
    PRIMARY KEY (EquipmentID, ClassEquipmentID),
    FOREIGN KEY (EquipmentID)
        REFERENCES BXT_EquipElement,
    FOREIGN KEY (ClassEquipmentID)
        REFERENCES BXT_EquipElement  )

CREATE TABLE BXT_EquipProperty (
    EquipmentID          CHAR (32)        NOT NULL,
    PropertyID           CHAR (32)        NOT NULL,
    PropertyValue        CHAR (255),
    EngrUnits            CHAR (32),
    Description          CHAR (255),
    PRIMARY KEY (EquipmentID, PropertyID),
    FOREIGN KEY (EquipmentID)
        REFERENCES BXT_EquipElement  )

CREATE TABLE BXT_EquipInterface (
    EquipmentID          CHAR (32)        NOT NULL,
    EPI_ID               CHAR (32)        NOT NULL,
    EPI_Definition       CHAR (32)        NOT NULL,
    Description          CHAR (255),
    PRIMARY KEY (EPI_ID, EquipmentID),
    FOREIGN KEY (EquipmentID)
        REFERENCES BXT_EquipElement  )

CREATE TABLE BXT_EquipInterfaceDefinition (
    EPI_Definition       CHAR (32)        NOT NULL,
    Description          CHAR (255),
    PRIMARY KEY (EPI_Definition)  )
```

```
CREATE TABLE BXT_EquipInterfaceParameter (
    EPI_Definition      CHAR (32)       NOT NULL,
    ParameterID         CHAR (32)       NOT NULL,
    ParentParamID       CHAR (32),
    Type                INTEGER         NOT NULL,
    EngrUnits           CHAR (32),
    EnumSet             CHAR (32),
    Scaled              INTEGER,
    DefaultValue        CHAR (128),
    Description         CHAR (255),
    PRIMARY KEY (EPI_Definition, ParameterID),
    FOREIGN KEY (EPI_Definition)
        REFERENCES BXT_EquipInterfaceDefinition )

CREATE TABLE BXT_ScheduleEntry (
    ScheduleEntryID     CHAR (64)       NOT NULL,
    ParentSchedID       CHAR (64),
    ExternalID          CHAR (64),
    RE_ID               CHAR (128),
    REVersion           CHAR (16),
    SE_Type             INTEGER,
    BatchID             CHAR (128),
    LotID               CHAR (128),
    CampaignID          CHAR (128),
    ProductID           CHAR (32),
    OrderID             CHAR (128),
    SE_Action           INTEGER,
    SchedStatus         INTEGER,
    StartCondition      CHAR (255),
    InitialMode         INTEGER,
    SchedStartTime      DATETIME,
    SchedEndTime        DATETIME,
    BatchPriority       INTEGER,
    BatchSize           FLOAT,
    EngrUnits           CHAR (32),
    SENote              CHAR (255),
    Description         CHAR (255),
    PRIMARY KEY (ScheduleEntryID))

CREATE TABLE BXT_ScheduleEquip (
    ScheduleEntryID     CHAR (64)       NOT NULL,
    RequirementID       CHAR (32)       NOT NULL,
    Description         CHAR (255),
    PRIMARY KEY (ScheduleEntryID, RequirementID))

CREATE TABLE BXT_ScheduleProperty (
    ScheduleEntryID     CHAR (64)       NOT NULL,
    RequirementID       CHAR (32)       NOT NULL,
    PropertyName        CHAR (32)       NOT NULL,
    PropertyValue       CHAR (255),
    EngrUnits           CHAR (32),
    Description         CHAR (255),
    PRIMARY KEY (ScheduleEntryID, RequirementID, PropertyName))
```

```
CREATE TABLE BXT_ScheduleParameter (
    ScheduleEntryID            CHAR (64)     NOT NULL,
    ParameterID                CHAR (32)     NOT NULL,
    ParentParameterID          CHAR (32),
    ParameterValue             CHAR (255),
    EngrUnits                  CHAR (32),
    ItemLocation               CHAR (128),
    EnumSet                    CHAR (32),
    Description                CHAR (255),
    PRIMARY KEY (ScheduleEntryID, ParameterID))

CREATE TABLE BXT_HistoryElement (
    HistoryElementID           INTEGER       NOT NULL,
    BatchID                    CHAR (128),
    MasterRecipeID             CHAR (128),
    MasterRecipeVersion        CHAR (16),
    ControlRecipeID            CHAR (28),
    ReferenceEquipProcedure    INTEGER,
    RecipeProcedure            CHAR (128),
    UnitProcedure              CHAR (128),
    UnitProcedureCounter       INTEGER,
    Operation                  CHAR (128),
    OperationCounter           INTEGER,
    Phase                      CHAR (128),
    PhaseCounter               INTEGER,
    EquipmentID                CHAR (32),
    EPI_ID                     CHAR (32),
    PRIMARY KEY (HistoryElementID))

CREATE TABLE BXT_HistoryLog (
    RecordID                   INTEGER       NOT NULL,
    UTC                        DATETIME,
    LocalTime                  DATETIME      NOT NULL,
    BatchID                    CHAR (128),
    HistoryElementID           INTEGER,
    EquipmentID                CHAR (32),
    EPI_ID                     CHAR (32),
    UserID                     CHAR (64),
    RecordSet                  INTEGER       NOT NULL,
    RecordSubSet               INTEGER,
    RecordAlias                CHAR (32),
    NewValue                   CHAR (128),
    OldValue                   CHAR (128),
    EngrUnits                  CHAR (32),
    PRIMARY KEY (RecordID))
```

Annex C (informative) — Abbreviations

The following are abbreviations that are used in this standard:

BXT	Batch exchange table
EPE	Equipment procedural element
ID	Identification
IEC	International Electrotechnical Commission
ISA	ISA–The Instrumentation, Systems, and Automation Society
ISO	International Organization for Standardization
MR	Master recipe
PFC	Procedure function chart
RE	Recipe element
SFC	Sequential function chart
SOP	Standard operating procedure
SQL	Structured query language
UML	Unified modeling language
UTC	Universal coordinated time

This page intentionally left blank.

Annex D (informative) — Language guidelines

A language is a set of symbols and the rules for their use in communication. Clause 6 describes guidelines for creating the symbols and rules that are needed for selected batch process-related communications.

Communication, involving machines and people, occurs among all six of the control activities that are described in Part 1. Both ends of any communication path must understand the same symbols and they must use the same rules.

Communication between people and electronic batch processing systems is usually done with a video or other graphic devices and various pointing and input devices. Text is frequently used for detailed communication, but visual symbols are often more effective for communicating complex relationships.

The central feature of batch control is the recipe. Graphic symbols related to the procedure and rules for their use are defined in clause 6. Other parts of the recipe (e.g., formula) contain detail that may be better represented visually as text.

D.1 PFC derivation

Three example depiction methods, Table format, Gantt Chart notation, and Sequential Function Charts (SFC), were discussed in ISA-TR88.00.03-1996, *Possible Recipe Procedure Presentation Formats*.

A table format is attractive because of its simplicity, intuitive interpretation, and flexibility (e.g., support of additional attributes listed in a tabular form, of inserts). However, the table format is limited to procedures that are linear in nature because it does not adequately address selections and parallel steps.

Gantt Chart notation is commonly used to depict time-oriented activities and it can be extended to depict recipe procedures in more complex cases than the table format. However, Gantt Chart notation does not lend itself to portraying conditional decisions.

Sequential function charts may be used to portray the conditional decisions in a procedure. However, there are certain aspects of procedures that are not adequately represented by SFCs or that become unnecessarily complicated to depict in SFCs.

Elements of these three depiction methods were combined to create a notation called Procedure Function Chart (PFC). The PFC is defined to graphically depict the procedure portion of the recipe and it is a derivation of Function Chart notation, as defined in IEC 60848:1988, that has been modified to make it usable in recipe depiction, and to add some of the benefits of Gantt Chart notation and the table format.

D.2 Recipe procedure

The separation of the recipe procedure (that defines desired process functionality) from the equipment procedure (that defines control execution) supports one of the goals stated clearly in Part 1: that recipes can be created without routine control engineering involvement. This goal defines a division of effort between the control engineering function and the recipe authoring function. This goal requires that the control engineering function defines and implements equipment procedural elements (e.g., equipment phases) and provides representations for use by the Master Recipe author, with appropriate constraints.

Part 1 defined the following four levels of procedural elements:

a) Procedure

b) Unit procedure

c) Operation

d) Phase

The procedure consists of some number of unit procedures. A unit procedure consists of an ordered set of operations that depict a contiguous production sequence that is to take place within a unit. An operation is made up of an ordered set of phases. Although there is no limit to the number of phases that may be simultaneously active in a unit, only one operation is presumed to be active in a unit at any time. Unit procedures are largely independent, but they encapsulate or reference lower-level operations and phases that may interact with operations and phases in other unit procedures.

D.3 Requirements for procedural control element depiction

A clear method of depicting procedural control elements in recipe procedures is essential. The following requirements have been identified for a clear depiction method:

a) Simple to follow: Easy for people to understand

b) Easy to build: Few syntax requirements and symbols to learn

c) Clearly defined boundaries: Standardized graphical symbol for Start and End

d) Unambiguous depiction of execution order: Sequence, parallelism, selection (divergence) and convergence

e) Expression of coordination relationships: Material transfers, Wait for, Synchronize

f) Hierarchical level: Standardized symbols for Procedure, Unit Procedure, Operation, Phase

g) Existence of levels: Standardized graphical symbol to show possible decomposition of an element of the hierarchy

h) Applicable to master recipes and control recipes

i) Applicable to all levels: Similar set of symbols and rules at all levels in a recipe

j) Independent of media: Equally usable and understandable whether implemented with pencil and paper or with full-color animated computer graphics

Annex E (informative) — Procedure function chart processing examples

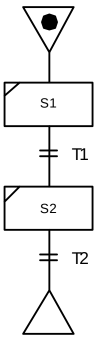
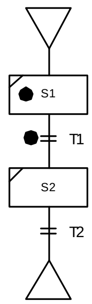
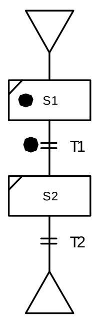
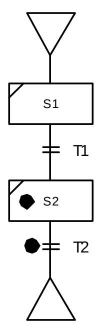
Many rules concerning Procedure Function Chart processing are system dependent. The following examples illustrate possible PFC processing systems.

In these examples, a dot is used to identify the active symbols. The dot is not part of the symbols and it is only used for explanatory purposes. Active means that the symbols are being evaluated or executed by the PFC processing system.

When a procedural element is active, the lower-level PFC or equipment procedural element that it represents is currently being processed according to its state model. For these examples, the ANSI/ISA-88.01-1995 example state model is used. Inactive means that the symbols are not being evaluated or executed by the PFC processing system. When a procedural element is inactive, the lower-level PFC or equipment procedural element that it represents has a state that is appropriate with its state model, but it is not being processed and, therefore, it cannot change its state.

When an explicit transition is active, its expression is being evaluated by the PFC processing system, or the expression has already been evaluated true and notification of this has been sent to the lower-level PFC or equipment procedural entity that the preceding recipe element represents.

Example 1: Phase completes after the trailing transition is evaluated to be true (see following figure).

	The PFC has started and the begin symbol is active.
	Recipe phase S1 becomes active. The corresponding equipment phase (not shown) has also become active. Explicit transition T1 becomes active and it is continually evaluated once recipe phase S1 becomes active.
	T1 is evaluated to be true. Equipment phase S1 receives input that T1 is true. Equipment phase S1 continues to be active until its logic results in it becoming inactive.
	Equipment phase S1 enters the complete state and it becomes inactive. This completion may be a result of the input from T1 and the successful accomplishment of housekeeping actions. Recipe phase S1 becomes inactive once equipment phase S1 becomes inactive. T1 stops being evaluated and it becomes inactive. Recipe phase S2 and its corresponding equipment phase S2 become active. Explicit transition T2 becomes active and it is continually evaluated.

Developing and promulgating technically sound consensus standards, recommended practices, and technical reports is one of ISA's primary goals. To achieve this goal, the Standards and Practices Department relies on the technical expertise and efforts of volunteer committee members, chairmen, and reviewers.

ISA is an American National Standards Institute (ANSI) accredited organization. ISA administers United States Technical Advisory Groups (USTAGs) and provides secretariat support for International Electrotechnical Commission (IEC) and International Organization for Standardization (ISO) committees that develop process measurement and control standards. To obtain additional information on the Society's standards program, please write:

ISA
Attn: Standards Department
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, NC 27709

ISBN 1-55617-745-3