

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

61512-2

Première édition
First edition
2001-11

**Contrôle-commande des processus
de fabrication par lots (batch) –**

**Partie 2:
Structures de données et règles générales
relatives aux langages**

Batch control –

**Part 2:
Data structures and guidelines for languages**



Numéro de référence
Reference number
CEI/IEC 61512-2:2001

Numérotation des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60000. Ainsi, la CEI 34-1 devient la CEI 60034-1.

Editions consolidées

Les versions consolidées de certaines publications de la CEI incorporant les amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

Informations supplémentaires sur les publications de la CEI

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique. Des renseignements relatifs à cette publication, y compris sa validité, sont disponibles dans le Catalogue des publications de la CEI (voir ci-dessous) en plus des nouvelles éditions, amendements et corrigenda. Des informations sur les sujets à l'étude et l'avancement des travaux entrepris par le comité d'études qui a élaboré cette publication, ainsi que la liste des publications parues, sont également disponibles par l'intermédiaire de:

- **Site web de la CEI** (www.iec.ch)
- **Catalogue des publications de la CEI**

Le catalogue en ligne sur le site web de la CEI (www.iec.ch/catlg-f.htm) vous permet de faire des recherches en utilisant de nombreux critères, comprenant des recherches textuelles, par comité d'études ou date de publication. Des informations en ligne sont également disponibles sur les nouvelles publications, les publications remplacées ou retirées, ainsi que sur les corrigenda.

- **IEC Just Published**

Ce résumé des dernières publications parues (www.iec.ch/JP.htm) est aussi disponible par courrier électronique. Veuillez prendre contact avec le Service client (voir ci-dessous) pour plus d'informations.

- **Service clients**

Si vous avez des questions au sujet de cette publication ou avez besoin de renseignements supplémentaires, prenez contact avec le Service clients:

Email: custserv@iec.ch
Tél: +41 22 919 02 11
Fax: +41 22 919 03 00

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site** (www.iec.ch)
- **Catalogue of IEC publications**

The on-line catalogue on the IEC web site (www.iec.ch/catlg-e.htm) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

This summary of recently issued publications (www.iec.ch/JP.htm) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

Email: custserv@iec.ch
Tel: +41 22 919 02 11
Fax: +41 22 919 03 00

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

61512-2

Première édition
First edition
2001-11

**Contrôle-commande des processus
de fabrication par lots (batch) –**

**Partie 2:
Structures de données et règles générales
relatives aux langages**

Batch control –

**Part 2:
Data structures and guidelines for languages**

© IEC 2001 Droits de reproduction réservés — Copyright - all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Electrotechnical Commission
Telefax: +41 22 919 0300

3, rue de Varembe Geneva, Switzerland
e-mail: inmail@iec.ch IEC web site <http://www.iec.ch>



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE **XD**

*Pour prix, voir catalogue en vigueur
For price, see current catalogue*

SOMMAIRE

AVANT-PROPOS	10
INTRODUCTION.....	14
1 Domaine d'application	16
2 Références normatives.....	16
3 Définitions	16
4 Modèle de données	18
4.1 Introduction	18
4.2 Vue d'ensemble du modèle	20
4.3 Modèle de recette.....	22
4.3.1 Entité de recette	22
4.3.2 Eléments des entités de recette.....	30
4.3.3 Relations entre entités de recette (structure procédurale).....	34
4.3.4 Blocs de construction de recette	34
4.3.5 Prescriptions d'équipement.....	38
4.3.6 Paramètres de recette	42
4.4 Modèle d'équipement	44
4.5 Planification et programmation de la production	48
4.6 Gestion de l'information de production.....	52
5 Tables relationnelles pour l'échange d'informations.....	58
5.1 Introduction	58
5.1.1 Méthode	60
5.1.2 Tables d'échange	60
5.1.3 Informations d'échange communes.....	62
5.2 Informations de recette maître.....	82
5.2.1 Définitions de recette.....	82
5.2.2 Structure de recette.....	82
5.2.3 Vue d'ensemble d'une table et contraintes d'intégrité	84
5.2.4 Sommaire des tables de recette	90
5.2.5 Définitions de tables de recette	90
5.3 Echange de modèle d'équipement de cellule de processus	108
5.3.1 Description d'équipement	108
5.3.2 Vue d'ensemble de table et contraintes d'intégrité.....	108
5.3.3 Vue d'ensemble de la table de description des équipements	110
5.3.4 Sommaire de la table d'information des équipements	110
5.3.5 Définitions de table d'équipements	112
5.4 Echange des informations de programmation	118
5.4.1 Vue d'ensemble des tables de programmation.....	118
5.4.2 Sommaire des tables de programmation.....	120
5.4.3 Définitions des tables de programmation	120

CONTENTS

FOREWORD.....	11
INTRODUCTION.....	15
1 Scope.....	17
2 Normative references	17
3 Definitions	17
4 Data model.....	19
4.1 Introduction	19
4.2 Overview model.....	21
4.3 Recipe model	23
4.3.1 Recipe entity	23
4.3.2 Parts of recipe entities.....	31
4.3.3 Recipe entity relation (procedural structure)	35
4.3.4 Recipe building blocks.....	35
4.3.5 Equipment requirements.....	39
4.3.6 Recipe parameters	43
4.4 Equipment model	45
4.5 Production planning and scheduling	49
4.6 Production information management	53
5 Relational tables for information exchange	59
5.1.1 Method	61
5.1.2 Exchange tables	61
5.1.3 Common exchange information.....	63
5.2 Master recipe information	83
5.2.1 Recipe definitions	83
5.2.2 Recipe structure	83
5.2.3 Table overview and integrity constraints	85
5.2.4 Recipe table summary	91
5.2.5 Recipe table definitions	91
5.3 Process cell equipment model exchange	109
5.3.1 Equipment description	109
5.3.2 Table overview and integrity constraints	109
5.3.3 Equipment description table overview.....	111
5.3.4 Equipment information table summary	111
5.3.5 Equipment table definitions.....	113
5.4 Schedule information exchange.....	119
5.4.1 Schedule table overview	119
5.4.2 Schedule table summary	121
5.4.3 Schedule table definitions.....	121

5.5	Echange d'informations de production	124
5.5.1	Informations de recette de contrôle.....	126
5.5.2	Informations d'équipement.....	126
5.5.3	Historique du batch.....	126
5.6	Domaines de table d'échange	132
6	Diagrammes fonctionnels de procédures	134
6.1	Notation de diagramme fonctionnel de procédure	136
6.1.1	Symboles	136
6.1.2	Initialisation de procédures et de procédures d'unité	158
6.1.3	Informations non procédurales de la recette maître	162
6.2	Représentation de la recette de contrôle	164
6.3	Traitement des exceptions.....	164
	Annexe A (normative) Technique de modélisation des données	166
	Annexe B (normative) Liste de définitions SQL	170
	Annexe C (informative) Abréviations	194
	Annexe D (informative) Règles de langage	196
	Annexe E (informative) Exemples de traitement de diagrammes fonctionnels de procédures.....	200
	Figure 1 – Vue d'ensemble du modèle	20
	Figure 2 – Entités recette.....	24
	Figure 3 – Eléments de l'entité de recette	32
	Figure 4 – Blocs de construction de recette	34
	Figure 5 – Principe des blocs de construction	38
	Figure 6 – Prescriptions d'équipement de l'entité recette	40
	Figure 7 – Modèle du paramètre	42
	Figure 8 – Structure de l'équipement	44
	Figure 9 – Relations entre entités d'équipement	46
	Figure 10 – Classes d'équipements	48
	Figure 11 – Programme de batch	50
	Figure 12 – Information de production.....	54
	Figure 13 – Transfert de données par des tables d'échange	60
	Figure 14 – Tables d'échange des informations communes.....	62
	Figure 15 – Les éléments de recette imbriqués constituent la recette.....	84
	Figure 16 – Relations entre les tables d'échange	86
	Figure 17 – Rapport entre les entrées des tables	88
	Figure 18 – Tables d'échange d'information d'équipement.....	108
	Figure 19 – Structure du programme.....	118
	Figure 21 – Symboles des éléments procéduraux de la recette	136
	Figure 22 – Eléments procéduraux qui encapsulent des éléments recette d'un niveau inférieur.....	138
	Figure 23 – Symbole de début	138

5.5	Production information exchange.....	125
5.5.1	Control recipe information.....	127
5.5.2	Equipment information.....	127
5.5.3	Batch history	127
5.6	Exchange table domains	133
6	Procedure function charts.....	135
6.1	Procedure function chart notation.....	137
6.1.1	Symbols	137
6.1.2	Procedure and unit procedure initiation	159
6.1.3	Non-procedural master recipe information	163
6.2	Control recipe depiction.....	165
6.3	Exception handling.....	165
Annex A (normative)	Data modelling technique.....	167
Annex B (normative)	SQL definition listing.....	171
Annex C (informative)	Abbreviations	195
Annex D (informative)	Language guidelines	197
Annex E (informative)	Procedure function chart processing examples.....	201
Figure 1	– Overview model	21
Figure 2	– Recipe entities	25
Figure 3	– Parts of recipe entities	33
Figure 4	– Recipe building block	35
Figure 5	– Building block concept	39
Figure 6	– Recipe entity equipment requirements	41
Figure 7	– Parameter model.....	43
Figure 8	– Equipment structure	45
Figure 9	– Equipment entity relations.....	47
Figure 10	– Equipment classes	49
Figure 11	– Batch schedule	51
Figure 12	– Production information	55
Figure 13	– Data transfer using exchange tables	61
Figure 14	– Common information exchange tables.....	63
Figure 15	– Nested recipe elements make up a recipe	85
Figure 16	– Exchange table relationships.....	87
Figure 17	– How entries relate in the tables	89
Figure 18	– Equipment information exchange tables	109
Figure 19	– Schedule structure	119
Figure 20	– Batch history.....	127
Figure 21	– Recipe procedural element symbols	137
Figure 22	– Procedural elements that encapsulate lower-level recipe procedural elements	139
Figure 23	– Begin symbol	139

Figure 24 – Symbole de fin	140
Figure 25 – Symbole d'allocation	140
Figure 26 – Exemples de synchronisation d'éléments	142
Figure 27 – Transition implicite	142
Figure 28 – Transition explicite	144
Figure 29 – Début de sélection de séquences	146
Figure 30 – Fin de sélection de séquences	146
Figure 31 – Début de séquences simultanées	148
Figure 32 – Fin de séquences simultanées	148
Figure 33 – Diagramme valide de sélection de séquences	150
Figure 34 – Diagramme valide de séquences simultanées	152
Figure 35 – Bouclage avec éléments procéduraux de recette explicites	154
Figure 36 – Diagramme fonctionnel de procédure invalide	156
Figure 37 – Représentation de la procédure et initialisation des procédures d'unité	158
Figure 38 – Liaisons relatives entre les entités procédurales	160
Figure 39 – Liaisons relatives entre les entités procédurales – Variante 1	162
Table 1 – Entité recette	24
Table 2 – Sous-classes – vue d'ensemble	26
Table 3 – Recette	26
Table 4 – Composant de recette	28
Table 5 – Bloc de construction de recette	28
Table 6 – Entité de recette générale	28
Table 7 – Entité de recette de site	28
Table 8 – Entité de recette maître	30
Table 9 – Entité de recette de contrôle	30
Table 10 – Paramètre	32
Table 11 – Prescription d'équipement	32
Table 12 – Autres informations	34
Table 13 – Élément structurel procédural	34
Table 14 – Élément procédural d'équipement	36
Table 15 – Entité d'équipement	40
Table 16 – Propriété d'équipement	40
Table 17 – Type de propriété d'équipement	42
Table 18 – Relation d'équipement	46
Table 19 – Classe d'équipement	48
Table 20 – Entrée de programme de batch	50
Table 21 – Paramètre de programme	50
Table 22 – Relation de programmation	52
Table 23 – Information de production	54
Table 24 – Information spécifique au batch	54
Table 25 – Historique du batch	54

Figure 24 – End symbol	141
Figure 25 – Allocation symbol	141
Figure 26 – Element synchronization examples	143
Figure 27 – Implicit transition	143
Figure 28 – Explicit transition	145
Figure 29 – Beginning of sequence selection	147
Figure 30 – End of sequence selection	147
Figure 31 – Beginning of simultaneous sequences	149
Figure 32 – End of simultaneous sequences	149
Figure 33 – Valid sequence selection diagram	151
Figure 34 – Valid simultaneous sequence diagram	153
Figure 35 – Looping with explicit recipe procedural elements	155
Figure 36 – Invalid procedure function chart	157
Figure 37 – Depiction of procedure and unit procedure initiation	159
Figure 38 – Relative relationship of procedural entities	161
Figure 39 – Relative relationship of procedural entities – Alternate 1	163
Table 1 – Recipe entity	25
Table 2 – Subclasses – overview	27
Table 3 – Recipe	27
Table 4 – Recipe component	29
Table 5 – Recipe building block	29
Table 6 – General recipe entity	29
Table 7 – Site recipe entity	29
Table 8 – Master recipe entity	31
Table 9 – Control recipe entity	31
Table 10 – Parameter	33
Table 11 – Equipment requirement	33
Table 12 – Other information	35
Table 13 – Procedural structural element	35
Table 14 – Equipment procedural element	37
Table 15 – Equipment entity	41
Table 16 – Equipment property	41
Table 17 – Equipment property type	43
Table 18 – Equipment relation	47
Table 19 – Equipment class	49
Table 20 – Batch schedule entry	51
Table 21 – Schedule parameter	51
Table 22 – Schedule relation	53
Table 23 – Production information	55
Table 24 – Batch specific information	55
Table 25 – Batch history	55

Table 26 – Information commune	56
Table 27 – Entité procédurale exécutée	58
Table 28 – Rapport de batch	58
Table 29 – BXT_Exchange	62
Table 30 – Contenus de la table BXT_Exchange	64
Table 31 – BXT_EnumerationSet	64
Table 32 – Enumérations standards	64
Table 33 – Enumeration_BXT	68
Table 34 – Enumérations standards	68
Table 35 – Tables d'échange de recette	90
Table 36 – BXT_MRecipeElement	92
Table 37 – BXT_MRecipeStep	94
Table 38 – BXT_MRecipeTransition	96
Table 39 – BXT_MRecipeLink	98
Table 40 – BXT_MRecipeElementParameter	100
Table 41 – Sous-paramètres standards	102
Table 42 – BXT_MRecipeStepParameter	102
Table 43 – BXT_MRecipeOtherInformation	104
Table 44 – BXT_MRecipeElementEquip	106
Table 45 – BXT_MRecipeStepEquip	106
Table 46 – Tables d'échange d'information d'équipement	110
Table 47 – BXT_EquipElement	112
Table 48 – BXT_EquipLink	112
Table 49 – BXT_EquipInclude	114
Table 50 – BXT_EquipProperty	114
Table 51 – BXT_EquipInterface	116
Table 52 – BXT_EquipInterfaceDefinition	116
Table 53 – BXT_EquipInterfaceParameter	116
Table 54 – Tables d'échange d'information de programmation	120
Table 55 – BXT_ScheduleEntry	120
Table 56 – BXT_ScheduleEquip	122
Table 57 – BXT_ScheduleProperty	124
Table 58 – BXT_ScheduleParameter	124
Table 59 – BXT_HistoryElement	128
Table 60 – BXT_HistoryLog	130
Table 61 – Domaines de table d'échange	132
Table A.1 – Notation UML	166
Table A.2 – Notation ERD	168

Table 26 – Common information	57
Table 27 – Executed procedural entity	59
Table 28 – Batch report	59
Table 29 – BXT_Exchange.....	63
Table 30 – BXT_Exchange table contents.....	65
Table 31 – BXT_EnumerationSet	65
Table 32 – Standard enumeration sets.....	65
Table 33 – BXT_Enumeration	69
Table 34 – Standard enumerations	69
Table 35 – Recipe exchange tables	91
Table 36 – BXT_MRecipeElement.....	93
Table 37 – BXT_MRecipeStep	95
Table 38 – BXT_MRecipeTransition	97
Table 39 – BXT_MRecipeLink	99
Table 40 – BXT_MRecipeElementParameter.....	101
Table 41 – Standard sub-parameters	103
Table 42 – BXT_MRecipeStepParameter	103
Table 43 – BXT_MRecipeOtherInformation	105
Table 44 – BXT_MRecipeElementEquip.....	107
Table 45 – BXT_MRecipeStepEquip	107
Table 46 – Equipment information exchange tables	111
Table 47 – BXT_EquipElement	113
Table 48 – BXT_EquipLink.....	113
Table 49 – BXT_EquipInclude.....	115
Table 50 – BXT_EquipProperty.....	115
Table 51 – BXT_EquipInterface	117
Table 52 – BXT_EquipInterfaceDefinition.....	117
Table 53 – BXT_EquipInterfaceParameter	117
Table 54 – Schedule information exchange tables	121
Table 55 – BXT_ScheduleEntry	121
Table 56 – BXT_ScheduleEquip.....	123
Table 57 – BXT_ScheduleProperty	125
Table 58 – BXT_ScheduleParameter	125
Table 59 – BXT_HistoryElement	129
Table 60 – BXT_HistoryLog	131
Table 61 – Exchange table domains	133
Table A.1 – UML notation	167
Table A.2 – ERD notation	169

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

CONTRÔLE-COMMANDE DES PROCESSUS DE FABRICATION PAR LOTS (BATCH) –

Partie 2: Structures de données et règles générales relatives aux langages

AVANT-PROPOS

- 1) La CEI (Commission Electrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI, entre autres activités, publie des Normes internationales. Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO) selon les conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux intéressés sont représentés dans chaque comité d'études.
- 3) Les documents produits se présentent sous la forme de recommandations internationales. Ils sont publiés comme normes, spécifications techniques, rapports techniques ou guides et agréés comme tels par les Comités nationaux.
- 4) Dans le but d'encourager l'unification internationale, les Comités nationaux de la CEI s'engagent à appliquer de façon transparente, dans toute la mesure du possible, les Normes internationales de la CEI dans leurs normes nationales et régionales. Toute divergence entre la norme de la CEI et la norme nationale ou régionale correspondante doit être indiquée en termes clairs dans cette dernière.
- 5) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand un matériel est déclaré conforme à l'une de ses normes.
- 6) L'attention est attirée sur le fait que certains des éléments de la présente Norme internationale peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.
- 7) Lorsqu'il est question de conformité, le lecteur doit avoir connaissance de la politique de la CEI et du style de la CEI dans l'expression des prescriptions. Ceci est décrit dans le paragraphe 6.6.1 des Directives ISO/CEI, Partie 3: 1997, «Règles pour la structure et la rédaction de Normes internationales».

La Norme internationale CEI 61512-2 a été établie par le sous-comité 65A: Aspects systèmes, du comité d'études 65 de la CEI: Mesure et commande dans les processus industriels.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65A/338/FDIS	65A/344/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/CEI, partie 3.

Les annexes A et B font partie intégrante de cette norme. Les annexes C, D et E sont données uniquement à titre d'information.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

BATCH CONTROL –**Part 2: Data structures and guidelines for languages****FOREWORD**

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.
- 7) When dealing with compliance issues, the reader needs to be aware of IEC policy and IEC style in expressing requirements. This is described in 6.6.1 of the ISO/IEC Directives, Part 3: 1997, "Rules for the structure and drafting of International Standards".

International Standard IEC 61512-2 has been prepared by subcommittee 65A: System aspects, of IEC technical committee 65: Industrial-process measurement and control.

The text of this standard is based on the following documents:

FDIS	Report on voting
65A/338/FDIS	65A/344/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 3.

Annexes A and B form an integral part of this part of this standard. Annexes C, D and E are for information only.

Le Comité a décidé que le contenu de cette publication ne sera pas modifié avant 2006. A cette date, cette publication sera:

- confirmée;
- annulée;
- remplacée par une nouvelle révisée, ou
- amendée.

The committee has decided that the contents of this publication will remain unchanged until 2006. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

INTRODUCTION

La partie 1 de cette Norme internationale relative au contrôle batch définit les modèles et la terminologie s'appliquant au contrôle batch. La partie 2 concerne les structures de données et les règles générales relatives aux langages. Les structures de données sont abordées par le modèle de données défini dans l'article 4 qui identifie plus précisément les objets et les relations abordés par les modèles et les concepts de la partie 1. Les structures de données sont également abordées par les tables relationnelles pour l'échange de l'information définies à l'article 5. Les langages sont abordés par une méthodologie de représentation de la recette définie à l'article 6.

L'utilisation proposée du modèle de données est de fournir un point de départ au développement de spécifications d'interfaces pour des composants logiciels concernant tout sous-ensemble de la partie 1 de cette norme. Le modèle de données aborde l'ensemble de la partie 1 de cette norme en tant que modèle objet intégré, mais il ne suppose ou n'exclut aucune architecture de système ou aucun échange d'informations spécifiques. Le modèle n'impose aucune division fonctionnelle spécifique entre les systèmes.

Une méthode spécifique pour l'échange de données sélectionnées est définie dans l'article 5. Les tables relationnelles sont utilisées comme méthode d'échange d'informations car, dans les limites de l'information traitée:

- elles utilisent des technologies largement disponibles;
- elles autorisent la migration vers d'autres technologies;
- elles sont adaptées;
- elles sont cohérentes avec d'autres sections de cette norme.

Il n'a pas été défini de multiples méthodes de transfert d'information, de même qu'il n'a pas été tenté d'identifier toute information susceptible d'être échangée. Dans le futur, des méthodes supplémentaires seront définies afin de fournir des moyens différents d'échange d'information.

L'article 6 présente les symboles et les règles d'un langage graphique qui peut être utilisé pour décrire les recettes. Les recettes représentent l'élément central du contrôle batch et peuvent présenter un large éventail de complexité, mais il n'existe aucune description qui soit idéale dans toutes les circonstances. Une simple table, par exemple, serait la forme de recette la plus appropriée pour des cas simples. Cette norme fournit une méthode pour la description des procédures de recette maître et de recette de contrôle qui peuvent être appliquées à un éventail plus large de complexité.

Bien que cette norme soit destinée essentiellement aux processus batchs, elle peut être de grande utilité pour d'autres types de processus.

INTRODUCTION

Part 1 of this International Standard on batch control provides models and terminology applicable to batch control. Part 2 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 were 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 part 1 of this standard. The data model addresses all of part 1 of this 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:

- utilise broadly available technologies;
- are amenable to translation to other technologies;
- are adequate;
- 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 of exchanging 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, it may be of considerable value for other types of processes.

CONTRÔLE-COMMANDE DES PROCESSUS DE FABRICATION PAR LOTS (BATCH) –

Partie 2: Structures de données et règles générales relatives aux langages

1 Domaine d'application

La présente partie de cette norme relative au contrôle batch définit les modèles de données qui décrivent le contrôle batch tel qu'il est appliqué dans les industries de processus, des structures de données pour faciliter les communications au sein d'un système et entre les systèmes mis en œuvre pour le contrôle batch et des règles pour les langages de représentation des recettes. Il convient de se référer à l'annexe A pour l'explication de la notation UML qui est utilisée dans cette partie de la norme. L'annexe B résume toutes les définitions SQL de l'article 5.

2 Références normatives

Les documents normatifs suivants contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente partie de la CEI 61512. Pour les références datées, les amendements ultérieurs ou les révisions de ces publications ne s'appliquent pas. Toutefois, les parties prenantes aux accords fondés sur la présente partie de la CEI 61512 sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des documents normatifs indiqués ci-après. Pour les références non datées, la dernière édition du document normatif en référence s'applique. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur.

CEI 60848:1988, *Etablissement des diagrammes fonctionnels pour systèmes de commande*

CEI 60050-351:1998, *Mesure et contrôle dans les processus industriels – Termes et définitions*

CEI 61131-3:1993, *Automates programmables – Partie 3: Langages de programmation*

CEI 61512-1:1997, *Contrôle-commande des processus de fabrication par lots – Partie 1: Modèles et terminologie*

ISO/CEI 9075:1992, *Technologies de l'information – Langages de base de données – SQL*

3 Définitions

Pour les besoins de la présente partie de la CEI 61512, les définitions suivantes s'appliquent. Les définitions et les concepts présentés dans la partie 1 de cette norme s'appliquent également, excepté dans les cas où les différences sont explicitement formulées dans cette partie. Les définitions de la CEI 60050-351 ont été également utilisées comme base.

3.1

symbole d'allocation

symbole graphique utilisé pour représenter l'encapsulation de l'allocation de ressource et des règles de désallocation pour un élément procédural de recette

BATCH CONTROL –

Part 2: Data structures and guidelines for languages

1 Scope

This part of this 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. Refer to Annex A for an explanation of the UML notation that is used in this part of this standard. Refer to Annex B for a summary of all of the SQL definitions from clause 5.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 61512. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of IEC 61512 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

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

IEC 60050-351:1998, *Industrial-process measurement and control – Terms and definitions*

IEC 61131-3:1993, *Programmable controllers – Part 3: Programming languages*

IEC 61512-1:1997, *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 part of IEC 61512, the following definitions as well as those expressed in part 1 of this International Standard apply, except where differences are explicitly stated in this part. Definitions in IEC 60050-351 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

bloc de construction

entité de recette qui existe en bibliothèque

3.3

énumération

liste de chaînes de caractères prédéfinies et leurs valeurs numériques correspondantes

3.4

table d'échange

table de base de données utilisée pour échanger les informations liées au batch entre les systèmes

3.5

lien

objet qui spécifie la connexion entre deux autres objets (par exemple la connexion entre les entités de recette ou entre les entités de recette et les transitions)

3.6

diagramme fonctionnel de procédure

représentation graphique d'une procédure de recette qui spécifie l'ordre de traitement pour les éléments procéduraux de recette

3.7

élément de recette

entité structurelle utilisée pour représenter les entités de recette et les symboles excepté les transitions et les liens orientés utilisés dans les diagrammes fonctionnels de procédures

3.8

entité de recette

combinaison d'un élément procédural avec une information de recette correspondante (par exemple en-tête, formules, prescriptions d'équipement, autres informations). Les recettes générale, de sites, maître et de contrôle sont également des entités de recette

4 Modèle de données

4.1 Introduction

Cet article définit les modèles de données qui précisent un ensemble d'objets et d'attributs ainsi que leurs relations de base en couvrant les concepts de la partie 1 de cette norme à un haut niveau d'abstraction. Les modèles s'appliquent aux interfaces avec les systèmes de contrôle batch indépendamment de la technologie. Les modèles ne sont pas conçus pour définir l'architecture interne des systèmes de contrôle batch.

L'utilisation prévue de ces modèles est de fournir un point de départ au développement de spécifications des interfaces pour des composants logiciels qui concernent tout sous-ensemble de la partie 1 de cette norme.

Les modèles abordent toute la partie 1 de cette norme en tant que modèle objet intégré, mais ils ne présument ou n'excluent aucune architecture de système ou d'échange d'information spécifique. Les modèles ne supposent aucune répartition fonctionnelle spécifique entre les systèmes.

Dans le cas où les objets et relations définis dans cet article sont présentés au travers d'une interface, alors cette dernière utilisera les noms des objets, les noms des attributs et les relations de cet article de manière compatible avec la technologie d'interface choisie et les capacités offertes. Un exemple d'une telle interface est l'interface basée sur des tables relationnelles SQL définies dans l'article 5.

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 (for example, 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 an associated recipe information (for example, 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 of this standard 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 the development of interface specifications for software components that address any subset of part 1 of this standard.

The models address all of part 1 of this standard 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.

Un format d'échange ou une spécification d'interface peut mettre en œuvre seulement certains objets ou parties d'objets (par exemple toutes les propriétés ne sont pas définies). Un format d'échange ou une spécification d'interface peut également fournir des objets ou des propriétés supplémentaires (par exemple une information de durée de la phase), y compris le développement de tout attribut de modèle de données en attributs multiples. Il est recommandé que de telles applications soient cohérentes avec le modèle de données présenté ici et avec les concepts de la partie 1 de cette norme.

Les modèles décrits dans cet article sont basés sur le Langage de Modélisation Unifié (UML) (voir article A.1).

Les tables décrivent seulement les attributs de classe des objets. Les relations entre objets sont décrites dans les figures.

4.2 Vue d'ensemble du modèle

Ce modèle (voir figure 1) donne une vue d'ensemble de haut niveau des classes principales qui sont définies et des relations entre ces classes pour le domaine du batch décrit par le modèle des activités de contrôle de la partie 1 de cette norme. Les classes d'objets individuelles sont décrites plus en détail dans les modèles suivants dans ce paragraphe.

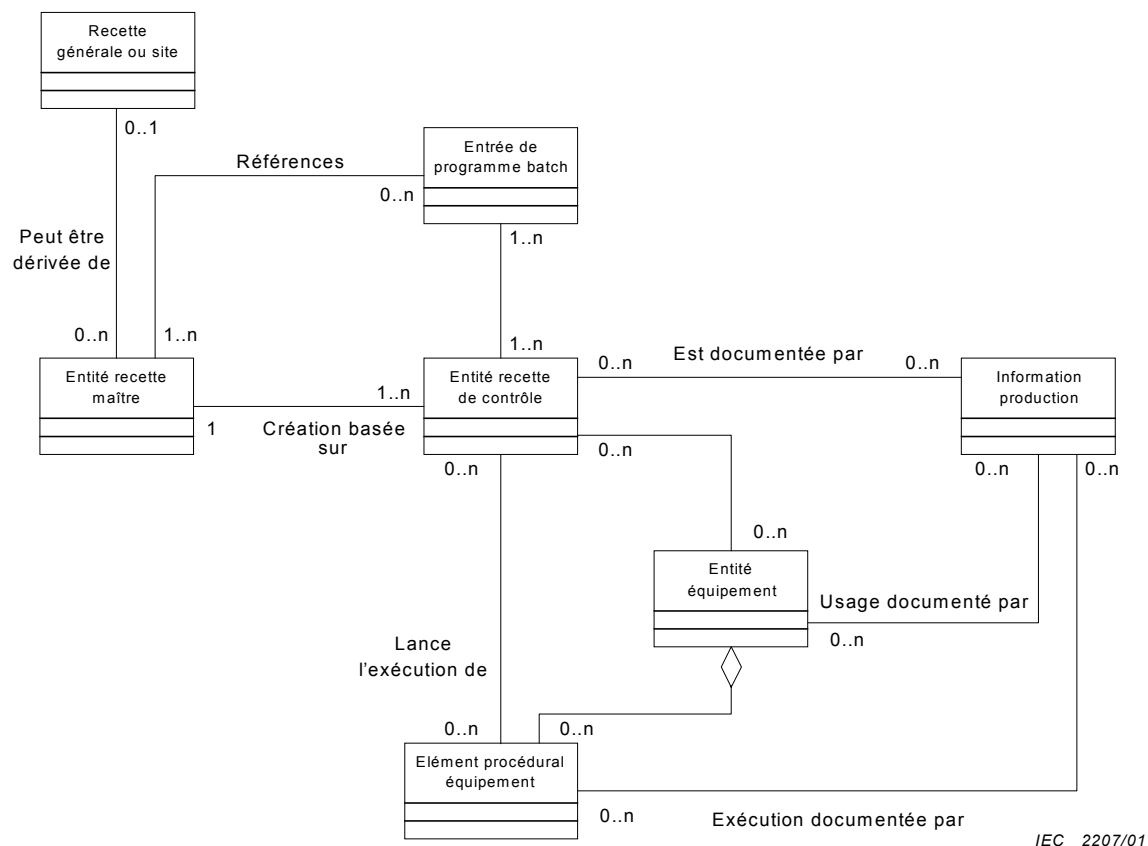


Figure 1 – Vue d'ensemble du modèle

Une recette générale ou de site est composée d'une hiérarchie d'entités de recette générale qui correspondent aux entités procédurales (stades de processus, opérations de processus et actions de processus).

Une recette maître peut être dérivée d'une recette générale ou de site. La recette maître elle-même peut être considérée comme une entité de recette maître de niveau supérieur. Une recette maître est composée d'une hiérarchie d'entités de recette maître qui correspondent aux entités procédurales (c'est-à-dire procédures, procédures d'unité, opérations, phases).

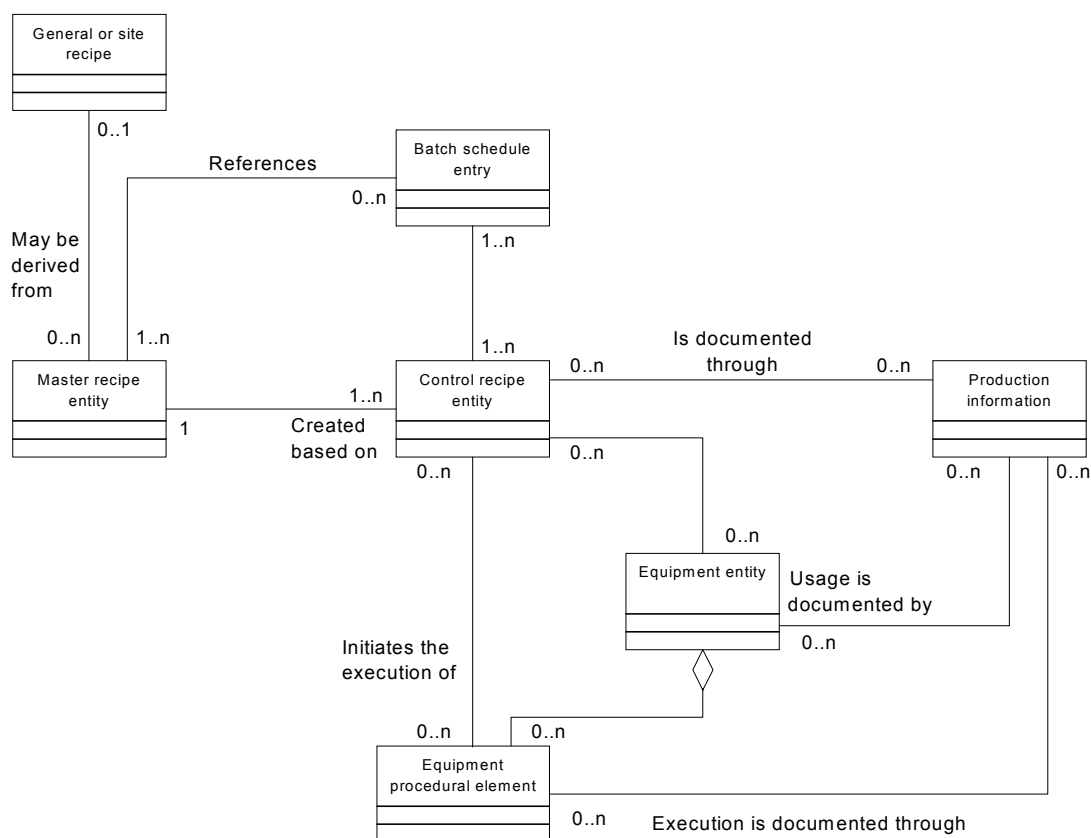
An exchange format or interface specification may implement only some objects or parts of objects (for example, not all properties are defined). An exchange format or interface specification may also provide additional objects or properties (for example, phase duration information), including the expansion 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 of this standard.

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 of this standard. The individual object classes are more thoroughly described in the subsequent models in this subclause.



IEC 2207/01

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 a 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).

Une *entrée de programme de batch* spécifie la production d'un batch spécifique à travers l'exécution d'une recette. Un programme de batch est, en principe, une liste qui spécifie la production de batchs; elle inclut les informations sur la chronologie. L'information spécifique au produit nécessaire à cet effet provient d'une *entité de recette maître* en relation.

Basée sur une *entrée de programme de batch*, une recette de contrôle débute comme la copie d'une version spécifique d'une recette maître; elle est ensuite modifiée pour créer la recette qui produira le batch. Une recette de contrôle inclut les informations nécessaires pour le contrôle de l'équipement.

La création des *entités de recette de contrôle* est basée sur les *entités de recette maître*. Une recette de contrôle peut être complétée par des informations supplémentaires (par exemple mise à l'échelle, assignation des équipements), et elle peut être modifiée (y compris la création et la suppression d'entités de recette de contrôle).

Les *entités d'équipement* sont sélectionnées et allouées aux *entités de recette de contrôle*.

Une *entité de recette de contrôle* peut être liée à une *entité procédurale d'équipement* au sein de l'*entité d'équipement* (normalement l'unité). L'*entité procédurale d'équipement* peut être initialisée, et ses paramètres peuvent être des valeurs assignées par la recette.

L'*information de production* est générée au cours de la production du batch. Cette information peut être liée aux entités de recette, aux entités d'équipement et/ou aux éléments procéduraux d'équipement.

4.3 Modèle de recette

4.3.1 Entité de recette

Les recettes sont organisées hiérarchiquement, avec diverses catégories d'informations à chaque niveau. L'entité de recette est la construction utilisée pour représenter l'association intime des données à un niveau particulier.

L'entité de recette est la structure fondamentale dans tous les types de recettes (voir figure 2). L'entité de recette, du point de vue de la structure, tient lieu d'élément procédural de recette comme défini dans la partie 1 de cette norme, mais elle peut comprendre un ou tous les composants de la recette: définitions procédurales, paramètres avec valeurs, prescriptions d'équipements et autres informations.

Les spécifications de classe sont montrées dans la table 1.

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 (for example, 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.

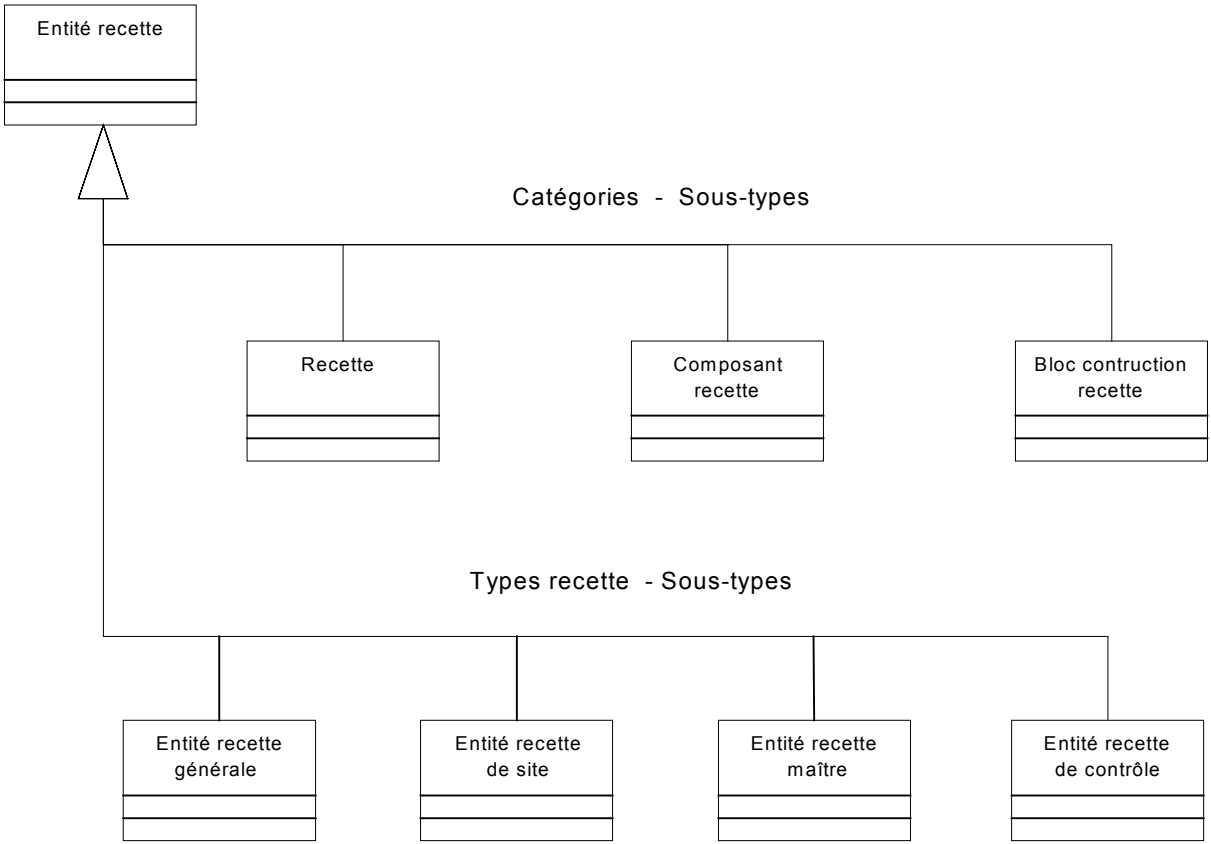
4.3 Recipe model

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 of this standard, 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.



IEC 2208/01

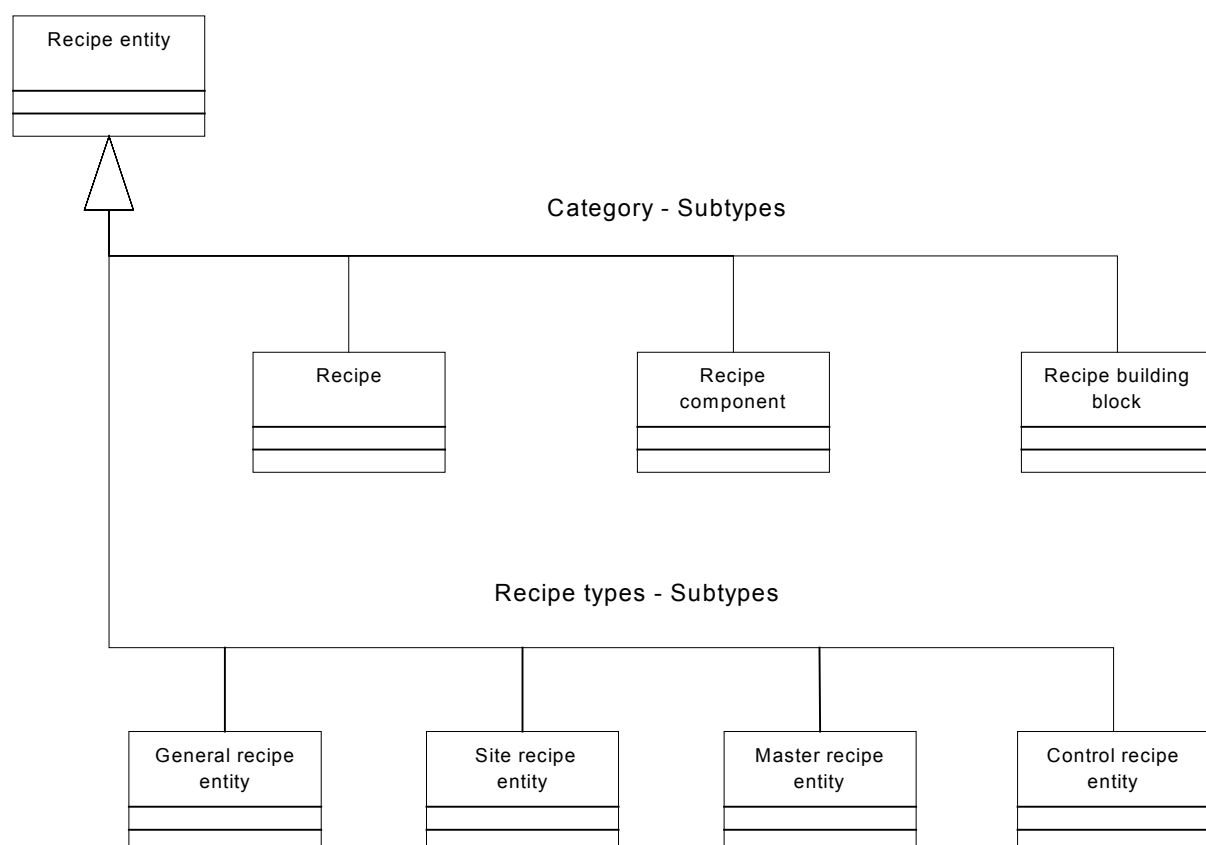
Figure 2 – Entités recette

Table 1 – Entité recette

NOM	ENTITE RECETTE
FunctionalDescription	Une entité de recette est la combinaison d'un élément procédural et de l'information de recette correspondante (par exemple en-tête, formule, prescriptions d'équipement et autres informations). Les recettes générale, de site, maître et de contrôle sont des entités de recette. Un exemple est une recette d'unité (CEI 61512-1, 5.3.2). «Recette d'unité: l'élément d'une recette de contrôle qui définit de façon unique les prescriptions de production contiguës pour une unité» (voir la CEI 61512-1, 3.62).
ATTRIBUTS	
RecipeEntityID	Fournit une identification unique

Une recette elle-même est considérée comme une entité de recette (catégorie: recette). Elle est composée d'entités de recette d'un niveau inférieur (par exemple recettes d'unité) (catégorie: composant). Lors de la construction d'une recette spécifique, les composants peuvent être créés à partir d'éléments de bibliothèque (catégorie: bloc de construction).

Le concept d'entité de recette est appliqué à tous les types de recettes: générale, de site, maître et de contrôle. Lorsqu'une recette est exécutée, la représentation dans l'historique du batch des entités de recette exécutées aura le plus souvent la même structure, et est par conséquent désignée comme une sous-classe. Une vue d'ensemble des sous-classes est montrée dans la table 2. Les catégories de sous-classes sont montrées dans les tables 3 à 5. Les types de sous-classes sont montrés dans les tables 6 à 9. Les recettes générale et de site ne sont pas davantage traitées dans ce paragraphe.



IEC 2208/01

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 (IEC 61512-1, 5.3.2). "Unit recipe: The part of a control recipe that uniquely defines the contiguous production requirements for a unit" (IEC 61512-1, 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 (for example, 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. Subclass categories are shown in tables 3 to 5. Subclass types are shown in tables 6 to 9. General and site recipes are not discussed further in this subclause.

Table 2 – Sous-classes – vue d'ensemble

	ENTITE DE RECETTE GENERALE	ENTITE DE RECETTE DE SITE	ENTITE DE RECETTE MAITRE	ENTITE DE RECETTE DE CONTROLE
RECETTE	Une recette générale complète et auto-contenue	Une recette de site complète et auto-contenue	Une recette maître complète et auto-contenue	Une recette de contrôle complète et auto-contenue
BLOC DE CONSTRUCTION DE RECETTE	Un type générique d'entité de recette générale qui peut être instancié dans une recette spécifique ou un autre bloc de construction	<i>Les blocs de construction pour recettes de sites peuvent ne pas exister. Les recettes de sites sont normalement modifiées par l'utilisation de blocs de construction de recette générale</i>	Un type générique d'entité de recette maître qui peut être instancié dans une recette spécifique ou un autre bloc de construction	<i>Les blocs de construction pour les recettes de contrôle n'existent pas. Les recettes de contrôle sont modifiées par l'utilisation de blocs de construction de recette maître</i>
COMPOSANT DE RECETTE	Un composant d'une recette générale ou d'un élément de bibliothèque – peut être une instanciation d'un bloc de construction	Un composant d'une recette de site ou d'un élément de bibliothèque – peut être une instanciation d'un bloc de construction d'une recette générale	Un composant d'une recette maître ou d'un élément de bibliothèque – peut être une instanciation d'un bloc de construction	Un composant d'une recette de contrôle – peut être une instanciation d'un bloc de construction d'une recette maître

Table 3 – Recette

NOM	RECETTE
FunctionalDescription	L'entité de recette du niveau le plus élevé
ATTRIBUTS	
RecipeID	Identifie la recette. Lorsqu'elle est combinée avec la «RecipeVersion», définit une instance unique d'une recette
RecipeVersion	Identifie la version d'une recette. Lorsqu'elle est combinée avec une «RecipeID», définit une instance unique d'une recette (par exemple Red Oak – A10.3)
VersionDate*	Identifie la date et l'heure auxquelles cette version de la recette a été créée ou modifiée
ApprovalDate*	Identifie la date et l'heure auxquelles cette version de la recette a été approuvée
EffectiveDate*	Identifie quand cette version de la recette est susceptible d'être utilisée au plus tôt
ExpirationDate*	Identifie la date et l'heure auxquelles cette version de la recette expire
ProductID*	Identifie le produit ou la famille de produits qui serait crée(e) par l'exécution de cette version de la recette (par exemple Premium Beer)
Author*	Identifie la personne ou le système qui est l'auteur de cette version de la recette (par exemple J. Smith)
ApprovedBy*	Identifie la personne ou le système qui a approuvé cette version de la recette
Description	Décrit cette version de la recette et / ou le produit (par exemple <i>Finest Premium Beer</i>)
Status*	Définit le statut de l'information (par exemple «approuvée pour production», «approuvée pour tests», «non approuvée», «inactive», «obsolète»)
*Non requis pour la recette de contrôle (disponible par référence à la recette maître).	

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	<i>Building blocks for site recipes may not exist. Site recipes are normally modified using general recipe building blocks</i>	A generic master recipe entity type that can be instantiated in a specific recipe or in another building block	<i>Building blocks for control recipes do not exist. Control recipes are modified using master recipe building blocks</i>
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., <i>Finest Premium Beer</i>)
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 – Composant de recette

NOM	COMPOSANT DE RECETTE
FunctionalDescription	Une entité de recette faisant partie d'une recette ou d'un bloc de construction (c'est-à-dire une instance d'un bloc de construction dans une recette particulière ou contenant une entité de bloc de construction de recette)
ATTRIBUTS	
Level	Indique le niveau de hiérarchie procédural (c'est-à-dire stade du processus, opération de processus ou action processus pour les recettes générale ou de site et la procédure d'unité, opération ou phase pour les recettes maître et de contrôle)
RE_Use	Définit si le composant de la recette est une copie d'un bloc de construction ou une référence à ce bloc

Table 5 – Bloc de construction de recette

NOM	BLOC DE CONSTRUCTION DE RECETTE
FunctionalDescription	Une entité de recette qui existe en bibliothèque. Un bloc de construction peut être paramétré et utilisé lors de la construction des recettes
ATTRIBUTS	
RecipeVersion	Identifie la version de l'entité de recette
VersionDate	Identifie la date et l'heure auxquelles cette version de recette a été créée ou modifiée
ApprovalDate	Identifie la date et l'heure auxquelles cette version de recette a été approuvée
Author	Identifie la personne ou le système qui est l'auteur de cette version de la recette (par exemple J. Smith)
ApprovedBy	Identifie la personne ou le système qui a approuvé cette version de la recette
Description	Décrit la fonction réalisée à travers l'exécution de cette version de l'entité de recette
Level	Indique le niveau de l'entité de recette
UsageConstraint	Définit les autres règles qui déterminent l'utilisation (par exemple toujours suivi de..., ou n'est jamais exécutée en parallèle avec....)
Status	Définit le statut de l'entité de recette (par exemple «approuvée pour production», «approuvée pour tests», «non approuvée», «inactive», «obsolète»)
Function	Définit comment l'entité de recette est exécutée (par exemple en référençant un élément procédural d'équipement (EPE), à travers l'exécution de la logique contenue)

Table 6 – Entité de recette générale

NOM	ENTITE DE RECETTE GENERALE
FunctionalDescription	Toute une recette générale ou de site, un composant d'une recette générale ou de site ou un bloc de construction pour la création d'une recette générale ou de site
ATTRIBUTS	
ScaleReference	Définit une échelle de référence pour les valeurs de paramètres

Table 7 – Entité de recette de site

NOM	ENTITE DE RECETTE SITE
FunctionalDescription	Toute une recette de site, un composant d'une recette de site ou un bloc de construction pour la création d'une recette de site
ATTRIBUTS	
ScaleReference	Définit une échelle de référence pour les valeurs de paramètres

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 a 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	RECIPE 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 the 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 the 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 the 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 the creation of a site recipe
ATTRIBUTES	
ScaleReference	Defines a reference scale for the parameter values

Table 8 – Entité de recette maître

NOM	ENTITE DE RECETTE MAITRE
FunctionalDescription	Toute une recette maître, un composant d'une recette maître ou un bloc de construction pour la création d'une recette maître
ATTRIBUTS	
ScaleReference	Définit une échelle de référence pour les valeurs de paramètres
ProcessCellID	Identifie la catégorie d'équipement pour laquelle l'entité de recette a été définie (par exemple cellule de processus ou cellules de processus pour lesquelles la recette maître a été définie)

Table 9 – Entité de recette de contrôle

NOM	ENTITE DE RECETTE DE CONTROLE
FunctionalDescription	Une entité de recette qui est tout ou partie d'une recette de contrôle
ATTRIBUTS	
BatchID	Spécifie l'ID du batch réel
BatchSize	Définit la taille requise, ou le facteur d'échelle, pour le batch, basée sur le facteur d'échelle pour le batch comme défini dans la recette maître
Status	Décrit le statut de l'exécution (par exemple pas encore activée, active ou achevée)

4.3.2 Éléments des entités de recette

Le modèle (voir figure 3 et tables 10 à 13) montre les catégories d'informations d'une recette comme spécifié dans la partie 1 de cette norme. Le modèle indique que ces composants peuvent exister à tout niveau de décomposition de la recette (par exemple une recette d'unité peut avoir ses propres prescriptions d'équipement).

- La catégorie d'information d'en-tête est contenue dans des attributs attachés à l'entité de recette elle-même, au lieu d'être une classe d'objet distincte dans ce modèle.
- La catégorie formule de la partie 1 de cette norme est modélisée comme un ensemble d'objets paramètre. Tous les niveaux de décomposition d'une recette peuvent avoir des paramètres, y compris la recette elle-même. Voir 4.3.6 pour les paramètres.
- La modélisation des prescriptions d'équipement est présentée en 4.3.5.
- La catégorie autres informations, comme défini dans la partie 1 de cette norme, est représentée en tant que classe d'objet unique, même si ces autres informations peuvent avoir de multiples éléments et des structures différentes.
- La catégorie procédure de la partie 1 de cette norme est modélisée comme un ensemble d'éléments structurels procéduraux.

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 the 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)

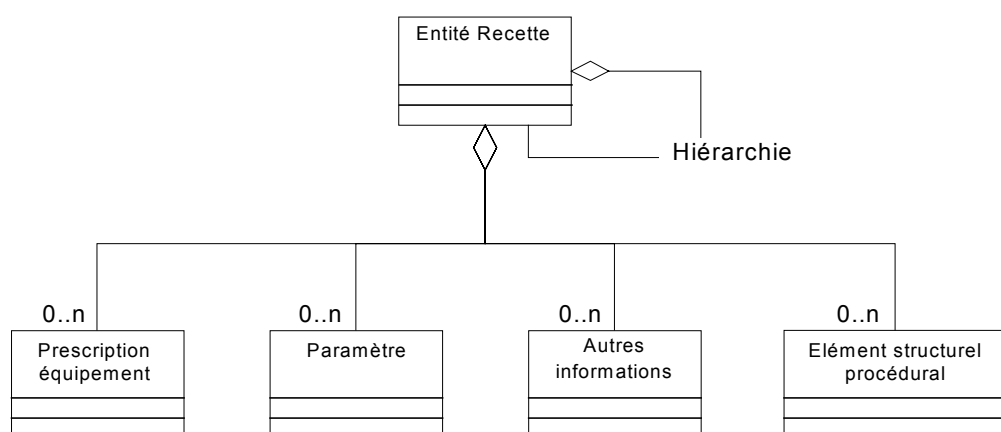
Table 9 – Control recipe entity

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 the 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 to 13) shows the categories of information of a recipe as specified in part 1 of this standard. The model indicates that these components may exist at any level of the decomposition of the recipe (for example, a unit recipe may have its own equipment requirements).

- The header category of information is contained in attributes to the recipe entity itself, instead of being a distinct object class in this model.
- The formula category of part 1 of this standard is modelled 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.
- The modeling of equipment requirements is discussed in 4.3.5.
- The other information category, as defined in part 1 of this standard, is represented as a single object class, even though other information may have multiple elements and different structures.
- The procedure category of part 1 of this standard is modeled as a set of procedural structural elements.



IEC 2209/01

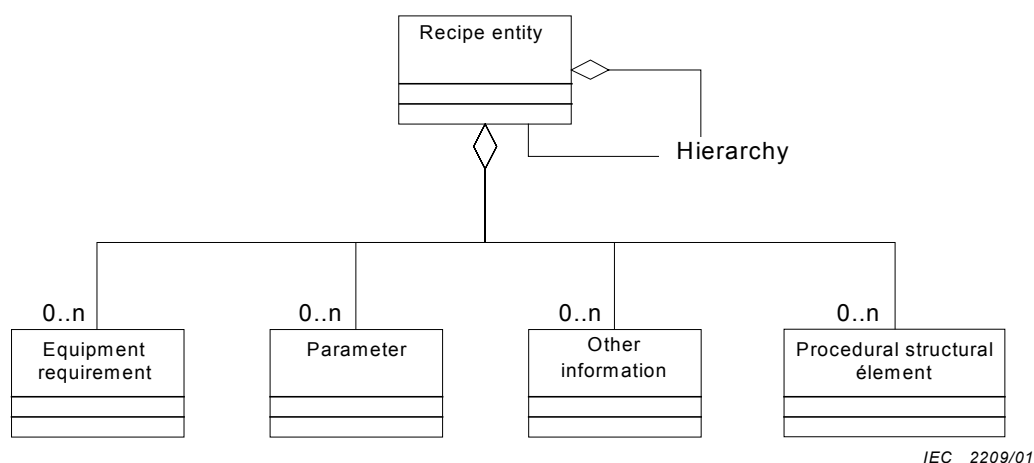
Figure 3 – Éléments de l'entité de recette

Table 10 – Paramètre

NOM	PARAMETRE
FunctionalDescription	Les valeurs de formule ou les emplacements pour des valeurs qui doivent être communiquées vers et depuis des entités de recette durant l'exécution
ATTRIBUTS	
ParameterID	Fournit une identification unique
ParameterType	Spécifie la façon dont la valeur du paramètre doit être interprétée (par exemple constante, équation de référence)
Description	Décrit le paramètre ou l'utilisation du paramètre
EngineeringUnits	Identifie les unités de mesure d'ingénierie pour la valeur (par exemple kg, livres)
Value	Contient la valeur de paramètre. Si la valeur est une relation, elle contient l'équation, la forme, la règle de renvoi ou tout ce qui relie les paramètres entre eux. Si la valeur est un paramètre de bloc de construction, cet attribut contiendra la valeur par défaut
Scaled	Spécifie la règle de mise à l'échelle. Cas le plus simple: ajusté ou non ajusté d'après la taille de référence du batch
Usage	Spécifie le paramètre comme entrée de processus, sortie de processus ou paramètre de processus

Table 11 – Prescription d'équipement

NOM	PRESCRIPTION D'EQUIPEMENT
FunctionalDescription	Représente une prescription d'équipement comme spécifié dans l'entité de recette
ATTRIBUTS	

**Figure 3 – Parts of recipe entities****Table 10 – Parameter**

NAME	PARAMETER
FunctionalDescription	Formula values or placeholders for values that are to be communicated to and from recipe entities during execution
ATTRIBUTES	
ParameterID	Provides a 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 the value is a relation, it contains the equation, form, deferral rule or whatever ties the related parameters together. If the 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 – Autres informations

NOM	AUTRES INFORMATIONS
FunctionalDescription	Une catégorie d'informations de recette qui peut contenir l'information de support de traitement du batch qui n'est pas contenue dans d'autres parties de la recette (par exemple l'information de conformité avec la réglementation, les informations pour la sécurité vis-à-vis du processus et des matières, les diagrammes de flux de processus, les informations sur le conditionnement et l'étiquetage)
ATTRIBUTS	

Table 13 – Élément structurel procédural

NOM	ELEMENT STRUCTUREL PROCEDURAL
FunctionalDescription	Les éléments procéduraux de recette et les informations pour ordonnancer leur exécution
ATTRIBUTS	

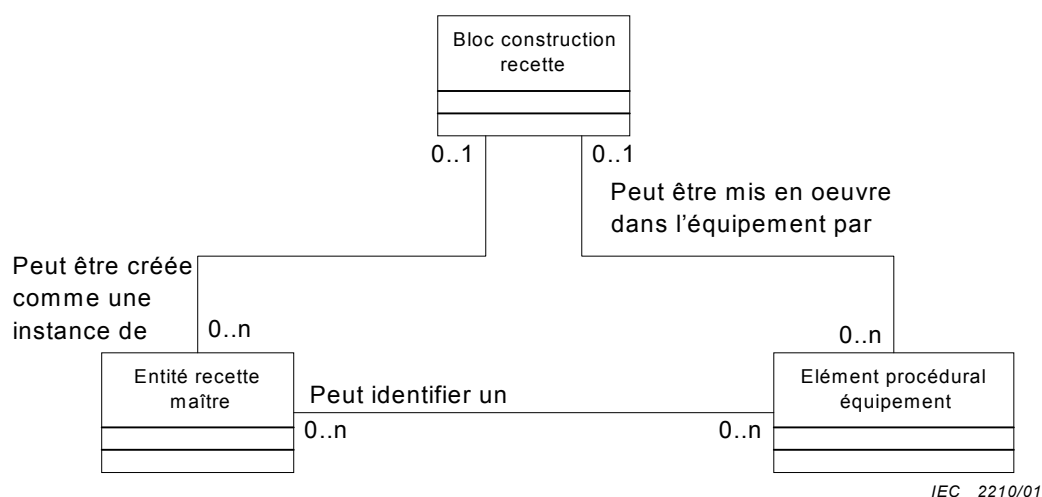
4.3.3 Relations entre entités de recette (structure procédurale)

Les entités de recette sont décomposées hiérarchiquement suivant les structures pour les entités procédurales comme défini dans la partie 1 de cette norme (c'est-à-dire une procédure de recette contient des procédures d'unité qui contiennent des opérations qui contiennent des phases). Cette hiérarchie est modélisée par une récursion de contenant. Des objets de niveau supérieur peuvent contenir des objets de niveau inférieur.

Les éléments structurels procéduraux comprennent les éléments procéduraux de recette et les connexions (par exemple lien, transition) qui sont utilisés pour les ordonner (par exemple les éléments structurels procéduraux d'une recette d'unité sont les opérations et l'ordonnancement de ces opérations qui y sont contenues). Les éléments structurels procéduraux sont susceptibles d'être liés à d'autres éléments structurels procéduraux.

4.3.4 Blocs de construction de recette

Les blocs de construction d'une recette sont un concept important dans le modèle de données (voir figure 4). Cette figure représente les relations à un seul niveau de la hiérarchie de la procédure.



IEC 2210/01

Figure 4 – Blocs de construction de recette

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	

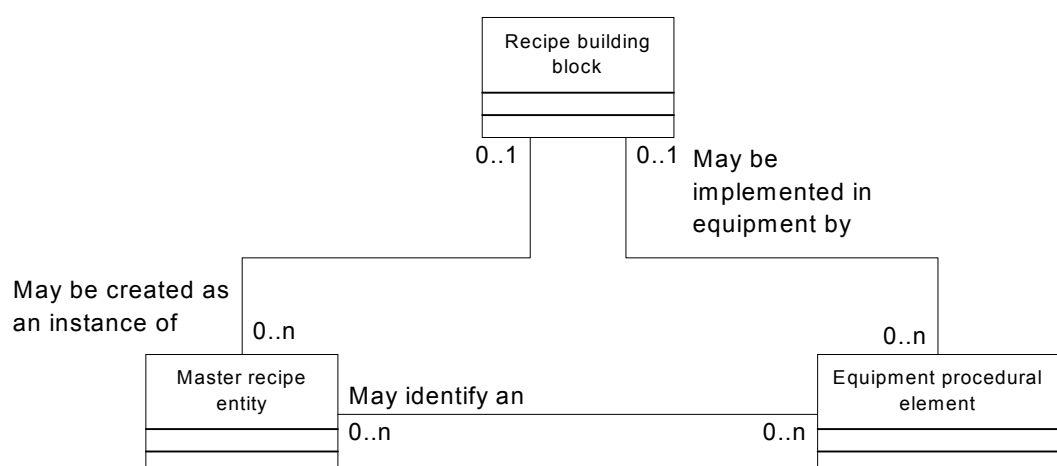
4.3.3 Recipe entity relationships (procedural structure)

Recipe entities are hierarchically decomposed along the structures for procedural entities as defined in part 1 of this standard (i.e., a recipe procedure contains unit procedures that contain operations that contain phases). This hierarchy is modelled using recursive containment. Higher-level objects may contain lower-level objects.

Procedural structural elements include the recipe procedural elements and the connections (for example, link, transition) that are used to order them (for example, 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.



IEC 2210/01

Figure 4 – Recipe building block

Les *blocs de construction de recette* sont les blocs de construction à partir desquels les recettes maîtres sont créées. Lorsqu'un *bloc de construction de recette* est instancié dans une recette maître comme *entité de recette maître*, il peut porter les paramètres, les prescriptions d'équipement et les autres informations qui peuvent être des valeurs spécifiques assignées à la recette maître. Les contenus de niveau inférieur du *bloc de construction de recette* (par exemple entités de recette subordonnées) peuvent être copiés dans les *entités de recette maître*. Ces mêmes contenus de niveau inférieur peuvent également être accessibles par référence au *bloc de construction de recette*.

Une fonctionnalité de *bloc de construction d'une recette* peut être mise en œuvre dans l'équipement à travers les *éléments procéduraux d'équipement* (voir table 14), ce qui est nécessaire pour l'exécution des entités de recette du niveau le moins élevé (c'est-à-dire les entités de recette qui sont destinées à être liées aux éléments procéduraux d'équipement).

Table 14 – Élément procédural d'équipement

NOM	ELEMENT PROCEDURAL D'EQUIPEMENT
FunctionalDescription	Un élément procédural associé à un élément d'équipement (par exemple une phase d'équipement ou une opération d'équipement)
ATTRIBUTS	
EquipmentProceduralElementID	Fournit une identification unique
Version	Identifie la version de l'élément procédural
VersionDate	Identifie la date et l'heure auxquelles cette version a été créée ou modifiée
ApprovalDate	Identifie la date et l'heure auxquelles cette version a été approuvée
Author	Identifie la personne ou le système auteur de cette version (par exemple J. Smith)
ApprovedBy	Identifie la personne ou le système qui a approuvé cette version
Description	Décrit la fonction exécutée à travers l'exécution de l'entité de recette
Level	Indique le niveau de l'entité d'équipement. L'entité d'équipement est susceptible d'être utilisée seulement à ce niveau
Mode	Indique le mode courant de l'élément procédural
State	Indique l'état courant de l'élément procédural

Les mécanismes peuvent être illustrés par l'exemple suivant (voir figure 5) qui pourrait représenter un modèle objet d'une application réelle partielle. Le concept générique du bloc de construction est instancié comme un bloc de construction spécifique «chauffe». La relation «est basé sur» entre blocs de construction et composants est remplacée par une relation de sous-classe (ceci est une des applications possibles, et elle indique que si le bloc de construction «chauffe» est changé, alors ce changement s'appliquera à toutes les recettes qui utilisent «chauffe»). Une autre application pourrait être que le bloc «chauffe» est simplement copié lorsqu'il est instancié.

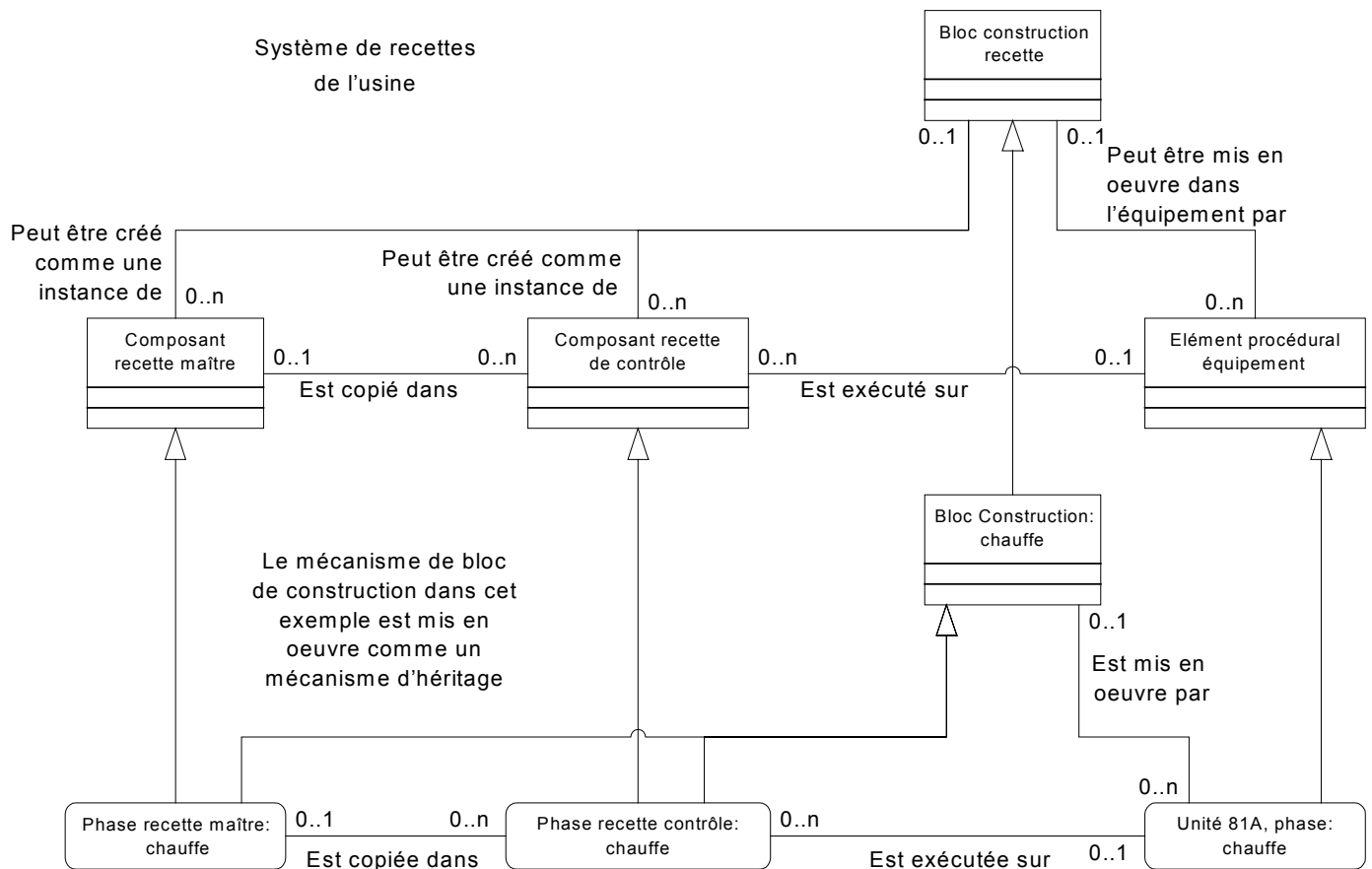
Recipe building blocks are the building blocks from which master recipes are created. When a *recipe building block* is instantiated in a master recipe as a *master recipe entity*, it may carry parameters, equipment requirements and other information that may be assigned master recipe specific values. Lower-level contents of the *recipe building block* (for example, 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 *recipe building block's* functionality may be implemented in equipment through *equipment 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	
EquipmentProceduralElementID	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 the 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 the building blocks and the 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.



IEC 2211/01

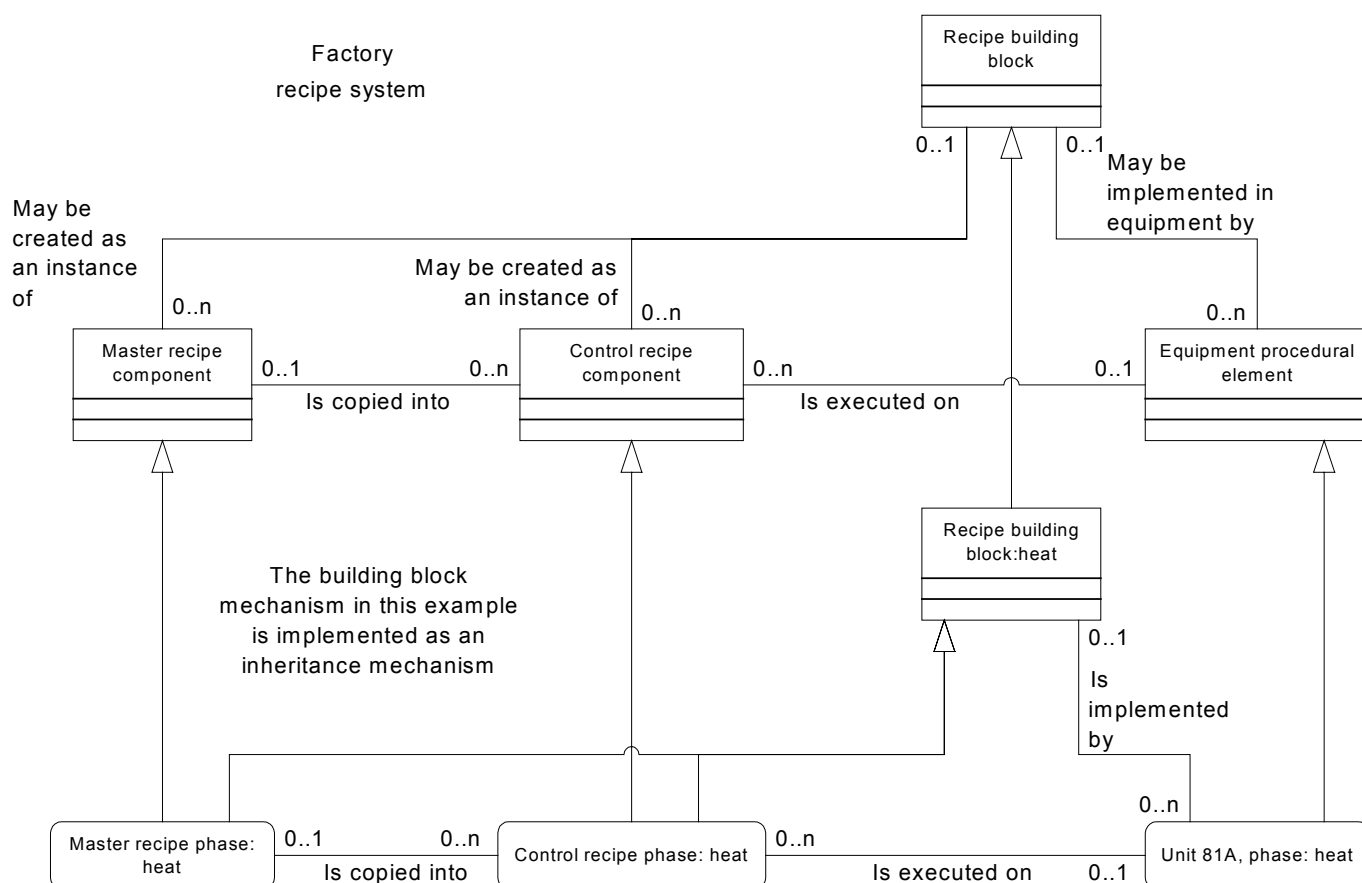
Figure 5 – Principe des blocs de construction

4.3.5 Prescriptions d'équipement

Les *entités de recette* peuvent contenir des *prescriptions d'équipement* (voir figure 6 et tables 15, 16 et 17). Les prescriptions d'équipement référencent un *type de propriété d'équipement* spécifique (par exemple un *type de propriété d'équipement* peut être la «taille de la cuve» ou le «revêtement de la cuve»). Une prescription d'équipement spécifique pourrait alors spécifier une valeur minimale pour la taille d'une cuve.

Cette prescription peut alors être satisfaite par un élément d'équipement possédant une certaine propriété d'équipement qui référence le même type de propriété d'équipement. Par exemple, une unité spécifique UNIT12 aurait une valeur pour le type de propriété «taille de la cuve».

Une entité d'équipement est un élément d'équipement spécifique, et elle est susceptible d'être remplacée par une classe d'équipement (classe d'équipements). Voir 4.4.



IEC 2211/01

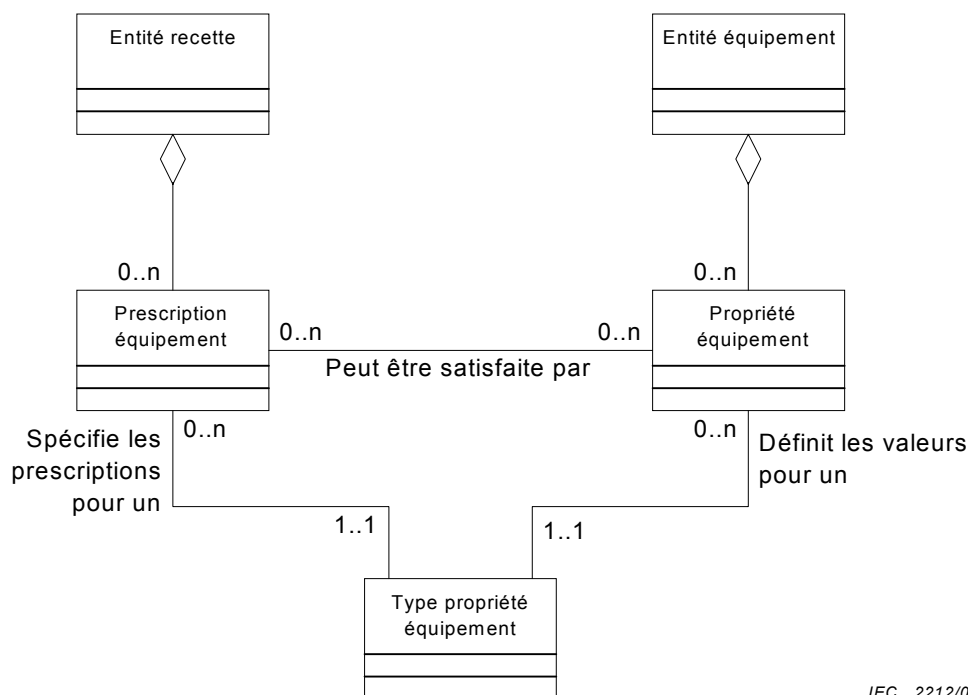
Figure 5 – Building block concept

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* (for example, 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.



IEC 2212/01

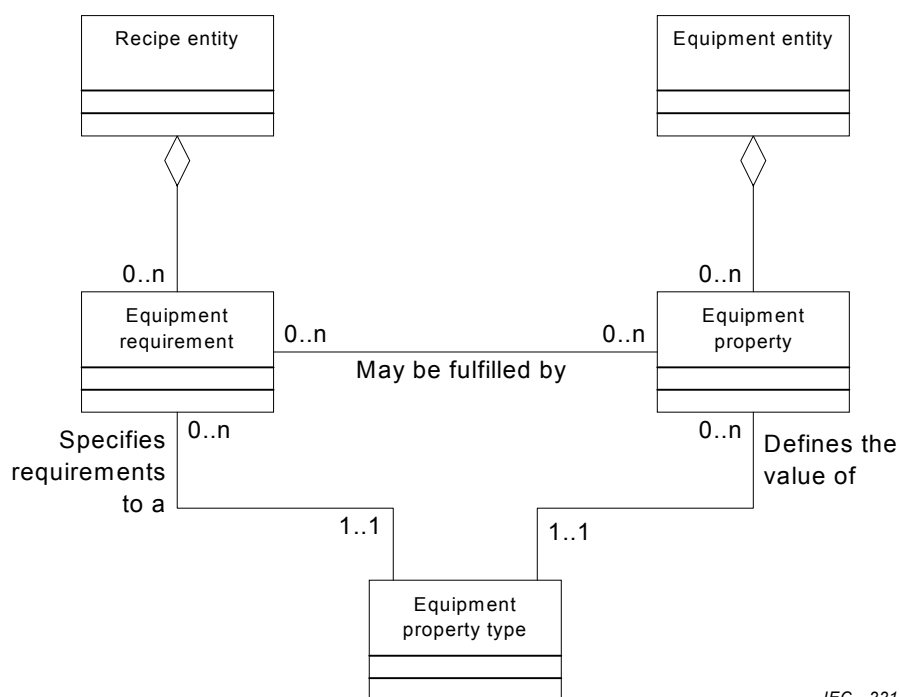
Figure 6 – Prescriptions d'équipement de l'entité recette

Table 15 – Entité d'équipement

NOM	ENTITE D'EQUIPEMENT
FunctionalDescription	L'ensemble formé par l'équipement physique de traitement et de contrôle ainsi que le contrôle d'équipement groupés pour accomplir une certaine fonction de contrôle ou un ensemble de fonctions de contrôle
ATTRIBUTS	
EquipmentEntityID	Fournit une identification unique
EquipmentLevel	Spécifie le niveau de hiérarchie physique (par exemple cellule de processus, unité, module d'équipement, module de contrôle)
Mode	Indique le mode courant de l'entité d'équipement
State	Indique l'état courant de l'entité d'équipement

Table 16 – Propriété d'équipement

NOM	PROPRIETE D'EQUIPEMENT
FunctionalDescription	Identifie une propriété que fournit l'entité ou la classe d'équipement. Ces propriétés sont spécifiques à l'application (par exemple type de revêtement, taille, capacité thermique, température de la vapeur)
ATTRIBUTS	
PropertyID	Fournit une identification unique
Value	Identifie la valeur de la propriété (par exemple verre, 50 000, 650)
ValueRange	Définit les limites ou les contraintes liées à la valeur
EngineeringUnits	Définit les unités d'ingénierie de la propriété
Description	Décrit le type de la propriété d'équipement



IEC 2212/01

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 (for example, 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, 50 000, 650)
ValueRange	Defines the limits or constraints that are related to the value
EngineeringUnits	Defines the engineering units of the property
Description	Describes the type of the equipment property

Table 17 – Type de propriété d'équipement

NOM	TYPE DE PROPRIETE D'EQUIPEMENT
FunctionalDescription	La classe générale des attributs d'équipement (par exemple type de revêtement, taille, capacité thermique, température de la vapeur)
ATTRIBUTS	

Les entités de recette de contrôle contiendront initialement les prescriptions d'équipement qui sont copiées à partir de l'entité de recette maître, et celles-ci nécessitent d'être satisfaites par la propriété correspondante d'une ou plusieurs entités d'équipement afin que l'équipement spécifique soit affecté. La prescription d'équipement initiale est susceptible d'être remplacée par des allocations d'équipement spécifiques. Ces allocations sont aussi modélisées comme des prescriptions d'équipement.

4.3.6 Paramètres de recette

Les paramètres sont des variables associées à des entités de recette. Ces variables peuvent être utilisées par des éléments procéduraux d'équipement, elles peuvent être utilisées par d'autres activités (par exemple programmation) ou être référencées par d'autres éléments de la recette (par exemple critères de transition) (voir figure 7).

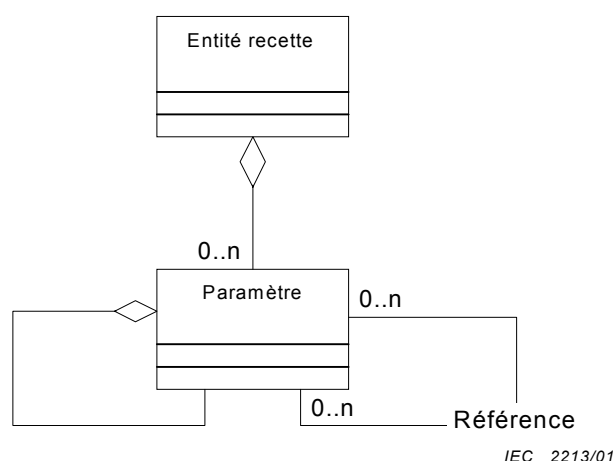


Figure 7 – Modèle du paramètre

Les paramètres peuvent être classés par catégories comme entrées de processus, sorties de processus ou paramètres de processus.

Les paramètres peuvent être composés d'un ensemble d'autres paramètres. Le modèle supporte le concept de paramètres structurés. Par conséquent, le modèle donne la possibilité d'inclure des paramètres de différents types (paramètres de processus, entrées de processus, sorties de processus) dans la même structure ainsi que de définir des structures de données d'un même type.

Les attributs de valeur de paramètre peuvent être organisés en définissant les types de valeur de paramètre. Les types de valeur de paramètre peuvent inclure:

- les types de données de base de la CEI 61131-3;
- les informations matricielles de compatibilité utilisées pour déterminer les exigences de nettoyage-en-place (NEP) ou de stérilisation-en-place (SEP);
- des ensembles de données qui définissent les transactions de matière (transfert, consommation, génération de matière);

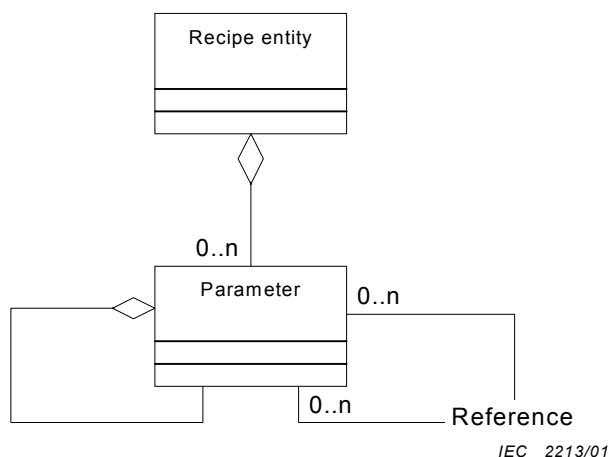
Table 17 – Equipment property type

NAME	EQUIPMENT PROPERTY TYPE
FunctionalDescription	The general class of equipment attributes (e.g., lining type, size, heat capability, steam temperature)
ATTRIBUTES	

The *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 modelled 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 (for example, scheduling) or they may be referenced by other parts of the recipe (for example, 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:

- IEC 61131-3 basic data types;
- compatibility matrix information that is used for determining clean-in-place (CIP) or sterilize-in-place (SIP) requirements;
- data sets that define material transactions (transfer, consumption, generation of material);

- des séries de données (par exemple un profil de la température qui sera suivi).

Les valeurs de paramètre peuvent être de simples valeurs, expressions ou références à des paramètres qui sont définis au même niveau ou à des niveaux supérieurs dans la hiérarchie procédurale. Les valeurs qui sont des expressions peuvent inclure des références à d'autres paramètres.

Les paramètres peuvent être liés de diverses manières, parmi lesquelles:

- équations algébriques ou booléennes;
- formulaires de saisie spécifiques au produit qui fonctionnent avec un ou plusieurs paramètres;
- procédures opératoires standards (SOPs) qui présentent ou utilisent des paramètres (par exemple valeurs dynamiques, valeurs de recette);
- renvoi de paramètres à différentes entités de recette (au même niveau ou à un niveau différent);
- applications externes qui utilisent des paramètres.

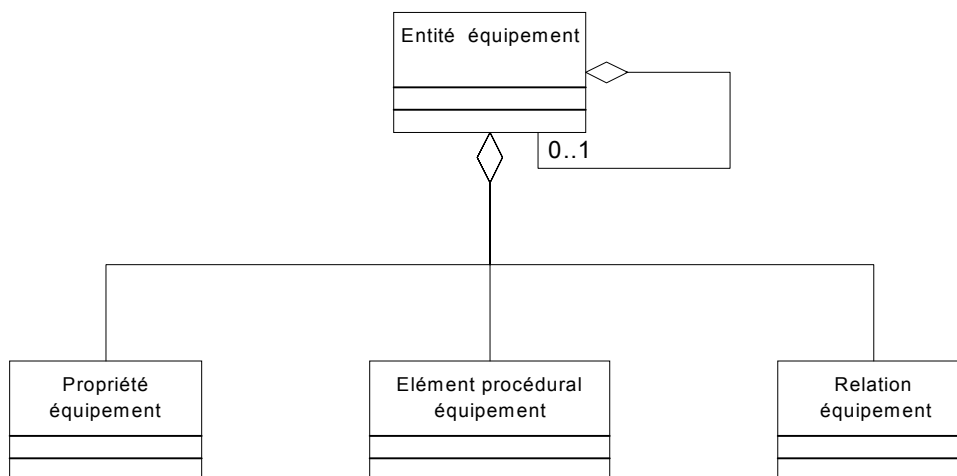
La formule est représentée dans le modèle de données en tant que paramètres de recette (voir table 10). Une formule de recette est un ensemble de paramètres sélectionnés pour la procédure de recette, et elle peut aussi inclure des paramètres qui sont définis pour des niveaux inférieurs de la hiérarchie procédurale.

Les paramètres sont souvent ajustés en se basant sur la taille du batch ou d'autres attributs clé. La mise à l'échelle peut être plus complexe qu'une simple relation linéaire. Des méthodes de mise à l'échelle plus complexes peuvent être élaborées en utilisant des relations et algorithmes définis pour l'utilisateur.

4.4 Modèle d'équipement

La structure physique de l'installation nécessite d'être prise en compte dans l'évaluation de la sélection de l'équipement lors de l'exécution de la recette (voir figure 8 et table 18). En particulier, les capacités de transfert entre équipements ou la faculté d'attribuer un équipement partagé sont importantes pour router un batch.

Les entités d'équipement sont définies conformément à la hiérarchie spécifiée dans la partie 1 de cette norme. Cette hiérarchie est modélisée à travers la nature récursive des objets. Cette construction supporte la configuration de l'extensibilité et de la réductibilité.



IEC 2214/01

Figure 8 – Structure de l'équipement

- data series (for example, 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:

- algebraic or boolean equations;
- product specific entry forms that use one or more parameters;
- standard operating procedures (SOPs) that display or otherwise utilize parameters (for example, dynamic values, recipe values);
- deferral of parameters to different recipe entities (at the same or another level);
- 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 of the recipe procedure, and it may also include parameters that are defined at lower levels of the procedural hierarchy.

The scaling of parameters is often 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 of this standard. This hierarchy is modelled through the recursive nature of the objects. This construct allows for the configuration of expandability and collapsibility.

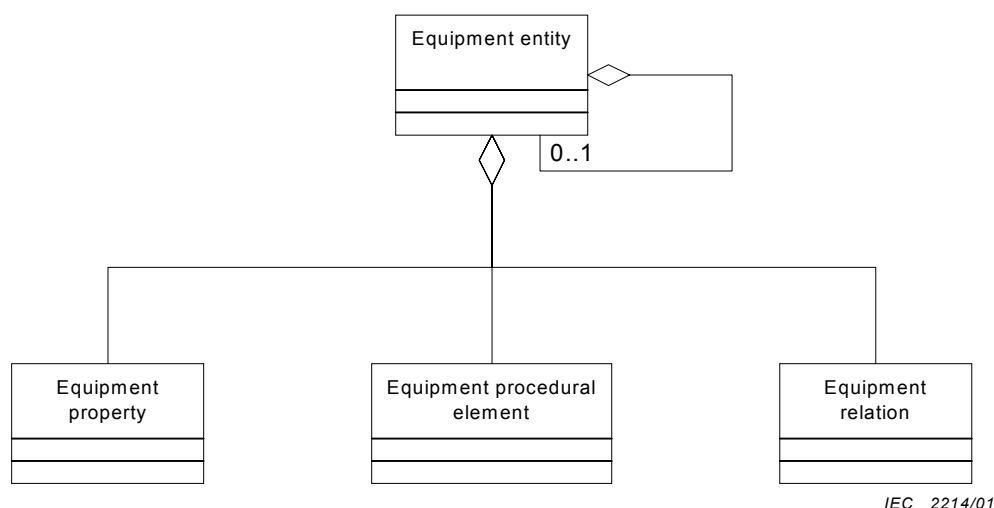


Figure 8 – Equipment structure

Table 18 – Relation d'équipement

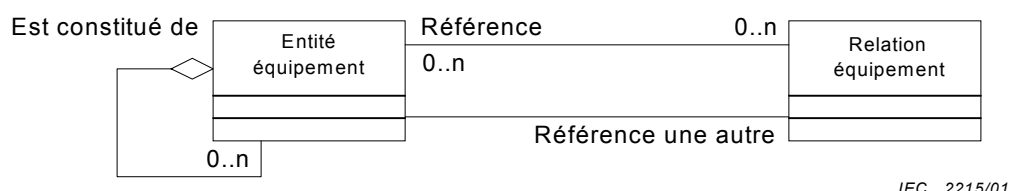
NOM	RELATION D'EQUIPEMNT
FunctionalDescription	Une représentation des connexions entre équipements (par exemple tuyaux, transporteurs, connexions flexibles), mais elle peut aussi être utilisée pour représenter d'autres sortes de relations entre équipements (par exemple relation avec un équipement à usage partagé)
ATTRIBUTS	
RelationID	Fournit une identification unique

Les équipements dans, par exemple, une cellule de processus (c'est-à-dire unités, modules d'équipement, modules de contrôle) sont normalement liés les uns aux autres par des tuyauteries ou d'autres connexions. Les connexions peuvent être modélisées comme des relations d'équipements (voir figure 9), et être optionnellement orientées (par exemple une direction de flux). Les relations (par exemple tuyaux) font partie de l'entité d'équipement de niveau supérieur. Les connexions peuvent être classées d'une manière commode par catégories dans les classes de relation, et ceci assure une évaluation cohérente. Les relations d'équipement comprennent les éléments suivants:

- connexion permanente;
- connexion temporaire;
- peut être utilisé comme ressource pour;
- traite toujours le même produit que.

Noter que des relations autres que les connexions peuvent être possibles.

L'équipement peut comporter des propriétés. Les propriétés sont spécifiques à chaque mise en œuvre par l'utilisateur. Les propriétés d'équipement peuvent être utilisées pour examiner les caractéristiques d'équipement et pour répondre aux prescriptions d'équipement d'une recette. Voir 4.3.5.



IEC 2215/01

Figure 9 – Relations entre entités d'équipement

Les classes d'équipement (voir figure 10 et table 19) fournissent un moyen de grouper les entités d'équipement par caractéristiques communes. Les entités d'équipement peuvent être membres d'une ou de plusieurs classes d'équipement, ou elles peuvent n'appartenir à aucune classe. Les classes d'équipement peuvent être utilisées pour spécifier des groupes d'unités, et elles peuvent être utilisées alternativement lors de la sélection d'équipements. Par exemple, si une recette nécessite un réacteur pour une procédure d'unité, ses prescriptions d'équipement pourront prescrire un réacteur spécifique (par exemple R-101), un ensemble d'unités de type réacteur (par exemple R-101, R-103) ou la classe réacteur (par exemple la classe «réacteur» qui contient les réacteurs R-101, R-102 et R-103).

Les entités d'équipement peuvent être membres d'une *classe d'équipement*, et la classe détermine certaines des propriétés des membres de la classe. Par exemple, certaines propriétés d'équipement (par exemple revêtement en verre) sont partagées avec la classe.

Table 18 – Equipment relation

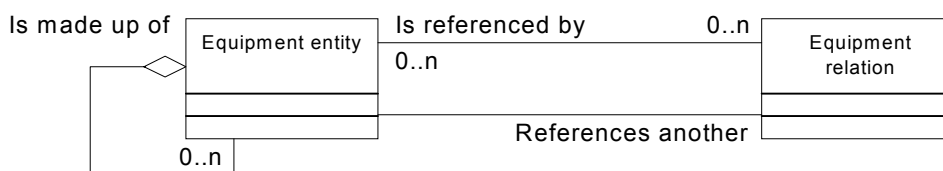
NAME	EQUIPMENT RELATION
FunctionalDescription	A representation of the connections between equipment (e.g., pipes, conveyors, flexible connections), but may also be used to represent other kinds of relations between equipment (e.g., relation to shared-use equipment)
ATTRIBUTES	
RelationID	Provides unique identification

The equipment in, for example, a process cell (i.e., units, equipment modules, control modules) is normally interrelated by pipes or other connections. The connections may be modelled as equipment relations (see figure 9), optionally with a direction (for example, a flow direction). The relations (for example, 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:

- permanent connection;
- temporary connection;
- may be used as resource for;
- always runs the same product as.

Note that relations other than connections may be possible.

The 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.



IEC 2215/01

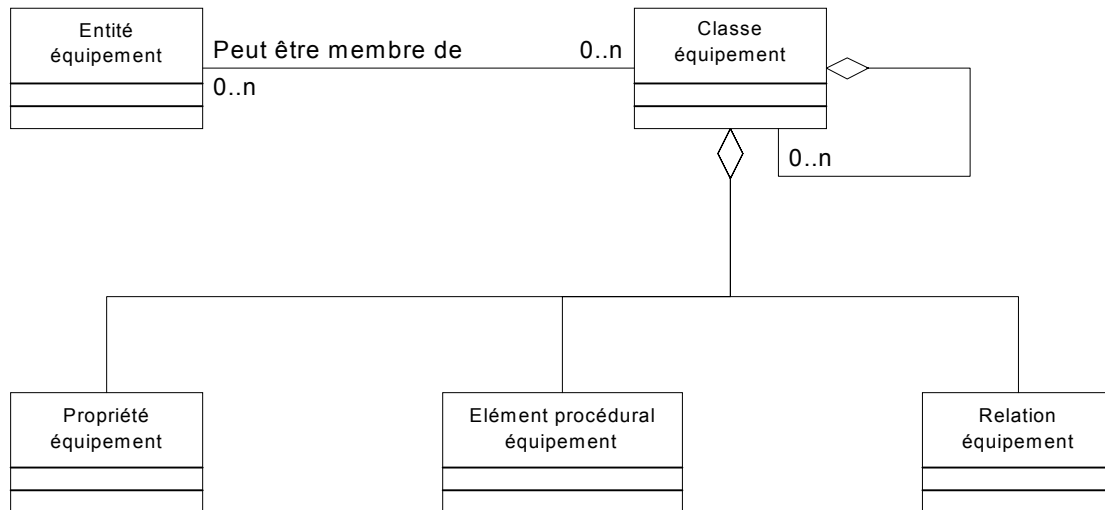
Figure 9 – Equipment entity relations

Equipment classes (see figure 10 and table 19) provide a means of grouping 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: its equipment requirements may specify a specific reactor (for example, R-101), a set of reactor units (for example, R-101, R-103) or the reactor class (for example, 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 (for example, glass lining) are shared with the class.

Les entités d'équipement peuvent appartenir à aucune ou plusieurs classes d'équipement (par exemple la capacité BV1 peut être à la fois un réacteur et un stockage intermédiaire).

Les classes d'équipement peuvent déterminer certaines ou toutes les propriétés d'équipement, les éléments procéduraux d'équipement et les relations des équipements des entités d'équipement référençantes.



IEC 2216/01

Figure 10 – Classes d'équipements

Table 19 – Classe d'équipement

NOM	CLASSE D'EQUIPMENT
FunctionalDescription	Une classe d'entité d'équipement
ATTRIBUTS	
EquipmentClass	Fournit une identification unique
EquipmentLevel	Spécifie le niveau de hiérarchie physique (par exemple cellule de processus, unité, module d'équipement, module de contrôle)

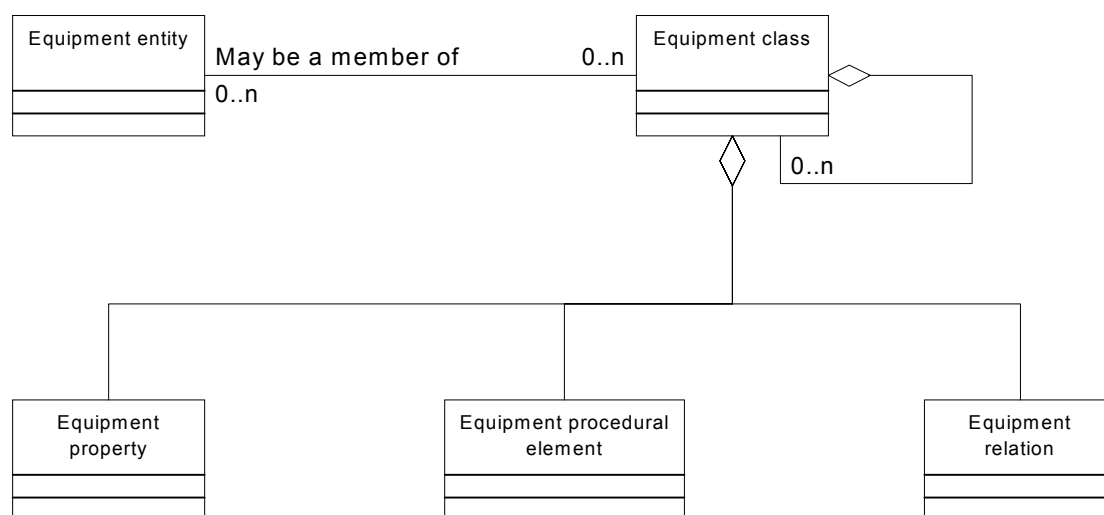
4.5 Planification et programmation de la production

L'entité centrale dans un programme de production (voir figure 11) est l'entrée de programme de batch. Cet objet définit l'exécution désirée d'un(e) ou plusieurs batchs/recettes de contrôle ou d'autres entités de recette de contrôle (typiquement, des procédures d'unité) (voir table 20). L'entrée de programme de batch peut aussi être utilisée pour programmer d'autres activités (par exemple mise en indisponibilité des équipements). Une entrée de programme de batch peut inclure des valeurs de formules/de paramètre qui sont à utiliser dans une recette de contrôle (voir table 21).

Il est admis que l'entrée de programme de batch soit utilisée pour représenter des entités de programmation d'un niveau supérieur (par exemple une campagne de production ou un ordre de production).

Equipment entities may belong to zero or more *equipment classes* (for example, 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.



IEC 2216/01

Figure 10 – Equipment classes

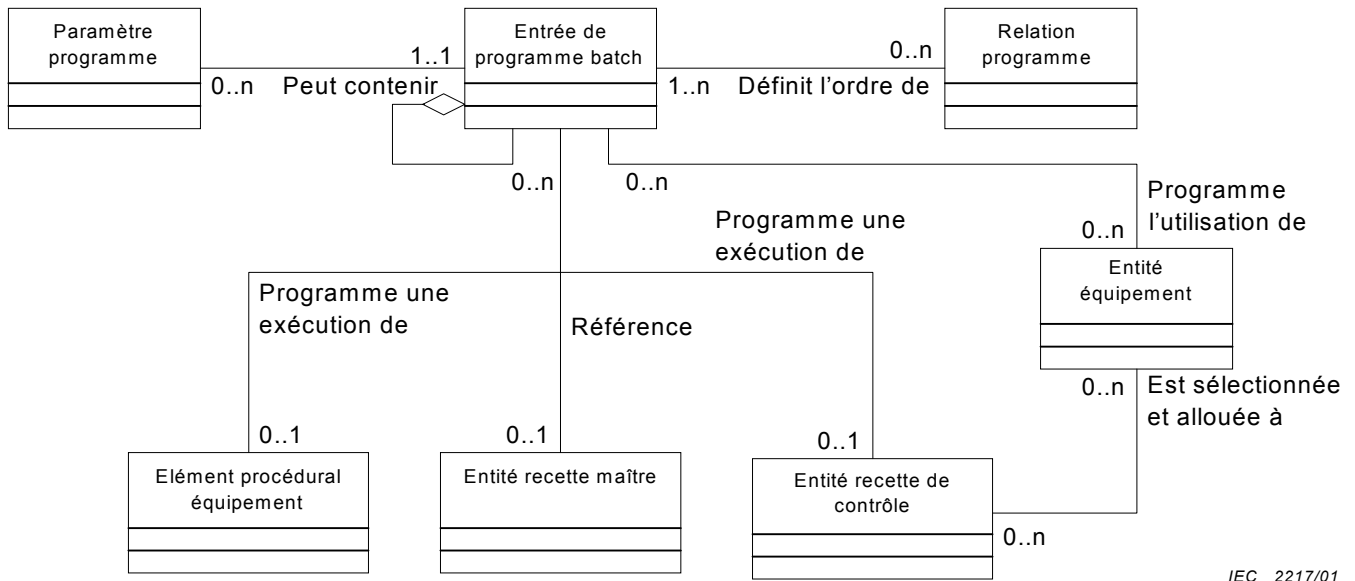
Table 19 – Equipment class

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 (for example, 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 (for example, a production campaign or a production order).



IEC 2217/01

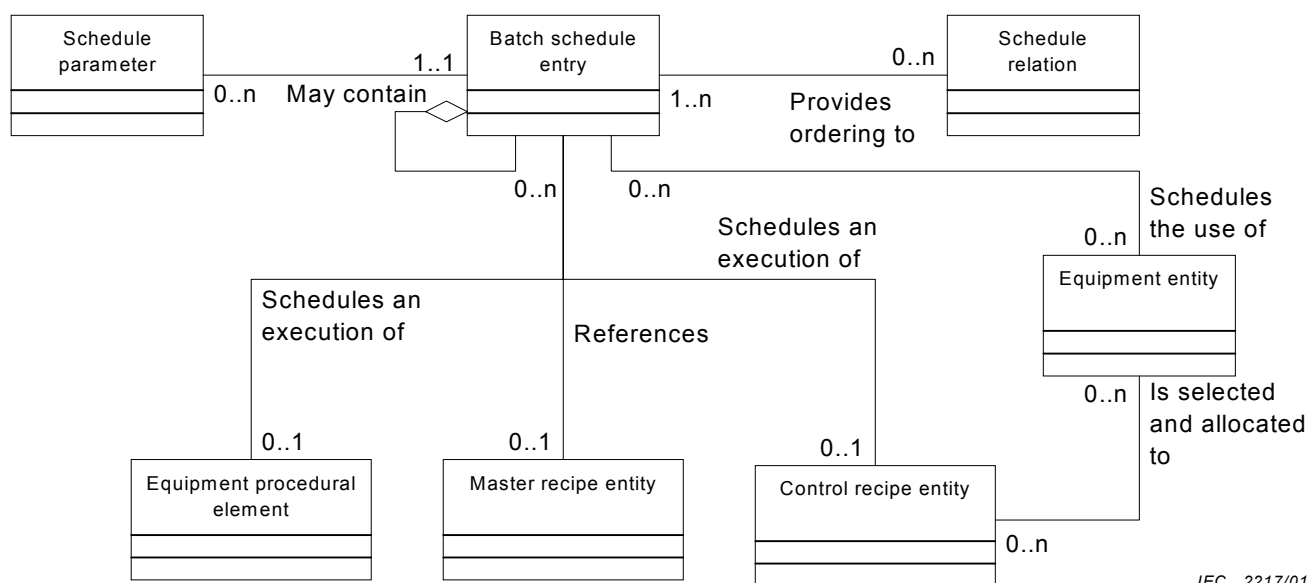
Figure 11 – Programme de batch

Table 20 – Entrée de programme de batch

NOM	ENTREE DE PROGRAMME DE BATCH
FunctionalDescription	Un élément de programmation qui représente une procédure d'unité dans un batch, un batch complet ou un ensemble de batchs (par exemple une campagne)
ATTRIBUTS	
ID	Fournit une identification unique (par exemple campagne actuelle, lot, ID du batch, ID d'entité procédurale)
Level	Spécifie le niveau de hiérarchie (par exemple campagne, batch, procédure d'unité)
BatchSize	Définit la taille requise, ou le facteur d'échelle, pour le batch, basée sur le facteur d'échelle pour le batch tel que défini dans la recette maître
Schedule	Définit les horodates d'exécution programmée (lancement/arrêt)
ResourceUsage	Spécifie l'utilisation de ressource pour cette entrée de programme
Status	Spécifie le statut du programme (par exemple proposé pour évaluation (par exemple analyse what-if), planifié, engagé, commencé, effectué)

Table 21 – Paramètre de programme

NOM	PARAMETRE DE PROGRAMME
FunctionalDescription	Valeurs de formule qui sont tenues d'être communiquées vers et depuis les entrées de programme de batch
ATTRIBUTS	
ParameterID	Fournit une identification unique
ParameterType	Inclut la façon dont la valeur doit être interprétée (par exemple constante, référence ou équation)
Description	Décrit le paramètre ou l'utilisation du paramètre
EngineeringUnits	Identifie les unités de mesure d'ingénierie pour la valeur (par exemple kg, livres)
Value	Contient la valeur du paramètre. Si c'est une relation, la valeur contient l'équation, la forme, la règle de renvoi ou tout ce qui relie les paramètres entre eux. Si la valeur est un paramètre de bloc de construction, cet attribut contiendra la valeur par défaut
Scaled	Définit la règle de mise à l'échelle. Cas le plus simple: ajusté ou non ajusté d'après la taille de référence du batch
Usage	Identifie le paramètre comme entrée, sortie ou paramètre de processus



IEC 2217/01

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 (e.g., 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, the value contains the equation, form, deferral rule or whatever ties the related parameters together. If the 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

Il est admis que les relations de programme soient utilisées pour représenter le sous-ensemble correspondant des relations de recette (par exemple les relations liées aux transferts de batch)(voir table 22). A des niveaux supérieurs, il est admis qu'elles soient utilisées pour représenter les relations requises ou désirables entre les entrées de programme (par exemple il convient que le batch xx suive le batch yy, ou qu'un nettoyage spécifique ait lieu entre les deux batchs). Les sous-classes et caractéristiques spécifiques des relations des entrées de programme ne sont pas modélisées ici.

Une entrée de programme de batch d'un niveau supérieur inclura les entrées et relations de programme du niveau inférieur (par exemple une campagne de programmation est susceptible d'inclure les batchs programmés et les relations entre ces batchs).

Dans le cas le plus simple, l'entrée de programme de batch représente un batch dans une liste ou une queue d'exécution de batch qui est supposée être initialisée en séquence. Les horodates de lancement prévus et les durées / horodates de fin projetées peuvent être ajoutées. De plus, les assignations des équipements et l'utilisation d'autres ressources peuvent être spécifiées par l'entrée de programme de batch. La programmation peut survenir à des niveaux plus détaillés (par exemple la programmation de recettes d'unité individuelles et leur assignation pour l'équipement, et potentiellement la programmation des opérations ou phases individuelles, leur durée prévue, et leur consommation de ressources – y compris les ressources d'utilisation partagée ou exclusive qui contraignent la programmation). Les unités et les modules d'équipement sont alloués ou désalloués comme requis par la recette de contrôle spécifique.

L'ensemble des entrées de programme peut être considéré de différentes manières:

- comme une liste ou un diagramme de batchs dans une cellule de processus ou dans un élément d'une cellule de processus, qui présente une vue d'ensemble de l'utilisation d'une cellule de processus;
- comme une liste ou un diagramme d'utilisations des ressources, qui présente un planning des ressources ou des équipements.

Table 22 – Relation de programmation

NOM	RELATION DE PROGRAMMATION
FunctionalDescription	Une représentation des relations entre entrées de programme (par exemple vérification de ligne requise, nettoyage entre éléments programmés, spécifications de relations séquentielles dans la procédure)
ATTRIBUTS	
ExecuteOrder	Spécifie dans quelle séquence il convient que les entrées de programme multiples soient traitées relativement les unes aux autres

4.6 Gestion de l'information de production

Ce paragraphe décrit les modèles qui définissent la collecte d'informations de production.

Les informations de production, y compris l'information pertinente sur la manière dont les choses ont progressé, peuvent inclure à la fois l'information spécifique au batch et l'information sélectionnée commune ou non spécifique au batch (voir figure 12 et tables 23 à 28).

The schedule relations can be used to represent the subset of recipe relations relevant to scheduling (for example, 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 (for example, 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 modelled here.

A higher-level batch schedule entry will include the lower level schedule entries and relations (for example, 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. Furthermore, equipment assignments and the use of other resources may be specified by the batch schedule entry. The scheduling may happen at more detailed levels (for example, 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 different ways:

- 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;
- as a list or chart of resource utilizations, which provides a resource or equipment schedule.

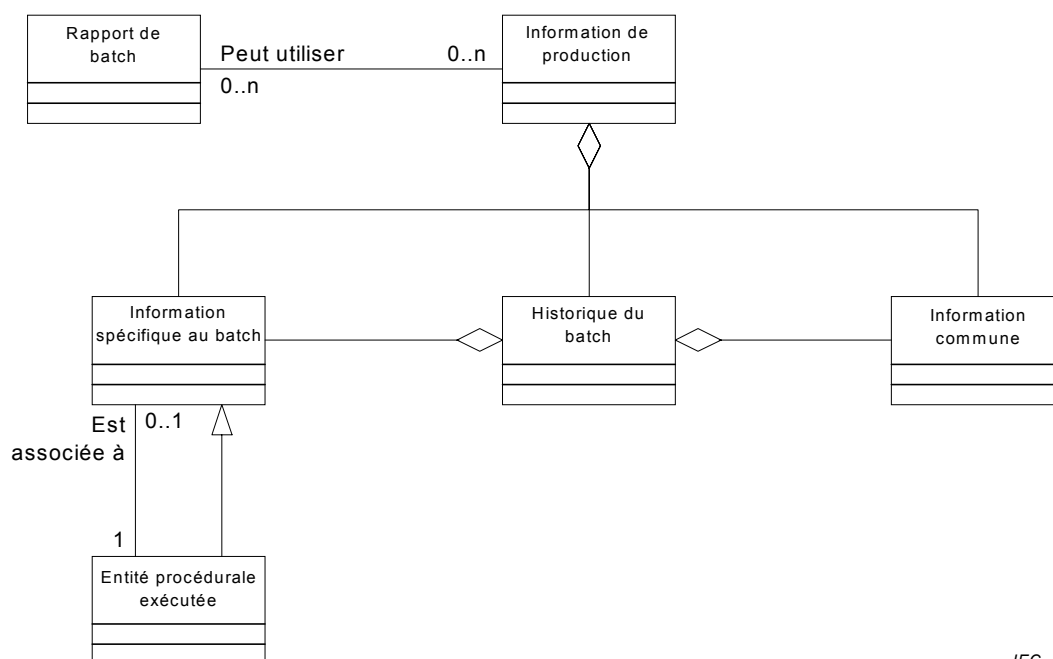
Table 22 – Schedule relation

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 to 28).



IEC 2218/01

Figure 12 – Information de production

Table 23 – Information de production

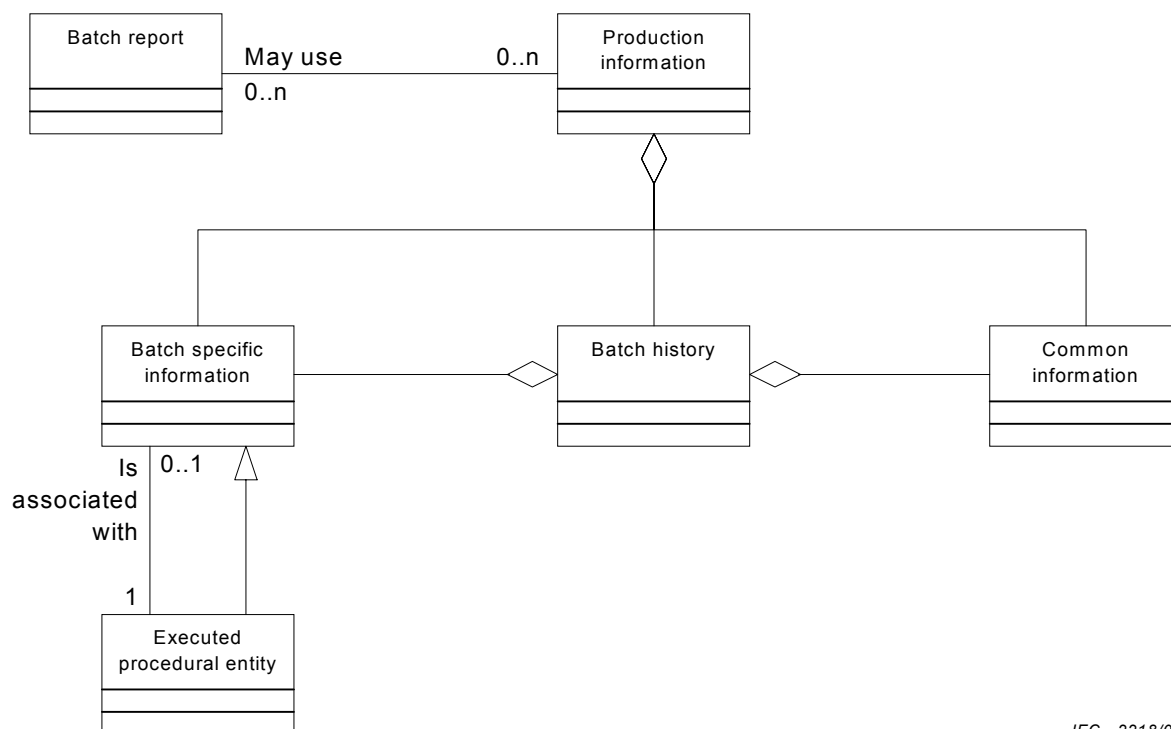
NOM	INFORMATION DE PRODUCTION
FunctionalDescription	L'information générée lors de la production d'un batch
ATTRIBUTS	

Table 24 – Information spécifique au batch

NOM	INFORMATION SPECIFIQUE AU BATCH
FunctionalDescription	Donnée qui est relative à une entrée d'historique du batch
ATTRIBUTS	
BatchID	Définit l'ID du batch réel
EntryID	Fournit une identification unique
NewValue	Indique la valeur courante réelle
EngineeringUnits	Définit les unités d'ingénierie, s'il y en a, qui sont appropriées pour la NewValue
EquipmentID	Identifie un élément d'équipement qui peut être associé avec l'entrée
UTC	Identifie le temps universel coordonné (TUC) et la date de l'entrée enregistrée
UserID	Identifie l'utilisateur, s'il y en a un, qui est associé au changement

Table 25 – Historique du batch

NOM	HISTORIQUE DU BATCH
FunctionalDescription	Un élément d'information qui documente la production du batch
ATTRIBUTS	
BatchID	Définit l'ID du batch réel



IEC 2218/01

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 co-ordinated 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 – Information commune

NOM	INFORMATION COMMUNE
FunctionalDescription	Donnée qui est liée à plus d'une entrée d'historique du batch (par exemple température de refroidissement de l'eau, pression atmosphérique, capacité de vapeur)
ATTRIBUTS	
EntryID	Fournit une identification unique
NewValue	Indique la valeur courante réelle
EngineeringUnits	Définit les unités d'ingénierie, s'il y en a, qui sont appropriées à la NewValue
EquipmentID	Identifie un élément d'équipement qui peut être associé avec l'entrée
UTC	Identifie le temps universel coordonné (TUC) et la date de l'entrée enregistrée
UserID	Identifie l'utilisateur, s'il y en a un, qui est associé au changement

Les informations de production sont susceptibles d'inclure certains des éléments suivants:

- une copie de la recette de contrôle;
- une copie de la recette maître;
- des informations sur les matières utilisées et produites;
- des informations de tendance;
- des alarmes et messages;
- les interactions de l'opérateur avec le batch (par exemple forçages, acquittements);
- des enregistrements différés et des enregistrements asynchrones (par exemple mesures sur un échantillon de laboratoire);
- des informations supplémentaires (par exemple allocation, lancement/arrêt).

Une entité procédurale exécutée est l'enregistrement d'une instance d'exécution d'une entité de recette ou d'une entité procédurale d'équipement (voir table 27). La donnée est associée à l'instance d'exécution, et cette donnée est maintenue dans les documents de l'historique du batch correspondant.

Les entités procédurales exécutées qui sont le résultat de l'exécution d'une recette de contrôle auront fréquemment la même structure que les entités de recette de contrôle. La structure des éléments procéduraux exécutés peut cependant différer de la recette de contrôle dans certains cas. Certains des exemples suivants sont typiques:

- Des instances répétées d'une entité de recette créées par un bouclage dans la logique procédurale.
- Des instances non exécutées en raison de branches ou de saut (GoTos) dans la logique procédurale.
- Des entités de recette qui sont insérées ou répétées manuellement.
- Des entités de recette qui sont activées sur place dans des unités qui sont déjà associées à un batch ou qui sont manuellement associées à un batch.

La relation d'entité de recette et le reste de la structure conceptuelle de l'entité de recette ne sont pas répétés ici, parce que l'historique concerne la capture des faits en tant que tels, et non pas la structure sous-jacente.

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 co-ordinated 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 copy of the control recipe;
- a copy of the master recipe;
- information about materials that are used and produced;
- trend information;
- alarms and messages;
- operator interactions with the batch (for example, overwrites, acknowledgements);
- late records and asynchronous records (for example, lab sample measurements);
- additional information (for example, 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 27). 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:

- Repeated instances of a recipe entity created through looping in the procedural logic.
- Instances not executed due to branches or GoTos in the procedural logic.
- Recipe entities that are inserted or repeated manually.
- 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 – Entité procédurale exécutée

NOM	ENTITE PROCEDURALE EXECUTEE
FunctionalDescription	Une représentation d'une entité de recette (par exemple un élément procédural d'équipement) qui a été exécutée
ATTRIBUTS	
ExecutedProcedural EntityID	Fournit une identification unique
ProceduralEntityCounter	Identifie uniquement une exécution répétée de la même entité procédurale

Les rapports de batch (voir table 28) sont considérés comme matérialisés par toute extraction de données de batch à partir de l'information de production en vue de leur présentation à l'écran ou sur papier ou pour leur transfert vers d'autres systèmes.

Table 28 – Rapport de batch

NOM	RAPPORT DE BATCH
FunctionalDescription	Un composant d'un rapport de batch
ATTRIBUTS	
ReportID	Fournit une identification unique

5 Tables relationnelles pour l'échange d'informations

5.1 Introduction

Cet article définit la structure de tables relationnelles SQL pour l'échange d'une information sélectionnée liée au contrôle de batch entre les systèmes. Il définit une spécification d'interface, répondant aux prescriptions de l'article 4, pour l'échange de l'information de batch dans les catégories suivantes:

- Informations de recette maître et de contrôle.
- Informations d'équipement de la cellule de processus.
- Informations de programmation.
- Informations de production.

Les tables d'échange doivent être construites en utilisant les noms de table, les noms des champs et les relations définis dans cet article. Toutes les tables ne nécessitent pas d'être mises en œuvre, mais tous les champs dans une table de mise en œuvre doivent être inclus. Toute mise en œuvre doit être compatible avec les définitions des tables présentées ici et avec les concepts de la partie 1 de cette norme.

Le format pour les tables d'échange définit seulement l'information standard qui peut être échangée. Il est admis que les définitions des tables soient étendues à travers l'inclusion d'attributs additionnels, de tables additionnelles en relation et d'énumérations supplémentaires. Les structures ainsi étendues peuvent être utilisées pour échanger de l'information entre des outils qui comprennent ces structures, mais cette information se trouve hors du domaine d'application de cette norme.

Des exemples d'informations supplémentaires peuvent être:

- L'addition de pictogrammes qui représentent différents éléments de la hiérarchie d'équipement, de telle manière que l'équipement puisse être représenté de façon cohérente par différents outils.

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 of clause 4, for the exchange of batch information in the following selected categories:

- Master and control recipe information.
- Process cell equipment information.
- Schedule information.
- 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 of this standard.

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:

- The addition of icons that represent different elements of the equipment hierarchy, so that the equipment may be represented consistently by different tools.

- L'addition d'adresses du système de contrôle pour les procédures d'élément d'équipement, les paramètres de procédure et les éléments de données.

5.1.1 Méthode

Les structures de tables relationnelles sont définies en notation SQL, telle que spécifiée dans la norme ISO/CEI 9075.

Le mécanisme d'échange est basé sur une structure commune pour les tables de base de données. Ces tables sont définies comme un schéma de base de données qui peut être défini dans des tables SQL. L'annexe B contient les définitions SQL des tables SQL d'échange.

La figure 13 illustre comment les tables d'échange seraient utilisées pour échanger une information de batch entre différents outils. Chaque outil aura généralement ses propres stockages de données pour l'information de batch, et chaque outil est responsable de leur importation depuis et de leur exportation vers les tables d'échange.

La plupart des autres modèles de cet article sont représentés par des diagrammes relationnels entre entités (ERSs) (voir article A.3).

Le type et l'utilisation des outils qui utilisent l'information de batch ne sont pas spécifiés dans cette norme. Les outils possibles incluent sans limitation particulière:

- des systèmes de création de recettes,
- des systèmes d'exécution de recette,
- des systèmes de documentation,
- des systèmes de gestion de configuration,
- des systèmes de simulation,
- des systèmes de contrôle batch,
- des systèmes de planification et d'ordonnancement,
- des systèmes de gestion de l'information.

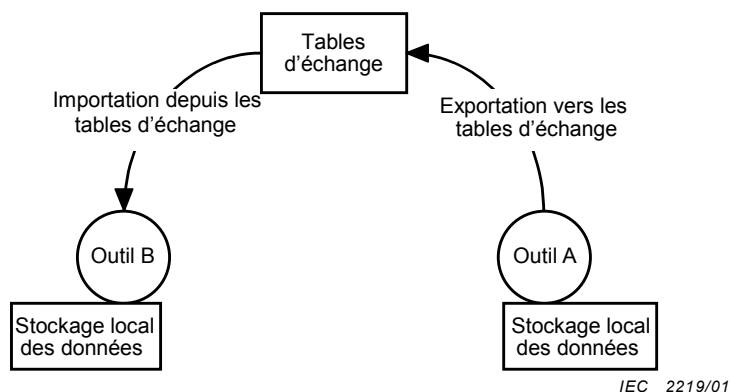


Figure 13 – Transfert de données par des tables d'échange

5.1.2 Tables d'échange

Cette norme ne spécifie pas quels outils créent et maintiennent les tables d'échange. Cette norme définit uniquement la structure des tables. Les outils pourraient être conçus de manière à ce qu'ils ne puissent que lire depuis des tables d'échange préexistantes, qu'écrire vers des tables d'échange préexistantes, lire depuis et écrire vers des tables préexistantes ou créer, lire depuis, et écrire vers des tables d'échange.

- 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.

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.

Most of the remaining models in this clause are represented using entity relationship diagrams (ERSs) (see clause A.3).

The type and use of the tools that use the batch information are not specified in this standard. Possible tools include, but are not limited to:

- recipe authoring systems,
- recipe execution systems,
- documentation systems,
- configuration management systems,
- simulation systems,
- batch control systems,
- planning and scheduling systems,
- information management systems.

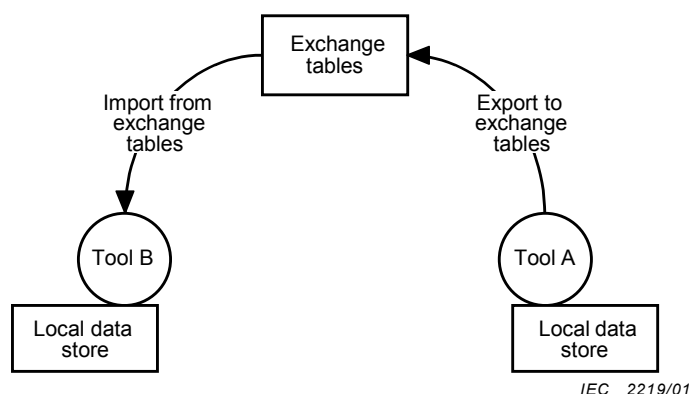


Figure 13 – Data transfer using exchange tables

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 from and write to pre-existing tables or create, read, and write to exchange tables.

La structure des tables d'échanges est conçue pour permettre à de multiples recettes d'être échangées dans le même ensemble de tables, et elle permet à de multiples versions de la même recette d'être échangées. La structure des tables d'échange permet également l'échange des capacités d'équipements et de l'information de spécification sur les cellules de processus et les équipements de la cellule de processus.

La structure des tables d'échange permet également le transfert de sous-ensembles complets ou partiels d'une recette maître, de la description des équipements, d'informations de programmation ou de l'historique de batch.

La syntaxe des chaînes de données dans les tables d'échange (par exemple éléments de formules calculés, conditions de transition) n'est pas définie dans cette norme. Il convient que les outils qui lisent et écrivent les tables d'échange SQL résolvent les différences de syntaxe.

Les longueurs des champs qui sont définies dans la définition SQL des tables d'échange représentent des valeurs par défaut, et elles ne sont pas destinées à imposer des longueurs de champs standards, minimales ou maximales. Les outils qui lisent et écrivent les tables d'échange SQL doivent résoudre les différences de longueur des champs.

5.1.3 Informations d'échange communes

Un ensemble de tables est défini pour contenir les informations qui décrivent le format d'échange et qui peuvent être communément utilisées dans les différents ensembles de tables d'échange. La figure 14 montre les tables associées aux informations communes d'échange.

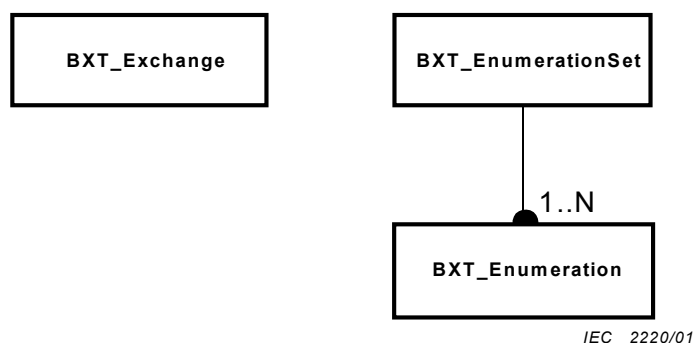


Figure 14 – Tables d'échange des informations communes

5.1.3.1 Informations d'échange

La table d'échange des informations communes, BXT_Exchange, contient toute l'information nécessaire une fois seulement dans l'échange de données (voir table 29).

La table contenue dans la table BXT_Exchange (voir table 30) contient un enregistrement pour chacun des éléments définis (par exemple SCHEMA et le caractère DELIMITER). Elle peut aussi contenir d'autres informations définies par l'utilisateur.

Table 29 – BXT_Exchange

Attribut	Description
<i>ExchangeID</i>	Identifie l'élément d'échange
ExchangeValue	Identifie la valeur pour l'information échangée

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 (for example, 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 minimum or maximum field length standard. Tools that read and write the SQL exchange tables shall resolve field length differences.

5.1.3 Common exchange information

A set of tables are 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.

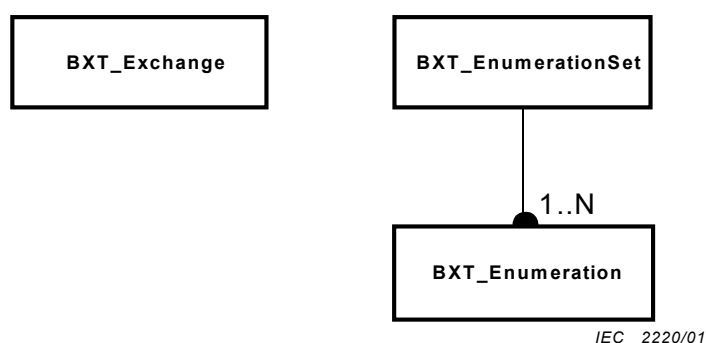


Figure 14 – Common information exchange tables

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 (for example, SCHEMA and the DELIMITER characters). It may also contain other user-defined information.

Table 29 – BXT_Exchange

Attribute	Description
<i>ExchangeID</i>	Identifies the exchange element
ExchangeValue	Identifies the value for the exchanged information

Table 30 – Contenus de la table BXT_Exchange

ExchangeID	Description
Schema	La version de la norme par rapport à laquelle le schéma de base de donnée a été défini. La valeur de la version initiale doit être le titre de cette norme (par exemple version CEI 61512-2:2001)
Délimiter	Le caractère utilisé pour séparer les noms des éléments dans le nom hiérarchique de l'élément de recette
ToolID	L'identification de l'outil qui a créé les tables d'échange
ToolVersion	La version de l'outil qui a créé les tables d'échange
ToolSchema	La version de schéma sur mesure créée par l'outil

5.1.3.2 Enumérations

Beaucoup de tables contiennent des champs qui peuvent contenir des éléments énumérés standards ou définis pour l'utilisateur. Ces éléments énumérés sont passés comme des nombres dans les tables d'échange, les chaînes étant contenues dans une table d'énumération. Les tables d'énumération (voir table 31) fournissent une localisation unique pour la traduction des chaînes entre différents langages. La table BXT_EnumerationSet définit les énumérations. La table BXT_Enumeration définit les membres de l'énumération et la valeur numérique correspondante.

Les tables d'énumération contiennent les énumérations standards et leurs valeurs. Celles-ci peuvent être étendues par l'utilisateur. Les extensions utilisateur des énumérations standards peuvent prendre les valeurs de 100 et plus. Des valeurs des membres d'énumération de 0 à 99 sont soit utilisées, soit réservées par cette norme. De plus, des énumérations définies par l'utilisateur et leur valeurs respectives peuvent également être définies dans les tables d'énumération. Par exemple, l'énumération «huiles de mélange» peut être définie pour des paramètres de phase avec des membres tels que «huile vierge», «huile mélangée», et «huile récupérée» ayant des valeurs respectives de 101, 102 et 103.

Table 31 – BXT_EnumerationSet

Attribut	Description
EnumSet	Identifie l'énumération standard
Description	Contient l'utilisation de l'énumération. (Fournie pour assister la traduction du TextString)

La table 32 définit les énumérations standards définies dans cette norme.

Table 32 – Enumérations standards

EnumSet	Description
Boolean	Définit un ensemble de valeurs booléennes
DirectionType	Définit comment un paramètre est destiné à être traité
EquipmentLevel	Définit le niveau hiérarchique d'équipement pour les éléments d'équipement
EquipmentType	Définit le type d'enregistrement d'équipement pour les éléments d'équipement
EvaluationRule	Définit les règles d'évaluation pour les propriétés d'équipement
FormulaType	Définit les types de formules de recette
FormulaSubType	Spécifie les définitions de sous-types de formule fournis par l'utilisateur

Table 30 – BXT_Exchange table contents

ExchangeID	Description
Schema	The version of the standard used in the definition of the database schema. The initial version value shall be the title of this standard (e.g., version IEC 61512-2:2001)
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) provide a single location for the 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 to 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

Table 32 – Enumérations standards (suite)

EnumSet	Description
LinkDepiction	Définit comment les liens entre les éléments de recette doivent être représentés
LinkToType	Définit si un lien fait référence à une étape ou à une transition
LinkType	Définit le type de lien
RE_Type	Définit l'élément de recette (RE), soit le niveau procédural de recette, soit un symbole d'allocation
RE_Use	Définit comment un RE est utilisé dans une recette
RecipeStatus	Définition des statuts possibles d'une recette
RecordSet	Définit l'énumération utilisée pour classer un enregistrement dans une catégorie d'information d'historique du batch
RecordSetControlRecipe	Fournit une classification additionnelle pour l'enregistrement de l'historique sous la catégorie ControlRecipe (recette de contrôle)
RecordSetMasterRecipe	Fournit une classification additionnelle pour l'enregistrement de l'historique sous la catégorie MasterRecipe (recette maître)
RecordSetExecutionInfo	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie de ExecutionInfo (infos exécution)
RecordSetMaterialInfo	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie MaterialInfo (infos matières)
RecordSetContinuousData	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie de ContinuousData (données continues)
RecordSetEvents	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie Events (événements)
RecordSetOperatorChange	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie OperatorChange (changements opérateur)
RecordSetOperatorComment	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie OperatorComment (commentaires opérateur)
RecordSetAnalysisData	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie of AnalysisData (données d'analyse)
RecordSetLateRecord	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie LateRecord (enregistrements différés)
RecordSetRecipeData	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie RecipeData (données de recette)
RecordSetRecipeSpecified	Fournit une classification d'enregistrement de l'historique supplémentaire sous la catégorie RecipeSpecified (spécifié par la recette)
RecordSetSummaryData	Fournit une classification additionnelle d'enregistrement de l'historique sous la catégorie SummaryData (données résumées)
ScheduleAction	Définit l'action voulue de l'enregistrement de programmation
ScheduleMode	Définit le mode dans lequel l'enregistrement de programmation débute l'exécution
ScheduleStatus	Définit le statut possible de l'enregistrement de programmation
SE_Type	Définit le type d'entité dans un enregistrement de programmation
ValueDataType	Définit le type de donnée d'une valeur de donnée correspondante
ValueType	Définit comment la valeur d'une chaîne est interprétée

Table 32 – Standard enumeration sets *(continued)*

EnumSet	Description
LinkDepiction	Defines how links between recipe elements are to be depicted
LinkToType	Defines if a link is referencing a step or a 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

La table 33 montre comment les énumérations sont définies par cette norme.

Table 33 – Enumeration_BXT

Attribut	Description
EnumSet	Identifie le nom de l'énumération
EnumValue	Spécifie la valeur numérique associée au membre de l'énumération
EnumString	Définit le texte associé au membre de l'énumération
Description	Contient l'utilisation du membre de l'énumération. (Fourni pour assister la traduction de TextString)

La table 34 contient la liste des membres des énumérations standards définis par cette norme.

Table 34 – Enumérations standards

Enumération	Valeur	Chaîne	Description
Boolean	0	FAUX	Définition d'une valeur booléenne
	1	VRAI	
DirectionType	0	Invalide	Entrée invalide
	1	Interne	Identifie comment un paramètre est traité. Interne = seulement disponible dans un élément de recette. Défini à la création ou créé comme valeur intermédiaire
	2	Entrée	L'élément de recette reçoit la valeur d'une source externe
	3	Sortie	L'élément de recette crée la valeur et la rend disponible pour une utilisation externe
	4	Entrée/Sortie	L'élément de recette et l'élément externe échangent la valeur, et peuvent changer sa valeur
	5-99		Réservé
	100+		Défini par l'utilisateur
EquipmentLevel	0	Invalide	Entrée invalide
	1	Entreprise	Identifie le niveau hiérarchique de l'équipement pour BXT_EquipElement
	2	Site	
	3	Zone	
	4	Cellule de Processus	
	5	Unité	
	6	Module d'Equipelement	
	7	Module de Contrôle	
	8-99		Réservé
	100+		Défini par l'utilisateur
EquipmentType	0	Invalide	Entrée invalide
	1	Classe	Identifie le type d'enregistrement pour BXT_EquipElement
	2	Elément	
	3-99		Réservé
	100+		Défini par l'utilisateur

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 (suite)

Enumération	Valeur	Chaîne	Description
EvaluationRule	0	Invalide	Entrée invalide
	1	=	Opérateur de comparaison d'égalité pour des propriétés d'équipement
	2	<>	Opérateur de comparaison d'inégalité pour des propriétés d'équipement
	3	<	Opérateur de comparaison «inférieur à» pour des propriétés d'équipement
	4	>	Opérateur de comparaison «supérieur à» pour des propriétés d'équipement
	5	<=	Opérateur de comparaison «inférieur ou égal à» pour des propriétés d'équipement
	6	>=	Opérateur de comparaison «supérieur ou égal à» pour des propriétés d'équipement
	7	Membre	Opérateur de comparaison «est un membre de» pour les propriétés d'équipement
	8	Non membre	Opérateur de comparaison «n'est pas un membre de» pour les propriétés d'équipement
	9	Non	Opérateur de négation pour des propriétés d'équipement
	10-99		Réservé
	100+		Défini par l'utilisateur
FormulaSubType	0	Invalide	Entrée invalide
	1-99		Réservé
	100+		Défini par l'utilisateur. Permet une classification utilisateur additionnelle d'un type de formule
FormulaType	0	Invalide	Entrée invalide
	1	Entrée de Processus	Type de formule de recette
	2	Sortie de Processus	
	3	Paramètre de Processus	
	4-99		Réservé
	100+		Défini par l'utilisateur
LinkDepiction	0	Invalide	Entrée invalide
	1	Aucun	Pas de représentation du lien
	2	Ligne	Lien montré seulement avec ligne
	3	ID	Lien montré seulement avec identificateur
	4	Ligne & ID	Lien montré avec ligne et identificateur
	5	Ligne & Flèche	Lien montré avec ligne et flèche de flux matière
	6	Ligne, Flèche & ID	Lien montré avec ligne, flèche de flux matière et identificateur
	7-99		Réservé
	100+		Défini par l'utilisateur

Table 34 (continued)

EnumSet	Enum Value	EnumString	Description
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 equal to" comparison operator for equipment properties
	6	>=	"Greater than or equal to" comparison operator for equipment properties
	7	Member	"Is a member of" comparison operator for equipment properties
	8	Not member	"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 (suite)

Enumération	Valeur	Chaîne	Description
LinkToType	0	Invalide	Entrée invalide
	1	Elément de Recette	Le lien fait référence à un enregistrement dans la table BXT_MRecipeElement
	2	Transition	Le lien fait référence à un enregistrement dans la table BXT_MRecipeTransition
	3-99		Réservé
	100+		Défini par l'utilisateur
LinkType	0	Invalide	Entrée invalide
	1	Lien de Contrôle	Définit un lien entre les éléments de recette qui indique un flux de contrôle procédural
	2	Lien de Transfert	Définit un lien entre les éléments de recette qui indique un transfert de matière
	3	Lien de Synchronisation	Définit un lien entre les éléments de recette où il y a une forme de synchronisation
	4-99		Réservé
	100+		Défini par l'utilisateur
RE_Type	0	Invalide	Entrée invalide
	1	Recette Maître	Spécifie le type d'élément de recette.
	2	Procédure	
	3	Procédure d'unité	
	4	Opération	
	5	Phase	
	6	Allocation	
	7	Début	
	8	Fin	
	9	Début de Parallélisme	
	10	Fin de Parallélisme	
	11	Début de Branche	
	12	Fin de Branche	
	13-99		Réservé
	100+		Défini par l'utilisateur
RE_Use	0	Invalide	Entrée invalide
	1	Lié	Un élément de recette (RE) peut être référencé par plusieurs étapes RE
	2	Imbriqué	Un RE n'est référencé que par un seul RE; un RE est défini pour chaque utilisation du RE
	3	Copié	Comme imbriqué, mais le RE spécifique a été modifié par rapport à sa définition d'origine
	4-99		Réservé
	100+		Défini par l'utilisateur

Table 34 (continued)

EnumSet	Enum Value	EnumString	Description
LinkToType	0	Invalid	Entry not valid
	1	Recipe Element	Link is referring to 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
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

Table 34 (suite)

Enumération	Valeur	Chaîne	Description
RecipeStatus	0	Invalide	Entrée invalide
	1	Approuvée pour Production	La recette est approuvée pour la production
	2	Approuvée pour Test	La recette est seulement approuvée pour des essais
	3	Non Approuvée	La recette n'est pas approuvée, ni pour une production ni pour des essais
	4	Inactive	La recette n'est pas active
	5	Obsolète	La recette est obsolète
	6-99		Réservé
	100+		Défini par l'utilisateur
RecordSet	0	Invalide	Entrée invalide
	1	RecordSetControlRecipe	Précise qu'un enregistrement d'information d'historique de batch fait partie de la catégorie ControlRecipe
	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		Réservé
	100+		Défini par l'utilisateur
RecordSet ControlRecipe	0	Invalide	Entrée invalide
	1	Recette de Contrôle Entière	L'enregistrement de l'historique est lié à la recette de contrôle dans son ensemble
	2-99		Réservé
	100+		Défini par l'utilisateur
RecordSet MasterRecipe	0	Invalide	Entrée invalide
	1	Recette Maître Entière	L'enregistrement de l'historique est lié à la recette maître dans son ensemble
	2-99		Réservé
	100+		Défini par l'utilisateur

Table 34 (continued)

EnumSet	Enum Value	EnumString	Description
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
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

Table 34 (suite)

Enumération	Valeur	Chaîne	Description
RecordSet ExecutionInfo	0	Invalide	Entrée invalide
	1	Allocation	
	2	Désallocation	
	3	Changement d'Etat	
	4	Commande d'Etat	
	5	Changement de Mode	
	6	Commande de Mode	
	7	Message d'Entité Procédurale	
	8	Alarme d'Entité Procédurale	
	9	Version d'Entité Procédurale	
	10	Message d'Invite d'Entité Procédurale	
	11	Réponse au Message d'Invite d'Entité Procédurale	
	12-99		Réservé
	100+		Défini par l'utilisateur
RecordSet MaterialInfo	0	Invalide	Entrée invalide
	1	Consommation de Matière	
	2	Production de Matière	
	3	Allocation de Matière	
	4	Désallocation de Matière	
	5-99		Réservé
	100+		Défini par l'utilisateur
RecordSet ContinuousData	0	Invalide	Entrée invalide
	1	Valeur de Donnée Continue	
	2	Association de Tendence	
	3	Dissociation de Tendence	
	4-99		Réservé
	100+		Défini par l'utilisateur
RecordSet Events	0	Invalide	Entrée invalide
	1	Événement Général	
	2-99		Réservé
	100+		Défini par l'utilisateur
RecordSet OperatorChange	0	Invalide	Entrée invalide
	1	Intervention Générale de l'Opérateur	
	2-99		Réservé
	100+		Défini par l'utilisateur
RecordSet OperatorComment	0	Invalide	Entrée invalide
	1	Commentaire Général de l'Opérateur	
	2-99		Réservé
	100+		Défini par l'utilisateur

Table 34 (continued)

EnumSet	Enum Value	EnumString	Description
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 Response	
	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
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

Table 34 (suite)

Enumération	Valeur	Chaîne	Description
RecordSetAnalysis Data	0	Invalide	Entrée invalide
	1	Message Général d'Analyse	
	2-99		Réservé
	100+		Défini par l'utilisateur
RecordSet LateRecord	0	Invalide	Entrée invalide
	1	Enregistrement Différé Général	
	2-99		Réservé
	100+		Défini par l'utilisateur
RecordSetRecipe Data	0	Invalide	Entrée invalide
	1	Donnée Générique de Recette	
	2	Changement de Valeur de Paramètre de Recette	
	3	Donnée de Résultat de Recette	
	4-99		Réservé
	100+		Défini par l'utilisateur
RecordSetRecipe Specified	0	Invalide	Entrée invalide
	1	Donnée Spécifiée Générique de Recette	
	2-99		Réservé
	100+		Défini par l'utilisateur
SummaryData	0	Invalide	Entrée invalide
	1	Donnée Résumée Générique	
	2	Consommation d'Utilités	
	3	Temps de Marche d'Equipeement	
	4-99		Réservé
	100+		Défini par l'utilisateur
ScheduleAction	0	Invalide	Entrée invalide
	1	Nouveau	Action de changement d'enregistrement de programmation
	2	Mise à Jour	
	3	Suppression	
	4-99		Réservé
	100+		Défini par l'utilisateur
ScheduleMode	0	Invalide	Entrée invalide
	1	Automatique	Mode de l'entrée de programmation
	2	Semi-automatique	
	3	Manuel	
	4	Non Spécifié	
	5-99		Réservé
	100+		Défini par l'utilisateur

Table 34 (continued)

EnumSet	Enum Value	EnumString	Description
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
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
RecordSetRecipe Specified	0	Invalid	Entry not valid
	1	Generic Recipe Specified Data	
	2-99		Reserved
	100+		User defined
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 (suite)

Enumération	Valeur	Chaîne	Description
ScheduleStatus	0	Invalide	Entrée invalide
	1	Effectué	Statut de l'entrée de programme de batch
	2	En cours	
	3	Programmé	
	4	Programmation suspendue	
	5	Non Spécifié	
	6-99		Réservé
	100+		Défini par l'utilisateur
SE_Type	0	Invalide	Entrée invalide
	1	Campagne	Définit le type d'entrée de programmation
	2	Batch	
	3	Procédure d'Unité	
	4	Opération	
	5	Phase	
	6-99		Réservé
	100+		Défini par l'utilisateur
ValueDataType	0	Invalide	Entrée invalide
	1	Booléen	Définit le type de donnée attendu pour une valeur associée
	2	Chaîne 8-Bit	
	3	Chaîne 16-Bit	
	4	Chaîne 32-Bit	
	5	Entier 8-Bit non signé	
	6	Entier 16-Bit non signé	
	7	Entier 32-Bit non signé	
	8	Entier 8-Bit signé	
	9	Entier 16-Bit signé	
	10	Entier 32-Bit signé	
	11	Flottant 32-Bit	
	12	Double flottant	
	13	Chaîne Octet	
	14	Date Heure	
	15-99		Réservé
	100+		Défini par l'utilisateur

Table 34 (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 (suite)

ValueType	0	Invalide	Entrée invalide
	1	Constant	Définit comment la valeur de type chaîne est interprétée. Elle contient une valeur fixe en tant que chaîne
	2	Référence	Définit comment la valeur de type chaîne est interprétée. Elle pointe sur la source de la valeur
	3	Equation	Précise que la valeur d'une chaîne représente une expression qui doit être évaluée pour déterminer la valeur
	4	Externe	La valeur est fournie par un moyen externe, et elle n'est pas contenue dans la recette (c'est-à-dire la valeur peut être fournie par un opérateur ou par un système d'ordonnancement)
	5-99		Réservé
	100+		Défini par l'utilisateur

5.2 Informations de recette maître

Ce paragraphe ne concerne que les *recettes maîtres*. L'information échangée est celle qui est nécessaire pour échanger une recette maître, telle qu'elle est définie dans la partie 1 de cette norme. Cette information contient les définitions de contrôle procédural, les définitions de valeurs de formules, les prescriptions d'équipement de la recette, les informations d'en-tête, les informations spécifiques aux cellules de processus, les autres informations et les prescriptions de contrôle de coordination.

L'information échangée est celle qui est nécessaire pour **échanger** une recette maître, mais cette information ne spécifie pas:

- comment l'information a été créée,
- comment la recette maître pourrait être utilisée dans un système.

5.2.1 Définitions de recette

La création d'une recette maître, telle que spécifiée dans la partie 1 de cette norme, peut utiliser l'information de la recette de site et la définition des capacités de traitement de la cellule de processus. Une recette maître est liée à la capacité de traitement de la cellule de processus, mais la définition de la capacité de traitement ne fait pas partie de l'information échangée avec la recette.

5.2.2 Structure de recette

Une recette est composée des catégories d'informations suivantes: en-tête, formules, procédure, prescriptions d'équipement et autres informations.

Le schéma d'échange de recette est récursif par nature. La structure de base de ce schéma, comme indiqué à la figure 15, est construite autour de ces cinq catégories d'informations. Chaque niveau de définition contient toutes ces catégories d'informations jusqu'à ce que la définition procédurale référence une entité procédurale d'équipement.

Le terme élément de recette (RE) est utilisé pour définir certaines des entités structurales dans une recette. Un élément de recette peut être la recette maître elle-même, la procédure de recette, la procédure d'unité de recette, l'opération de recette, la phase de recette, le symbole d'allocation ou d'autres symboles graphiques tels que définis dans la table 34.

Table 34 (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 of this standard. 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 co-ordination control requirements.

The information that is exchanged is what is needed to **exchange** a master recipe, but this information does not specify the following:

- how the information was created,
- 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 of this standard, may use information from the site recipe and the definition of the process cell's processing capability. A master recipe is linked to the process cell's processing capability, but the definition of the processing capability is not part of the recipe-exchanged information.

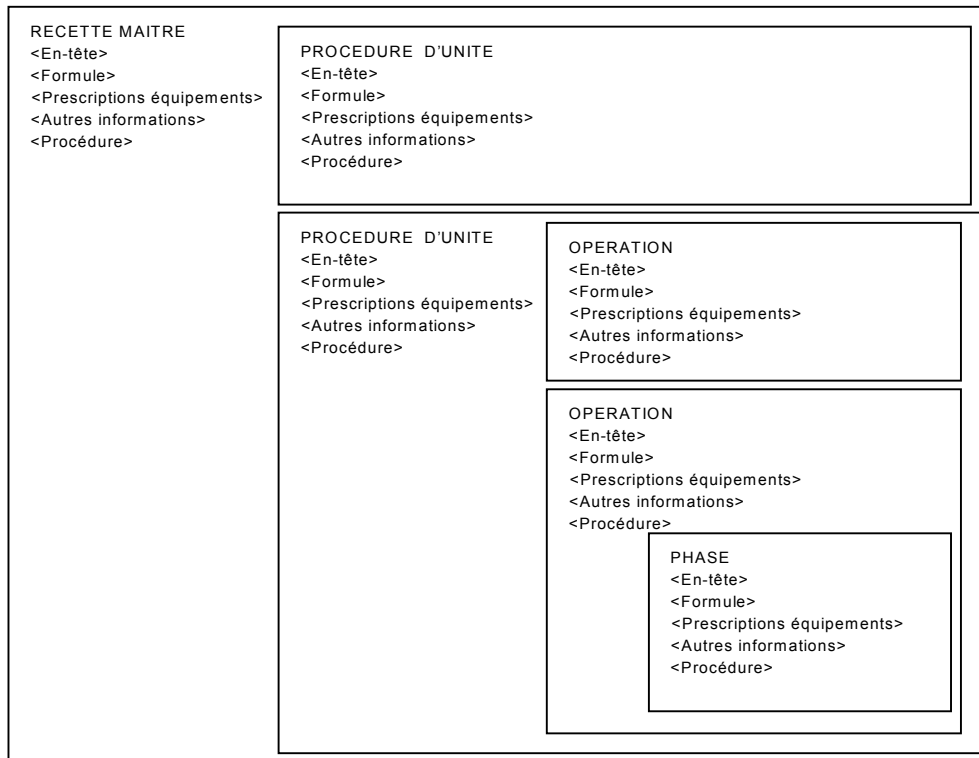
5.2.2 Recipe structure

A recipe is made up of the following information categories: header, formulae, procedure, equipment requirements and other information.

The recipe exchange scheme is recursive by nature. The basic structure of that scheme, 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.

Dans le modèle récursif du schéma, les éléments de recette contiennent soit des éléments de recette d'un niveau inférieur, soit des références aux éléments procéduraux d'équipement. Alors que la définition d'un élément de recette est contenue dans la table des éléments de recette, chaque utilisation d'un élément de recette est appelée une étape (step).



IEC 2221/01

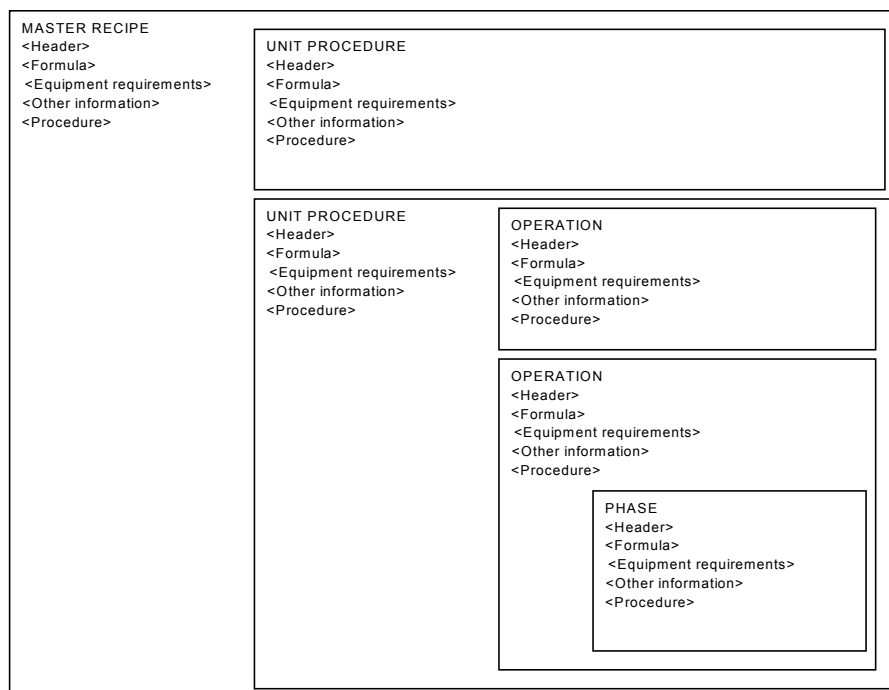
Figure 15 – Les éléments de recette imbriqués constituent la recette

5.2.3 Vue d'ensemble d'une table et contraintes d'intégrité

La figure 16 illustre les tables utilisées pour échanger des recettes et leurs relations, et elle définit les contraintes d'intégrité associées entre les tables.

Des relations non directes, comme entre LINK et RE, ne sont pas représentées; cependant, elles sont définies à travers l'ensemble de champs clé communs dans la table. Les entrées «NOT NULL» dans les tables SQL correspondantes ne sont utilisées que pour appliquer les contraintes d'intégrité du diagramme entités-relations.

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.



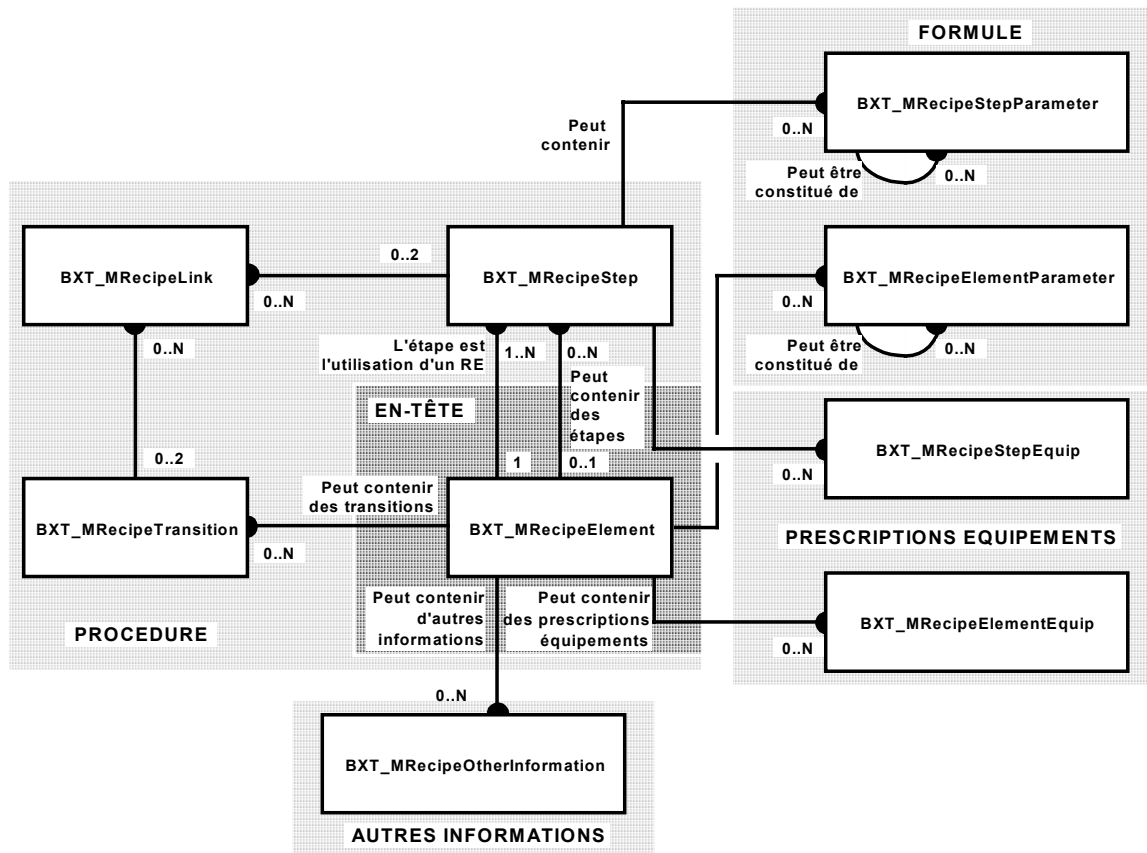
IEC 2221/01

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.



IEC 2222/01

Figure 16 – Relations entre les tables d'échange

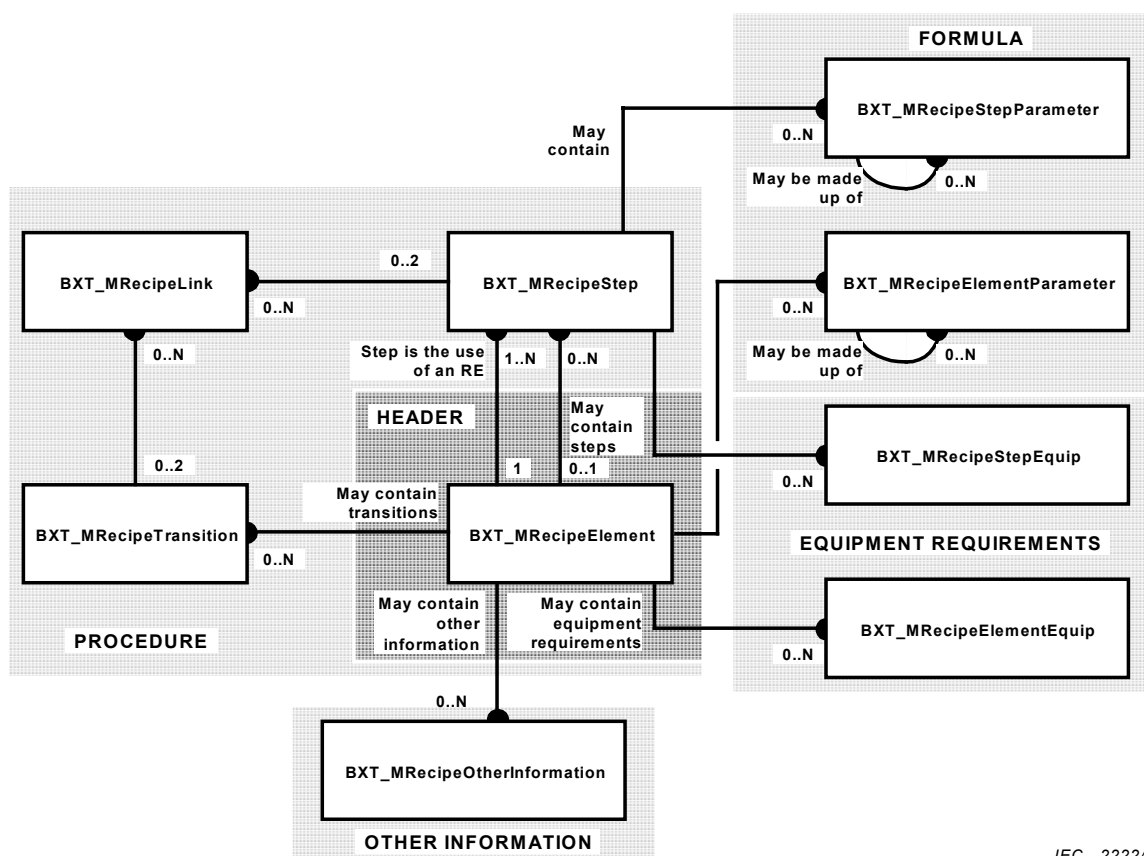
La définition récursive des éléments de recette (REs) se fait à travers les associations entre les BXT_MrecipeStep et les BXT_MrecipeElement entités. Chaque utilisation définie d'un élément de recette est enregistrée dans une table BXT_MrecipeStep. Chaque définition d'un élément de recette est enregistrée dans la table BXT_MrecipeElement. Une association présente les étapes qui sont contenues dans un élément de recette. L'autre association représente l'élément qui est référencé par une étape. Chaque RE qui est référencé par le RE propriétaire fait l'objet d'un enregistrement dans la table etape (step). La table Etape référence ensuite la définition du RE réel. Les tables supportent une définition de RE unique par Etape, et elles supportent de multiples etapes qui référencent le même RE.

Les relations entre etapes et transitions sont maintenues dans la table BXT_MRecipeLink.

La définition de la formule de recette maître est l'ensemble de tous les enregistrements des paramètres de la recette maître, et elle décrit les paramètres d'entrée de processus, de sortie de processus et de processus à l'intérieur de la recette. Les valeurs de formule sont contenues dans la table BXT_MRecipeStepParameter, et elles ont leur définition dans la table BXT_MRecipeElementParameter.

Les prescriptions d'équipement de la recette sont contenues dans les tables BXT_MRecipeStepEquip et BXT_MRecipeElementEquip.

La figure 17 montre comment les entrées dans chaque table sont liées les unes aux autres pour les tables BXT_MRecipeStep, BXT_MRecipeElement, BXT_MRecipeElementParameter et BXT_MRecipeStepParameter. Un enregistrement de BXT_MRecipeElement existe pour chaque version d'une recette échangée. Une relation entre cet enregistrement de BXT_MRecipeElement et un enregistrement unique de la table BXT_MRecipeStep existe, et cette relation contient l'information de recette spécifique, y compris les valeurs de formule dans la table BXT_MRecipeStepParameter.



IEC 2222/01

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.

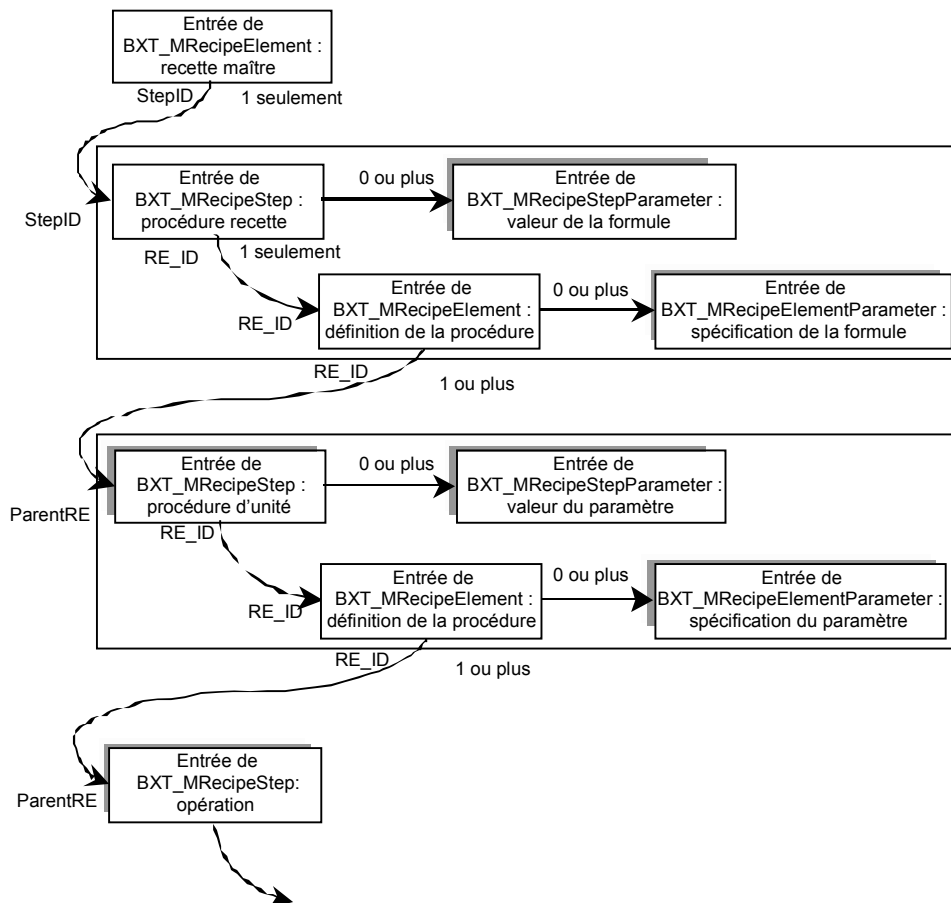
The recipe's equipment requirements are contained in the BXT_MRecipeStepEquip and BXT_MRecipeElementEquip tables.

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.

La table BXT_MRecipeStep contient une clé dans un enregistrement unique dans la table BXT_MRecipeElement. Cet enregistrement de table BXT_MRecipeElement contient la définition de la procédure de recette, y compris la définition des valeurs de formule dans la table BXT_MRecipeElementParameter.

La table BXT_MRecipeElement pour cette procédure contient une clé pour de multiples enregistrements dans la table BXT_MRecipeStep, une pour chaque procédure d'unité. (D'autres éléments de recette du niveau procédure sont omis dans cet exemple pour plus de simplicité). L'enregistrement de BXT_MRecipeStep pour ces procédures d'unité contient une clé pour l'enregistrement de BXT_MRecipeElement qui définit la procédure d'unité. L'enregistrement de BXT_MRecipeElement pour chaque procédure d'unité contient une clé pour la table BXT_MRecipeStep pour chaque opération. Cette structure se poursuit en descendant jusqu'aux définitions des phases.

Ce format de table utilise les tables BXT_MRecipeStep et BXT_MRecipeElement pour contenir la hiérarchie procédurale de la procédure de recette.



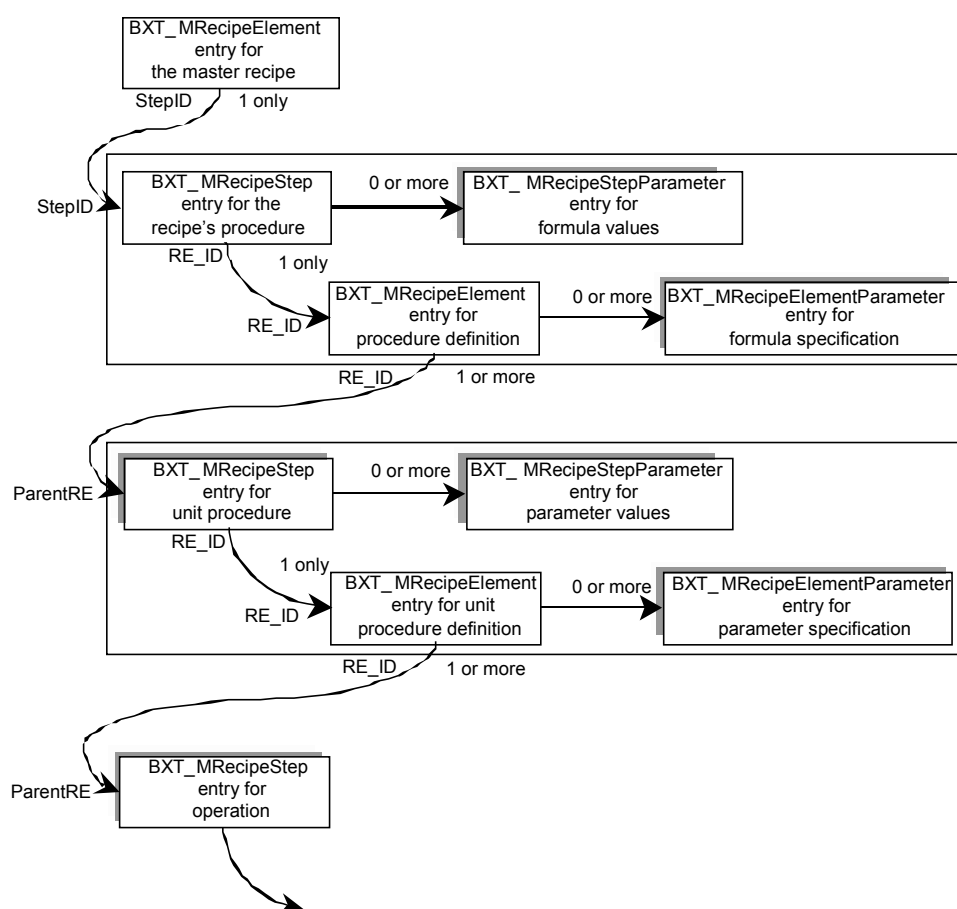
IEC 2223/01

Figure 17 – Rapport entre les entrées des tables

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_MRecipeElement Parameter 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.



IEC 2223/01

Figure 17 – How entries relate in the tables

5.2.4 Sommaire des tables de recette

Les tables qui sont définies pour l'échange des recettes sont présentées dans la table 35.

Table 35 – Tables d'échange de recette

Nom de la table	Description
BXT_MRecipeStep	Un enregistrement pour chaque utilisation d'un RE dans un RE
BXT_MRecipeElement	Un enregistrement pour chaque élément de recette échangé
BXT_MRecipeTransition	Un enregistrement pour chaque transition utilisée dans un RE
BXT_MRecipeLink	Un enregistrement pour chaque lien entre étapes et transitions
BXT_MRecipeElementParameter	Un enregistrement pour chaque paramètre pour chaque RE
BXT_MRecipeStepParameter	Un enregistrement pour chaque paramètre pour chaque étape
BXT_MRecipeOtherInformation	Un enregistrement pour chaque élément d'autres informations
BXT_MRecipeElementEquip	Un enregistrement pour chaque prescription de propriété pour un RE
BXT_MRecipeStepEquip	Un enregistrement pour chaque valeur pour une propriété d'équipement définie dans une étape

5.2.5 Définitions de tables de recette

5.2.5.1 Information d'en-tête

L'information d'en-tête de recette maître est transférée comme champs d'un enregistrement dans la table BXT_MRecipeElement. La table BXT_MRecipeElement (voir table 36) contient un élément pour chaque recette maître qui sera échangée. La combinaison d'ER_ID et de REVersion définit la recette maître échangée.

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 master recipe.

Table 36 – BXT_MRecipeElement

Attribut	Description
<i>RE_ID</i>	Identifie l'élément de recette qui sera échangé (par exemple <i>Red Oak</i>). Lorsqu'il est combiné à la «version», ce champ définit une instance unique d'un RE. Quand l'enregistrement représente la recette maître, ce champ contient l'ID de la recette maître
<i>REVersion</i>	Identifie la version du RE. Lorsqu'elle est combinée à une «RE_ID,» ce champ définit une instance unique d'un RE (par exemple <i>V10.3</i>)
<i>VersionDate</i>	Identifie la date et l'heure auxquelles cette version du RE a été modifiée pour la dernière fois
<i>ApprovalDate</i>	Identifie la date et l'heure auxquelles cette version de recette a été approuvée
<i>EffectiveDate</i>	Identifie la date et l'heure auxquelles cette version de recette est effective
<i>ExpirationDate</i>	Identifie la date et l'heure auxquelles cette version de recette expire
<i>Author</i>	Identifie la personne ou le système auteur de cette version (par exemple <i>J. Smith</i>)
<i>ApprovedBy</i>	Identifie la personne ou le système qui a approuvé cette version de la recette
<i>ProcessCellID</i>	Identifie la cellule de processus ou la classe de cellules de processus pour laquelle cette version de la recette maître a été définie
<i>ProductID</i>	Identifie le produit ou la famille de produit qui serait créé(e) par l'exécution de cette version de la recette (par exemple <i>Premium Beer</i>)
<i>UsageConstraint</i>	Définit d'autres règles qui déterminent l'utilisation (par exemple doit être suivi par..., ou ne doit pas fonctionner en parallèle avec...)
<i>Description</i>	Décrit l'élément de recette
<i>Status</i>	Définit le statut de l'information qui est échangée comme membre de l'énumération «recipe status»
<i>RE_Type</i>	Identifie le type de RE, membre de l'énumération «RE_Type»
<i>RE_Function</i>	Contient une référence optionnelle aux tables d'échange d'information d'équipement. Le format pour cette information n'est pas défini dans cette norme. Un exemple est une référence à un élément procédural d'équipement (voir table BXT_EquipInterface). Si cette entrée est NULL, alors la fonction de cet RE est définie par les entrées BXT_MRecipeStep et BXT_MRecipeTransition qui ont un ParentRE qui correspond au RE_ID de table
<i>RE_Use</i>	Identifie la relation entre le RE et BXT_MRecipeStep, membre de l'énumération «RE_Use». «Lié» spécifie qu'il peut y avoir de multiples utilisations BXT_MRecipeStep de la définition du RE. Lié serait utilisé quand le BXT_MRecipeElement est un bloc de construction de bibliothèque. «Imbriqué» spécifie que le RE n'a qu'un RE de référence, et un RE est défini pour chaque utilisation du RE. Dans ce cas, le RE est «imbriqué» dans la définition de recette d'une étape unique de recette, et n'est pas utilisé ailleurs. «Copié» spécifie la même chose qu'imbriqué, mais le RE a été modifié à partir d'une définition d'origine. Copié serait utilisé lorsque le RE était un bloc de construction de bibliothèque complètement reproduit dans la recette, et que son lien à la bibliothèque a été rompu
<i>DerivedRE</i>	Identifie l'élément de recette à partir duquel cet élément de recette a été dérivé
<i>DerivedVersion</i>	Identifie la version de l'élément de recette à partir duquel cet élément de recette a été dérivé

Table 36 – BXT_MRecipeElement

Attribute	Description
<i>RE_ID</i>	Identifies the recipe element that will be exchanged (e.g., <i>Red Oak</i>). 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
<i>REVersion</i>	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a RE (e.g., <i>V10.3</i>)
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., <i>J. Smith</i>)
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., <i>Premium Beer</i>)
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 Information de procédure

Les éléments procéduraux d'une recette maître sont inclus ou contenus dans une combinaison ou un ensemble de tables. Ces enregistrements définissent:

- les étapes des éléments de contrôle procédural ;
- les éléments de contrôle procédural;
- les symboles d'allocation et autres symboles de représentation graphique;
- le paramétrage des éléments procéduraux avec limites;
- la relation entre les éléments, et
- les définitions des transitions entre les éléments.

Les tables étape (step), transition, et lien (link) contiennent la définition des RE qui contiennent des RE d'un niveau inférieur. La table BXT_MRecipeStep (voir table 37) contient chaque instance d'utilisation d'un RE. Chaque BXT_MRecipeStep contient les paramètres utilisés quand un RE est utilisé. Comme un RE peut être utilisé plusieurs fois dans une recette, il n'y a qu'un enregistrement dans la table BXT_MRecipeElement, mais il y a plusieurs enregistrements dans la table BXT_MRecipeStep, un enregistrement pour chaque utilisation du RE.

Table 37 – BXT_MRecipeStep

Attribut	Description
<i>ParentRE</i>	Identifie le RE ou la recette maître avec lequel/laquelle cette étape est associée
<i>ParentVersion</i>	Identifie la version du RE ou de la recette maître avec laquelle cette étape est associée. Lorsqu'il est combiné avec un «ParentRE,» ce champ définit une instance unique d'un RE
<i>StepID</i>	Identifie l'instance d'exécution unique du RE référencé, avec un nom unique pour le domaine d'application de son RE parent. (Un exemple pourrait être simplement le numéro d'étape dans le RE)
RE_ID	Identifie le nom du RE dont cette étape est une instance, avec un nom unique pour le domaine d'application de son RE parent
REVersion	Identifie la version du RE dont cette étape est une instance
VerticalStart	Spécifie la position de départ verticale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
VerticalStop	Spécifie la position d'arrêt verticale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
HorizontalStart	Spécifie la position de départ horizontale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
HorizontalStop	Spécifie la position d'arrêt horizontale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
ScaleReference	Spécifie la taille de référence pour les éléments de recette; toutes les valeurs de formule sont basées sur cette taille de référence (par exemple 1 234,5 kg)
ScaleEngrUnits	Spécifie les unités de ScaleReference
MaximumScale	Spécifie le facteur d'échelle maximum, ou taille, de l'élément de recette
MinimumScale	Spécifie le facteur d'échelle minimum, ou taille, de l'élément de recette

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:

- the procedural control element steps;
- the procedural control elements;
- allocation symbols and other graphical representation symbols;
- the parameterization of procedural elements with limits;
- the linkage between the elements, and
- 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 more than once 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
<i>ParentRE</i>	Identifies the RE or master recipe with which this step is associated
<i>ParentVersion</i>	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
<i>StepID</i>	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 that 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 co-ordinates 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 co-ordinates 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 co-ordinates 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 co-ordinates 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., 1 234,5 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 Elément de recette

La table BXT_MRecipeElement (voir table 36) contient un enregistrement pour chaque élément procédural référencé dans la recette maître échangée. Cette table contient la définition de l'élément, mais pas l'utilisation de l'élément. Dans la table, un enregistrement existe pour la procédure, pour chaque procédure d'unité, pour chaque opération et pour chaque phase de recette qui est échangée. Les tables BXT_MRecipeElement et BXT_MRecipeElementParameter contiennent les spécifications pour l'utilisation du RE, le nombre et les types de paramètres qui sont passés et les valeurs par défaut pour les paramètres.

Les RE doivent être uniques à un RE parent. Le RE_ID est le nom complètement qualifié d'un RE sous ses RE parents; par conséquent, le RE_ID est suffisant pour contenir tous les RE_ID des parents.

5.2.5.2.2 Transitions

La table BXT_MRecipeTransition contient un enregistrement pour chaque connexion de transition définie par les RE (voir table 38). Ces enregistrements correspondent aux transitions dans les diagrammes fonctionnels de procédures (voir article 6).

Table 38 – BXT_MRecipeTransition

Attribut	Description
<i>RE_ID</i>	Identifie le RE dans lequel est contenue la transition
<i>REVersion</i>	Identifie la version du RE. Lorsqu'elle est combinée avec une «RE_ID», elle définit une instance unique d'une recette maître
<i>TransitionID</i>	Identifie l'instance d'exécution unique de cet élément de transition. L'ID contient la hiérarchie complète de noms d'instance des RE parents desquels cette transition est un élément membre
Condition	Contient l'expression ou la condition qui, si elle est VRAIE cause ou permet la transition
VerticalStart	Position de départ verticale pour la présentation de cet élément dans la vue procédurale du RE, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
VerticalStop	Position d'arrêt verticale pour la présentation de cet élément dans la vue procédurale du RE, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
HorizontalStart	Position de départ horizontale pour la présentation de cet élément dans la vue procédurale du RE, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
HorizontalStop	Position d'arrêt horizontale pour la présentation de cet élément dans la vue procédurale du RE, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite

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
<i>RE_ID</i>	Identifies the RE the transition is contained in
<i>REVersion</i>	Identifies the version of the RE. When combined with a "RE_ID" defines a unique instance of a master recipe
<i>TransitionID</i>	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 which, if TRUE causes or allows the transition
VerticalStart	Vertical starting position in the presentation of this element in the procedural view of the RE, in normalized co-ordinates of (0,0) upper left to (1,1) lower right
VerticalStop	Vertical stopping position in the presentation of this element in the procedural view of the RE, in normalized co-ordinates of (0,0) upper left to (1,1) lower right
HorizontalStart	Horizontal starting position in the presentation of this element in the procedural view of the RE, in normalized co-ordinates of (0,0) upper left to (1,1) lower right
HorizontalStop	Horizontal stopping position in the presentation of this element in the procedural view of the RE, in normalized co-ordinates of (0,0) upper left to (1,1) lower right

5.2.5.2.3 Liens

La table BXT_MRecipeLink contient un enregistrement pour chaque connexion définie dans les RE et/ou les transitions (voir table 39). Ces enregistrements correspondent aux lignes qui connectent les éléments dans les diagrammes fonctionnels de procédures décrits dans l'article 6.

Table 39 – BXT_MRecipeLink

Attribut	Description
<i>RE_ID</i>	Identifie le RE avec lequel l'étape et/ou la transition est associée
<i>REVersion</i>	Identifie la version du RE. Lorsqu'elle est combinée avec une «RE_ID,» ce champ définit une instance unique d'un RE
<i>LinkID</i>	Spécifie une ID unique pour le lien, qui simplifie l'accès à la table
FromType	Conservé en tant qu'énumération, précise si le FromElement spécifie un StepID pour une étape ou une TransitionID pour une transition. Membre de l'énumération «LinkToType»
FromElement	Spécifie le nom de l'étape ou la TransitionID. L'ID contient la hiérarchie complète des noms d'instance d'un RE parent dans lequel cet élément est contenu. Ceci doit correspondre au nom de l'étape ou de la transition dans la table d'étape ou de transition
ToType	Conservé en tant qu'énumération, précise si le ToElement spécifie un StepID pour une étape ou une TransitionID pour une transition. Membre de l'énumération «LinkToType»
ToElement	Spécifie le nom de l'étape ou la TransitionID. L'ID contient la hiérarchie complète des noms d'instance d'un RE parent dans lequel cet élément est contenu. Ceci doit correspondre au nom de l'étape ou de la transition dans la table d'étape ou de transition
LinkType	Spécifie si le lien est un flux de contrôle procédural ou une association de transfert de matière. Membre de l'énumération «LinkType» comme Lien de Contrôle ou Lien de Transfert
VerticalStart	Spécifie la position de départ verticale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
VerticalStop	Spécifie la position d'arrêt verticale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
HorizontalStart	Spécifie la position de départ horizontale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
HorizontalStop	Spécifie la position d'arrêt horizontale pour la présentation de cet élément dans la vue procédurale du RE parent, coordonnées normalisées de (0,0) en haut à gauche à (1,1) en bas à droite
Depiction	Définit comment le lien est présenté, membre de l'énumération «LinkDepiction»
Evaluation Order	Définit l'ordre spécifique d'évaluation du lien (si requis) pour répondre à l'évaluation de gauche à droite des vérifications des transitions du PFC spécifiées dans l'article 6. Tous les liens à partir de la même étape vers de multiples transitions sont supposés être évalués dans l'ordre spécifié par le champ d'ordre. Les nombres les plus petits sont évalués en premier

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
<i>RE_ID</i>	Identifies the RE with which the step and/or transition is associated
<i>REVersion</i>	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a RE
<i>LinkID</i>	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 in 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 co-ordinates 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 co-ordinates 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 co-ordinates 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 co-ordinates of (0,0) upper left to (1,1) lower right
Depiction	Defines how the link is presented, from the enumeration set "LinkDepiction"
Evaluation Order	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 Paramètres

La table BXT_MRecipeElementParameter contient un enregistrement pour chaque paramètre pour chaque RE défini (voir table 40). Par exemple, un RE appelé *CHARGE* peut être défini avec deux paramètres: le type de matière à charger et la quantité à charger. Un enregistrement existerait dans la table BXT_MRecipeElement et deux enregistrements existeraient dans la table BXT_MRecipeElementParameter pour le RE *CHARGE*.

Table 40 – BXT_MRecipeElementParameter

Attribut	Description
<i>RE_ID</i>	Identifie le RE avec lequel le paramètre est associé
<i>REVersion</i>	Identifie la version du RE. Lorsqu'il est combiné à une «RE_ID,» ce champ définit une instance unique d'une recette maître
<i>ParameterID</i>	Identifie le paramètre de l'élément procédural. Noter que si le paramètre fait partie d'une structure, le champ ParentParamID est utilisé et le paramètre structuré fait alors partie du nom du paramètre. Par exemple, pour un paramètre structuré MINOR_CHARGES et un paramètre de BLUE_DYE, le ParameterID serait MINOR_CHARGES.BLUE_DYE si le caractère délimiteur est un point
ParentParamID	Identifie le paramètre structuré parent duquel ce paramètre est un membre. Ce champ sera NULL s'il ne s'agit pas d'un paramètre structuré
DataInterpretation	Définit comment la valeur de paramètre par défaut est interprétée (c'est-à-dire constante, référence ou équation), membre de l'énumération «ValueType»
DataDirection	Définit comment la valeur de paramètre est destinée à être manipulée (c'est-à-dire entrée, sortie, entrée/sortie, aucune), membre de l'énumération «DirectionType»
DefaultValue	Contient la valeur par défaut du paramètre utilisée si l'instance d'exécution ne spécifie pas de valeur, et elle peut être un membre d'une énumération d'utilisateur
ValueType	Définit le type de donnée de la valeur, membre de l'énumération «ValueDataTypes»
Description	Décrit le paramètre ou l'utilisation du paramètre dans le RE
EngrUnits	Identifie les unités de mesure d'ingénierie pour la valeur (par exemple <i>kg</i> , <i>livres</i>)
EnumSet	Identifie l'énumération dont cet élément est un membre (si non NULL)
DefaultScaling	Spécifie la sélection par défaut lorsqu'elle est définie comme une instance dans la table BXT_MrecipeStepParameter. Membre de l'énumération «Boolean». Si VRAI, la valeur de formule sera ajustée quand la taille du batch est ajustée. Si FAUX, la valeur de formule ne sera pas ajustée quand la taille du batch est ajustée. Une mise à l'échelle non-linéaire peut être effectuée grâce à l'utilisation d'expressions dans les valeurs de formule
ParamType	Spécifie la sélection par défaut lorsqu'elle est définie comme une instance dans la table BXT_MRecipeStepParameter. Ce champ identifie l'utilisation de la valeur de formule comme membre de l'énumération «FormulaType». (c'est-à-dire entrée de processus, sortie de processus ou un paramètre de processus). L'énumération est extensible par l'utilisateur
ParamSubType	Spécifie la sélection par défaut lorsqu'elle est définie comme une instance dans la table BXT_MRecipeStepParameter. Ce champ identifie l'utilisation de la valeur de formule comme membre de l'énumération FormulaSubType. Les éléments dans cet ensemble sont tous définis par l'utilisateur

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 be charged 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
<i>RE_ID</i>	Identifies the RE with which the parameter is associated
<i>REVersion</i>	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a master recipe
<i>ParameterID</i>	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., <i>kg</i> , <i>pounds</i>)
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. 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 Sous-paramètres standards

Cette norme reconnaît un ensemble de sous-paramètres qui peuvent être liés à une valeur de paramètre pour qualifier plus en détail sa définition et son utilisation. Par exemple, il peut être utile de définir et de transférer les limites haute et basse qu'un paramètre est autorisé à prendre. Ce type d'information est transféré dans les tables d'échange en définissant des sous-paramètres pour le paramètre en question. Les sous-paramètres sont définis pour un paramètre donné en créant un nouvel enregistrement de table avec le ParameterID en question utilisé comme ParentParamID. L'ensemble de sous-paramètres standards et la définition de leur utilisation est donnée dans la table 41. D'autres sous-paramètres peuvent aussi être définis par l'utilisateur. Cet ensemble de sous-paramètres devrait être supporté pour l'utilisation avec les tables BXT_MRecipeElementParameter, BXT_Mrecipe StepParameter (voir 5.2.5.3), BXT_EquipInterfaceParameter (voir 5.3.5.7) et BXT_Schedule Parameter (voir 5.4.3.4).

Table 41 – Sous-paramètres standards

ParameterID	Description
<i>HighValueLimit</i>	Spécifie la plus grande valeur que le paramètre associé peut prendre
<i>LowValueLimit</i>	Spécifie la plus petite valeur que le paramètre associé peut prendre
<i>HighTolerance</i>	Spécifie la plus grande déviation vers le haut permise de la valeur du paramètre associé
<i>LowTolerance</i>	Spécifie la plus grande déviation vers le bas permise de la valeur du paramètre associé

5.2.5.3 Formules

Les valeurs de composants qui sont à utiliser dans la production ou l'exécution d'un batch sont passées comme paramètres d'étape avec leurs limites. La table BXT_Mrecipe Step Parameter contient un enregistrement pour chaque paramètre utilisé pour chaque étape définie (voir table 42). Parce qu'il est permis que les RE évaluent les valeurs par défaut des paramètres, tous les paramètres d'un RE n'ont pas besoin d'être définis dans l'étape.

Table 42 – BXT_MRecipeStepParameter

Attribut	Description
<i>ParentRE</i>	Partie de la CLE de la table BXT_MRecipeStep pour ce paramètre
<i>ParentVersion</i>	Partie de la CLE de la table BXT_MRecipeStep pour ce paramètre
<i>StepID</i>	Identifie l'instance d'exécution unique du RE, avec un nom unique pour le domaine d'application de son RE parent
<i>ParameterID</i>	Identifie le paramètre du RE
<i>ParentParameterID</i>	Identifie l'ID du paramètre parent si défini. (Permet les paramètres structurés)
<i>ParameterValue</i>	Contient la valeur du paramètre, les nombres sont transférés comme représentations ASCII du nombre. Si le type de paramètre est une énumération d'utilisateur, alors ceci peut être un membre de l'énumération
<i>DataInterpretation</i>	Définit comment la valeur de paramètre est interprétée, comme un membre (constante, référence, externe, ou équation) de l'énumération «ValueType»
<i>Scaled</i>	Conservé en tant qu'énumération, membre de l'énumération «Boolean», si VRAI, alors la valeur de paramètre sera ajustée quand la taille du batch est ajustée. Si FAUX, alors la valeur de formule ne sera pas ajustée quand la taille du batch est ajustée

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 effected 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), the BXT_EquipInterfaceParameter table (see 5.3.5.7) and BXT_ScheduleParameter table (see 5.4.3.4).

Table 41 – Standard sub-parameters

ParameterID	Description
<i>HighValueLimit</i>	Specifies the largest value that the associated parameter is allowed to take on
<i>LowValueLimit</i>	Specifies the smallest value that the associated parameter is allowed to take on
<i>HighTolerance</i>	Specifies the largest upward deviation from the value of the associated parameter that is allowed
<i>LowTolerance</i>	Specifies the largest downward deviation from the value of the associated parameter that is allowed

5.2.5.3 Formulae

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
<i>ParentRE</i>	Part of the KEY to the BXT_MRecipeStep table for this parameter
<i>ParentVersion</i>	Part of the KEY to the BXT_MRecipeStep table for this parameter
<i>StepID</i>	Identifies the unique execution instance of the RE, with a name unique to the scope of its parent RE
<i>ParameterID</i>	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 Autres informations

Les autres informations doivent être transférées typiquement avec la recette (voir table 43). La signification des informations n'est pas spécifiée dans la définition de l'échange, mais c'est un accord entre l'expéditeur et le destinataire. Ces autres informations sont généralement une documentation supplémentaire ou une information descriptive qui peut être nécessaire pour échanger une recette maître valide, mais ce n'est pas une information nécessaire pour exécuter une recette. Des exemples d'autres informations peuvent inclure une documentation de conformité, des schémas de structure moléculaire ou même des images du produit attendu. La table autres informations du RE, nommée BXT_Mrecipe OtherInformation, contient ces données.

Parce que la structure et la forme de ces données sont si variables, le langage d'échange de recette fournit un moyen pour référencer, ou désigner, cette information. L'information réelle peut être échangée dans le champ DataValue si c'est une information textuelle, ou elle peut référencer d'autres fichiers dont les noms se trouvent dans les champs DataValue. L'échange des autres fichiers est en dehors du domaine d'application de cette norme.

Table 43 – BXT_MRecipeOtherInformation

Attribut	Description
<i>RE_ID</i>	Identifie le RE avec lequel l'«autre» donnée est associée
<i>REVersion</i>	Identifie la version du RE. Lorsqu'il est combiné avec une «RE_ID,» ce champ définit une instance unique du RE
<i>StepID</i>	Identifie l'instance d'exécution unique du RE si l'autre information est associée avec l'étape. Si NULL, alors l'autre information est associée avec le RE
<i>DataID</i>	Définit l'identification d'un élément de donnée avec d'autres informations
<i>DataType</i>	Identifie le type de valeur de donnée membre de l'énumération ValueDataType
<i>DataValue</i>	Spécifie la valeur des autres informations. Ce champ peut être le nom d'un fichier qui contient les données réelles. L'outil d'importation devrait avoir accès à ce fichier
<i>Description</i>	Décrit le type d'élément de donnée

5.2.5.5 Prescriptions d'équipement

Les prescriptions d'équipement du RE contiennent les contraintes et conditions d'équipement de la recette, et elles sont définies par des tables et des relations similaires à celles utilisées pour définir les capacités d'équipement en 5.3.

Les contraintes et les conditions d'équipement peuvent être associées à tout élément de recette dans la hiérarchie des éléments de recette. Les prescriptions d'équipement des RE définissent les contraintes et les conditions d'équipement applicables à ce niveau (par exemple au niveau de la procédure d'unité ou de l'opération).

Les prescriptions d'équipement sont contenues dans les tables BXT_MRecipeElementEquip et BXT_MRecipeStepEquip. La table BXT_MRecipeElementEquip contient la définition et la valeur par défaut de la propriété, et la table BXT_MRecipeStepEquip contient la valeur de propriété spécifique requise pour une étape.

5.2.5.5.1 Prescription d'équipement d'un RE

Un RE peut avoir des prescriptions d'équipement (par exemple «nom du réacteur» ou «revêtement du réacteur»), et celles-ci spécifient les propriétés prescrites pour le RE à exécuter. Les valeurs autorisées spécifiques pour la propriété pour une étape spécifique sont définies dans la table BXT_MRecipeStepEquip.

5.2.5.4 Other information

Typically other information has to be transferred with the recipe (see table 43). The meaning of the information is not specified in the exchange definition, but is agreed upon 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 is so variable, the recipe exchange language provides a means to reference, or designate, 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. The exchange of the other files is outside the scope of this standard.

Table 43 – BXT_MRecipeOtherInformation

Attribute	Description
<i>RE_ID</i>	Identifies the RE with which the "other" data is associated
<i>REVersion</i>	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of the RE
<i>StepID</i>	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
<i>DataID</i>	Defines the identification of a data element with other information
<i>DataType</i>	Identifies the type of the data value from the enumeration set ValueDataType
<i>DataValue</i>	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
<i>Description</i>	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 (for example, 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 (for example, "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.

Les prescriptions d'équipement peuvent définir des éléments de données que fournit la cellule de processus pour leur utilisation dans la recette. Ces éléments peuvent être utilisés dans les conditions de transition de la recette ou ailleurs dans des expressions. Des exemples d'éléments de données sont «PressionBallon» disponibles à partir d'une unité et «PressionVapeur» à partir d'un module d'équipement.

La table 44, nommée BXT_MRecipeElementEquip, définit cet ensemble de prescriptions d'équipement en relation.

Table 44 – BXT_MRecipeElementEquip

Attribut	Description
<i>RE_ID</i>	Identifie le RE pour lequel les prescriptions d'équipement sont nécessaires. Lorsqu'il est combiné avec la «version,» ce champ définit une instance unique du RE
<i>REVersion</i>	Identifie la version du RE. Lorsqu'il est combiné avec une «RE_ID,» ce champ définit une instance unique d'un RE
<i>PropertyID</i>	Définit la propriété d'équipement nécessaire pour exécuter le RE (par exemple «classe de réacteur» ou «tare de pression du ballon»)
DefaultValue	Définit la valeur par défaut pour la propriété, si aucune n'est spécifiée dans les étapes (par exemple «réacteur exothermique» or «pression ballon»)
DataInterpretation	Défini en tant qu'énumération, spécifie comment la valeur est interprétée (c'est-à-dire constante, référence, externe, ou équation). Membre de l'énumération «ValueType»
EvaluationRule	Défini en tant qu'énumération, spécifie comment la valeur est comparée à la valeur de propriété d'équipement. Membre de l'énumération «EvaluationRule»
EngrUnits	Identifie les unités de mesure d'ingénierie pour la valeur (par exemple <i>kg</i> , <i>livres</i>)
Description	Décrit ce qu'est la propriété, et peut inclure pourquoi elle est nécessaire pour la recette

5.2.5.5.2 Prescription d'équipement d'une étape d'un RE

Un BXT_MRecipeStep peut avoir une valeur spécifique pour une propriété d'équipement (par exemple «classe réacteur» = «exothermique» ou «revêtement réacteur» = «verre»). La table 45, nommée BXT_MRecipeStepEquip, définit cet ensemble de prescriptions d'équipement en relation.

Table 45 – BXT_MRecipeStepEquip

Attribut	Description
<i>ParentRE</i>	Partie de la CLE de la table BXT_MRecipeStep pour ce paramètre
<i>ParentVersion</i>	Partie de la CLE de la table BXT_MRecipeStep pour ce paramètre
<i>StepID</i>	Identifie l'instance d'exécution unique du RE, avec un nom unique pour le domaine d'application de son RE parent
<i>PropertyID</i>	Identifie la propriété du RE
PropertyValue	Contient la valeur de la propriété, les nombres sont transférés comme représentations ASCII du nombre. Si le type de paramètre est une énumération d'utilisateur, alors ceci peut être un membre de l'énumération

La «signification» des contraintes n'est pas importante pour la définition du langage; cependant, elle correspond à un accord sur la dénomination des ensembles ou entités d'équipement.

The equipment requirements 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
<i>RE_ID</i>	Identifies the RE for which the equipment requirements are needed. When combined with the "version," this field defines a unique instance of the RE
<i>REVersion</i>	Identifies the version of the RE. When combined with a "RE_ID," this field defines a unique instance of a RE
<i>PropertyID</i>	Defines the equipment property needed to execute the RE (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, is defined as an enumeration (i.e., constant, reference, external, or equation). From the enumeration set "ValueType"
EvaluationRule	Specifies how the value is compared to the equipment property value, is defined as an enumeration. From the enumeration set "EvaluationRule"
EngrUnits	Identifies the engineering units of measure for the value (e.g., <i>kg</i> , <i>pounds</i>)
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 (for example, "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
<i>ParentRE</i>	Part of the KEY to the BXT_MRecipeStep table for this parameter
<i>ParentVersion</i>	Part of the KEY to the BXT_MRecipeStep table for this parameter
<i>StepID</i>	Identifies the unique execution instance of the RE, with a name unique to the scope of its parent RE
<i>PropertyID</i>	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 Echange de modèle d'équipement de cellule de processus

Un ensemble de tables est défini qui décrit les capacités de l'équipement dans une cellule de processus. Cette information reflète les capacités réelles d'une cellule de processus. L'échange de capacités de cellule de processus peut être utile en tant qu'échange d'information même s'il n'est pas utilisé conjointement avec des recettes.

5.3.1 Description d'équipement

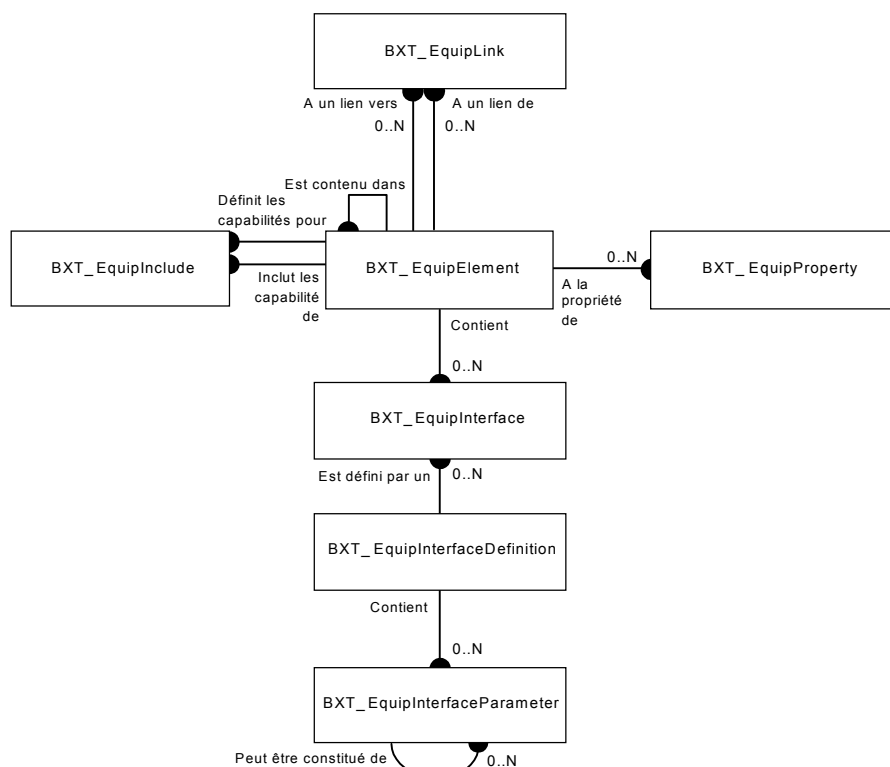
Les tables de capacités des cellules de processus sont organisées autour d'une hiérarchie de descriptions d'équipement. Les éléments dans la hiérarchie correspondent aux éléments dans la partie 1 de la hiérarchie d'équipement.

Les tables d'échange d'information d'équipement incluent les informations suivantes:

- Elément d'équipement – ceci définit un élément d'équipement spécifique ou une classe d'équipement.
- Interface d'élément procédural d'équipement – ceci définit l'interface d'un élément procédural disponible dans l'élément d'équipement.
- Propriété d'élément d'équipement – ceci définit les propriétés de l'équipement d'une manière qui reflète la spécification de propriété des propriétés d'équipement du RE.
- Lien d'éléments d'équipement – ceci définit la liaison entre équipements d'une manière qui reflète la spécification de propriété des liens d'équipement de RE de recette maître.

5.3.2 Vue d'ensemble de table et contraintes d'intégrité

La figure 18 montre la structure des tables d'information d'équipement.



IEC 2224/01

Figure 18 – Tables d'échange d'information d'équipement

5.3 Process cell equipment model exchange

A set of tables that describes the capabilities of the equipment in a process cell is defined. 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

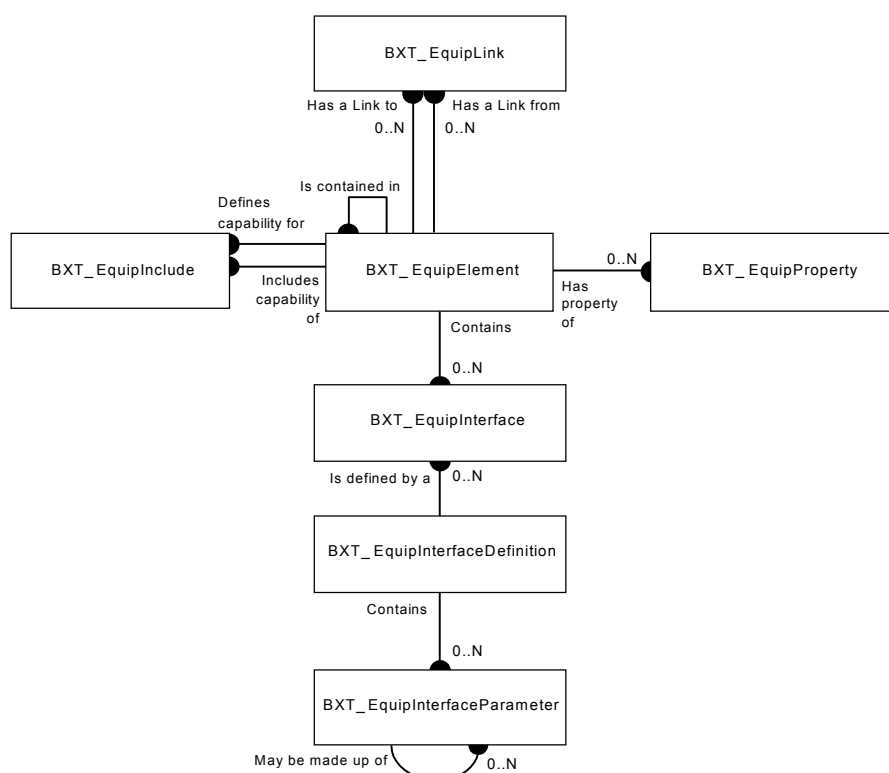
The process cell capability tables are organized around a hierarchy of equipment descriptions. The elements in the hierarchy correspond to the elements in the part 1 equipment hierarchy.

The equipment information exchange tables include the following information:

- Equipment element – this defines a specific equipment element or a class of equipment.
- Equipment procedural element interface – this defines the interface to a procedural element that is available in the equipment element.
- Equipment element property – this defines the properties of the equipment in a manner that mirrors the property specification of the RE equipment properties.
- 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.



IEC 2224/01

Figure 18 – Equipment information exchange tables

5.3.3 Vue d'ensemble de la table de description des équipements

5.3.3.1 Hiérarchie d'équipement

Les tables d'information d'équipement contiennent la définition hiérarchique des équipements, à travers l'attribut «ContainedIn» de la table BXT_EquipElement. Cette table décrit quel équipement est contenu à l'intérieur d'un autre élément d'équipement. Par exemple, un élément d'équipement d'une cellule de processus peut contenir des éléments d'équipement composés d'unités et/ou de modules d'équipement.

Chaque niveau peut avoir des spécifications de propriétés associées, des spécifications de données et des spécifications de lien. Chaque niveau peut aussi avoir des éléments procéduraux associés pour supporter le modèle complet défini dans la partie 1 de cette norme, mais ceux-ci seront associés typiquement avec des unités ou des modules d'équipement.

5.3.3.2 Classes d'équipement

Les tables d'information d'équipement permettent la spécification de classes d'équipement à travers l'association «inclus la capacité de» entre les éléments. Un élément d'équipement peut aussi inclure la capacité définie dans un ou plusieurs éléments d'équipement.

Par exemple, une unité peut être un membre d'une classe d'unité parce qu'elle applique un ensemble de procédures que définit la classe, parce qu'elle contient un ensemble de propriétés de la classe ou toute combinaison cette information. Un enregistrement existe dans la table BXT_EquipElement pour chaque définition d'instance d'équipement, et un enregistrement existe dans la table BXT_EquipElement pour chaque définition de classe.

5.3.4 Sommaire de la table d'information des équipements

Les tables utilisées pour l'échange d'information d'équipement sont énumérées dans la table 46.

Table 46 – Tables d'échange d'information d'équipement

Nom de la table	Nom complet	Description
BXT_EquipElement	Élément d'équipement	Un enregistrement pour chaque élément d'équipement ou classe d'équipement
BXT_EquipLink	Spécification de lien d'élément d'équipement	Un enregistrement pour chaque lien entre éléments d'équipement
BXT_EquipInclude	Élément d'équipement inclut	Un enregistrement pour chaque élément d'équipement qui appartient à une classe d'éléments d'équipement
BXT_EquipProperty	Spécification de propriété d'élément d'équipement	Un enregistrement pour chaque spécification de propriété et sa valeur pour un élément d'équipement
BXT_EquipInterface	Interface d'élément procédural d'équipement	Un enregistrement pour chaque élément procédural d'équipement défini à l'intérieur d'un élément d'équipement
BXT_EquipInterfaceDefinition	Définition d'un interface d'élément procédural d'équipement	Un enregistrement pour chaque classe d'interface d'élément procédural d'équipement définie à l'intérieur d'un élément d'équipement
BXT_EquipInterfaceParameter	Paramètre d'interface d'élément procédural d'équipement	Un enregistrement pour chaque entrée d'élément de données vers, ou de sortie de l'élément procédural d'équipement défini à l'intérieur d'un élément d'équipement

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 model defined in part 1 of this standard, 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 "includes 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 it's 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 Définitions de table d'équipements

Le paragraphe suivant contient les spécifications détaillées des tables d'information des équipements.

5.3.5.1 Élément d'équipement

La table BXT_EquipElement contient un enregistrement pour chaque entité d'équipement (c'est-à-dire cellule de processus, unité, module d'équipement, module de contrôle) (voir table 47). La table BXT_EquipElement contient les définitions des entités et des classes d'entité (par exemple «Racteur101», «Filtre20», «Réacteur», «Filtre»). La table BXT_EquipInclude définie ci-dessous est utilisée pour contenir les relations de classe.

Table 47 – BXT_EquipElement

Attribut	Description
<i>EquipmentID</i>	Identifie un élément d'équipement ou une classe d'équipement
EE_Type	Identifie le type d'enregistrement, membre (c'est-à-dire définition de classe ou d'élément) de l'énumération «EquipmentType»
EE_Level	Identifie le niveau de l'équipement, membre (c'est-à-dire zone, unité, module d'équipement, module de contrôle) de l'énumération «EquipmentLevel»
ContainedIn	Identifie l'élément d'équipement dans lequel cet équipement est contenu (par exemple la cellule de processus dans laquelle est contenue une unité). Ce champ peut être NULL si l'équipement n'est contenu dans aucun équipement ou si l'équipement contenant n'est pas défini
Description	Décrit l'élément d'équipement

5.3.5.2 Liens d'élément d'équipement

Un élément d'équipement peut avoir une spécification de liaison d'équipement (par exemple cet équipement peut alimenter d'autres équipements). Ces liens définissent typiquement les transferts de matière que supporte la cellule de processus (ou les unités). Par exemple, une recette peut spécifier qu'un MIXEUR est relié à un REACTEUR. La table BXT_EquipLink contient les liens d'équipements (voir table 48).

Table 48 – BXT_EquipLink

Attribut	Description
<i>EquipmentID</i>	Identifie un élément d'équipement ou une classe d'équipement
<i>ToEquipmentID</i>	Identifie un élément d'équipement ou une classe d'équipement que relie «EquipmentID»
Description	Décrit le type du lien d'élément d'équipement

5.3.5.3 Un élément d'équipement inclut la capacité

Un élément d'équipement peut inclure les capacités d'une ou plusieurs classes d'éléments d'équipement. Par exemple, il peut exister un élément d'équipement qui représente une classe de réacteurs avec un ensemble spécifique d'éléments procéduraux qui représentent la capacité de traitement des réacteurs. Une unité spécifique inclurait la capacité de la classe du réacteur, et elle peut y ajouter des capacités ou des spécifications supplémentaires. La table BXT_EquipInclude contient la relation entre la classe d'équipement et l'instance d'équipement (voir table 49).

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 (for example, "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
<i>EquipmentID</i>	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 (for example, 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
<i>EquipmentID</i>	Identifies an equipment element or class of equipment
<i>ToEquipmentID</i>	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 that represents a class of reactors with a specific set of procedural elements that represent the processing capability of reactors may exist. 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

Attribut	Description
<i>EquipmentID</i>	Identifie un élément d'équipement ou une classe d'équipement
<i>ClassEquipmentID</i>	Identifie une classe d'équipement à l'intérieur de laquelle l'EquipmentID est contenue
Description	Décrit l'association

5.3.5.4 Propriétés d'éléments d'équipement

Les propriétés d'éléments d'équipement définissent la capacité disponible à l'intérieur d'un élément d'équipement, et ceci est obtenu en spécifiant quel équipement est disponible, sous la même forme que celle spécifiée dans les prescriptions de recette, comme définition de la spécification et la valeur pour la spécification.

Les spécifications peuvent être associées avec tout élément d'équipement dans la hiérarchie d'équipement, et elles définissent les spécifications applicables au niveau considéré (par exemple au niveau de la cellule de processus, de l'unité ou du module d'équipement).

La table BXT_EquipProperty (voir table 50) contient un enregistrement pour chaque propriété que l'élément d'équipement met à disposition (par exemple «type de revêtement» et «Unité revêtue de verre», «Taille» et «50,000 gallon»). Les spécifications peuvent aussi inclure des éléments de données d'équipement disponibles à partir de l'équipement (par exemple «TempératureVapeur» d'un module d'équipement de chauffage, «PressionBallon» d'une unité). Les éléments de données peuvent être disponibles pour utilisation dans les transitions et expressions de la recette.

Table 50 – BXT_EquipProperty

Attribut	Description
<i>EquipmentID</i>	Identifie un élément d'équipement ou une classe d'équipement
<i>PropertyID</i>	Identifie une propriété que fournit cet équipement (par exemple «Type de revêtement», «Taille», «Capacité de chaleur», «Température de la vapeur»)
PropertyValue	Identifie la valeur pour la propriété (par exemple «Verre», «50 000», «650»)
EngrUnits	Spécifie les unités d'ingénierie de la propriété (par exemple «gallons», «BTU/hr»)
Description	Décrit le type de la propriété d'élément d'équipement

5.3.5.5 Interface d'élément procédural d'équipement

La table BXT_EquipInterface (voir table 51) contient un enregistrement pour chaque procédure d'élément d'équipement définie à l'intérieur d'un élément d'équipement. Chaque enregistrement BXT_EquipInterface fournit une correspondance (*mapping*) avec la définition de l'interface pour la procédure d'élément d'équipement associée. Comme de multiples BXT_EquipInterfaces (par exemple phases d'équipement) peuvent avoir exactement la même définition d'interface externe, ils peuvent tous référencer une définition d'interface BXT_EquipInterface unique. Cette structure permet la définition de BXT_EquipInterfaces fonctionnellement équivalents afin de prendre en compte les recettes basées sur une classe.

Table 49 – BXT_EquipInclude

Attribute	Description
<i>EquipmentID</i>	Identifies an equipment element or class of equipment
<i>ClassEquipmentID</i>	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, and 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 (for example, 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 (for example, "lining type" and "glass lined unit", "size" and "50,000 gallon"). Specifications may also include equipment data elements that are available from the equipment (for example, "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
<i>EquipmentID</i>	Identifies an equipment element or class of equipment
<i>PropertyID</i>	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", "50 000", "650")
EngrUnits	Specifies the 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 (for example, 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

Attribut	Description
<i>EquipmentID</i>	Identifie un élément d'équipement ou une classe d'équipement
<i>EPI_ID</i>	Identifie l'interface d'élément procédural d'équipement d'un élément procédural d'équipement
<i>EPI_Definition</i>	Identifie la définition du BXT_EquipInterface
Description	Décrit le type de l'élément procédural d'équipement

5.3.5.6 Définition de l'interface d'élément procédural d'équipement

La table BXT_EquipInterfaceDefinition (voir table 52) contient un enregistrement pour chaque interface d'élément procédural d'équipement défini. La table BXT_EquipInterfaceDefinition contient la définition des paramètres d'entrée et de sortie de BXT_EquipInterface.

Table 52 – BXT_EquipInterfaceDefinition

Attribut	Description
<i>EPI_Definition</i>	Identifie la définition de l'interface d'élément procédural d'équipement
Description	Décrit le comportement prévu du BXT_EquipInterface

5.3.5.7 Paramètre d'interface d'élément procédural d'équipement

La table BXT_EquipInterfaceParameter (voir table 53) contient un enregistrement pour chaque élément de donnée requis par, généré par ou modifié par une exécution de l'élément procédural d'équipement défini à l'intérieur d'un élément d'équipement. La table BXT_EquipInterfaceParameter contient la définition du type et des unités de l'élément de donnée requis, et une référence optionnelle à une énumération.

La table BXT_EquipInterfaceParameter indique également si la valeur d'entrée est susceptible d'être ajustée, et elle fournit une valeur par défaut qui serait utilisée si aucune valeur réelle n'est passée à l'élément procédural d'équipement.

Table 53 – BXT_EquipInterfaceParameter

Attribut	Description
<i>EPI_Definition</i>	Identifie la classe d'interface d'élément procédural d'équipement
<i>ParameterID</i>	Identifie le nom du paramètre utilisé par un élément procédural d'équipement
<i>ParentParamID</i>	Identifie le paramètre structuré parent duquel ce paramètre est un membre, et ce champ est NULL s'il ne s'agit pas d'un paramètre structuré
Type	Identifie le type de la donnée utilisé par l'élément procédural d'équipement, membre de l'énumération ValueDataType
EngrUnits	Identifie les unités d'ingénierie de la donnée utilisée par l'élément procédural d'équipement
EnumSet	Identifie l'énumération dont cet élément est un membre (si non NULL)
Scaled	Spécifie si le paramètre peut être mis à l'échelle avant qu'il ne soit passé à l'élément procédural d'équipement. Membre de l'énumération «Boolean»: Si ce champ est VRAI, le paramètre peut être ajusté
DefaultValue	Identifie la valeur par défaut utilisée si aucune valeur n'est passée à l'élément procédural d'équipement
Description	Décrit le paramètre de l'élément procédural d'équipement

Table 51 – BXT_EquipInterface

Attribute	Description
<i>EquipmentID</i>	Identifies an equipment element or class of equipment
<i>EPI_ID</i>	Identifies the equipment procedural element interface of an equipment procedural element
<i>EPI_Definition</i>	Identifies the definition of the BXT_EquipInterface
<i>Description</i>	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
<i>EPI_Definition</i>	Identifies the equipment procedural element interface definition
<i>Description</i>	Describes the expected behaviour 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
<i>EPI_Definition</i>	Identifies the equipment procedural element interface class
<i>ParameterID</i>	Identifies the name of the parameter that is used by the equipment procedural element
<i>ParentParamID</i>	Identifies the parent parameter set of which this parameter is a member, and this field is NULL if there is no parameter set
<i>Type</i>	Identifies the type of the data that is used by the equipment procedural element, from the enumeration set ValueDataType
<i>EngrUnits</i>	Identifies the engineering units of the data that is used by the equipment procedural element
<i>EnumSet</i>	Identifies the enumeration set of which this element is a member (i.e., if not NULL)
<i>Scaled</i>	Specifies if the parameter may be scaled before it is passed to the equipment procedural element. From the enumeration set "Boolean". If this field is TRUE, the parameter may be scaled
<i>DefaultValue</i>	Identifies the default value that is used if no value is passed to the equipment procedural element
<i>Description</i>	Describes the equipment procedural element parameter

5.4 Echange des informations de programmation

Les tables de programmation définissent un ou plusieurs batchs programmés pour une cellule de processus donnée. Chaque batch programmé contient des informations supplémentaires spécifiques au batch qui peuvent être utilisées avec l'information de la recette maître afin de créer une recette de contrôle. Chaque batch programmé peut définir un ensemble de valeurs de paramètre nécessaires à la création d'une recette, et un ensemble de prescriptions d'équipement. Les tables d'échange d'information de programmation ne fournissent pas nécessairement toute l'information utile pour exécuter une recette de contrôle; des informations complémentaires peuvent être fournies par le système de contrôle ou l'opérateur.

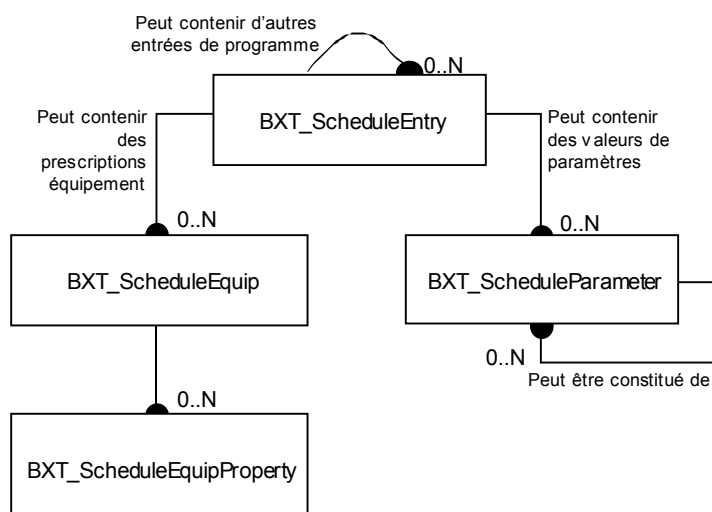
Les activités de programmation peuvent exiger d'autres informations en plus de celles fournies par ces tables. La programmation peut nécessiter l'utilisation d'autres moyens pour obtenir certaines données (par exemple le statut d'équipement, les stocks de matière, les conditions d'utilités). Cette information sera avantageusement échangée par d'autres méthodes.

5.4.1 Vue d'ensemble des tables de programmation

Les tables d'échange d'information de programmation permettent d'inclure les informations de multiples batchs programmés dans un ensemble unique de tables.

Les tables d'échange d'information de programmation ne spécifient pas comment l'information est créée ou comment elle est utilisée. Les outils qui utilisent l'information peuvent inclure des logiciels d'ordonnancement, des logiciels d'automatisation de batch, des logiciels de visualisation des opérations et des logiciels de suivi des ordres de travaux. Les outils d'importation et d'exportation doivent déterminer l'utilisation appropriée des informations de programmation dans les tables.

La figure 19 montre les cinq tables qui forment les tables d'échange d'information de programmation.



IEC 2225/01

Figure 19 – Structure du programme

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.

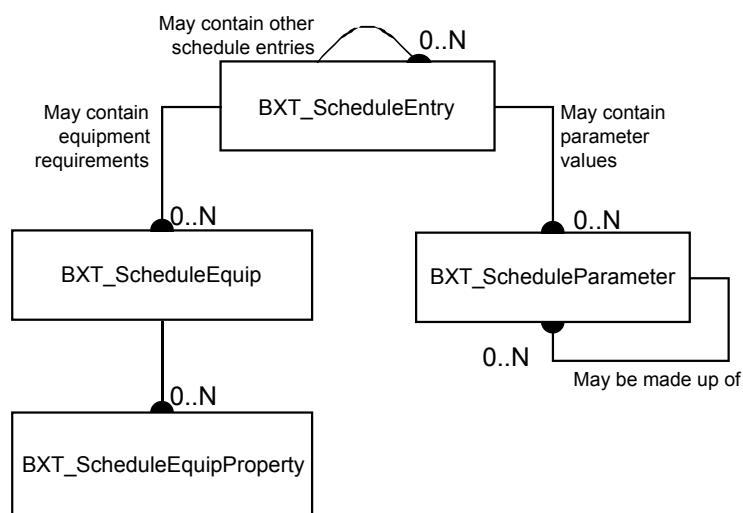
Scheduling activities may require information in addition to that provided by these tables. Scheduling may need to use other means in order to obtain some of the data (for example, the status of equipment, material inventories, utility conditions). This information is best exchanged by other methods.

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.



IEC 2225/01

Figure 19 – Schedule structure

5.4.2 Sommaire des tables de programmation

Les tables utilisées pour l'échange d'information de programmation sont présentées dans la table 54.

Table 54 – Tables d'échange d'information de programmation

Nom de la table	Description
BXT_ScheduleEntry	Un enregistrement pour chaque élément programmé (c'est-à-dire un batch, une recette d'unité ou un événement de nettoyage)
BXT_ScheduleEquip	Un enregistrement pour chaque prescription de sélection d'équipement. Des sélections d'équipement permises peuvent être définies
BXT_ScheduleProperty	Un enregistrement pour chaque spécification de propriété pour chaque prescription d'équipement
BXT_ScheduleParameter	Un enregistrement pour chaque élément de paramètre de l'enregistrement de programmation

5.4.3 Définitions des tables de programmation

5.4.3.1 Entrée de programme

La table BXT_ScheduleEntry (voir table 55) contient un élément pour chaque événement programmé. Des entrées de programme peuvent représenter un batch ou une autre activité de traitement (par exemple une recette d'unité). Si l'événement programmé est un batch, la table contient l'identification du batch associée au batch programmé (c'est-à-dire si attribuée à ce moment) dans le champ ScheduleEntryID, et la recette maître qui est associée au batch programmé est identifiée dans le champ RE_ID. Le champ ScheduleEntryString est utilisé pour identifier de façon unique des entrées de programme quand leur ID «réelle» n'a pas encore été attribuée.

Table 55 – BXT_ScheduleEntry

Attribut	Description
<i>ScheduleEntryID</i>	Identifie l'ID unique (à l'intérieur de cette table) de l'entrée de programme. Peut être une ID de campagne, une ID de batch, une ID de procédure d'unité, ou une chaîne unique sans aucune signification externe
ParentSchedID	Identifie l'enregistrement d'un élément programmé parent auquel cet enregistrement est lié en utilisant le ScheduleEntryString du parent
<i>ExternalID</i>	Définit l'identificateur utilisé par la gestion pour identifier cette entrée de programme
<i>RE_ID</i>	Identifie l'élément de recette (par exemple Red Oak). Lorsqu'il est combiné avec la «version», ce champ définit une instance unique d'un RE. Quand l'enregistrement représente la recette maître, ce champ contient l'ID de la recette maître. (Identifie l'élément de recette référencé par cette entrée de programme)
<i>REVersion</i>	Identifie la version de la recette maître
SE_Type	Définit le type d'entité que représente cet enregistrement de programmation, membre de l'énumération «SE_Type». Cette définition permet à un enregistrement de batch d'avoir une information de programmation plus détaillée à des niveaux inférieurs dans la hiérarchie de procédure (c'est-à-dire les prescriptions d'équipement et de propriété peuvent être identifiées pour chaque procédure d'unité dans la recette) ainsi que de traiter l'ordonnement de campagnes et de groupes de batches
BatchID	Identifie l'ID du batch duquel cet élément de programme fait partie
LotID	Identifie l'ID du lot duquel cet élément de programme fait partie
CampaignID	Identifie l'ID de la campagne de production de laquelle cet élément de programme fait partie
ProductID	Identifie le produit à fabriquer
OrderID	Identifie l'ordre de production ou la ou les commandes du client avec lesquels cet enregistrement de programme est lié

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 (for example, 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
<i>ScheduleEntryID</i>	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
<i>ExternalID</i>	Defines the identifier that is used by the business to identify this schedule entry
<i>RE_ID</i>	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)
<i>REVersion</i>	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 of which this schedule item is a part
LotID	Identifies the ID of the lot of which this schedule item is a part
CampaignID	Identifies the ID of the production campaign of which this schedule item is a part
ProductID	Identifies the product to be made
OrderID	Identifies the production order or customer order(s) to which this schedule record is related

Table 55 (suite)

Attribut	Description
SE_Action	Définit l'action prévue par l'outil récepteur, membre (c'est-à-dire nouveau, mise à jour, suppression, défini par l'utilisateur) de l'énumération «ScheduleAction»
SchedStatus	Définit le statut de l'enregistrement de programmation, (c'est-à-dire effectué, en cours, programmé, programmation suspendue) membre de l'énumération «ScheduleStatus»
StartCondition	Spécifie la condition de départ prévue de l'enregistrement de programme, si elle est connue (par exemple «démarré avant...», «suit...»)
InitialMode	Définit le mode dans lequel l'enregistrement de programme commence l'exécution, (c'est-à-dire automatique, semi-automatique, manuel) membre de l'énumération «ScheduleMode»
SchedStartTime	Spécifie le moment de départ prévu de l'enregistrement de programme, si connu
SchedEndTime	Spécifie le moment de fin prévu de l'enregistrement de programme, si connu
BatchPriority	Spécifie une priorité placée sur l'enregistrement de programme, si connue. Les plus petits nombres ont la plus haute priorité (par exemple la priorité 1 est plus importante que la priorité 7)
BatchSize	Définit la taille requise, ou le facteur d'échelle, pour le batch, basé sur le facteur d'échelle pour le batch comme défini dans la recette maître
EngrUnits	Identifie de façon optionnelle les unités de mesure d'ingénierie pour le BatchSize
SENote	Fournit une information ou des instructions aux opérations
Description	Décrit l'élément de programmation et/ou le produit (par exemple <i>Premium Beer</i>)

5.4.3.2 Prescription d'équipement dans un enregistrement de programme

La table BXT_ScheduleEquip (voir table 56) contient un élément pour chaque prescription d'équipement pour un enregistrement de programme. La table BXT_ScheduleEquipProperty relative contient une définition de propriété spécifique que cet équipement doit satisfaire.

Des prescriptions typiques pour un batch peuvent être des sélections d'équipement. Des prescriptions et des propriétés d'équipement d'un enregistrement de programme correspondront généralement aux prescriptions et propriétés d'équipement d'une recette maître. Par exemple, un logiciel d'ordonnancement peut spécifier qu'une unité particulière soit utilisée dans la production. Le BXT_ScheduleEquip spécifierait l'identité de l'équipement comme défini dans un RE, et le BXT_ScheduleProperty spécifierait le nom de l'unité sélectionnée.

Table 56 – BXT_ScheduleEquip

Attribut	Description
<i>ScheduleEntryID</i>	Identifie l'élément de programme échangé
<i>RequirementID</i>	Fournit un nom unique pour la prescription d'équipement d'un élément du programme. Ce nom peut se référer à un élément d'équipement individuel, une classe d'équipement, une liste d'équipement permis ou d'autres groupes d'équipement (par exemple un train)
Description	Décrit la prescription (par exemple <i>Première unité de réaction dans le batch</i>)

5.4.3.3 Prescription de propriété d'équipement dans un enregistrement de programme

La table BXT_ScheduleProperty (voir table 57) contient un élément pour chaque spécification de propriété pour chaque prescription d'équipement pour un enregistrement de programme.

Comme une prescription d'équipement unique pour un élément programmé donné peut avoir plusieurs critères de propriété (par exemple matériau de construction et volume), une table séparée est utilisée pour énumérer ces prescriptions.

Table 55 (Continued)

Attribute	Description
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...", "follows...")
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 finishing 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., <i>Premium Beer</i>)

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
<i>ScheduleEntryID</i>	Identifies the scheduled item that is exchanged
<i>RequirementID</i>	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., <i>First reactor unit in the batch</i>)

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 (for example, material of construction and volume), a separate table is used to list these requirements.

Table 57 – BXT_ScheduleProperty

Attribut	Description
<i>ScheduleEntryID</i>	Identifie l'élément de programme échangé
<i>RequirementID</i>	Identifie l'ensemble de prescriptions associées au batch, typiquement un équipement ou une classe de matière
<i>PropertyName</i>	Identifie le nom de la propriété pour l'élément de programme
PropertyValue	Spécifie la valeur de la propriété pour l'élément de programme
EngrUnits	Identifie de façon optionnelle les unités de mesure d'ingénierie pour la PropertyValue
Description	Une description de la propriété (ex. Utiliser UNIT345 comme première unité de réaction dans le batch)

5.4.3.4 Paramètre de programmation

La table BXT_ScheduleParameter (voir table 58) contient un élément pour chaque paramètre pour un élément de programme. Les paramètres dans un élément de programme sont typiquement des paramètres dans la recette maîtresse, mais ils peuvent aussi être une information pour l'opérateur ou pour d'autres utilisateurs de l'information de programmation.

Les limites pour les éléments de paramètre de programmation peuvent être définies de la même manière que les sous-paramètres définissent plus en détail les paramètres dans des recettes.

Table 58 – BXT_ScheduleParameter

Attribut	Description
<i>ScheduleEntryID</i>	Identifie l'élément de programme échangé
<i>ParameterID</i>	Identifie le paramètre pour le batch programmé
ParentParameterID	Identifie l'ensemble de paramètres parents duquel cet élément de paramètre est un membre (NULL s'il n'y en a aucun)
ParameterValue	Spécifie la valeur pour le paramètre pour le batch programmé
EngrUnits	Identifie de façon optionnelle les unités de mesure d'ingénierie pour la ParameterValue
ItemLocation	Définit où, dans la structure de la recette, cet élément de paramètre est appliqué. Cet enregistrement est utilisé lorsque le ParameterID est un simple alias, et que le ParameterID ne fournit pas une information suffisante pour identifier où il est appliqué
EnumSet	Identifie l'énumération de laquelle cet élément est un membre (si non NULL)
Description	Décrit le paramètre pour l'élément de programmation en relation et/ou le produit (par exemple <i>Premium Beer</i>)

5.5 Echange d'informations de production

L'échange d'informations de production fournit une structure afin d'échanger toute l'information sur l'exécution d'un batch.

Beaucoup d'outils (par exemple systèmes d'automatisation de batch, systèmes de gestion de l'information de laboratoire, systèmes d'édition de rapports de batch, systèmes d'analyse de batch, systèmes d'ordonnancement, systèmes de simulation) peuvent utiliser cette information.

La structure des tables d'échange permet d'échanger dans les mêmes tables des données relatives à de multiples batches.

Table 57 – BXT_ScheduleProperty

Attribute	Description
<i>ScheduleEntryID</i>	Identifies the scheduled item exchanged
<i>RequirementID</i>	Identifies the requirement set associated with the batch, typically an equipment or material class
<i>PropertyName</i>	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
<i>ScheduleEntryID</i>	Identifies the scheduled item that is exchanged
<i>ParameterID</i>	Identifies the parameter for the scheduled batch
ParentParameterID	Identifies the parent parameter set of which this parameter item is a member (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., <i>Premium Beer</i>)

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 (for example, 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.

L'information de production est contenue dans trois domaines:

- a) la recette de contrôle;
- b) les équipements sur lesquels la recette a été exécutée;
- c) une capture de tous les événements qui se sont produits durant l'exécution de la recette.

5.5.1 Informations de recette de contrôle

L'information de la recette de contrôle peut être échangée en utilisant les tables MR, avec ProductID comme représentation de l'identification du batch.

Parce que la recette de contrôle débute comme la copie d'une recette maître, les tables MR peuvent contenir la recette maître utilisée pour la production, et la table d'événement peut contenir tous changements appliqués à la recette de contrôle, ou les tables MR peuvent contenir la recette de contrôle après modification. Dans chacun de ces cas, les informations de RecipeID et REVersion servent à identifier la recette maître spécifique qui a été utilisée pour créer la recette de contrôle.

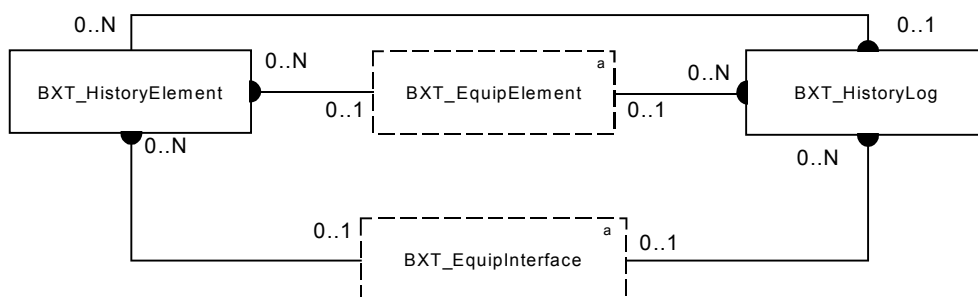
5.5.2 Informations d'équipement

Les équipements utilisés dans la production du batch peuvent être échangés en utilisant les tables BXT_EquipElement (entité d'équipement). Les tables peuvent contenir la définition de la cellule de processus complète qui a été utilisée au cours de la production, ou elles peuvent contenir seulement le sous-ensemble d'équipement réel qui a été utilisé dans la production du batch.

5.5.3 Historique du batch

L'historique du batch est contenu dans deux tables, la table BXT_HistoryElement et la table BXT_HistoryLog.

La table BXT_HistoryElement est l'enregistrement de chaque exécution d'un élément procédural de recette et/ou de l'élément procédural d'équipement équivalent. Cette table contient un élément pour chaque exécution d'un RE ou d'un EPE. La figure 20 montre les éléments d'historique du batch et la relation aux tables d'élément d'équipement définies précédemment. La table BXT_HistoryElement (voir table 59) contient des enregistrements qui peuvent être utilisés pour référencer l'équipement et l'élément procédural d'équipement associés.



^a Défini dans 5.5.2

IEC 2226/01

Figure 20 – Historique du batch

Production information is contained in three areas:

- a) the control recipe;
- b) the equipment on which the recipe ran;
- 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.

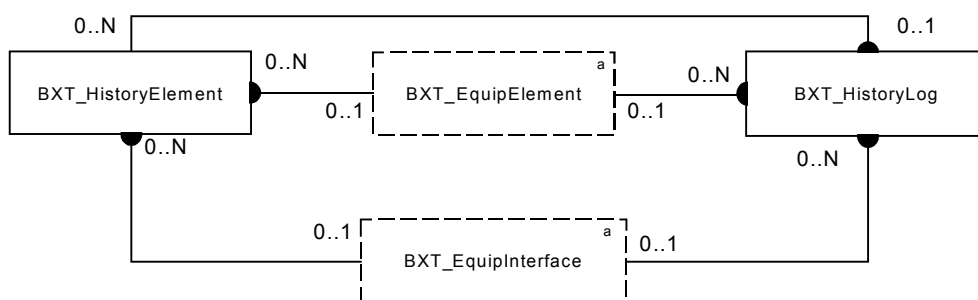
5.5.2 Equipment information

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 that was actually 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.



^a Defined in 5.5.2

IEC 2226/01

Figure 20 – Batch history

Les tables BXT_HistoryLog et BXT_HistoryElement contiennent une liste d'enregistrements horodatés créés pendant la production du batch, un enregistrement pour chaque événement enregistré dans le batch. La table BXT_HistoryElement contient toutes les instances d'utilisation d'un élément procédural de recette ou d'un élément procédural d'équipement. La table BXT_HistoryLog contient un enregistrement pour chaque événement qui s'est déroulé pour l'élément procédural (par exemple le début d'exécution de l'élément procédural, un changement de mode, un changement d'état, une valeur rapportée par l'élément).

Les tables d'historique du batch sont conçues avec plusieurs objectifs:

- Une information qui serait dupliquée dans plusieurs enregistrements est déplacée vers la table BXT_HistoryElement, ce qui permet une réduction considérable de la taille de la table BXT_HistoryLog.
- Une information qui peut être décrite dans les tables d'équipement est seulement référencée par une valeur clé. L'enregistrement BXT_HistoryElement contient une identification de l'élément d'équipement à l'équipement associé et une identification de l'élément procédural d'équipement associé.
- La table BXT_HistoryLog inclut également une référence à l'élément d'équipement et à l'élément procédural d'équipement afin de simplifier l'utilisation de la table BXT_HistoryLog, même si cette information est dupliquée dans la table BXT_HistoryElement.

5.5.3.1 Élément d'historique

La table BXT_HistoryElement contient un enregistrement pour chaque élément de recette (voir table 59).

Table 59 – BXT_HistoryElement

Attribut	Description
<i>HistoryElementID</i>	Fournit un nombre d'identification généré requis pour l'intégrité relationnelle
BatchID	Spécifie une identification unique du batch associé à l'enregistrement BXT_HistoryElement, et ceci est une information dupliquée avec l'enregistrement de la table BXT_HistoryLog
MasterRecipeID	Spécifie l'identification de la recette maître associée au batch
MasterRecipeVersion	Spécifie l'identification de la version de la recette maître associée
ControlRecipeID	Identifie l'ID de la recette de contrôle. Dans certains cas, cette identification peut être différente de l'ID du batch
ReferenceEquip Procedure	Identifie si la hiérarchie de contrôle procédural est définie pour faire référence à une recette ou un équipement, membre de l'énumération Boolean avec VRAI identifiant une référence à un équipement
RecipeProcedure	Identifie la procédure associée à l'enregistrement BXT_HistoryElement
UnitProcedure	Identifie la procédure d'unité associée à l'enregistrement BXT_HistoryElement
UnitProcedureCounter	Spécifie une instance de compteur d'exécution, qui est le nombre de fois que la procédure d'unité a été exécutée. Ce compteur est nécessaire parce que la recette peut exécuter la même procédure d'unité plusieurs fois en raison d'une intervention manuelle ou d'un bouclage
Operation	Identifie l'opération associée à l'enregistrement BXT_HistoryElement
OperationCounter	Spécifie une instance de compteur d'exécution, qui est le nombre de fois que la l'opération a été exécutée. Ce compteur est nécessaire parce que la recette peut exécuter la même opération plusieurs fois en raison d'une intervention manuelle ou d'un bouclage
Phase	Identifie la phase de recette associée à l'enregistrement BXT_HistoryElement

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 table contains each instance of a 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 (for example, 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:

- 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.
- 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.
- 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
<i>HistoryElementID</i>	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, and 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 the 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
ReferenceEquip Procedure	Identifies whether the procedural control hierarchy is defined in order to refer 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

Table 59 (Suite)

Attribut	Description
PhaseCounter	Spécifie une instance de compteur d'exécution, qui est le nombre de fois que la phase a été exécutée. Ce compteur est nécessaire parce que la recette peut exécuter la même phase plusieurs fois en raison d'une intervention manuelle ou d'un bouclage
EquipmentID	Identifie l'élément d'équipement qui peut être associé à l'enregistrement BXT_HistoryElement, ceci est une information dupliquée avec l'enregistrement de la table BXT_HistoryLog
EPI_ID	Identifie l'élément procédural d'équipement qui peut être associé à l'enregistrement, et ceci est une information dupliquée avec l'enregistrement de la table BXT_HistoryLog

5.5.3.2 Capture de l'historique

La table BXT_HistoryLog (voir table 60) contient cinq séries d'informations sur l'événement qui devrait être enregistré:

- le moment de l'événement,
- l'information du batch et de la recette associés à l'événement,
- l'équipement associé à l'événement,
- l'opérateur associé à l'événement, et
- l'information sur l'événement.

La table BXT_HistoryLog contient des événements d'information de production. Les événements d'information de production sont classés par catégories en utilisant des énumérations dans les champs RecordSet et RecordSubSet pour faciliter la manipulation de l'information (par exemple filtrage et classement).

Table 60 – BXT_HistoryLog

Attribut	Description
RecordID	Spécifie un nombre d'identification généré requis pour l'intégrité relationnelle
UTC	Identifie le temps universel coordonné (TUC) et la date de l'enregistrement
LocalTime	Identifie l'heure et la date locales de l'enregistrement
BatchID	Fournit une identification unique du batch associé à l'enregistrement
HistoryElementID	Fournit une identification unique de l'instance d'exécution ou de l'élément de recette associé ou de l'élément procédural d'équipement qui peut être associé à l'enregistrement. Ce champ est une clé dans la table BXT_HistoryElement
EquipmentID	Identifie un élément d'équipement qui peut être associé à l'enregistrement
EPI_ID	Identifie l'élément procédural d'équipement qui peut être associé à l'enregistrement
UserID	Spécifie le nom de l'utilisateur, s'il y en a un, qui est associé à l'enregistrement
RecordSet	Spécifie le type de l'enregistrement, membre de l'énumération RecordSet
RecordSubSet	Spécifie le sous-type de l'enregistrement membre de l'énumération spécifié par l'enregistrement RecordSet
RecordAlias	Définit une spécification d'enregistrement indépendante de l'équipement (par exemple «température de haut de cuve»)
NewValue	Spécifie la valeur de donnée associée au type et sous-type d'enregistrement
OldValue	Définit un champ qui peut contenir la valeur de donnée précédente
EngrUnits	Spécifie les unités d'ingénierie, s'il y en a, appropriées pour NewValue et OldValue.

Table 59 (Continued)

Attribute	Description
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:

- the time of the event,
- the batch and recipe information that is associated with the event,
- the equipment that is associated with the event,
- the operator that is associated with the event, and
- the event information.

The BXT_HistoryLog table contains production information events. The production information events are categorized using enumerations in the RecordSet and RecordSubSet fields to aid information handling (for example 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 co-ordinated 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 Domaines de table d'échange

La même information est contenue dans des tables multiples pour des champs clé, et la plupart de ces champs dans les tables ont le même domaine (c'est-à-dire type de donnée et étendue spécifiques). La norme ISO/CEI 9075 pour SQL ne définit pas de domaines, ainsi la table 61 contient la définition des domaines pour des attributs de table sélectionnés dans les tables d'échange.

Table 61 – Domaines de table d'échange

Nom du domaine	Type	Description
BXT_MRecipeElement-Author	CHAR (32)	Nom ou ID de l'auteur
BXT_MRecipeStep-StepID BXT_MRecipeStepParameter-StepID BXT_MRecipeStepEquip-StepID BXT_ScheduleEntry-StepID	CHAR (128)	Identifie une étape à l'intérieur d'un élément procédural de recette
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)	Spécification d'unités d'ingénierie
BXT_MRecipeElement-Status BXT_ScheduleEntry-SchedStatus	INTEGER	Statut de la recette ou élément procédural comme énumération
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)	Identifie un élément procédural de recette
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)	Identificateur de version pour tous les éléments avec des versions

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). ISO/IEC 9075 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

Table 61 (suite)

Nom du domaine	Type	Commentaire
BXT_MRecipeElementParameter-ParameterID BXT_MRecipeElementParameter-ParentParamID BXT_MRecipeStepParameter-ParameterID BXT_MRecipeStepParameter-ParentParamID BXT_EquipInterfaceParameter-ParameterID BXT_EquipInterface-ParentParamID BXT_ScheduleParameter-ParameterID	CHAR (32)	Identifie un paramètre d'un RE, définition d'utilisation
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)	Identifie une cellule de processus ou une autre entité d'équipement

6 Diagrammes fonctionnels de procédures

Cet article définit une méthode pour la représentation graphique des recettes maître et de contrôle. La représentation de la procédure est appelée un diagramme fonctionnel de procédure (PFC). Cet article décrit aussi les prescriptions pour la représentation de la formule, des prescriptions d'équipement, de l'en-tête et des autres informations. Le langage PFC tel que défini dans cette norme est conçu pour supporter des recettes avec des procédures complexes (par exemple étapes parallèles, sélections) qui varient d'un produit à l'autre.

Les diagrammes fonctionnels de procédures sont développés à partir des diagrammes fonctionnels tels que définis dans la CEI 60848. Dans les diagrammes fonctionnels de procédures, on présume que les étapes suivent les transitions et les transitions suivent les étapes comme décrit dans la CEI 60848. Cependant, il existe des différences significatives entre cette norme et la présente qui sont nécessaires afin de satisfaire aux exigences du contrôle procédural par opposition à une documentation. Dans le contrôle batch, les éléments procéduraux d'équipement contiennent la logique procédurale qui régule quand et comment ils s'accomplissent. De plus, beaucoup de phases d'équipement dans un contrôle batch sont conçues pour accomplir une tâche et ensuite se terminer. L'approche PFC supporte la séparation des éléments procéduraux de recette et des éléments procéduraux d'équipement en reconnaissant que ces éléments procéduraux d'équipement, une fois lancés, s'exécutent indépendamment. Une autre différence qui doit être prise en compte dans un diagramme fonctionnel de procédure est la structure multi-niveaux des éléments procéduraux de recette; par conséquent, on doit voir clairement si un symbole représente une phase de recette, une opération de recette, une procédure d'unité de recette ou une procédure de recette complète.

Table 61 — (Continued)

Domain Name	Type	Comment
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
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 the 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 (for example, 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 which 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; it must therefore be clear whether a symbol represents a recipe phase, a recipe operation, a recipe unit procedure or an entire recipe procedure.

6.1 Notation de diagramme fonctionnel de procédure

Les diagrammes fonctionnels de procédures décrivent la logique procédurale en utilisant une série de symboles qui sont interconnectés par des liens directs afin de définir la séquence d'exécution des éléments procéduraux. L'exécution des éléments procéduraux peut se faire en série ou en parallèle, et l'exécution peut être dépendante d'une logique conditionnelle. Les activités qui sont décrites incluent l'exécution souhaitée de procédures d'unité, d'opérations, de phases et l'évaluation de transitions. En général, le flux d'exécution va du haut vers le bas et de la gauche vers la droite. Les diagrammes fonctionnels de procédures sont utilisés pour décrire la logique procédurale pour tous les niveaux de la recette: procédure de recette, procédure d'unité de recette et opération de recette.

6.1.1 Symboles

Un diagramme fonctionnel de procédure est défini par une série de symboles pour:

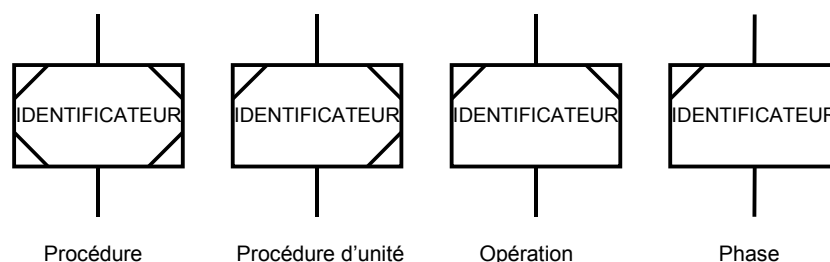
- les éléments (c'est-à-dire éléments procéduraux de recette);
- les points de début et de fin;
- l'allocation des ressources;
- la synchronisation d'éléments;
- les transitions de recette;
- les structures de base (c'est-à-dire liens dirigés, sélection de séquence, séquences simultanées).

Seule la représentation globale des symboles est imposée; les dimensions et les détails (par exemple épaisseur des ligne et polices de caractères) sont laissés à l'appréciation de chaque mise en œuvre.

6.1.1.1 Éléments

Les symboles doivent être utilisés pour représenter une phase de recette, une opération de recette, une procédure d'unité de recette ou une procédure de recette. Une indication graphique à l'intérieur du symbole doit être utilisée pour identifier le symbole comme représentant une phase de recette, une opération de recette, une procédure d'unité de recette ou une procédure de recette. Un exemple pour identifier des éléments procéduraux est proposé dans la figure 21.

La partie 1 de cette norme définit seulement quatre niveaux dans la hiérarchie procédurale, et seuls ces quatre niveaux sont décrits dans ce document. Cependant, des niveaux supplémentaires sont possibles, et ils peuvent être définis à des fins diverses. Alors que la présente norme ne concerne que les quatre niveaux d'éléments procéduraux définis dans la partie 1 de cette norme, tous niveaux supplémentaires définis séparément peuvent être identifiés soit graphiquement en suivant les mêmes principes généraux décrits dans cette norme, soit par une chaîne de caractères qui commence dans le coin supérieur gauche du rectangle.



IEC 2227/01

Figure 21 – Symboles des éléments procéduraux de la recette

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 the procedural elements. The execution of procedural elements may occur in series or in 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:

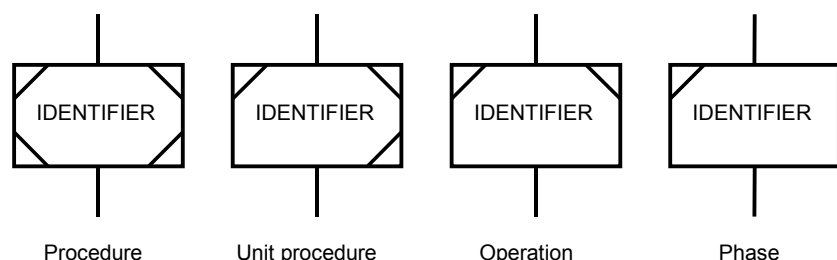
- elements (i.e., recipe procedural elements);
- begin and end points;
- resource allocation;
- element synchronization;
- recipe transitions;
- basic structures (i.e., directed links, sequence selection, simultaneous sequences).

Only the global representation of the symbols is imposed; dimensions and details (for example, 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.

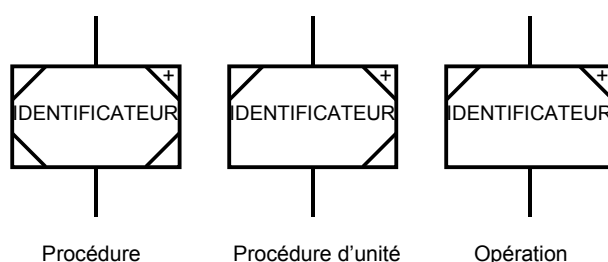
Part 1 of this standard defines only four levels in the procedural hierarchy, and only those four levels are described in this document. However, additional levels are possible, and they may be defined for various purposes. While this document only addresses the four levels of procedural elements that are defined in part 1 of this Standard, 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.



IEC 2227/01

Figure 21 – Recipe procedural element symbols

Un élément procédural au-dessus du niveau d'une phase peut représenter une encapsulation d'autres éléments procéduraux au niveau inférieur suivant dans la hiérarchie de contrôle procédural. Un élément procédural qui représente une encapsulation, où les éléments procéduraux de recette du niveau inférieur ne sont pas montrés, doit être identifié par un signe plus (+) dans le coin en haut à droite du rectangle représentant l'élément procédural (voir figure 22). Les symboles d'éléments procéduraux qui exposent l'encapsulation doivent être identifiés par un signe moins (–) (voir figure 38). Les symboles d'éléments procéduraux qui référencent un élément procédural d'équipement ne doivent porter aucune d'indication.



IEC 2228/01

Figure 22 – Éléments procéduraux qui encapsulent des éléments procéduraux de recette d'un niveau inférieur

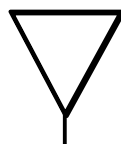
Lorsqu'un élément procédural représente une encapsulation d'éléments procéduraux subordonnés, un diagramme fonctionnel de procédure séparé de niveau inférieur qui spécifie les éléments procéduraux subordonnés et les symboles d'organisation associés peut être utilisé pour les définir. L'icône représentant l'élément procédural de recette encapsulant peut également être étirée pour permettre au diagramme fonctionnel de procédure du niveau inférieur d'être décrit à l'intérieur des limites du symbole encapsulant. Le processus d'expansion des symboles élémentaires peut continuer jusqu'à ce qu'il n'y ait plus de niveau subordonné. Une procédure d'équipement peut éventuellement être montrée comme une expansion de l'élément procédural de recette qui lui fait référence.

6.1.1.2 Points de début et de fin

Les diagrammes fonctionnels de procédures doivent avoir au moins un point de début et au moins un point de fin, par opposition à un diagramme fonctionnel séquentiel qui peut boucler continuellement.

6.1.1.2.1 Début

Au moins un symbole de début (voir figure 23) doit être utilisé pour désigner le commencement de chaque diagramme fonctionnel de procédure et/ou chaque diagramme fonctionnel de procédure subordonné.



IEC 2229/01

Figure 23 – Symbole de début

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.

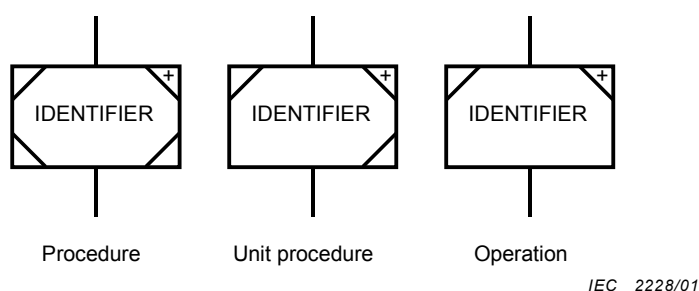


Figure 22 – Procedural elements that encapsulate lower-level recipe procedural elements

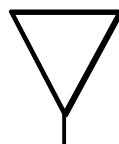
When a procedural element represents an encapsulation of subordinate procedural elements, a separate, lower-level procedure function chart that specifies the subordinate procedural elements and associated ordering symbols may be used to define it. The icon representing the encapsulating recipe procedural 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.



IEC 2229/01

Figure 23 – Begin symbol

6.1.1.2.2 Fin

Au moins un symbole de fin (voir figure 24) doit être utilisé pour désigner la fin de chaque diagramme fonctionnel de procédure et/ou chaque diagramme fonctionnel de procédure subordonné.

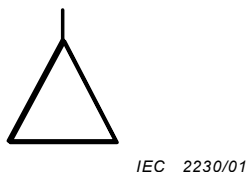


Figure 24 – Symbole de fin

6.1.1.3 Allocation de ressource

Certaines ressources sont allouées à un batch avant que le batch ne puisse les utiliser. Contrôler la planification de cette allocation peut être important pour la recette ou pour la programmation; par conséquent, les règles d'allocation des ressources et les conditions de départ sont nécessaires. L'allocation, si elle est montrée, doit être représentée par un pictogramme ovale (voir figure 25) qui représente l'encapsulation des prescriptions d'allocation de ressource pour une entité de recette. Le symbole d'allocation est un élément procédural de recette approprié au niveau du diagramme fonctionnel de procédure, et il peut être utilisé à tout niveau de la hiérarchie procédurale.

Le symbole d'allocation représente la donnée et/ou la logique qui détermine ce qui sera alloué au batch (par exemple quelle unité spécifique ou critère pour la sélection des unités, quels modules d'équipement, matières, personnel) et quand ils seront attribués au batch (par exemple deux heures après le départ d'une autre procédure d'unité). Le symbole d'allocation peut contenir une logique qui doit être exécutée par la gestion de processus ou à travers des éléments procéduraux d'équipement qui ne sont pas nécessairement associés à un équipement physique réel. Une transition explicite qui suit le symbole d'allocation peut être utilisée pour spécifier la condition de départ pour l'entité de recette qui suit.

Le symbole d'allocation peut aussi être utilisé pour spécifier une désallocation explicite. Dans ce cas, il convient d'utiliser une annotation de texte adéquate pour indiquer cette utilisation du symbole.

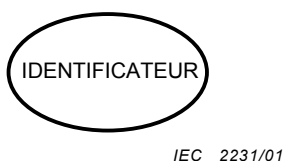


Figure 25 – Symbole d'allocation

6.1.1.4 Synchronisation d'éléments

On peut avoir besoin de représenter la synchronisation entre éléments de recette (voir figure 26). La synchronisation, si elle est montrée, doit être représentée par un rectangle qui sort de l'un quelconque des côtés du symbole pour chacun des éléments de recette impliqués dans la synchronisation. Les symboles de synchronisation correspondants peuvent être connectés par une ligne pointillée ou par un autre type de ligne qui peut être distinguée d'un lien direct lorsque l'emplacement des symboles permet une telle notation sans confusion. Si la synchronisation représente un transfert de matière, une flèche doit être ajoutée pour indiquer la direction du mouvement de matière désiré. Si aucune ligne n'est utilisée, un identificateur unique qui identifie l'interaction spécifique doit être fourni à chaque élément de recette impliqué dans la synchronisation.

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.

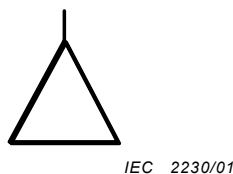


Figure 24 – End symbol

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; resource allocation rules and start conditions are therefore 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 (for example, which specific unit or criteria for unit selection, equipment modules, materials, personnel) and when it will be allocated to the batch (for example, 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 deallocation.

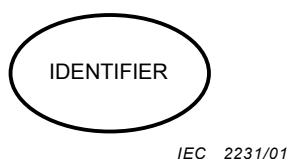
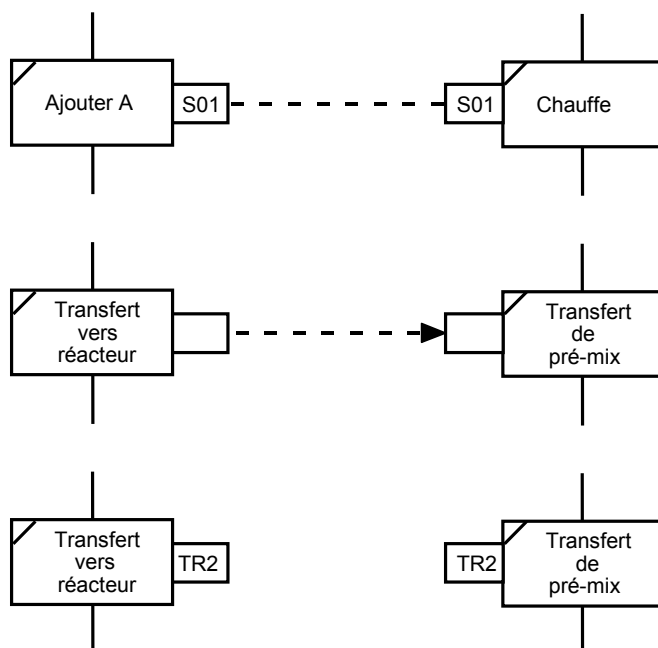


Figure 25 – Allocation symbol

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.



IEC 2232/01

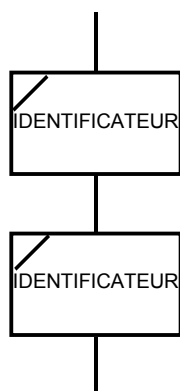
Figure 26 – Exemples de synchronisation d'éléments

6.1.1.5 Transitions de recette

Les diagrammes fonctionnels de procédure représentent deux types de transitions: une transition implicite et une transition explicite.

6.1.1.5.1 Transitions implicites

Un lien direct qui consiste en une simple ligne entre des entités de recette (voir figure 27) doit indiquer une transition dont la seule condition doit être que les entités précédant directement la transition ont terminé leur exécution. Aucune condition logique ne doit être saisie pour ce type de transition.



IEC 2233/01

Figure 27 – Transition implicite

6.1.1.5.2 Transition explicite

Un lien direct qui consiste en une simple ligne entre des entités de recette avec deux barres courtes et peu espacées, perpendiculaires à la ligne de liaison (voir figure 28) doit être utilisé pour indiquer la transition de recette explicite.

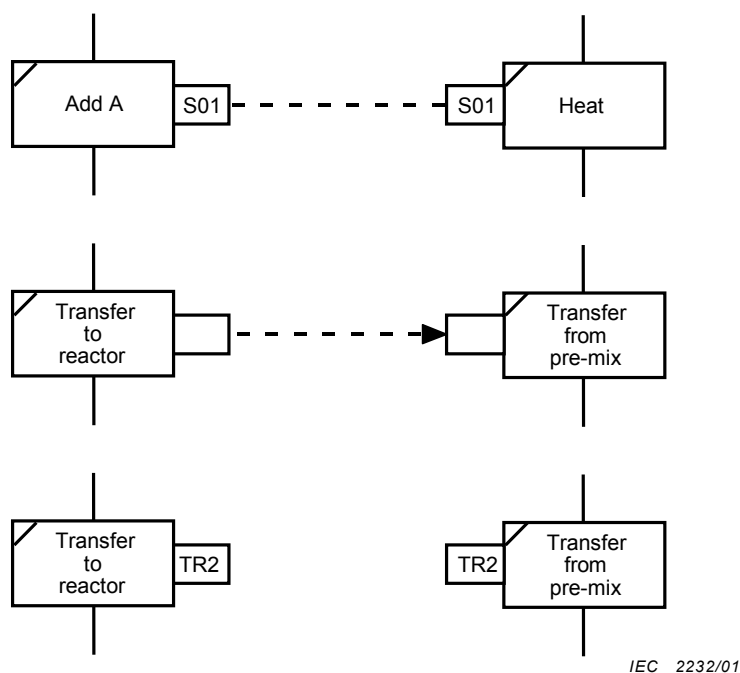


Figure 26 – Element synchronization examples

6.1.1.5 Recipe transitions

Procedure function 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.

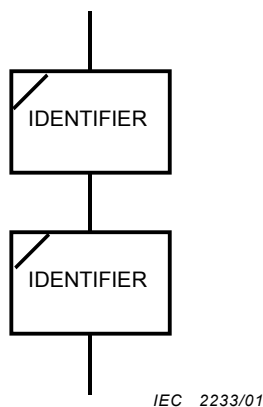


Figure 27 – Implicit transition

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.

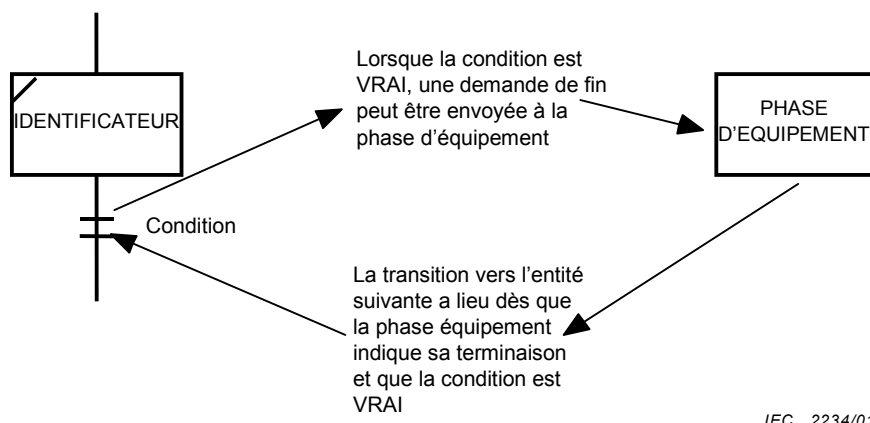


Figure 28 – Transition explicite

Cette transition est définie par une expression qui est évaluée soit à vraie, soit à faux. La transition est évaluée continuellement une fois que l'entité immédiatement précédente devient active.

La transition explicite est utilisée pour accomplir deux fonctions:

- interrompre l'exécution d'une branche de la logique procédurale de la recette;
- demander l'achèvement de tous les éléments procéduraux immédiatement précédents (par exemple procédures d'unité, opérations, phases).

L'achèvement de l'élément procédural immédiatement précédent peut être une condition de l'expression. Toutefois, un langage conditionnel pour l'expression n'est pas défini dans cette norme.

Lorsqu'une transition devient vraie, les éléments procéduraux actifs précédant immédiatement la transition doivent recevoir une notification d'achèvement. Lorsque des éléments procéduraux immédiatement précédents s'achèvent avant que la transition ne soit évaluée vraie, la transition doit continuer à évaluer sa logique jusqu'à ce qu'elle soit vraie.

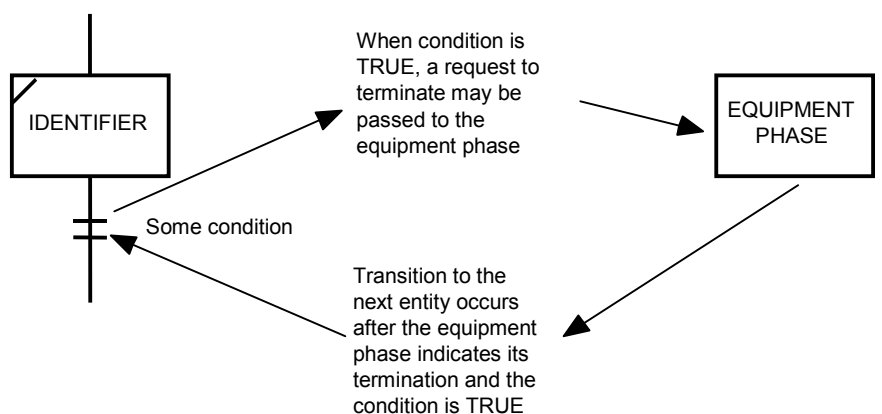
Une entité qui suit immédiatement une transition doit être activée seulement après que la condition de transition a été évaluée à vrai et que les éléments procéduraux précédant la transition s'achèvent.

6.1.1.6 Structures de base

Les structures définissent le fil de l'exécution désiré des éléments de recette. Le cas le plus simple est une série d'éléments procéduraux de recette qui seront activés l'un après l'autre. Des structures plus complexes comprennent la sélection de séquences et les séquences simultanées.

6.1.1.6.1 Début de sélection de séquence

Le début d'une sélection de séquence est présenté sur la figure 29. Chaque branche d'une sélection de séquence doit débuter par une transition. La sélection d'un chemin parmi plusieurs possibilités est représentée par autant de transitions (en fait, *sous* la ligne horizontale) qu'il y a de possibilités. Une seule séquence doit être sélectionnée parmi l'ensemble des séquences sous la ligne. Les transitions doivent être évaluées selon une priorité de gauche à droite. La séquence sous la transition qui devient vraie en premier, lorsqu'elle est évaluée de cette manière, doit devenir la séquence sélectionnée.



IEC 2234/01

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;
- b) to request the termination of all immediately preceding procedural elements (for example, 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 the 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 is evaluated to be 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.

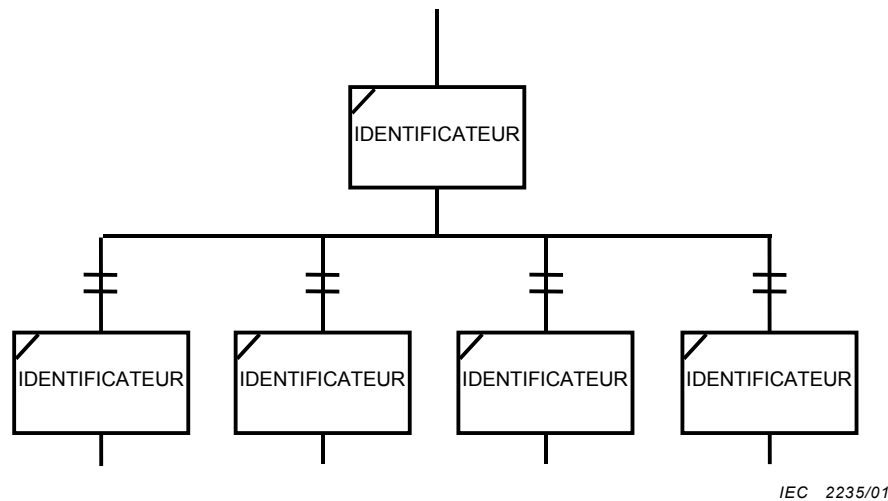


Figure 29 – Début de sélection de séquences

6.1.1.6.2 Fin de sélection de séquence

La fin d'une sélection de séquence montre la jonction des fils d'exécution possibles d'une sélection de séquence (voir figure 30).

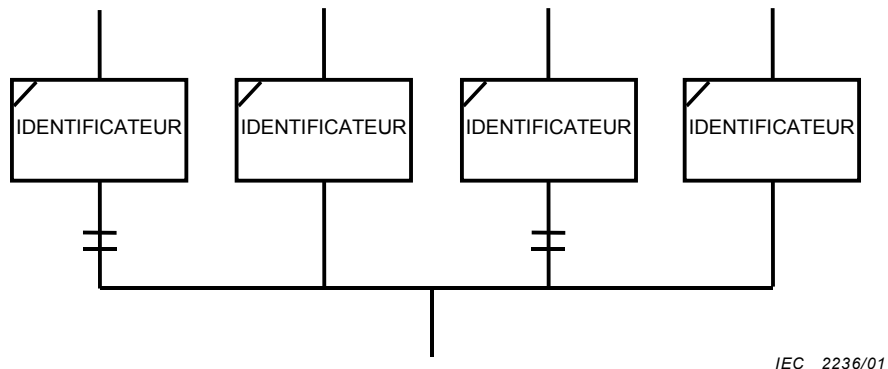


Figure 30 – Fin de sélection de séquences

6.1.1.6.3 Début de séquences simultanées

Le début de séquences simultanées (voir figure 31) montre le départ des fils indépendants d'exécution des éléments de recette, et il y a un fil d'exécution pour chaque cheminement sous un début de séquences simultanées. Tous les fils d'exécution doivent être joints à un fil d'exécution unique dans l'entité de recette. Le début et la fin des fils d'exécution ne correspondent pas nécessairement. Si une transition explicite est nécessaire, elle doit alors se trouver au-dessus des lignes parallèles.

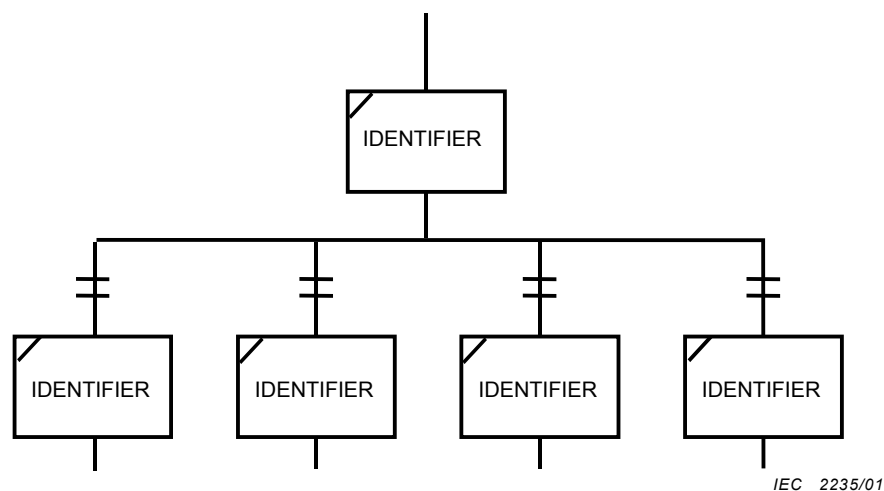


Figure 29 – Beginning of sequence selection

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).

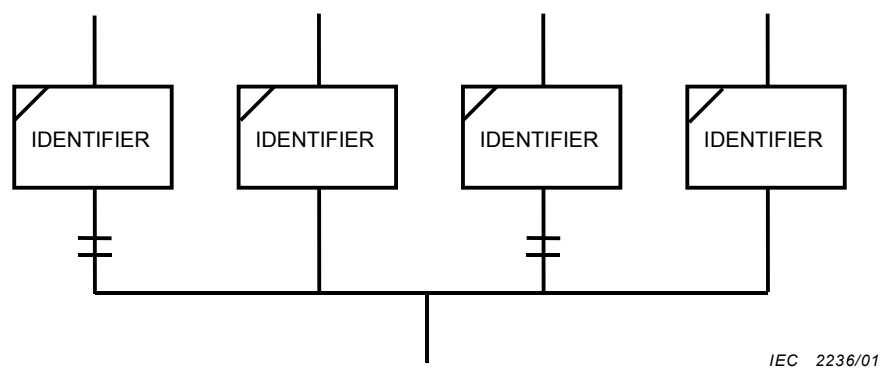


Figure 30 – End of sequence selection

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.

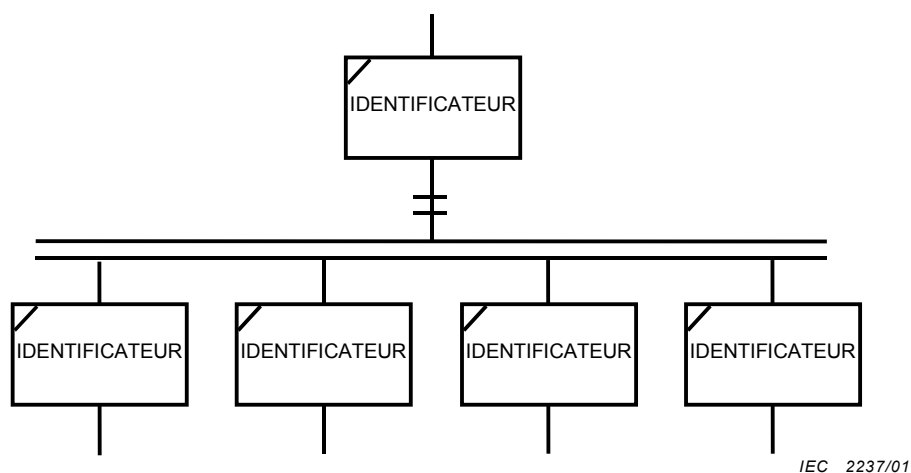


Figure 31 – Début de séquences simultanées

6.1.1.6.4 Fin de séquences simultanées

La fin de séquences simultanées montre la jonction de fils d'exécution indépendants de l'élément de recette (voir figure 32). La transition qui suit immédiatement les lignes parallèles est évaluée seulement lorsque toutes les entités qui précèdent immédiatement les lignes parallèles sont actives ou bien qu'elles ont terminé leur exécution.

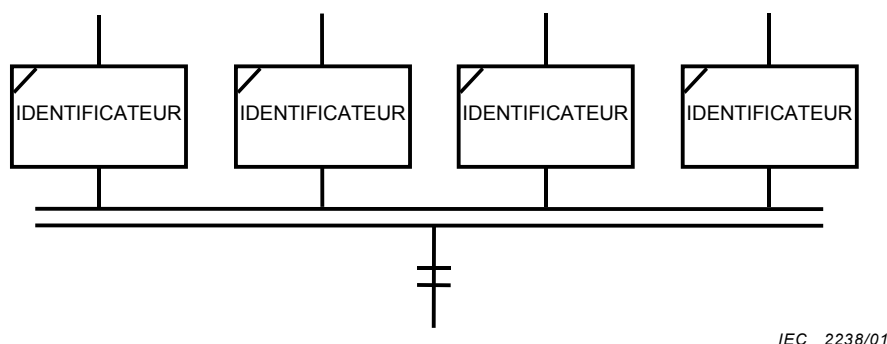
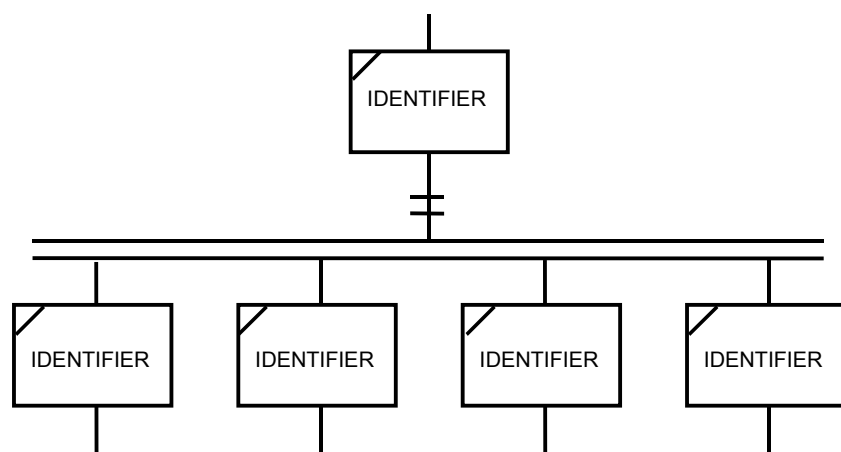


Figure 32 – Fin de séquences simultanées

6.1.1.6.5 Règles pour des diagrammes valides

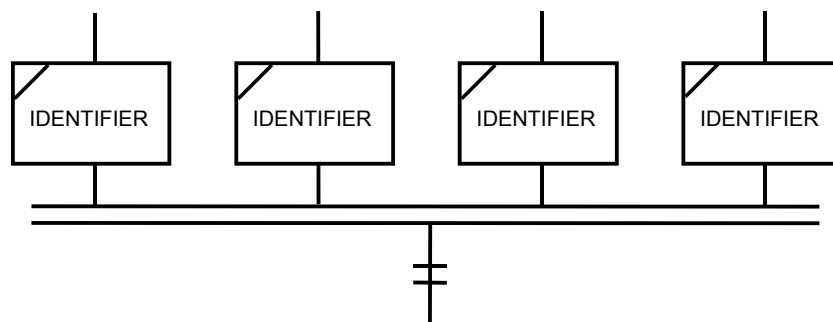
Les diagrammes valides doivent suivre des règles cohérentes pour les fils d'exécution. Des fils d'exécution simultanés indépendants doivent être joints. Il n'est pas admis que la fin de sélection de séquence soit utilisée pour joindre des fils d'exécution simultanés. La figure 33 donne un exemple d'un segment de diagramme valide avec sélection de séquences et fin de séquence.



IEC 2237/01

Figure 31 – Beginning of simultaneous sequences**6.1.1.6.4 End of simultaneous sequences**

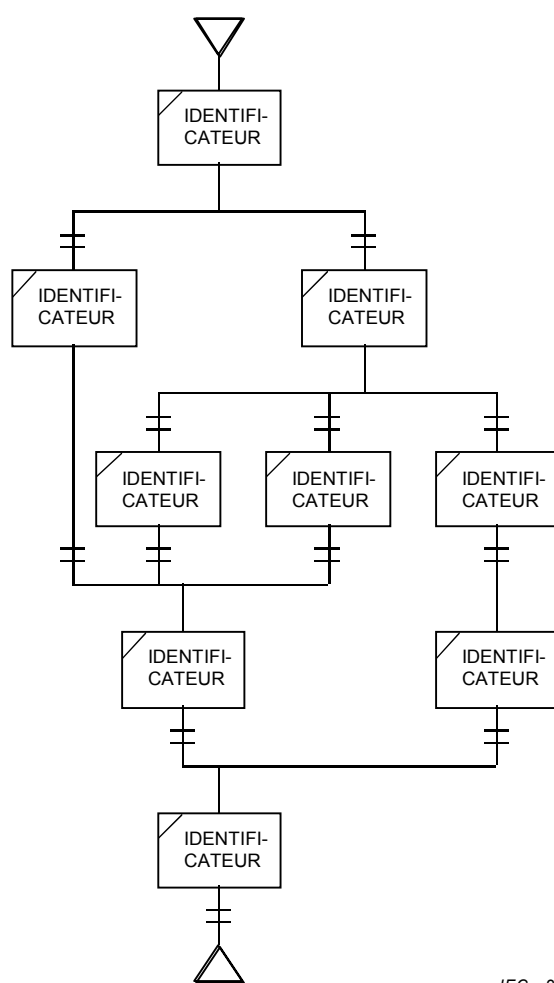
The end of simultaneous sequences shows the joining of independent threads of execution of the recipe element (see figure 32). The transition that immediately follows the parallel lines is evaluated only when all of the entities that immediately precede the parallel lines are either active or have completed.



IEC 2238/01

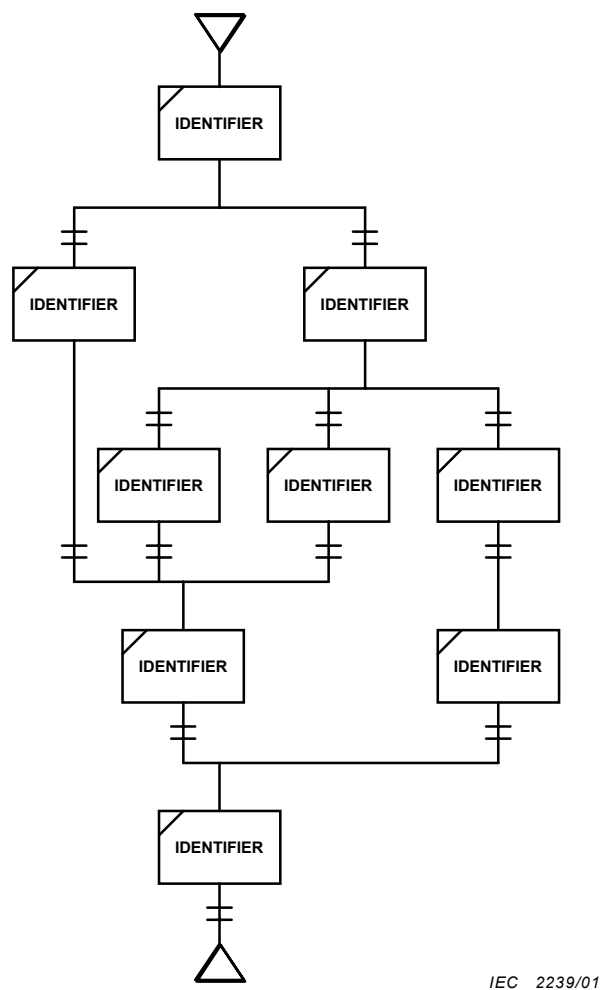
Figure 32 – End of simultaneous sequences**6.1.1.6.5 Rules for valid diagrams**

Valid diagrams shall follow consistent rules for threads of execution. Independent simultaneous threads of execution shall be joined. The end of a 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.



IEC 2239/01

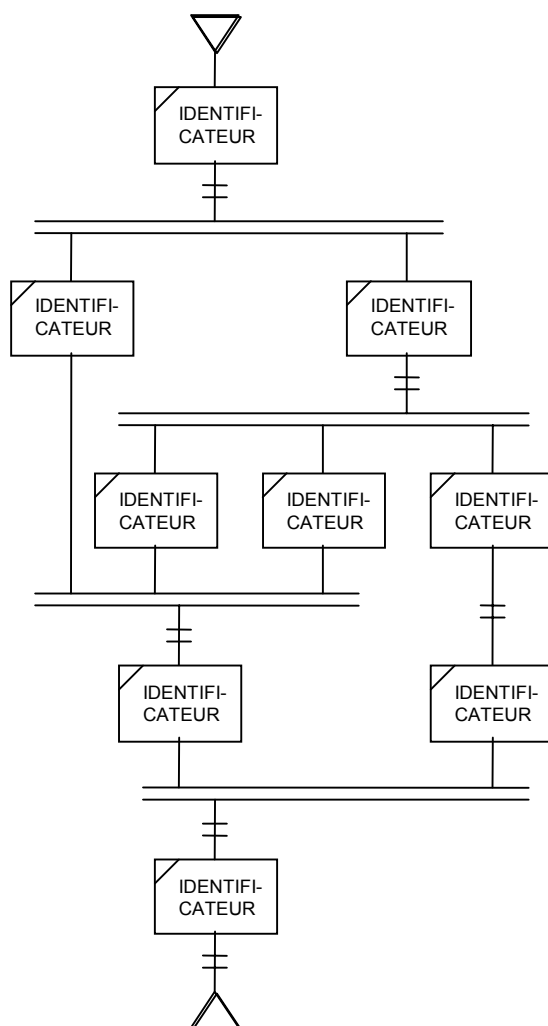
Figure 33 – Diagramme valide de sélection de séquences



IEC 2239/01

Figure 33 – Valid sequence selection diagram

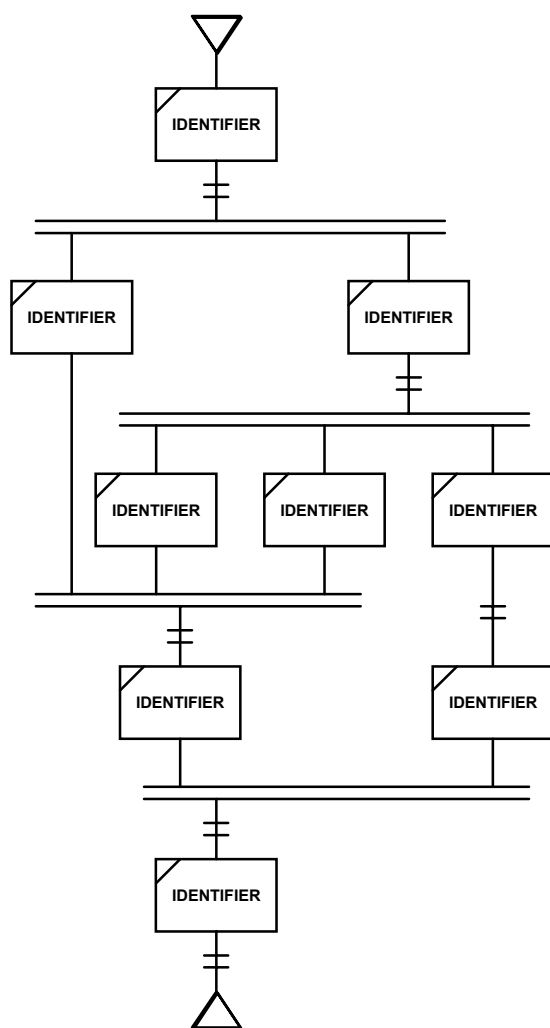
La figure 34 donne un exemple d'un segment de diagramme valide qui montre des départs et fins de séquences simultanées.



IEC 2240/01

Figure 34 – Diagramme valide de séquences simultanées

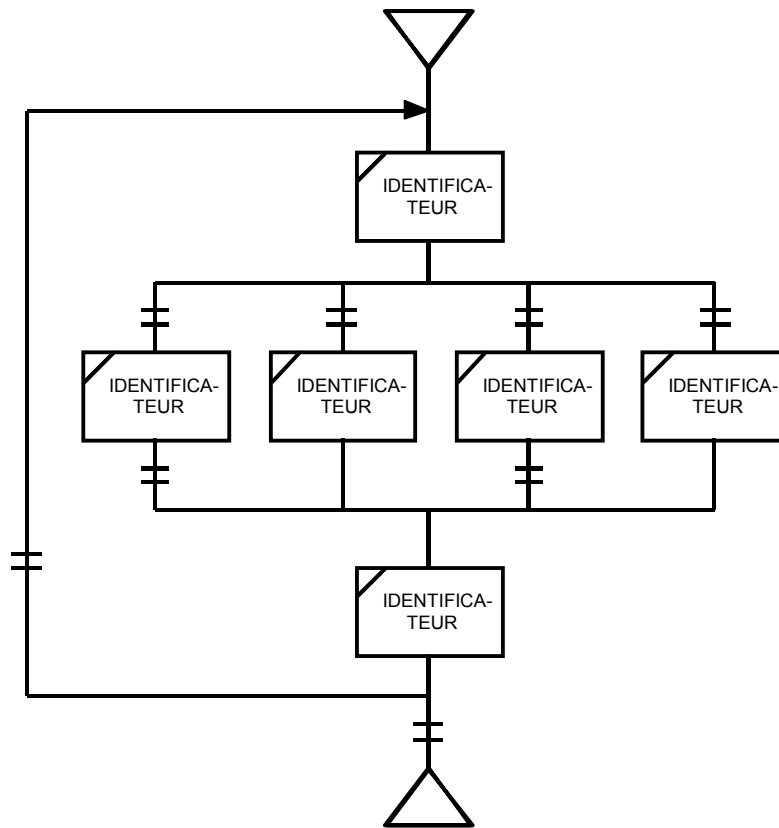
Figure 34 shows an example of a valid diagram segment that shows the start and end of simultaneous sequences.



IEC 2240/01

Figure 34 – Valid simultaneous sequence diagram

Un bouclage permet la réexécution d'entités basées sur des conditions de transition (voir figure 35). Ceci permet une exécution dynamique d'entités basée sur des conditions différentes.

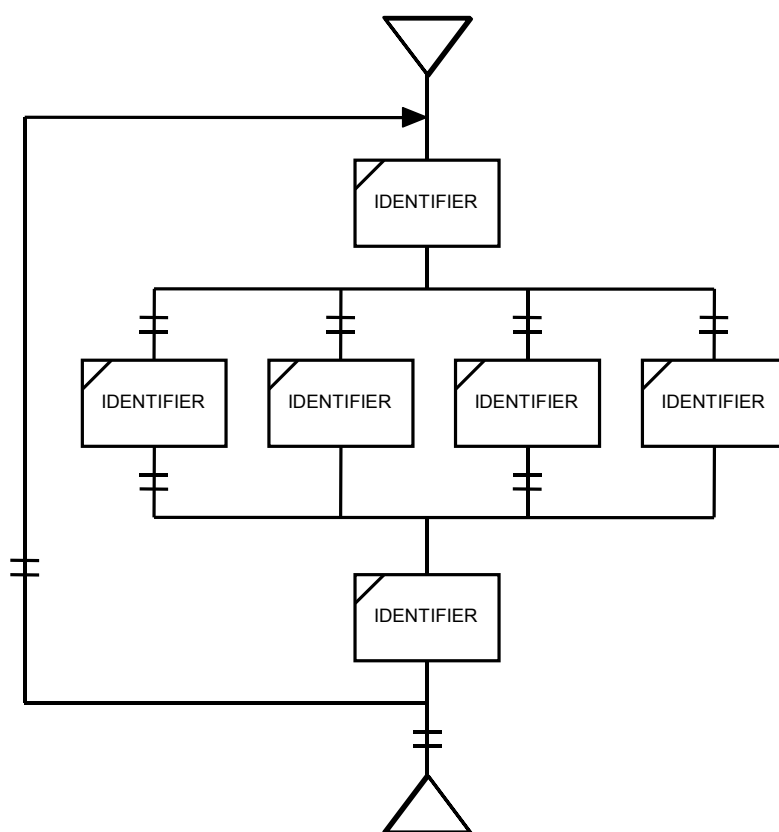


IEC 2241/01

Figure 35 – Bouclage avec éléments procéduraux de recette explicites

Cette norme ne peut pas définir tous les diagrammes fonctionnels de procédures valides et non valides. Les PFCs peuvent être construits avec des entités procédurales inaccessibles ou avec un cheminement d'exécution non valide (par exemple dans la figure 36, le fil à travers la «phase 1» peut ne jamais être mené à son terme si le fil qui passe par la «phase 5» est exécuté).

Looping provides for the re-execution of entities based upon transition conditions (see figure 35). This allows the dynamic execution of entities based upon differing conditions.



IEC 2241/01

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, (for example, in figure 36, the thread through "phase 1" may never be complete if the thread through "phase 5" is executed).

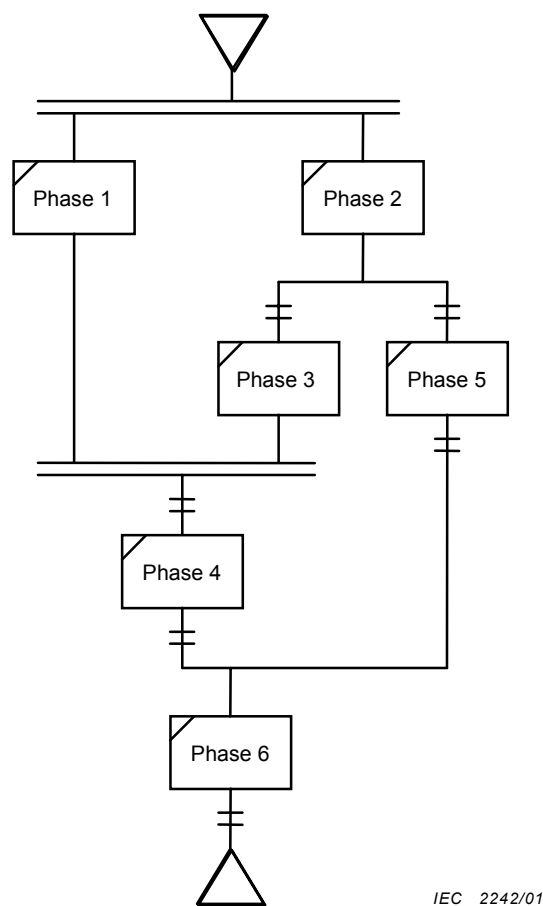
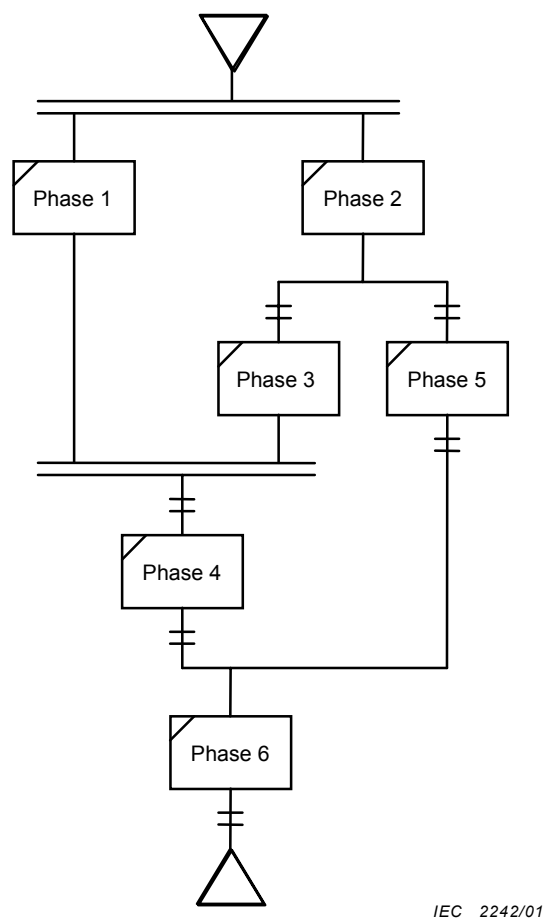


Figure 36 – Diagramme fonctionnel de procédure invalide

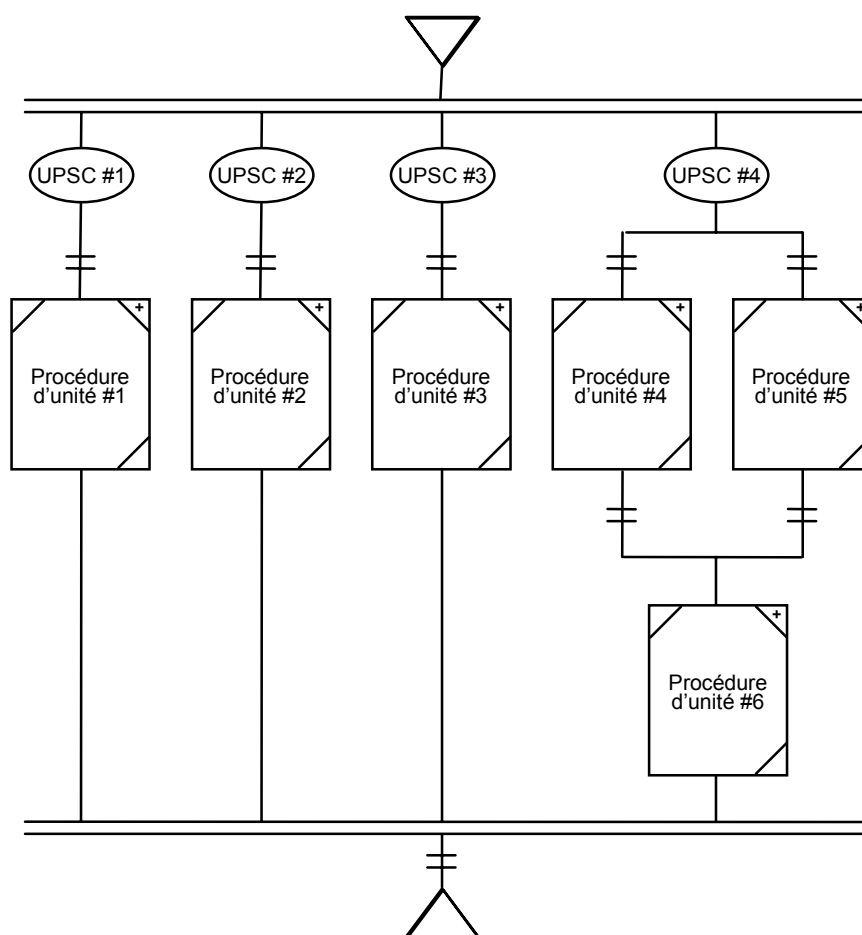


IEC 2242/01

Figure 36 – Invalid procedure function chart

6.1.2 Initialisation de procédures et de procédures d'unité

Une représentation du début d'une procédure de recette est illustrée sur la figure 37. A chaque niveau sous la procédure d'unité, des liens directs indiquent clairement l'ordre dans lequel les éléments procéduraux de recette deviennent actifs ou sont initialisés. Il est probable que l'initialisation d'une procédure de recette soit liée à des prescriptions de programmation; il faut donc des règles de départ, et certaines d'entre elles peuvent être basées sur la programmation. Une procédure d'unité devient active après que la transition suivant le symbole d'allocation est évaluée vraie.



IEC 2243/01

Figure 37 – Représentation de la procédure et initialisation des procédures d'unité

6.1.2.1 Procédure, procédure d'unité, et achèvement d'opération

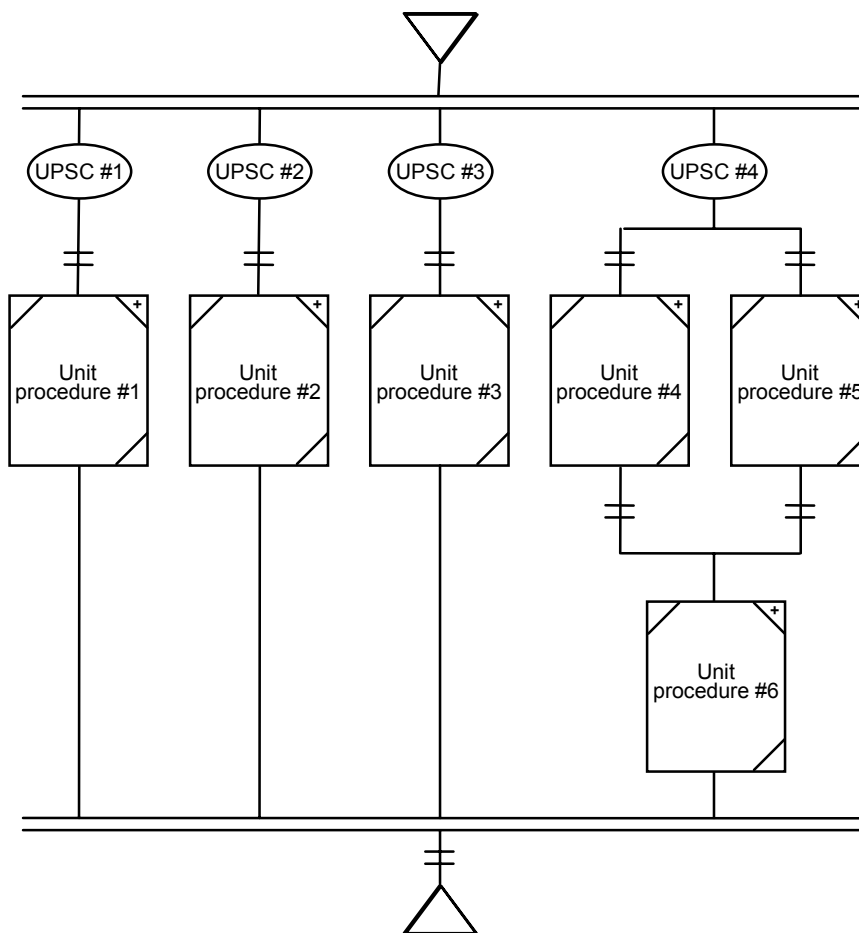
Quand le symbole de fin dans un diagramme fonctionnel de procédure est atteint, l'élément procédural encapsulant est achevé.

6.1.2.2 Liaisons relatives entre entités procédurales

Les figures 38 et 39 montrent deux méthodes pour représenter la relation relative entre les éléments procéduraux dans les diagrammes fonctionnels de procédures. Il est admis que cette relation soit assurée en organisant les éléments procéduraux verticalement en relation les uns avec les autres. La taille verticale des éléments procéduraux peut être modulée afin de montrer des relations relatives entre des éléments représentés. Les lignes discontinues horizontales montrent la synchronisation qui intervient entre les opérations à l'intérieur de chaque procédure d'unité.

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; it therefore needs 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.



IEC 2243/01

Figure 37 – Depiction of procedure and unit procedure initiation

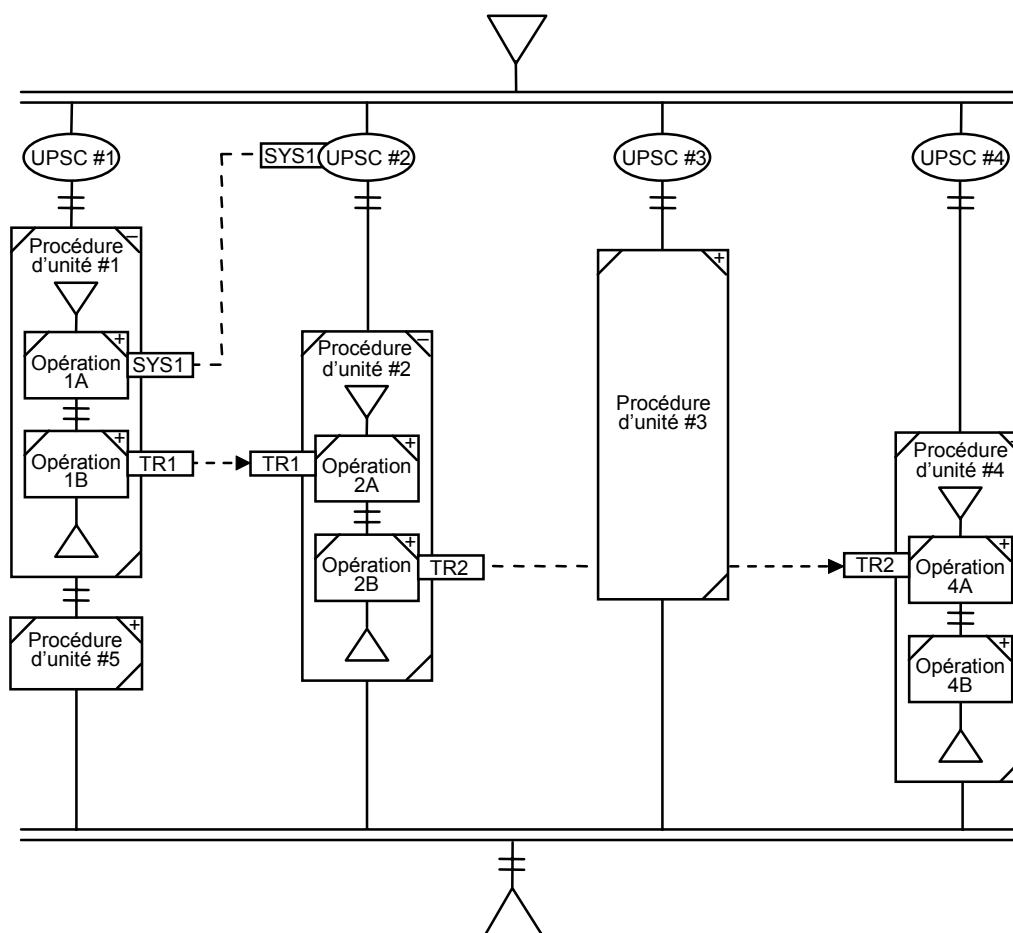
6.1.2.1 Procedure, unit procedure, and operation completion

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.

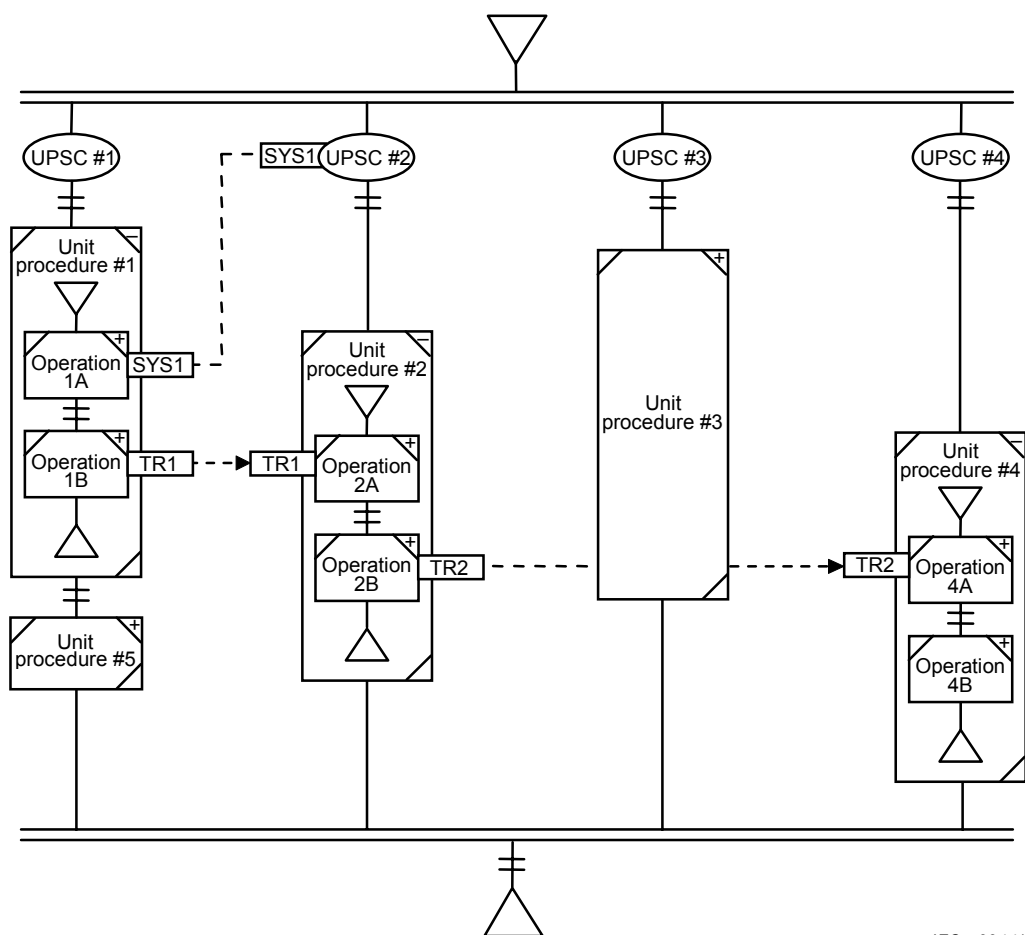
Un système conforme doit appliquer au moins une des méthodologies décrites dans les figures 38 et 39.



IEC 2244/01

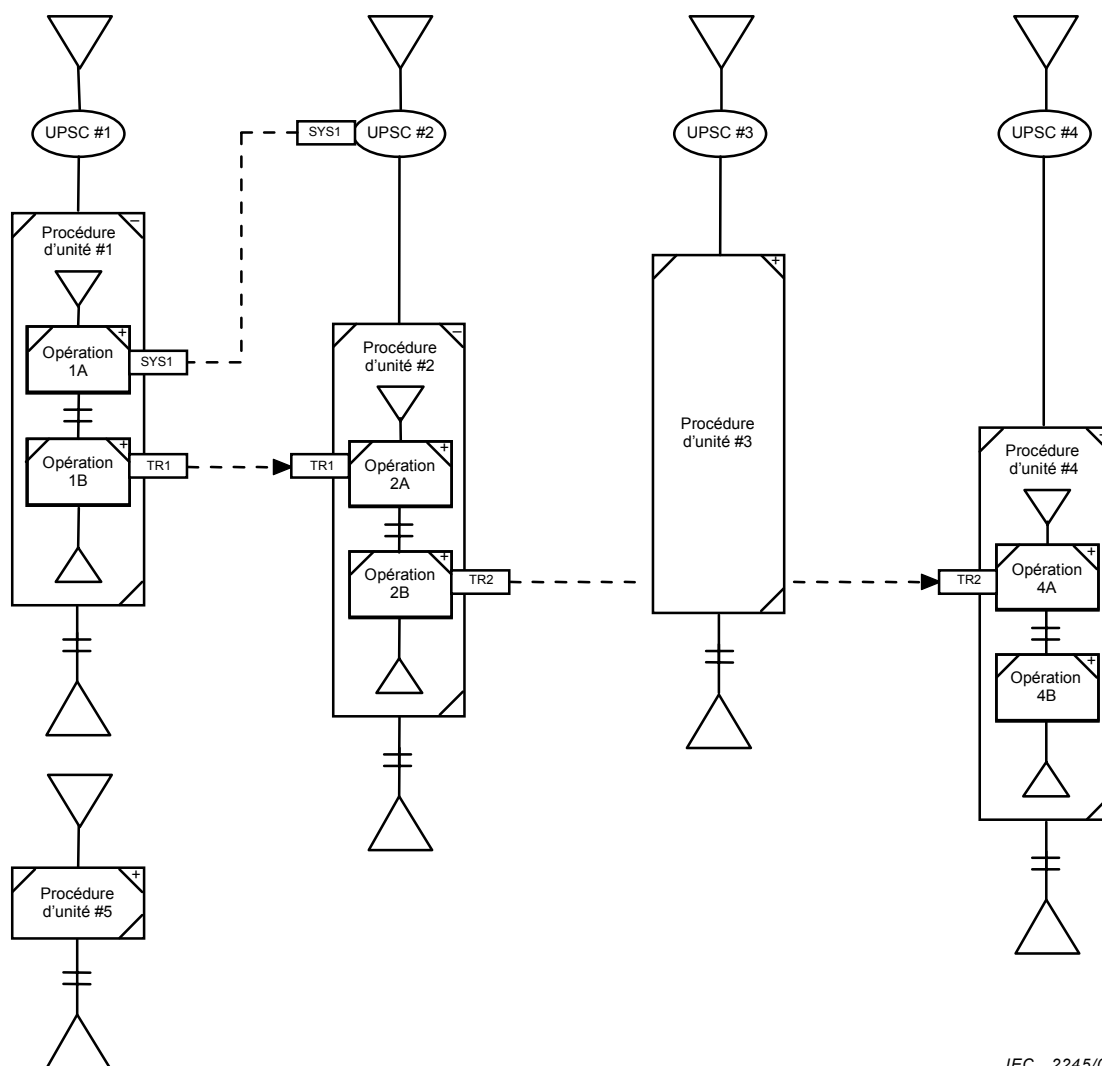
Figure 38 – Liaisons relatives entre les entités procédurales

A compliant system shall implement at least one of the methodologies that are depicted in figures 38 and 39.



IEC 2244/01

Figure 38 – Relative relationship of procedural entities



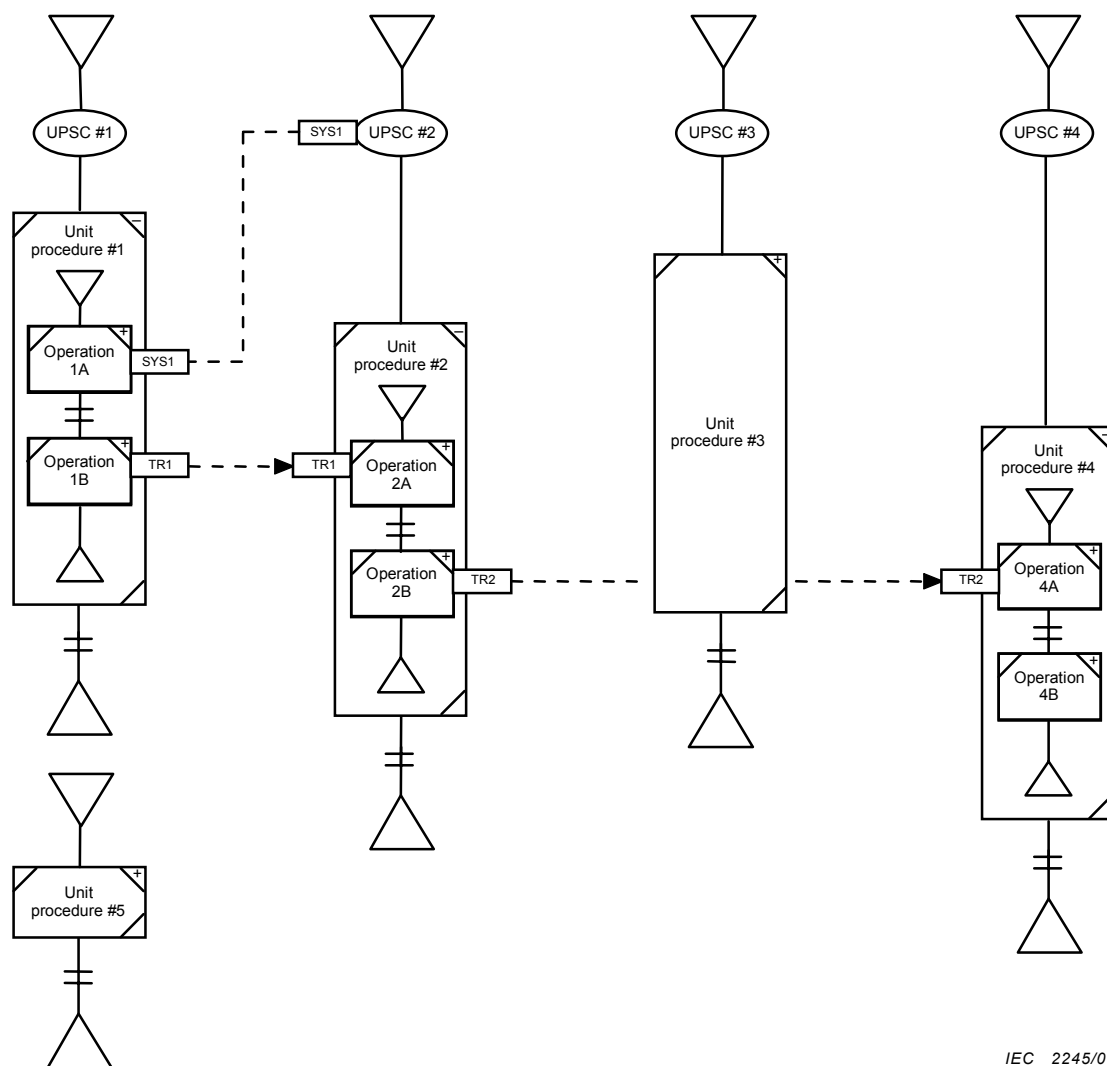
IEC 2245/01

Figure 39 – Liaisons relatives entre les entités procédurales – Variante 1

Lorsque des éléments procéduraux à deux niveaux différents (par exemple procédures d'unité et opérations) sont représentés sur un dessin de ce type, une boîte qui montre le même groupement vu comme un élément procédural de niveau supérieur (c'est-à-dire procédures d'unité) doit comprendre les éléments procéduraux de niveau inférieur c'est-à-dire opérations). La procédure d'unité #1 à la figure 38 illustre un élément procédural qui est représenté avec le PFC de niveau inférieur qu'il encapsule.

6.1.3 Informations non procédurales de la recette maître

Toute autre information qui fait partie d'une recette maître est liée à un élément ou un symbole spécifique dans la procédure de recette maître. Cette norme ne spécifie pas intentionnellement comment cette liaison ou cette référence est appliquée. Dans une mise en œuvre au crayon et papier, par exemple, la référence pourrait être assurée par une représentation quelque peu similaire à une note de bas de page, ou bien l'information pourrait être écrite à côté de l'élément procédural en question. Dans une réalisation électronique, des boîtes de dialogue ou quelque autre mécanisme qui n'a pas encore été inventé pourrait être le choix retenu pour la mise en œuvre. Cependant, la relation doit être clairement apparente, et elle doit être logique dans le contexte de chaque application.



IEC 2245/01

Figure 39 – Relative relationship of procedural entities – Alternative 1

When procedural elements at two different levels (for example, 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 Formule de la recette maître

L'information de formule est composée d'entrées de processus, de paramètres de processus et de sorties de processus. L'information de formule doit pouvoir être représentée dans son intégralité (par exemple associée à une procédure de recette), en parties (par exemple entrées de processus seulement ou bien pour une procédure d'unité spécifique) ou comme une agrégation des formules de niveau moins élevé comme cela paraît appropriée dans le contexte d'utilisation présumé. Lorsqu'elles sont décrites, les formules doivent être associées à un élément procédural de recette.

Par exemple, la quantité de produit que génère une recette peut être associée à la procédure de recette tandis que la quantité de matière à ajouter à un réacteur peut être associée à une phase spécifique. L'utilisation de la formule permet une sommation de tous les paramètres d'éléments procéduraux de recette qui ont été identifiés en tant qu'entrées de processus de telle façon qu'une liste des entrées de processus pour une recette complète, une procédure d'unité ou une opération puisse être fournie.

6.1.3.2 Prescriptions d'équipement d'une recette maître

Les prescriptions d'équipement sont spécifiques à l'exécution d'éléments procéduraux de recette. La représentation doit fournir une méthode pour que l'utilisateur puisse observer les prescriptions d'équipement qui sont associées à chaque élément procédural individuellement ou pour tous les éléments globalement.

6.1.3.3 En-tête et autres informations

L'information d'en-tête et la catégorie «autres informations» de l'information de recette peuvent être liées à la recette en général (par exemple ID recette, statut réglementaire) ou à des entités procédurales de recette spécifiques (par exemple prescriptions d'équipement de protection, information sur les risques chimiques). Les données d'en-tête et les «autres informations» doivent pouvoir être représentées dans leur intégralité ou associées à l'entité procédurale à laquelle les informations sont liées.

6.2 Représentation de la recette de contrôle

Lorsque des diagrammes fonctionnels de procédures sont utilisés pour la description d'une recette de contrôle, ils doivent suivre les mêmes principes définis pour l'utilisation de diagrammes fonctionnels de procédures pour la description d'une recette maître. Dans ce cas, cependant, la représentation de la recette de contrôle prendra généralement la forme d'un affichage actif dans les systèmes automatisés. Comme la liaison entre les éléments procéduraux de recette et les éléments procéduraux d'équipement est connue pendant l'exécution de la recette de contrôle, des couleurs et/ou d'autres moyens de délimitation peuvent être utilisés pour indiquer le statut des éléments procéduraux.

6.3 Traitement des exceptions

Un traitement spécial peut être inclus dans la recette afin de résoudre les exceptions spécifiques aux produits, en plus de toute logique d'équipement utilisée pour le traitement des exceptions qui sont liées à l'équipement. Ces exceptions spécifiques aux produits sont généralement liées à la méthode de fabrication d'un produit, souvent à sa qualité, et elles ne sont pas spécifiques à l'équipement. Elles sont étroitement liées à la procédure normale; par conséquent, on ne les distingue pas aisément du traitement normal, et elles deviennent seulement actives dans le cas d'une exception.

L'étendue de l'effet du traitement d'une exception dans la procédure de recette est normalement confinée dans les procédures d'unité parce que les unités fonctionnent indépendamment. Dans certains cas, cependant, les commandes, les états et/ou les modes sont propagés vers une autre unité lorsqu'il s'agit d'un problème commun (par exemple un transfert ou un traitement concurrent avec des contraintes de temps communes). Dans certaines circonstances, une recette complète peut devoir être contrainte à un état et/ou un mode spécifié.

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 (for example, associated with a recipe procedure), in parts (for example, 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 a formula allows a summation of all recipe procedural element 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 (for example, recipe ID, regulatory status) or to specific recipe procedural entities (for example, protective equipment requirements, chemical hazards 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, colours 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; they are therefore 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 (for example, 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.

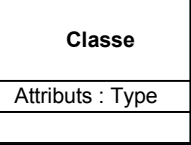
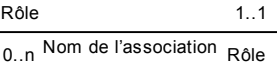

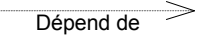
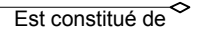

Annexe A (normative)

Technique de modélisation des données

A.1 Notation UML

La table A.1 définit la notation UML utilisée dans cette norme [1]¹.

Table A.1 – Notation UML

Symbole		Définition
	Classe	Définit une classe d'objets, chacune avec les mêmes types d'attributs. Chaque objet est identifiable de façon unique ou énumérable. Aucune opération ou méthode n'est citée pour les classes. Les attributs dont les noms sont précédés de «-» indiquent qu'ils sont généralement optionnels dans toute utilisation de la classe
	Attributs : Type	
		Une association entre éléments d'une classe et éléments d'une autre ou de la même classe. Chaque association est identifiée. Peut avoir un nombre prévu ou un choix de membres de la sous-classes, où 'n' indique un nombre indéterminé (par exemple 0,n signifie qu'aucun ou plusieurs membres de la sous-classe peuvent exister)
		Une généralisation (La flèche pointe vers la super classe) montre qu'un élément de la classe est un type spécialisé de la super classe
		Une dépendance (c'est-à-dire relation intime entre les éléments) montre qu'un élément de la classe dépend d'un élément d'une autre classe
		Une agrégation (c'est-à-dire composé de) montre qu'un élément de la classe est composé d'éléments d'autres classes
		Une classe d'objets qui est une instance d'une autre classe d'objets

A.2 Définitions

A.2.1 classe

description d'un ensemble d'objets qui partagent les mêmes attributs, comportements, relations, et sémantique

A.2.2 encapsulation

technique qui sépare les aspects externes et internes d'un objet des détails de mise en œuvre de l'objet (également appelée masquage d'information)

A.2.3 instance

terme utilisé pour désigner un objet qui appartient à une classe particulière mais qui n'est pas lui-même une classe ou une sous-classe. Par exemple, «réacteur401» est une instance de la classe «réacteur»

¹ Les chiffres en crochets renvoient à la bibliographie.

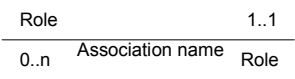
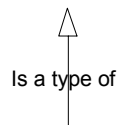
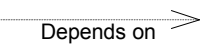
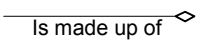
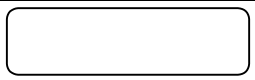
Annex A (normative)

Data modelling technique

A.1 UML notation

Table A.1 defines the UML notation that is used in this standard [1]¹.

Table A.1 – UML notation

Symbol	Definition
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <div style="text-align: center; border-bottom: 1px solid black; margin-bottom: 5px;">Class</div> <div style="border-bottom: 1px solid black; margin-bottom: 5px;">Attributes : Type</div> <div style="border-bottom: 1px solid black; height: 15px;"></div> </div>	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"

¹ Figures in square brackets refer to the bibliography.

A.2.4 modèle

représentation formelle abstraite d'un système. Un modèle est généralement présenté comme un ensemble de diagrammes et un dictionnaire de données

A.2.5 objet

entité comprenant un état et un comportement. L'état est la valeur de tous les attributs à un moment donné. Un attribut est une information qui qualifie l'objet. Le comportement d'un objet est la fonctionnalité contenue dans un objet qui est nécessaire pour manipuler les attributs

A.2.6 sous-classe

classe qui est un cas spécial d'une classe plus générale (par exemple «réacteur revêtu de verre» est une sous-classe d'une classe «réacteur»)


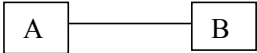
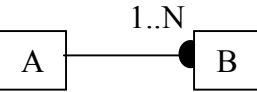
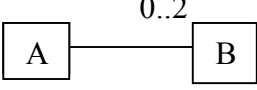
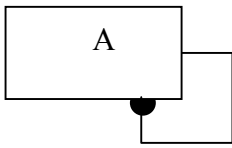
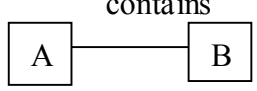
A.2.7 langage de modélisation unifié (UML)

langage utilisé pour spécifier, visualiser, construire et documenter les objets de systèmes logiciels, ainsi que pour la modélisation de processus de gestion ou d'autres systèmes non logiciels.

A.3 Notation ERD

La table A.2 définit la notation ERD qui est utilisée dans cette norme.

Table A.2 – Notation ERD

Symbole	Définition
	Définit une entité
	Pour chaque occurrence de A, il y a une, et une seule, occurrence de B. L'association à B peut également être marquée par le chiffre 1
	Association spécifiée numériquement. Dans cet exemple, pour chaque occurrence de A, il peut exister une ou plusieurs occurrences de B. Un autre exemple est 0..N. Si aucune association numérique n'est donnée, c'est 0..N qui est implicite
	Association spécifiée numériquement de 0 à un certain nombre positif. Dans cet exemple, pour chaque occurrence de A, il peut exister de 0 à 2 occurrences de B
	Association bouclée. Une occurrence de A peut résulter d'aucune ou plusieurs occurrences d'entités du même type. Une association optionnelle est qu'une occurrence de l'entité A peut contenir aucune ou plusieurs occurrences d'entités du même type. Une autre utilisation est qu'une occurrence de A peut contenir aucune ou plusieurs occurrences d'entités du même type
	L'association entre entités est étiquetée de façon à spécifier la nature de la relation. L'étiquette s'applique à l'entité la plus proche. Dans cet exemple, l'association est la suivante: chaque occurrence de A contient une et une seule occurrence de B

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 (for example, glass-lined reactor is a subclass of the “reactor” class)


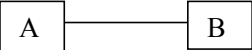
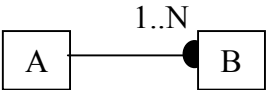
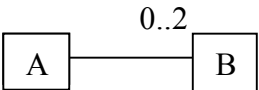
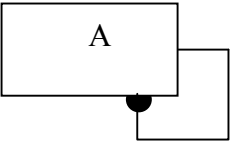
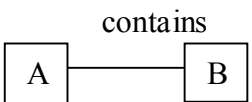
A.2.7 unified modeling language (UML)

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: from 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

Annexe B (normative)

Liste de définitions SQL

Cette annexe contient la définition standard ISO/CEI 9075:1992 de toutes les tables définies dans l'article 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', 'CEI 61512-2 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', 'Définit un ensemble de valeurs Booléennes')
VALUES ('DirectionType',
        'Définit comment un paramètre est destiné à être manipulé')
VALUES ('EquipmentLevel',
        'Définit le niveau hiérarchique d'équipement pour les éléments d'équipement')
VALUES ('EquipmentType',
        'Définit le type d'enregistrement d'équipement pour les éléments d'équipement')
VALUES ('EvaluationRule',
        'Définit les règles d'évaluation pour les propriétés d'équipement')
VALUES ('FormulaType', 'Définit les types de formules de recette')
VALUES ('FormulaSubType', 'Définitions des sous-types de formule fournis par
        l'utilisateur')
VALUES ('LinkDepiction',
        'Définit comment les liens entre éléments de recette doivent être représentés')
VALUES ('LinkToType', 'Définit si un lien référence est une étape ou une transition')
VALUES ('LinkType', 'Définit le type de lien')
VALUES ('RE_Type',
        'Définit l'élément de recette, soit le niveau procédural de la recette soit l'entité
        d'allocation')
```

Annex B (normative)

SQL definition listing

This annex contains the ISO/IEC 9075:1992 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', 'IEC 61512-2 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', 'Définit comment un élément de recette est utilisé dans une
recette')

VALUES ('RecipeStatus', 'Définit le statut possible d'une recette')

VALUES ('RecordSet',
'Définit l'énumération utilisée pour classier un enregistrement dans une
catégorie d'informations d'historique du batch.')

VALUES ('RecordSetControlRecipe',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie ControlRecipe.')

VALUES ('RecordSetMasterRecipe',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie MasterRecipe.')

VALUES ('RecordSetExecutionInfo',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie ExecutionInfo.')

VALUES ('RecordSetMaterialInfo',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie MaterialInfo.')

VALUES ('RecordSetContinuousData',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie ContinuousData.')

VALUES ('RecordSetEvents',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie Events.')

VALUES ('RecordSetOperatorChange',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie OperatorChange.')

VALUES ('RecordSetOperatorComment',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie OperatorComment.')

VALUES ('RecordSetAnalysisData',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie AnalysisData.')

VALUES ('RecordSetLateRecord',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie LateRecord.')

VALUES ('RecordSetRecipeData',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie RecipeData.')

VALUES ('RecordSetRecipeSpecified',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie RecipeSpecified.')

VALUES ('RecordSetSummaryData',
'Fournit une classification de l'enregistrement de l'historique supplémentaire
sous la catégorie SummaryData.')

VALUES ('ScheduleAction', 'Définit l'action voulue de l'entrée de programmation')

VALUES ('ScheduleMode',
'Définit le mode dans lequel l'entrée de programmation commence l'exécution')

VALUES ('ScheduleStatus', 'Définit le statut possible d'un programme')

VALUES ('SE_Type',
'Définit le type d'entité dans un enregistrement de programme')

VALUES ('ValueDataType',
'Définit comment une valeur est représentée (ex. Booléenne, flottant, etc.)')

VALUES ('ValueType', 'Définit comment la valeur d'une chaîne est interprétée')

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 (for example Boolean, float, etc.)')

VALUES ('ValueType', 'Defines how a value string is interpreted')

```

INSERT INTO BXT_Enumeration (EnumSet, EnumValue, EnumString, Description)
VALUES ('Boolean', 0, 'FAUX', 'Définit une valeur Booléenne')
VALUES ('Boolean', 1, 'VRAI', '')
VALUES ('DirectionType', 0, 'Invalide', 'Entrée Invalide')
VALUES ('DirectionType', 1, 'Interne',
        'Identifie comment un paramètre est manipulé.')
VALUES ('DirectionType', 2, 'Entrée',
        'L'Element de Recette reçoit la valeur d'une source externe.')
VALUES ('DirectionType', 3, 'Sortie',
        'L'Element de Recette crée la valeur et la rend disponible pour une utilisation
        externe.')
VALUES ('DirectionType', 4, 'Entrée/Sortie',
        'L'Element de Recette et l'élément externe échangent la valeur, et peuvent
        changer sa valeur.')
VALUES ('EquipmentLevel', 0, 'Invalide', 'Entrée invalide')
VALUES ('EquipmentLevel', 1, 'Entreprise',
        'Identifie le niveau hiérarchique d'équipement for BXT_EquipElement')
VALUES ('EquipmentLevel', 2, 'Site', '')
VALUES ('EquipmentLevel', 3, 'Zone', '')
VALUES ('EquipmentLevel', 4, 'Cellule de Processus', '')
VALUES ('EquipmentLevel', 5, 'Unité', '')
VALUES ('EquipmentLevel', 6, 'Module d'Equipeement', '')
VALUES ('EquipmentLevel', 7, 'Module de Contrôle', '')
VALUES ('EquipmentType', 0, 'Invalid', 'Entrée invalide')
VALUES ('EquipmentType', 1, 'Classe',
        'Identifie le type d'enregistrement pour BXT_EquipElement')
VALUES ('EquipmentType', 2, 'Elément', '')
VALUES ('EvaluationRule', 0, 'Invalide', 'Entrée invalide')
VALUES ('EvaluationRule', 1, '=',
        'Opérateur de comparaison d'égalité pour des propriétés d'équipement')
VALUES ('EvaluationRule', 2, '<>',
        'Opérateur de comparaison d'inégalité pour des propriétés d'équipement')
VALUES ('EvaluationRule', 3, '<',
        'Opérateur de comparaison «inférieur à» pour des propriétés d'équipement')
VALUES ('EvaluationRule', 4, '>',
        'Opérateur de comparaison «supérieur à» pour des propriétés d'équipement')
VALUES ('EvaluationRule', 5, '<=',
        'Opérateur de comparaison «inférieur ou égal à» pour des propriétés
        d'équipement')
VALUES ('EvaluationRule', 6, '>=',
        'Opérateur de comparaison «supérieur ou égal à» pour des propriétés
        d'équipement')
VALUES ('EvaluationRule', 7, 'Membre',
        'Opérateur de comparaison «Est un membre de» pour les propriétés
        d'équipement')
VALUES ('EvaluationRule', 8, 'Non membre',
        'Opérateur de comparaison «N'est pas un membre de» pour les propriétés
        d'équipement')
VALUES ('EvaluationRule', 9, 'Non',
        'Opérateur de négation pour des propriétés d'équipement')
VALUES ('FormulaType', 0, 'Invalide', 'Entrée invalide')
VALUES ('FormulaType', 1, 'Entrée de Processus', 'Type de formule de recette')
VALUES ('FormulaType', 2, 'Sortie de Processus', '')
VALUES ('FormulaType', 3, 'Paramètre de Processus', '')
VALUES ('FormulaSubType', 0, 'Invalide', 'Entrée invalide')
VALUES ('LinkDepiction', 0, 'Invalide', 'Entrée invalide')
VALUES ('LinkDepiction', 1, 'Aucun', 'Pas de représentation du lien')
VALUES ('LinkDepiction', 2, 'Ligne', 'Lien montré seulement avec ligne')
VALUES ('LinkDepiction', 3, 'ID', 'Lien montré seulement avec identificateur')
VALUES ('LinkDepiction', 4, 'Ligne & ID',
        'Lien montré avec ligne et identificateur ')

```



```

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, 'Ligne & Flèche',
 'Lien montré avec ligne et flèche du flux de la matière')

VALUES ('LinkDepiction', 6, 'Ligne, Flèche, & ID',
 'Lien montré avec ligne, flèche du flux matière et identification')

VALUES ('LinkToType', 0, 'Invalide', 'Entrée invalide')

VALUES ('LinkToType', 1, 'Élément de Recette',
 'Lien référençant une entrée dans la table BXT_MRecipeElement')

VALUES ('LinkToType', 2, 'Transition',
 'Lien référençant une entrée dans la table BXT_MRecipeTransition')

VALUES ('LinkType', 0, 'Invalide', 'Entrée invalide')

VALUES ('LinkType', 1, 'Lien de Contrôle',
 'Définit un lien entre éléments de recette qui indique un flux de contrôle
 procédural.')

VALUES ('LinkType', 2, 'Lien de Transfert',
 'Définit un lien entre éléments de recette qui indique un transfert de matière.')

VALUES ('LinkType', 3, 'Lien de Synchronisation',
 'Définit un lien entre éléments de recette où il y a une forme de
 synchronisation.')

VALUES ('RE_Type', 0, 'Invalide', 'Entrée invalide')

VALUES ('RE_Type', 1, 'Recette Maître',
 'Spécifie le type d'élément de recette.')

VALUES ('RE_Type', 2, 'Procédure', '')

VALUES ('RE_Type', 3, 'Procédure d'Unité', '')

VALUES ('RE_Type', 4, 'Opération', '')

VALUES ('RE_Type', 5, 'Phase', '')

VALUES ('RE_Type', 6, 'Allocation', '')

VALUES ('RE_Type', 7, 'Début', '')

VALUES ('RE_Type', 8, 'Fin', '')

VALUES ('RE_Type', 9, 'Début de Parallélisme', '')

VALUES ('RE_Type', 10, 'Fin de Parallélisme', '')

VALUES ('RE_Type', 11, 'Début de Branche', '')

VALUES ('RE_Type', 12, 'Fin de Branche', '')

VALUES ('RE_Use', 0, 'Invalide', 'Entrée invalide')

VALUES ('RE_Use', 1, 'Lié',
 'Un élément de recette (RE) peut être référencé par plusieurs étapes RE.')

VALUES ('RE_Use', 2, 'Imbriqué',
 'Un RE n'est référencé que par un RE, un RE est défini pour chaque utilisation
 du RE')

VALUES ('RE_Use', 3, 'Copié',
 'Comme Imbriqué, mais le RE spécifique a été modifié par rapport à sa définition
 d'origine')

VALUES ('RecipeStatus', 0, 'Invalide', 'Entrée invalide')

VALUES ('RecipeStatus', 1, 'Approuvée pour Production',
 'La recette a été approuvée pour la production.')

VALUES ('RecipeStatus', 2, 'Approuvée pour Test',
 'La recette a seulement été approuvée pour des tests')

VALUES ('RecipeStatus', 3, 'Non Approuvée',
 'La recette n'a pas été approuvée pour la production ni pour des tests')

VALUES ('RecipeStatus', 4, 'Inactive', 'La recette n'est pas active')

VALUES ('RecipeStatus', 5, 'Obsolète', 'La recette est obsolète')

VALUES ('RecordSet', 0, 'Invalide', 'Entrée invalide')

VALUES ('RecordSet', 1, 'RecordSetControlRecipe',
 'Définit qu'un enregistrement des informations de l'historique du batch fait partie
 de la catégorie ControlRecipe.')

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 ('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', 9, 'RecordSetAnalysisData', '')
VALUES ('RecordSet', 10, 'RecordSetLateRecord', '')
VALUES ('RecordSet', 11, 'RecordSetRecipeData', '')
VALUES ('RecordSet', 12, 'RecordSetRecipeSpecified', '')
VALUES ('RecordSet', 13, 'RecordSetSummaryData', '')
VALUES ('RecordSetControlRecipe', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetControlRecipe', 1, 'Recette de Contrôle Entière',
'L'enregistrement de l'historique est lié à la recette de contrôle dans son
ensemble.')

VALUES ('RecordSetMasterRecipe', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetMasterRecipe', 1, 'Recette Maître Entière',
'L'enregistrement de l'historique est lié à la recette maître dans son ensemble.')

VALUES ('RecordSetExecutionInfo', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetExecutionInfo', 1, 'Allocation', '')
VALUES ('RecordSetExecutionInfo', 2, 'Dés-allocation', '')
VALUES ('RecordSetExecutionInfo', 3, 'Changement d'Etat', '')
VALUES ('RecordSetExecutionInfo', 4, 'Commande d'Etat', '')
VALUES ('RecordSetExecutionInfo', 5, 'Changement de Mode', '')
VALUES ('RecordSetExecutionInfo', 6, 'Commande de Mode', '')
VALUES ('RecordSetExecutionInfo', 7, 'Message d'Entité Procédurale', '')
VALUES ('RecordSetExecutionInfo', 8, 'Alarme d'Entité Procédurale', '')
VALUES ('RecordSetExecutionInfo', 9, 'Version d'Entité Procédurale', '')
VALUES ('RecordSetExecutionInfo', 10, 'Message d'Invite d'Entité Procédurale', '')
VALUES ('RecordSetExecutionInfo', 11, 'Réponse au Message d'Invite d'Entité
Procédurale', '')

VALUES ('RecordSetMaterialInfo', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetMaterialInfo', 1, 'Consommation de Matière', '')
VALUES ('RecordSetMaterialInfo', 2, 'Production de Matière', '')
VALUES ('RecordSetMaterialInfo', 3, 'Allocation de Matière', '')
VALUES ('RecordSetMaterialInfo', 4, 'Dés-allocation de Matière', '')
VALUES ('RecordSetContinuousData', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetContinuousData', 1, 'Valeur de Donnée Continue', '')
VALUES ('RecordSetContinuousData', 2, 'Association de Tendance', '')
VALUES ('RecordSetContinuousData', 3, 'Dissociation de Tendance', '')
VALUES ('RecordSetEvents', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetEvents', 1, 'Événement Général', '')
VALUES ('RecordSetOperatorChange', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetOperatorChange', 1, 'Intervention Général de l'Opérateur', '')
VALUES ('RecordSetOperatorComment', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetOperatorComment', 1, 'Commentaire Général de l'Opérateur', '')
VALUES ('RecordSetAnalysisData', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetAnalysisData', 1, 'Message Général d'Analyse', '')
VALUES ('RecordSetLateRecord', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetLateRecord', 1, 'Enregistrement Différé Général', '')
VALUES ('RecordSetRecipeData', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetRecipeData', 1, 'Donnée Générique de Recette', '')
VALUES ('RecordSetRecipeData', 2, 'Changement de Valeur de Paramètre de Recette', '')
VALUES ('RecordSetRecipeData', 3, 'Donnée de Résultat de Recette', '')
VALUES ('RecordSetRecipeSpecified', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetRecipeSpecified', 1, 'Donnée Spécifiée Générique de Recette', '')
VALUES ('RecordSetSummaryData', 0, 'Invalide', 'Entrée invalide')
VALUES ('RecordSetSummaryData', 1, 'Donnée Résumée Générique', '')
VALUES ('RecordSetSummaryData', 2, 'Consommation d'Utilités', '')
VALUES ('RecordSetSummaryData', 3, 'temps de Marche d'Équipement', '')
VALUES ('ScheduleChange', 0, 'Invalide', 'Entrée invalide')
VALUES ('ScheduleChange', 1, 'Nouveau', 'Action de changement d'enregistrement de
programme', '')
VALUES ('ScheduleChange', 2, 'Mise à Jour', '')
VALUES ('ScheduleChange', 3, 'Suppression', '')
VALUES ('ScheduleMode', 0, 'Invalide', 'Entrée invalide')
VALUES ('ScheduleMode', 1, 'Automatique', 'Mode de l'enregistrement de programme')
VALUES ('ScheduleMode', 2, 'Semi-Automatique', '')

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 Response', '')
 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 ('ScheduleMode', 3, 'Manuel', '')
VALUES ('ScheduleMode', 4, 'Non Spécifié', '')
VALUES ('ScheduleStatus', 0, 'Invalide', 'Entrée invalide')
VALUES ('ScheduleStatus', 1, 'Effectué', 'Statut de l'enregistrement de programme de batch')
VALUES ('ScheduleStatus', 2, 'En cours', '')
VALUES ('ScheduleStatus', 3, 'Programmé', '')
VALUES ('ScheduleStatus', 4, 'Programmation Suspendue', '')
VALUES ('ScheduleStatus', 5, 'Non Spécifié', '')
VALUES ('SE_Type', 0, 'Invalide', 'Entrée invalide')
VALUES ('SE_Type', 1, 'Campagne',
'Définit le type d'Entrée Programmée')
VALUES ('SE_Type', 2, 'Batch', '')
VALUES ('SE_Type', 3, 'Procédure d'Unité', '')
VALUES ('SE_Type', 4, 'Opération', '')
VALUES ('SE_Type', 5, 'Phase', '')
VALUES ('ValueDataType', 0, 'Invalide', 'Entrée invalide')
VALUES ('ValueDataType', 1, 'Booleen',
'Définit le type de donnée attendu pour une valeur associée.')

VALUES ('ValueDataType', 2, 'Chaîne 8-Bit', '')
VALUES ('ValueDataType', 3, 'Chaîne 16-Bit', '')
VALUES ('ValueDataType', 4, 'Chaîne 32-Bit', '')
VALUES ('ValueDataType', 5, 'Entier non signé 8-Bit', '')
VALUES ('ValueDataType', 6, 'Entier non signé 16-Bit', '')
VALUES ('ValueDataType', 7, 'Entier non signé 32-Bit', '')
VALUES ('ValueDataType', 8, 'Entier signé 8-Bit', '')
VALUES ('ValueDataType', 9, 'Entier signé 16-Bit', '')
VALUES ('ValueDataType', 10, 'Entier signé 32-Bit', '')
VALUES ('ValueDataType', 11, 'Flottant 32-Bit', '')
VALUES ('ValueDataType', 12, 'Double flottant', '')
VALUES ('ValueDataType', 13, 'Chaîne Octet', '')
VALUES ('ValueDataType', 14, 'DateHeure', '')
VALUES ('ValueType', 0, 'Invalide', 'Entrée invalide')
VALUES ('ValueType', 1, 'Constant',
'Définit comment la valeur d'une chaîne est interprétée. Elle contient une valeur fixe en tant que chaîne.')

VALUES ('ValueType', 2, 'Référence',
'Définit comment la valeur d'une chaîne est interprétée. Elle pointe sur la source de la valeur.')

VALUES ('ValueType', 3, 'Equation',
'Définit que la valeur d'une chaîne est interprétée comme une expression qui doit être évalué pour déterminer la valeur.')

VALUES ('ValueType', 4, 'External',
'La valeur est fournie par un moyen externe, et elle n'est pas contenue dans la recette (c'est-à-dire la valeur peut être fournie par une entrée d'opérateur ou par un système d'ordonnancement).')

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_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_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_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_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_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))
```

```

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))

```

Annexe C (informative)

Abréviations

Les abréviations suivantes sont utilisées dans cette norme:

BXT: Table d'échange de batch

EPE: Élément procédural d'équipement

ID: Identification

ISA: Instrument Society of America

MR: Recette maître

PFC: Diagramme fonctionnel de procédure

RE: Élément de recette

SFC: Diagramme fonctionnel en séquence

SOP: Procédure opératoire standard

SQL: Langage de requête structuré

UML: Langage de modélisation unifié

TUC: Temps universel coordonné

Annex C

(informative)

Abbreviations

The following are abbreviations that are used in this standard:

BXT: Batch exchange table

EPE: Equipment procedural element

ID: Identification

ISA: Instrument Society of America

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

Annexe D (informative)

Règles de langage

D.1 Généralités

Un langage est un ensemble de symboles et des règles pour leur utilisation dans la communication. L'article 6 décrit les règles pour créer les symboles et les règles nécessaires pour les communications relatives au processus batch.

La communication, qui implique des machines et des hommes, se produit dans les six activités de contrôle décrites dans la partie 1 de cette norme. Chaque extrémité d'un cheminement de communication doit comprendre les mêmes symboles, et ils doivent utiliser les mêmes règles.

La communication entre les hommes et les systèmes de traitement électronique du batch est généralement assurée avec un écran vidéo ou d'autres appareils graphiques et divers dispositifs de pointage d'entrée. Le texte est fréquemment utilisé pour une communication détaillée, mais les symboles visuels sont souvent plus efficaces pour traduire des relations complexes.

La caractéristique principale du contrôle batch est la recette. Les symboles graphiques relatifs à la procédure, et les règles pour leur utilisation, sont définis dans l'article 6. Les autres éléments de la recette (par exemple formules) contiennent les détails susceptibles d'être mieux représentés visuellement par du texte.

D.2 Dérivation PFC

Trois exemples de méthodes de représentation, le format tabulaire, la notation des diagrammes de Gantt, et les diagrammes fonctionnels en séquence (SFC), ont été décrits dans le rapport ISA TR88.00.03 :1996 [2].

Le format tabulaire est intéressant pour sa simplicité, son interprétation intuitive et sa flexibilité (par exemple support d'attributs supplémentaires énumérés sous forme tabulaire, support d'inserts). Cependant, le format tabulaire est limité aux procédures linéaires par nature parce qu'il ne décrit pas correctement les sélections et les étapes parallèles.

La notation des diagrammes de Gantt est communément utilisée pour représenter les activités qui se déroulent dans le temps, et elle peut être étendue à la représentation des procédures de recette dans des cas plus complexes que le format tabulaire. Cependant, la notation des diagrammes de Gantt ne se prête pas à la représentation de décisions conditionnelles.

Les diagrammes fonctionnels en séquence peuvent être utilisés pour représenter les décisions conditionnelles dans une procédure. Cependant, certains aspects de procédures ne sont pas convenablement représentés par les SFCs ou deviennent inutilement compliqués à décrire avec des SFCs.

Les éléments de ces trois méthodes de représentation ont été combinés pour créer une notation appelée diagramme fonctionnel de procédure (PFC). Le PFC est défini pour représenter graphiquement la partie procédurale de la recette, et il apparaît comme une dérivation de la notation du diagramme fonctionnel tel que défini dans CEI 60848, modifié pour être utilisable dans la représentation de recettes et pour bénéficier de certains avantages de la notation des diagrammes de Gantt et du format tabulaire.

Annex D (informative)

Language guidelines

D.1 General

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 of this standard. 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 device 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 (for example, formula) contain detail that may be better represented visually as text.

D.2 PFC derivation

Three example depiction methods, table format, Gantt chart notation, and sequential function charts (SFC), were discussed in ISA TR88.00.03 :1996 [2].

A table format is attractive because of its simplicity, intuitive interpretation and flexibility (for example, support of additional attributes listed in a tabular form, support 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 the procedure function chart (PFC). The PFC is defined to graphically depict the procedure portion of the recipe. It is a derivation of the function chart notation as defined in IEC 60848 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.3 Procédure de recette

La séparation de la procédure de recette (qui définit la fonctionnalité de processus désirée) de la procédure d'équipement (qui définit l'exécution du contrôle) supporte un des objectifs clairement formulé dans la partie 1: il est admis que les recettes puissent être créées sans la participation de l'ingénierie du contrôle habituelle. Cet objectif définit une division des efforts entre la fonction d'ingénierie du contrôle et la fonction de création de recettes. Cet objectif impose que la fonction d'ingénierie du contrôle définisse et mette en œuvre les éléments procéduraux d'équipement (par exemple les phases d'équipement) et fournisse des représentations utilisables par l'auteur des recettes maîtres, avec des contraintes appropriées.

La partie 1 définit les quatre niveaux suivants d'éléments procéduraux:

- procédure;
- procédure d'unité;
- opération;
- phase.

La procédure consiste en un certain nombre de procédures d'unité. Une procédure d'unité consiste en un ensemble ordonné d'opérations qui représentent une séquence de production contiguë qui s'effectue dans une unité. Une opération est composée d'un ensemble ordonné de phases. Bien qu'il n'y ait pas de limite au nombre de phases qui peuvent être simultanément actives dans une unité, une seule opération est présumée être active dans une unité à un moment donné. Les procédures d'unité sont largement indépendantes, mais elles encapsulent ou référencent des opérations de niveau inférieur et des phases qui peuvent avoir une interaction avec des opérations et des phases dans d'autres procédures d'unité.

D.4 Prescriptions pour la représentation d'éléments de contrôle procéduraux

Une méthode claire pour représenter les éléments de contrôle procéduraux dans une recette est essentielle. Les exigences suivantes ont été identifiées pour une méthode de représentation claire:

- Simple à suivre: facile à comprendre pour les personnes.
- Facile à construire: peu d'obligations syntaxiques et de symboles à apprendre.
- Limites clairement définies: symbole graphique standardisé pour le début et la fin.
- Représentation non ambiguë de l'ordre d'exécution: séquence, parallélisme, sélection (divergence) et convergence.
- Expression de relations de coordination: transferts de matière, attente de, synchronise.
- Niveau hiérarchique: symboles standardisés pour les procédures, procédure d'unité, opération, phase.
- Existence de niveaux: symbole graphique standardisé pour montrer la décomposition possible d'un élément de la hiérarchie.
- Applicable aux recettes maîtres et aux recettes de contrôle.
- Applicable à tous les niveaux: ensemble de symboles et de règles similaire à tous les niveaux dans une recette.
- Indépendant du media: utilisable et compréhensible qu'elle soit mise en œuvre avec un crayon et du papier ou par des graphiques animés en couleur sur ordinateur.

D.3 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 (for example, 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:

- procedure;
- unit procedure;
- operation;
- phase.

The procedure consists of a 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.4 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:

- Simple to follow: easy for people to understand.
- Easy to build: few syntax requirements and symbols to learn.
- Clearly defined boundaries: standardised graphical symbol for start and end.
- Unambiguous depiction of execution order: sequence, parallelism, selection (divergence) and convergence.
- Expression of coordination relationships: material transfers, wait for, synchronise.
- Hierarchical level: standardised symbols for procedure, unit procedure, operation, phase.
- Existence of levels: standardised graphical symbol to show possible decomposition of an element of the hierarchy.
- Applicable to master recipes and control recipes.
- Applicable to all levels: similar set of symbols and rules at all levels in a recipe.
- Independent of media: equally usable and understandable whether implemented with pencil and paper or with full-colour animated computer graphics.

Annexe E (informative)

Exemples de traitement de diagrammes fonctionnels de procédures

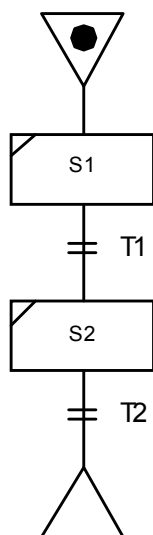
Beaucoup de règles concernant le traitement de diagrammes fonctionnels de procédures sont dépendantes du système. Les exemples suivants illustrent les systèmes de traitement possibles du PFC.

Dans ces exemples, un point est utilisé pour identifier les symboles actifs. Le point ne fait pas partie des symboles, il est utilisé seulement à des fins explicatives. Actif signifie que les symboles sont en cours d'évaluation ou d'exécution par le système de traitement du PFC.

Lorsqu'un élément procédural est actif, le PFC du niveau inférieur ou l'élément procédural d'équipement qu'il représente est en cours de traitement selon son graphe d'état. Pour ces exemples, le graphe d'état donné à titre d'exemple dans la CEI 61512-1 est utilisé. Inactif signifie que les symboles ne sont pas en cours d'évaluation ou d'exécution par le système de traitement du PFC. Lorsqu'un élément procédural est inactif, le PFC de niveau inférieur ou l'élément procédural d'équipement qu'il représente est dans un état compatible avec son graphe d'état, mais n'étant pas en cours de traitement, il ne peut pas changer d'état.

Lorsqu'une transition explicite est active, son expression est en cours d'évaluation par le système de traitement du PFC, ou bien l'expression a déjà été évaluée VRAIE et une notification de ce fait a été envoyée aux PFC du niveau inférieur ou à l'entité procédurale d'équipement que représente l'élément de recette précédent.

Exemple 1: La phase s'achève après que la transition qui la suit a été évaluée VRAIE.



Le PFC a démarré et le symbole de début est actif.

Annex E (informative)

Procedure function chart processing examples

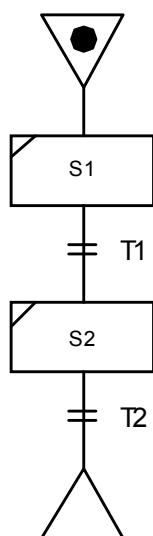
Many rules concerning procedure function chart processing are system dependent. The following examples illustrate possible PFC processing systems.

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.

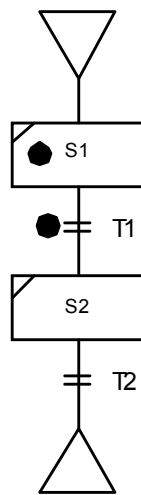
When a procedural element is active, the lower-level PFC or equipment procedural element that it represents is currently being processed according to its state model. For these examples, the IEC 61512-1 example state model is used. Inactive means that the symbols are not being evaluated or executed by 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.

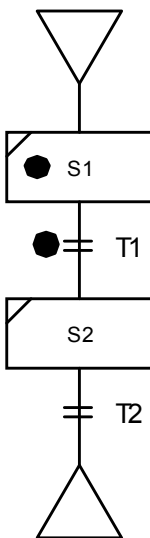


The PFC has started and the begin symbol is active.



La phase de recette S1 devient active. La phase d'équipement correspondante (non représentée) est également devenue active.

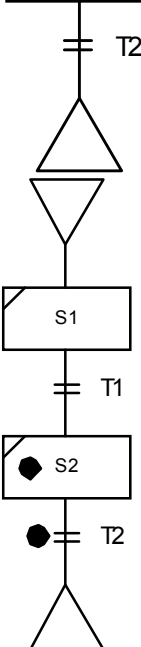
La transition explicite T1 devient active, et elle est continuellement évaluée une fois que la phase de recette S1 devient active.



T1 est évaluée VRAIE.

La phase d'équipement S1 est informée du fait que T1 est VRAIE.

La phase d'équipement S1 continue d'être active jusqu'à ce que sa propre logique la rende inactive.



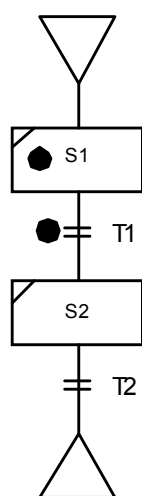
La phase d'équipement S1 passe à l'état effectué, et elle devient inactive. Cet achèvement peut être le résultat d'une notification par T1 et du succès de ses traitements «ménagers» internes.

La phase de recette S1 devient inactive une fois que la phase d'équipement S1 devient inactive.

T1 cesse d'être évaluée et devient inactive.

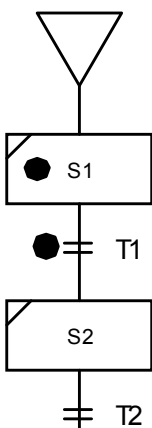
La phase de recette S2 et la phase d'équipement S2 correspondante deviennent actives.

La transition explicite T2 devient active, et elle est continuellement évaluée.



Recipe phase S1 becomes active. The corresponding equipment phase (not shown) has also become active.

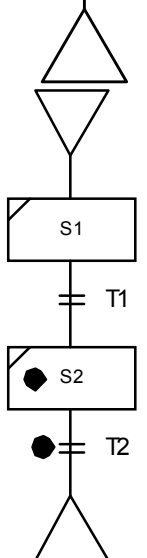
Explicit transition T1 becomes active, and it is continually evaluated once recipe phase S1 becomes active.



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.

Bibliographie

- [1] Rumbaugh, J. Jacobsen, I and J Booch, *The unified modelling Language Manual*, 1999, Addison-Wesley)
 - [2] ISA TR88.0.03:1996, *Possible recipe procedure presentation formats*
-

Bibliography

- [1] Rumbaugh, J., Jacobsen, I. And G. Booch, *The Unified Modeling Language Reference Manual*, 1999, Addison-Wesley
 - [2] ISA TR88.0.03:1996, *Possible recipe procedure presentation formats*
-

ISBN 2-8318-6064-4



ICS 35.240.50

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND