

ANSI/ISA-88.00.02-2001



Batch Control Part 2: Data Structures and Guidelines for Languages

NOTICE OF COPYRIGHT

This is a copyrighted document and may not be copied or distributed in any form or manner without the permission of ISA. This copy of the document was made for the sole use of the person to whom ISA provided it and is subject to the restrictions stated in ISA's license to that person. It may not be provided to any other person in print, electronic, or any other form. Violations of ISA's copyright will be prosecuted to the fullest extent of the law and may result in substantial civil and criminal penalties.



ISA—The Instrumentation,
Systems, and
Automation Society

Approved 7 February 2001



Documents
Document Sharing Hub

ANSI/ISA-88.00.02-2001

Batch Control Part 2: Data Structures and Guidelines for Languages

Copyright © 2001 by ISA-The Instrumentation, Systems, and Automation Society. All rights reserved. Not for resale. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), without the prior written permission of the Publisher.

ISBN: 1-55617-745-3

ISA
67 Alexander Drive
PO Box 12277
Research Triangle Park, NC 27709
USA

Preface

This preface, as well as all footnotes and annexes, is included for information purposes and is not part of ANSI/ISA-88.00.02-2001.

The standards referenced within this document may contain provisions that, through reference in this text, constitute requirements of this document. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated within this document. Members of IEC and ISO maintain registers of currently valid International Standards. ANSI maintain registers of currently valid U.S. National Standards.

This document has been prepared as part of the service of ISA—The Instrumentation, Systems, and Automation Society, toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static, but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms and asks that they be addressed to the Secretary, Standards and Practices Board; ISA; 67 Alexander Drive; P. O. Box 12277; Research Triangle Park, NC 27709; Telephone (919) 549-8411; Fax (919) 549-8288; E-mail: standards@isa.org.

The ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards. The Department is further aware of the benefits to USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, this Department will endeavor to introduce SI-acceptable metric units in all new and revised standards, recommended practices, and technical reports to the greatest extent possible. *Standard for Use of the International System of Units (SI): The Modern Metric System*, published by the American Society for Testing & Materials as IEEE/ASTM SI 10-97, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

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

CAUTION — ISA ADHERES TO THE POLICY OF THE AMERICAN NATIONAL STANDARDS INSTITUTE WITH REGARD TO PATENTS. IF ISA IS INFORMED OF AN EXISTING PATENT THAT IS REQUIRED FOR USE OF THE STANDARD, IT WILL REQUIRE THE OWNER OF THE PATENT TO EITHER GRANT A ROYALTY-FREE LICENSE FOR USE OF THE PATENT BY USERS COMPLYING WITH THE STANDARD OR A LICENSE ON REASONABLE TERMS AND CONDITIONS THAT ARE FREE FROM UNFAIR DISCRIMINATION.

EVEN IF ISA IS UNAWARE OF ANY PATENT COVERING THIS STANDARD, THE USER IS CAUTIONED THAT IMPLEMENTATION OF THE STANDARD MAY REQUIRE USE OF TECHNIQUES, PROCESSES, OR MATERIALS COVERED BY PATENT RIGHTS. ISA TAKES NO POSITION ON THE EXISTENCE OR VALIDITY OF ANY PATENT RIGHTS THAT MAY BE INVOLVED IN IMPLEMENTING THE STANDARD. ISA IS NOT RESPONSIBLE FOR IDENTIFYING ALL PATENTS THAT MAY REQUIRE A LICENSE BEFORE IMPLEMENTATION OF THE STANDARD OR FOR INVESTIGATING THE VALIDITY OR SCOPE OF ANY PATENTS BROUGHT TO ITS ATTENTION. THE USER SHOULD CAREFULLY INVESTIGATE RELEVANT PATENTS BEFORE USING THE STANDARD FOR THE USER'S INTENDED APPLICATION.

HOWEVER, ISA ASKS THAT ANYONE REVIEWING THIS STANDARD WHO IS AWARE OF ANY PATENTS THAT MAY IMPACT IMPLEMENTATION OF THE STANDARD NOTIFY THE ISA STANDARDS AND PRACTICES DEPARTMENT OF THE PATENT AND ITS OWNER.

ADDITIONALLY, THE USE OF THIS STANDARD MAY INVOLVE HAZARDOUS MATERIALS, OPERATIONS OR EQUIPMENT. THE STANDARD CANNOT ANTICIPATE ALL POSSIBLE APPLICATIONS OR ADDRESS ALL POSSIBLE SAFETY ISSUES ASSOCIATED WITH USE IN HAZARDOUS CONDITIONS. THE USER OF THIS STANDARD MUST EXERCISE SOUND PROFESSIONAL JUDGMENT CONCERNING ITS USE AND APPLICABILITY UNDER THE USER'S PARTICULAR CIRCUMSTANCES. THE USER MUST ALSO CONSIDER THE APPLICABILITY OF ANY GOVERNMENTAL REGULATORY LIMITATIONS AND ESTABLISHED SAFETY AND HEALTH PRACTICES BEFORE IMPLEMENTING THIS STANDARD.

This standard is structured to follow IEC (International Electrotechnical Commission) guidelines. Therefore, the first three clauses discuss the *Scope* of the standard, *Normative References*, and *Definitions*, in that order.

Clause 4 is entitled *Data Model*. The intent of this clause is to describe a data structure for batch control systems using an object model approach.

Clause 5 is entitled *Relational Tables for Information Exchange*. The intent of this clause is to discuss a data format that can be used to share recipes and other batch information across different systems.

Clause 6 is entitled *Procedure Function Charts*. The intent of this clause is to describe a symbolic language for recipe depiction.

This standard is intended for people who are

- a) involved in designing and/or operating batch manufacturing plants;
- b) responsible for specifying controls and the associated application programs for batch manufacturing plants; or
- c) involved in the design and marketing of products in the area of batch control.

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

NAME	COMPANY
L. Craig, Chairman	Rohm and Haas Co.
T. Fisher, Editor	The Lubrizol Corp.
M. Albano	Honeywell, Inc.
J. Barrault	Siemens AG
D. Brandl	Sequencia Corp.
B. Braunstein	Exxon Chemical Co.
E. Bristol	The Foxboro Co.
L. Charpentier	GSE Systems Inc.
T. Crowl	Siemens Moore Process Automation
C. Eaves	Intellution, Inc.
D. Emerson	Yokogawa Corp. of America
L. Falkenau	E.I. du Pont de Nemours and Co.
M. Fersky	ABB Automation Inc.
D. Fleming	Rockwell Automation
A. Ghosh	ARC

R. Hall
 W. Hawkins
 N. Haxthausen
 B. Jensen
 B. Korkmaz
 D. Kraska
 T. McFarlane
 R. Mergen
 D. Moffatt
 P. Moylan
 T. Müller-Heinzerling
 L. Natiello
 P. Nelson
 L. Noble
 L. Pillai
 S. Prichard
 T. Province
 R. Salisbury
 M. Saucier
 D. Sweeny
 B. Sykes
 T. Tom
 K. Unger
 M. Van Epps
 J. Vieille
 A. Webster
 A. Weidenbach
 R. Winslow
 B. Winters
 D. Wolin

Sequencia Corp.
 HLQ Ltd.
 Novo Nordisk Engineering
 Yokogawa Corp. of America
 Automation Vision Inc.
 Geon Co.
 Neles Automation Inc.
 The Lubrizol Corp.
 Wonderware Corp.
 Rockwell Automation
 Siemens AG
 Merck
 Dow Corning Corp.
 ABB Automation, Inc.
 Pharmacia & Upjohn, Inc.
 GE Fanuc Automation
 Eastman Chemical Co.
 ABB Industrial Systems Inc.
 Sequencia Corp.
 PharmComm Inc.
 Yokogawa Corp. of America
 Siemens Moore Process Automation
 Real Enterprise Solutions
 Pharmacia Corp.
 Jean Vieille Conseil
 E.I. du Pont de Nemours and Co.
 Eastman Chemical Co.
 Sterling Electronics
 Honeywell IAC
 Siemens Moore Process Automation

This standard was approved for publication by the ISA Standards and Practices Board on 03 January 2001.

NAME

M. Zielinski
 D. Bishop
 M. Cohen
 M. Coppler
 B. Dumortier
 W. Holland
 E. Iccayan
 A. Iverson
 R. Jones
 V. Maggioli
 T. McAviney
 A. McCauley, Jr.
 G. McFarland
 D. Rapley
 R. Reimer
 J. Rennie
 H. Sasajima
 I. Verhappen

COMPANY

Fisher-Rosemount Systems, Inc.
 Consultant
 Senior Flexonics, Inc.
 Ametek, Inc.
 Schneider Electric SA
 Southern Company
 Advanced Control & Engineering Solutions
 Ivy Optiks
 Dow Chemical Co.
 Feltronics Corp.
 Bateman Engineering, Inc.
 Chagrin Valley Controls, Inc.
 Westinghouse Process Control Inc.
 Rapley Consulting Inc.
 Rockwell Automation
 Factory Mutual Research Corp.
 Yamatake Corp.
 Syncrude Canada Ltd.



R. Webb
W. Weidman
J. Weiss
M. Widmeyer
R. Wiegler
C. Williams
G. Wood

Altran Corp.
Parsons Energy & Chemicals Group
EPRI
EG&G Defense Materials
CANUS Corp.
Eastman Kodak Co.
Graeme Wood Consulting

ANSI/ISA-88.00.02-2001



Contents

1	Scope	13
2	Normative references	13
3	Definitions	13
4	Data model	14
4.1	Introduction	15
4.2	Overview model	15
4.3	Recipe model	17
4.4	Equipment model	29
4.5	Production planning and scheduling	31
4.6	Production information management	34
5	Relational tables for information exchange	37
5.1	Introduction	37
5.2	Master recipe information.....	52
5.3	Process cell equipment model exchange.....	64
5.4	Schedule information exchange.....	71
5.5	Production information exchange	75
5.6	Exchange table domains.....	78
6	Procedure function charts.....	80
6.1	Procedure function chart notation	81
6.2	Control recipe depiction	99
6.3	Exception handling	99
	Annex A (normative) — Data modeling technique	101
	Annex B (normative) — SQL definition listing	105
	Annex C (informative) — Abbreviations	117
	Annex D (informative) — Language guidelines	119
	Annex E (informative) — Procedure function chart processing examples	121

This page intentionally left blank.

Foreword

- 1) The formal decisions or agreements of the IEC on technical matters, prepared by technical committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.
- 2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.
- 3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules insofar as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rules should, as far as possible, be clearly indicated in the latter.
- 4) The IEC has not laid down any procedure concerning marking as an indication of approval and has no responsibility when an item of equipment is declared to comply with one of its recommendations.

This standard has been prepared by IEC SC65A WG11 and ISA SP88.

It forms Part 2 of a series, the other part being ANSI/ISA-88.01-1995, Batch Control Part 1: Models and Terminology.

Annexes A and B form an integral part of this standard. Refer to annex A for an explanation of the UML notation that is used in this standard. Refer to annex B for a summation of all of the SQL definitions from clause 5. Annexes C and D are for information only.



This page intentionally left blank.

Introduction

ANSI/ISA-88.01-1995, Batch Control Part 1: Models and Terminology (referred to as Part 1 throughout this standard), provides models and terminology applicable to batch control. This standard (referred to as Part 2 throughout the standard) addresses data structures and guidelines for languages. Data structures are addressed by the data model that is defined in clause 4 that more precisely identifies objects and relationships that are addressed by models and concepts of Part 1. Data structures are also addressed by relational tables for information exchange that are defined in clause 5. Languages are addressed by a recipe depiction methodology that is defined in clause 6.

The intended use of the data model is to provide a starting point for developing interface specifications for software components that address any subset of the Part 1 standard. The data model addresses all of the Part 1 standard as an integrated object model, but it does not presume or preclude any specific system architecture or information exchange. The model does not assume any specific division of functionality between systems.

A specific method for the exchange of selected data is defined in clause 5. Relational tables are used as the information exchange method because, within the bounds of the information treated, they

- a) utilize broadly available technologies;
- b) are amenable to translation to other technologies;
- c) are adequate; and
- d) are consistent with other sections of the standard.

Multiple methods of information transfer have not been defined, nor has there been an attempt to identify all information that might be exchanged. In the future, additional methods may be defined to provide alternate ways to exchange data.

Clause 6 defines the symbols and rules for a graphical language that can be used to depict recipes. Recipes are the central feature of batch control, and they can address a wide range of complexity, but there is no one depiction that is ideal for all circumstances. A simple table, for example, might be the most appropriate recipe form for simple cases. This standard specifies a method for depiction of master and control recipe procedures that can be applied over a broader range of complexity.

Although this standard is intended primarily for batch processes, there may be considerable value for other types of processes.

...

This page intentionally left blank.



Documents
Document Sharing Hub

1 Scope

This Part 2 standard on Batch Control defines data models that describe batch control as applied in the process industries, data structures for facilitating communications within and between batch control implementations, and language guidelines for representing recipes.

2 Normative references

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

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

IEC 60902:1987, *Industrial-process measurement and control: Terms and definitions*

NOTE — IEC 60902:1987 has been replaced by IEC 60050-351:1998, International electrotechnical vocabulary-Part 351: Automatic control

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

IEC 61512-1:1997, *Batch control – Part 1: Models and terminology*

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

ISO/IEC 9075:1992, *Information processing systems – Database language SQL with integrity enhancement*

3 Definitions

ISA hereby grants a non-exclusive, royalty-free, limited license under ISA's copyright in the standard, to copy, display and distribute this section of this ISA standard (including software included in or defined by such section), as follows:

1. Producers of products or services intended to comply with the standard may incorporate this designated section, but only to the extent reasonably necessary to make, use, and distribute any product or service (including product documentation) that is compliant with this standard.
2. End users of a product or service made by a producer acting under the preceding license may reproduce and use the designated section, but only to the extent reasonably necessary to enjoy the intended functions of the product or service and to maintain, configure, or reconfigure systems to be compliant.
3. Persons providing education on or promotion of the standard may copy, display and distribute the designated section, but only to the extent reasonably necessary to provide information related to the standard.

Except as expressly permitted, all other reproduction and distribution without permission of ISA is prohibited. All copies of this section of the standard made or distributed under this license must cite the standard and include the following notice of copyright:

Copyright © 2001 by ISA—The Instrumentation, Systems, and Automation Society. All rights reserved. Used with permission of ISA.

For the purposes of this standard, the following definitions apply. Definitions and concepts expressed in ANSI/ISA-88.01-1995 apply, except where differences are explicitly stated in this Part 2 standard. Definitions in IEC 60902:1987 were also used as a basis.

3.1 allocation symbol:

a graphical symbol that is used to represent the encapsulation of the resource allocation and de-allocation rules for a recipe procedural element.

3.2 building block:

a recipe entity that exists in a library.

3.3 enumeration set:

a list of predefined strings and their associated numerical values.

3.4 exchange table:

a database table that is used to exchange batch-related information between systems.

3.5 link:

an object that specifies the connection between two other objects (e.g., the connection between recipe entities or between recipe entities and transitions).

3.6 procedure function chart:

a graphical representation of a recipe procedure that specifies the processing order for recipe procedural elements.

3.7 recipe element:

a structural entity that is used to represent recipe entities and symbols, except transitions and directed links, that are used in procedure function charts.

3.8 recipe entity:

the combination of a procedural element with associated recipe information (e.g., header, formula, equipment requirements, other information). General, site, master, and control recipes are also recipe entities.

4 Data model

ISA hereby grants a non-exclusive, royalty-free, limited license under ISA's copyright in the standard, to copy, display and distribute this section of this ISA standard (including software included in or defined by such section), as follows:

1. Producers of products or services intended to comply with the standard may incorporate this designated section, but only to the extent reasonably necessary to make, use, and distribute any product or service (including product documentation) that is compliant with this standard.
2. End users of a product or service made by a producer acting under the preceding license may reproduce and use the designated section, but only to the extent reasonably necessary to enjoy the intended functions of the product or service and to maintain, configure, or reconfigure systems to be compliant.

3. **Persons providing education on or promotion of the standard may copy, display and distribute the designated section, but only to the extent reasonably necessary to provide information related to the standard.**

Except as expressly permitted, all other reproduction and distribution without permission of ISA is prohibited. All copies of this section of the standard made or distributed under this license must cite the standard and include the following notice of copyright:

Copyright © 2001 by ISA–The Instrumentation, Systems, and Automation Society. All rights reserved. Used with permission of ISA.

4.1 Introduction

This clause defines data models that specify a set of objects, attributes, and their basic relationships that cover the concepts of Part 1 at a high level of abstraction. The models apply to interfaces to batch control systems in a technology-independent manner. The models are not intended to address the internal system architecture of batch control systems.

The intended use of these models is to provide a starting point for developing interface specifications for software components that address any subset of Part 1.

The models address all of Part 1 as an integrated object model, but they do not presume or preclude any specific system architecture or information exchange. The models do not assume any specific division of functionality between systems.

In the cases where the objects and relationships defined in this clause are presented through an interface, then that interface shall use the object names, the attribute names, and the relationships of this clause commensurate with the interface technology chosen and the capabilities offered. An example of such an interface is the SQL relational table interface that is defined in clause 5.

An exchange format or interface specification may implement only some objects or parts of objects (e.g., not all properties are defined). An exchange format or interface specification may also provide additional objects or properties (e.g., phase duration information), including the expansion of any data model attribute into multiple attributes. Any such implementation shall be consistent with the data model that is presented here and the concepts of Part 1.

4.1.1 Modeling techniques

The models that are described in this clause are based on the Unified Modeling Language (UML) (see clause A.1).

The tables describe only the class attributes of the objects. The relationships between objects are described in the figures.

4.2 Overview model

This model (see figure 1) gives a high level overview of the main classes that are defined and the relations between these classes for the batch domain that is described by the Control Activity Model of Part 1. The individual object classes are more thoroughly described in the subsequent models in this subclause.

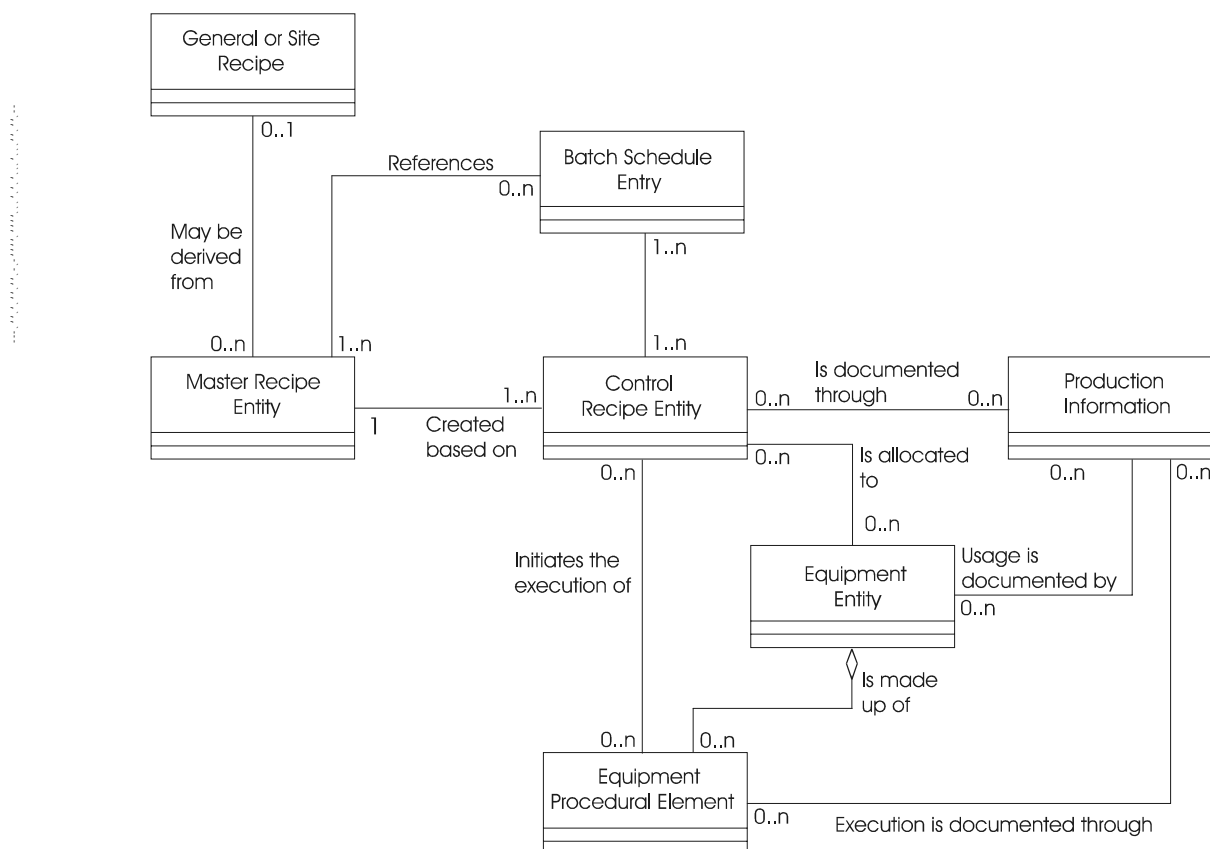


Figure 1 — Overview model

A general or site recipe is made up of a hierarchy of *general recipe entities* that correspond to the procedural entities (process stages, process operations, and process actions).

A master recipe may be derived from a general or site recipe. The master recipe itself may be seen as a top-level master recipe entity. A master recipe is made up of a hierarchy of *master recipe entities* that correspond to the procedural entities (i.e., procedures, unit procedures, operations, phases).

A *batch schedule entry* specifies production of a specific batch through the execution of a recipe. A batch schedule is, in principle, a list that specifies the production of batches, and it includes information about timing. The product-specific information necessary for this purpose is derived from a related *master recipe entity*.

Based on a *Batch Schedule Entry*, a control recipe starts as a copy of a specific version of a master recipe, and it is then modified to create the recipe that will produce the batch. A control recipe includes the information necessary for equipment control.

Control recipe entities are created based on *master recipe entities*. A control recipe may be enhanced with additional information (e.g., scaling, equipment assignment), and it may be modified (including creating or removing control recipe entities).

Equipment entities are selected and allocated to the *control recipe entities*.

A *control recipe entity* may be linked to an *equipment procedural entity* within the *equipment entity* (normally the unit). The *equipment procedural entity* may be initiated, and its parameters may be assigned recipe values.

Production information is generated during the production of the batch. This information may be related to recipe entities, equipment entities, and/or equipment procedural elements.

4.3 Recipe model

4.3.1 Recipe entity

Recipes are organized hierarchically, with various categories of information at each level. The recipe entity is the construct that is used to represent the tight coupling of the data at a particular level.

The recipe entity is the fundamental structure in all kinds of recipes (see figure 2). The recipe entity structurally takes the place of the recipe procedural element as defined in Part 1, but it may include any or all of the recipe components: procedural definitions, parameters with values, equipment requirements and other information.

The class specifications are shown in table 1.

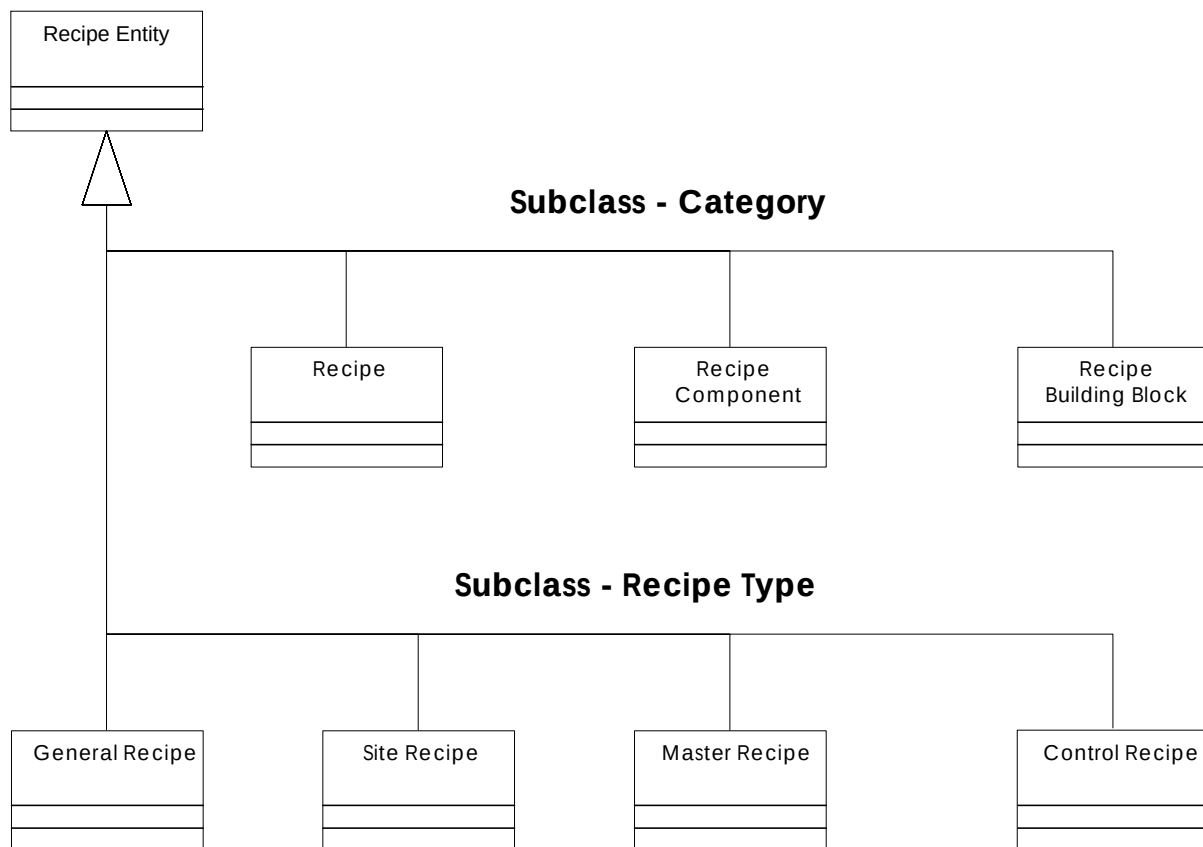


Figure 2 — Recipe entities

Table 1 — Recipe entity

NAME	RECIPE ENTITY
FunctionalDescription	A recipe entity is the combination of a procedural element with associated recipe information (e.g., header, formula, equipment requirements and other information). General, site, master, and control recipes are recipe entities. One example is a unit recipe (ANSI/ISA-88.01-1995, 5.3.2). "unit recipe: the part of a control recipe that uniquely defines the contiguous production requirements for a unit." (ANSI/ISA-88.01-1995, 3.62).
ATTRIBUTES	
RecipeEntityID	Provides unique Identification.

A recipe itself is considered to be a recipe entity (category: recipe). The recipe is built up of lower-level recipe entities (e.g., unit recipes) (category: component). When constructing a specific recipe, the components may be created from library elements (category: building block).

The recipe entity concept is applied to all recipe types: general, site, master and control. When a recipe is executed, the representation in the batch history of the executed recipe entities will have much the same structure, and it is therefore shown as a subclass. An overview of the subclasses is shown in table 2. Subclasses by category are shown in tables 3 to 5. Subclasses by type are shown in tables 6 to 9. General and site recipes are not discussed further in this subclause.

Table 2 — Subclasses – overview

	GENERAL RECIPE ENTITY	SITE RECIPE ENTITY	MASTER RECIPE ENTITY	CONTROL RECIPE ENTITY
RECIPE	A complete and self-contained general recipe.	A complete and self-contained site recipe.	A complete and self-contained master recipe.	A complete and self-contained control recipe.
RECIPE BUILDING BLOCK	A generic general recipe entity type that can be instantiated in a specific recipe or in another building block.	<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., North Carolina's Finest Premium Beer).
Status*	Defines the Status of the information (e.g., "Approved for Production", "Approved for Test", "Not Approved", "Inactive", "Obsolete").
*Not required for control recipe (available by reference to master recipe).	

Table 4 — Recipe component

NAME	RECIPE COMPONENT
FunctionalDescription	A recipe entity that is part of a recipe or building block (i.e., an instance of a building block in a particular recipe or containing building block recipe entity).
ATTRIBUTES	
Level	Indicates the procedural hierarchy level (i.e., process stage, process operation, or process action for general and site recipes and unit procedure, operation, or phase for master and control recipes).
RE_Use	Defines if the recipe component is a copy of a building block or a reference to it.

Table 5 — Recipe building block

NAME	RECEIPE BUILDING BLOCK
FunctionalDescription	A recipe entity that exists in a library. A building block can be parameterized and used when building recipes.
ATTRIBUTES	
RecipeVersion	Identifies the version of the recipe entity.
VersionDate	Identifies the date and time that this version of the recipe was created or modified.
ApprovalDate	Identifies the date and time that this version of the recipe was approved.
Author	Identifies the person or system that authored this version of the recipe (e.g., J. Smith).
ApprovedBy	Identifies the person or system that approved this version of the recipe.
Description	Describes the function that is achieved through execution of this version of the recipe entity.
Level	Indicates the level of the recipe entity.
UsageConstraint	Defines other rules that determine the usage (e.g., "always succeeded by..." or "never runs in parallel with...").
Status	Defines the Status of the recipe entity (e.g., "Approved for Production", "Approved for Test", "Not Approved", "Inactive", "Obsolete").
Function	Defines how the recipe entity is executed (e.g., by referencing an equipment procedural element, through execution of contained logic).

Table 6 — General recipe entity

NAME	GENERAL RECIPE ENTITY
FunctionalDescription	All of a general or site recipe, a component of a general or site recipe, or a building block for creation of a general or site recipe.
ATTRIBUTES	
ScaleReference	Defines a reference scale for the parameter values.

Table 7 — Site recipe entity

NAME	SITE RECIPE ENTITY
FunctionalDescription	All of a site recipe, a component of a site recipe, or a building block for creation of a site recipe.
ATTRIBUTES	
ScaleReference	Defines a reference scale for the parameter values.

Table 8 — Master recipe entity

NAME	MASTER RECIPE ENTITY
FunctionalDescription	All of a master recipe, a component of a master recipe, or a building block for creation of a master recipe.
ATTRIBUTES	
ScaleReference	Defines a reference scale for the parameter values.
ProcessCellID	Identifies the equipment category for which the recipe entity was defined (e.g., process cell or process cells for which this master recipe was defined).

Table 9 — Control recipe entity

NAME	CONTROL RECIPE ENTITY
FunctionalDescription	A recipe entity that is all of or a part of a control recipe.
ATTRIBUTES	
BatchID	Specifies the actual Batch ID.
BatchSize	Defines the requested size or scale factor for the batch, based on the scale factor for the batch as defined in the master recipe.
Status	Describes execution status (e.g., not yet activated, active, or completed).

4.3.2 Parts of recipe entities

The model (see figure 3 and tables 10 through 13) shows the categories of information of a recipe as specified in Part 1. The model indicates that these components may exist at any level of the decomposition of the recipe (e.g., a unit recipe may have its own equipment requirements).

- 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 is modeled as a set of parameter objects. All levels of the recipe decomposition may have parameters, including the recipe itself. See 4.3.6 on parameters.
- The modeling of Equipment Requirements is discussed in 4.3.5.
- The Other Information category, as defined in Part 1, is represented as a single object class, even though other information may have multiple elements and different structures.
- The Procedure category of Part 1 is modeled as a set of procedural structural elements.

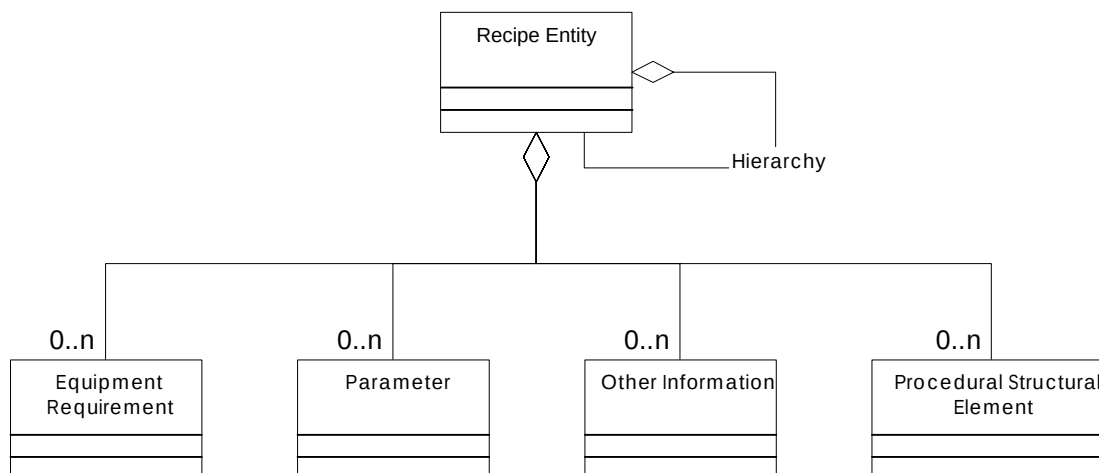


Figure 3 — Parts of recipe entities

Table 10 — Parameter

NAME	PARAMETERS
FunctionalDescription	Formula values or placeholders for values that are to be communicated to and from recipe entities during execution.
ATTRIBUTES	
ParameterID	Provides unique identification.
ParameterType	Specifies how the parameter value is interpreted (e.g., constant, reference equation).
Description	Describes the parameter or use of the parameter.
EngineeringUnits	Identifies the engineering units of measure for the Value (e.g., kg, pounds).
Value	Contains the parameter value. If Value is a relation, it contains the equation, form, deferral rule or whatever ties the related parameters together. If Value is a building block parameter, this attribute will hold the default value.
Scaled	Specifies the scaling rule. Simplest case: scaled or not scaled with batch reference size.
Usage	Specifies the parameter as a process input, process output, or process parameter.

Table 11 — Equipment requirement

NAME	EQUIPMENT REQUIREMENT
FunctionalDescription	Represents an equipment requirement as specified in the recipe entity.
ATTRIBUTES	

Table 12 — Other information

NAME	OTHER INFORMATION
FunctionalDescription	A category of recipe information that may contain batch processing support information that is not contained in other parts of the recipe (e.g., regulatory compliance information, materials and process safety information, process flow diagrams, packaging/labeling information).
ATTRIBUTES	

Table 13 — Procedural structural element

NAME	PROCEDURAL STRUCTURAL ELEMENT
FunctionalDescription	The recipe procedural elements and the ordering information for their execution.
ATTRIBUTES	

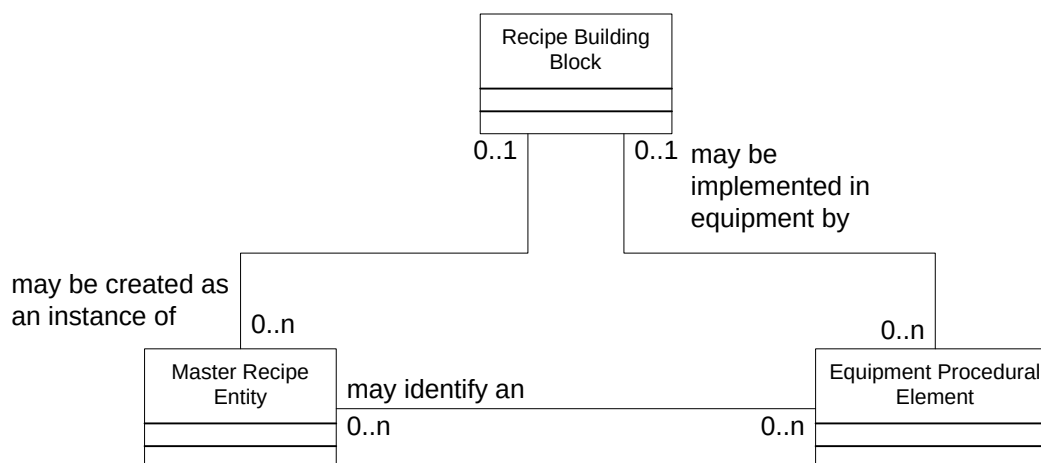
4.3.3 Recipe entity relation (procedural structure)

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

Procedural structural elements include the recipe procedural elements and the connections (e.g., link, transition) that are used to order them (e.g., a unit recipe's procedural structural elements are the operations and the ordering of those operations contained within it). The procedural structural elements may be related to other procedural structural elements.

4.3.4 Recipe building blocks

Recipe building blocks are an important concept in the data model (see figure 4). This figure represents the relations at a single level of the procedure hierarchy.

**Figure 4 — Recipe building block**

Recipe building blocks are the building blocks from which master recipes are created. When a *recipe building block* is instantiated in a master recipe as a *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* (e.g., subordinate recipe entities) may be copied into *master recipe entities*. These same lower-level contents may also be accessible by reference to the *recipe building block*.

A *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 execution of the recipe entity.
Level	Indicates the level of the equipment entity. The equipment entity may only be used at this level.
Mode	Indicates the current mode of the procedural element.
State	Indicates the current state of the procedural element.

The mechanisms may be illustrated by the following example (see figure 5) that could represent an object model of a part of an actual application. The generic concept of a building block is instantiated as a specific building block "heat." The "is based on" relation between building blocks and components is replaced by a subclass relation (this is one possible implementation, and it indicates that if the "heat" building block is changed, then the change will propagate to all recipes that use "heat"). Another implementation could be that "heat" is just copied when instantiated.

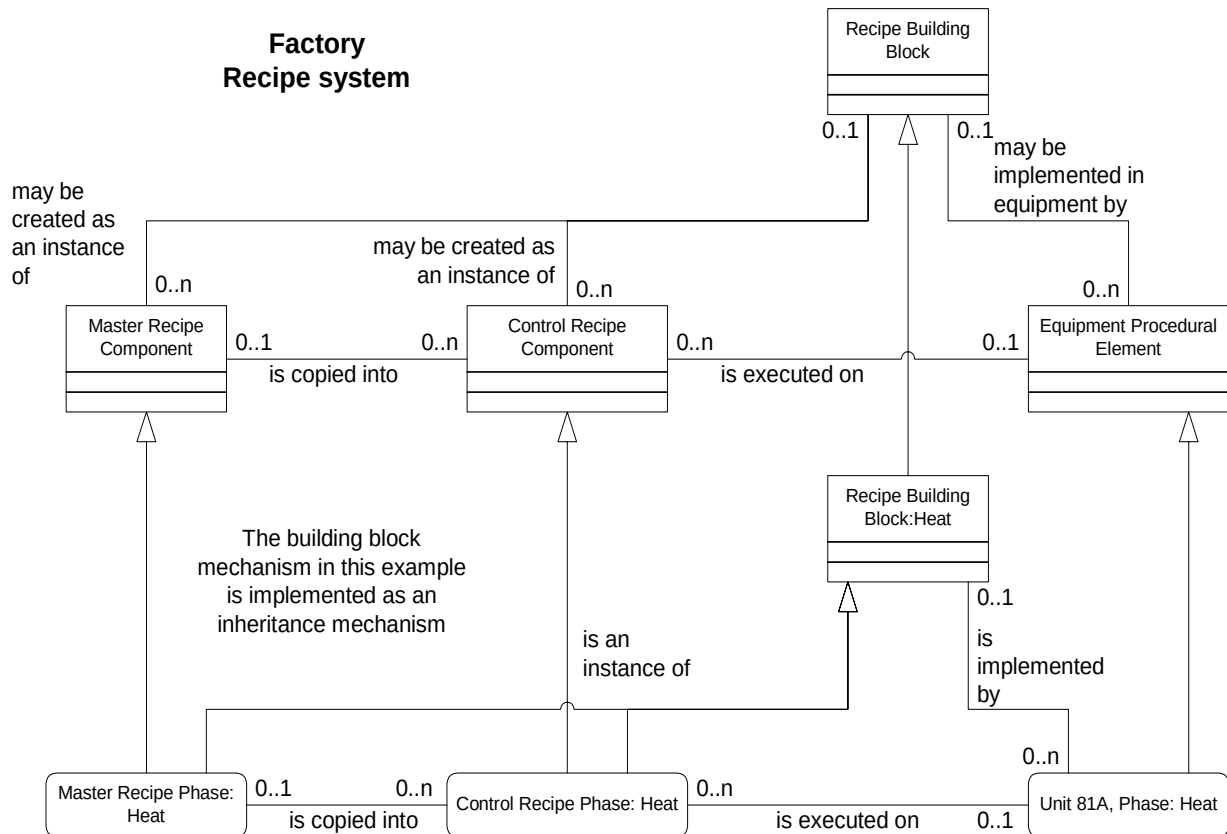


Figure 5 — Building block concept

4.3.5 Equipment requirements

Recipe entities may contain *equipment requirements* (see figure 6 and tables 15, 16, and 17). Equipment requirements reference a specific *equipment property type* (e.g., an *equipment property type* may be the 'size of vessel' or the 'lining of vessel'). A specific equipment requirement could then specify a minimum value for the size of a vessel.

This requirement may then be met by a piece of equipment with a certain equipment property that references the same equipment property type. For example, a specific unit, UNIT12, would have a value for the property type 'size of vessel'.

An equipment entity is a specific piece of equipment, and it may be replaced by a class of equipment (equipment class). See 4.4.1.

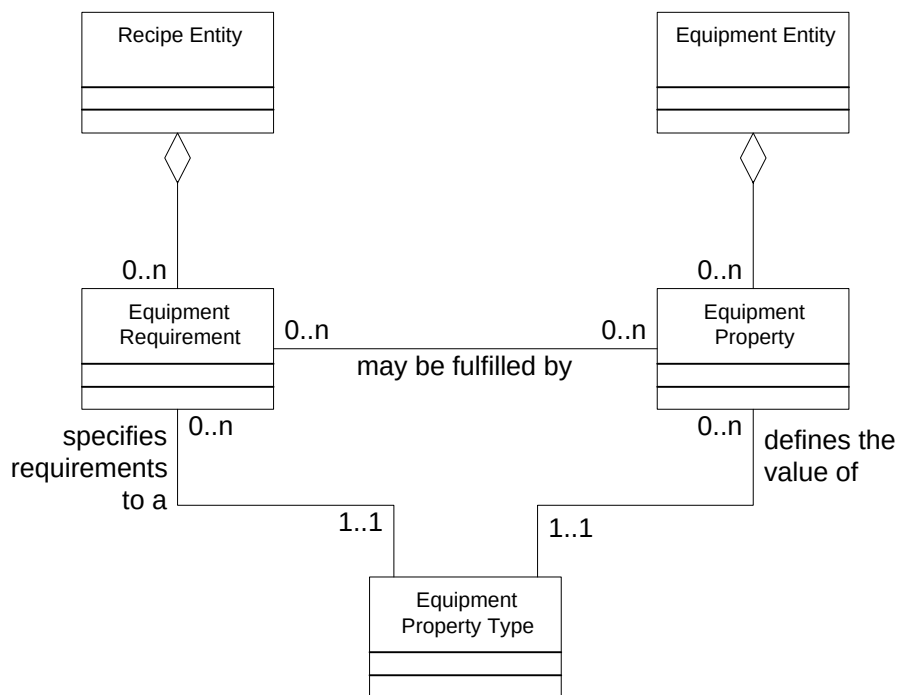


Figure 6 — Recipe entity equipment requirements

Table 15 — Equipment entity

NAME	EQUIPMENT ENTITY
FunctionalDescription	A collection of physical processing and control equipment and equipment control that is grouped together to perform a certain control function or set of control functions.
ATTRIBUTES	
EquipmentEntityID	Provides unique identification.
EquipmentLevel	Specifies the physical hierarchy level (e.g., process cell, unit, equipment module, control module).
Mode	Indicates the current mode of the equipment entity.
State	Indicates the current state of the equipment entity.

Table 16 — Equipment property

NAME	EQUIPMENT PROPERTY
FunctionalDescription	Identifies a property that the equipment entity or class supplies. These properties are application specific (e.g., lining type, size, heat capability, steam temperature).
ATTRIBUTES	
PropertyID	Provides unique identification.
Value	Identifies the value of the property (e.g., glass, 50000, 650).
ValueRange	Defines limits or constraints that are related to Value.
EngineeringUnits	Defines the engineering units of the property.
Description	Describes the type of the equipment property.

Table 17 — Equipment property type

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

Control recipe entities will initially contain the equipment requirements that are copied from the *master recipe entity*, and these need to be fulfilled by the corresponding property of one or more equipment entities in order for specific equipment to be allocated. The initial *equipment requirement* may be replaced by specific equipment allocations. These allocations are also modeled as equipment requirements.

4.3.6 Recipe parameters

Parameters are variables associated with recipe entities. These variables may be used by equipment procedural elements, they may be used by other activities (e.g., scheduling), or they may be referenced by other parts of the recipe (e.g., transition criteria) (see figure 7).

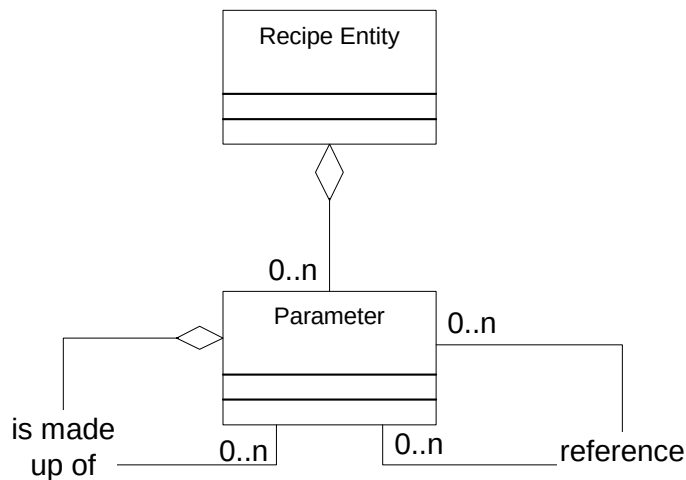


Figure 7 — Parameter model

Parameters may be categorized as process inputs, process outputs, or process parameters.

Parameters may be made up of a collection of other parameters. The model supports the concept of structured parameters. Therefore, the model allows the possibility of including parameters of different types (process parameters, process inputs, process outputs) in the same structure as well as defining single type data structures.

Parameter value attributes may be organized by defining parameter value types. Parameter value types could include

- a) IEC 61131-3 basic data types;
- b) compatibility matrix information that is used for determining clean-in-place (CIP) or sterilize-in-place (SIP) requirements;
- c) data sets that define material transactions (transfer, consumption, generation of material); or
- d) data series (e.g., a temperature profile that will be tracked).

Parameter values may be simple values, expressions, or references to parameters that are defined at the same level or higher levels in the procedural hierarchy. Values that are expressions may include references to other parameters.

Parameters may be related in a number of ways, including the following:

- a) Algebraic or Boolean equations
- b) Product specific entry forms that work on one or more parameters
- c) Standard operating procedures (SOPs) that display or otherwise utilize parameters (e.g., dynamic values, recipe values)
- d) Deferral of parameters to different recipe entities (at the same or another level)

e) External applications that use parameters

The formula is represented in the data model as recipe parameters (see table 10). A recipe's formula is a collection of selected parameters to the recipe procedure, and it may also include parameters that are defined at lower levels of the procedural hierarchy.

Parameters are often scaled, based on batch size or other key attributes. Scaling may be more complex than a simple linear relationship. More complex scaling methods can be accommodated with user-defined algorithms and relationships.

4.4 Equipment model

The physical structure of the plant needs to be taken into account in the evaluation of equipment selection during recipe execution (see figure 8 and table 18). In particular, the transfer capabilities between equipment or the ability to allocate shared equipment are important to routing a batch.

Equipment entities are defined as in the hierarchy that is specified in Part 1. This hierarchy is modeled through the recursive nature of the objects. This construct allows for the configuration of expandability and collapsibility.

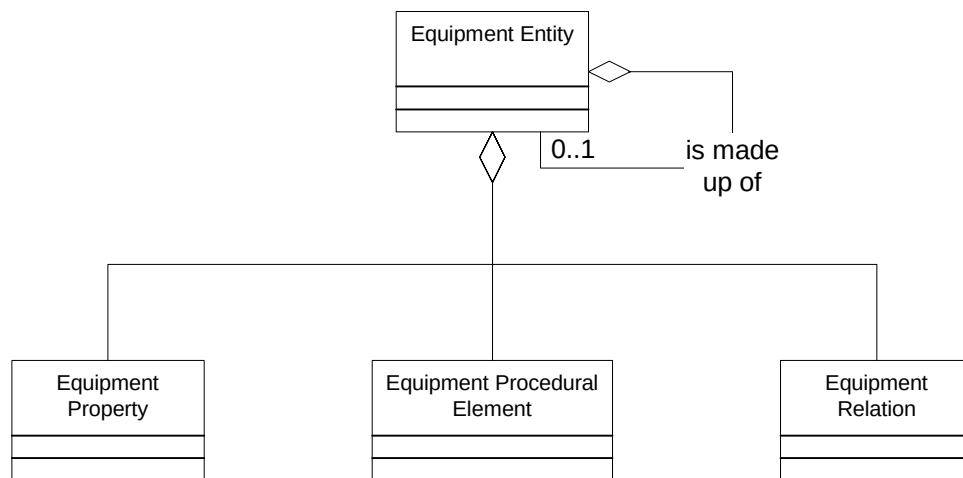


Figure 8 — Equipment structure

Table 18 — Equipment relation

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

The equipment in, for example, a process cell (i.e., units, equipment modules, control modules) is normally related to each other by pipes or other connections. The connections may be modeled as equipment relations (see figure 9), optionally with a direction (e.g., a flow direction). The relations (e.g., pipes) are part

of the higher-level equipment entity. The connections may conveniently be categorized in relation classes and this ensures a consistent evaluation. Equipment relations include

- a) permanent connection;
- b) temporary connection;
- c) may be used as resource for; and
- d) always runs the same product as.

Note that relations other than connections may be possible.

Equipment may have properties. The properties are specific to each user implementation. Equipment properties may be used for examining equipment characteristics and matching recipe equipment requirements. See 4.3.5.

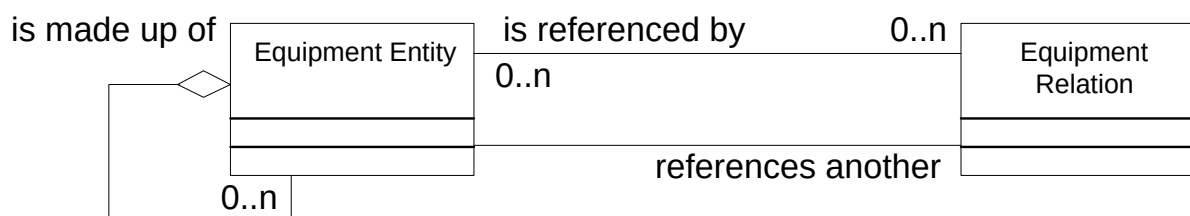


Figure 9 — Equipment entity relations

4.4.1 Equipment classes

Equipment classes (see figure 10 and table 19) provide a means to group equipment entities by common characteristics. Equipment entities may be a member of one or more equipment classes, or they may not belong to a class at all. Equipment classes may be used to specify groups of units and they may be used as alternatives during equipment selection. For example, a recipe may require a reactor for a unit procedure, so its equipment requirements may specify a specific reactor (e.g., R-101), a set of reactor units (e.g., R-101, R-103) or the reactor class (e.g., the class “Reactor” that contains reactors R-101, R-102 and R-103).

Equipment entities may be members of an *equipment class*, and the class determines some of the properties of the class members. As an example, certain equipment properties (e.g., glass lining) are shared with the class.

Equipment entities may belong to zero or more *equipment classes* (e.g., vessel BV1 may be both a reactor and a holding tank).

Equipment classes may determine some or all *equipment properties*, *equipment procedural elements* and *equipment relations* of the referencing equipment entities.

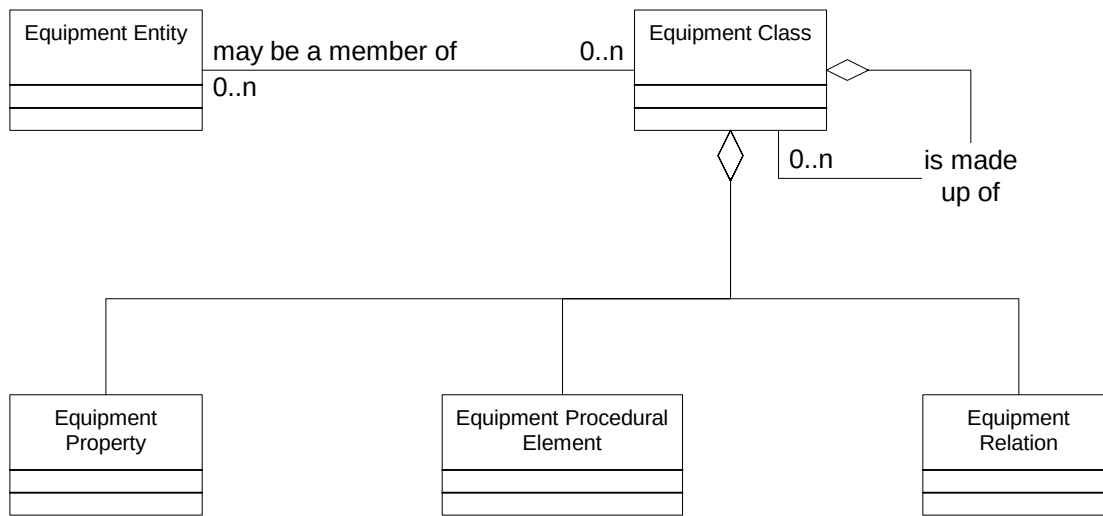


Figure 10 — Equipment classes

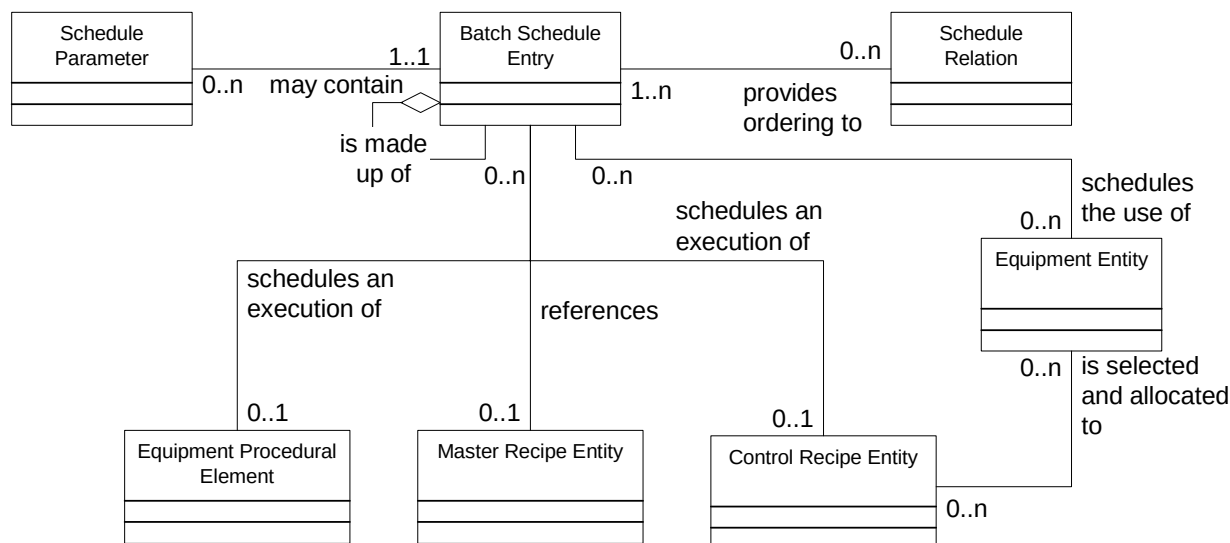
Table 19 — Equipment class

NAME	EQUIPMENT CLASS
FunctionalDescription	An equipment entity class.
ATTRIBUTES	
EquipmentClass	Provides unique identification.
EquipmentLevel	Specifies the physical hierarchy level (e.g., process cell, unit, equipment module, control module).

4.5 Production planning and scheduling

The central entity in a schedule (see figure 11) is the batch schedule entry. This object defines an intended execution of one or more batches/control recipes, or other control recipe entities (typically unit procedures)(see table 20). The batch schedule entry may also be used to schedule other activities (e.g., equipment downtime). A batch schedule entry may include formula/parameter values that are to be used in a control recipe (see table 21).

The batch schedule entry can be used to represent higher-level schedule entities (e.g., a production campaign or a production order).

**Figure 11 — Batch schedule****Table 20 — Batch schedule entry**

NAME	BATCH SCHEDULE ENTRY
FunctionalDescription	A scheduled item that represents a unit procedure in a batch, a complete batch, or a set of batches (e.g., a campaign).
ATTRIBUTES	
ID	Provides unique identification (e.g., actual campaign, lot, batch ID, procedural entity ID).
Level	Specifies the hierarchy level (e.g., campaign, batch, unit procedure).
BatchSize	Defines the requested size or scale factor for the batch, based on the scale factor for the batch as defined in the master recipe.
Schedule	Defines scheduled execution times (start/stop).
ResourceUsage	Specifies resource usage for this schedule entry.
Status	Specifies schedule status (e.g., proposed for evaluation (such as what-if analysis), planned, committed, started, completed).

Table 21 — Schedule parameter

NAME	SCHEDULE PARAMETER
FunctionalDescription	Formula values that are to be communicated to and from batch schedule entries.
ATTRIBUTES	
ParameterID	Provides unique identification.
ParameterType	Includes how the Value is interpreted (e.g., constant, reference, or equation).
Description	Describes the parameter or use of the parameter.
EngineeringUnits	Identifies the engineering units of measure for the Value (e.g., kg, pounds).
Value	Contains the parameter value. If it is a relation, Value contains the equation, form, deferral rule or whatever ties the related parameters together. If Value is a building block parameter, this attribute will hold the default value.
Scaled	Defines the scaling rule. Simplest case: scaled or not scaled with batch reference size.
Usage	Identifies the parameter as a process input, process output, or process parameter.

The schedule relations can be used to represent the scheduling-relevant subset of recipe relations (e.g., the relations related to batch transfers)(see table 22). At higher levels, they can be used to represent required or desirable relations between schedule entries (e.g., batch xx should follow batch yy or a specific cleaning should take place between the two batches). The specific subclasses and characteristics of schedule entry relations are not modeled here.

A higher-level batch schedule entry will include the lower-level schedule entries and relations (e.g., a scheduled campaign may include scheduled batches and relations between these batches).

In the simplest case, the batch schedule entry represents a batch in a batch execution list or queue that is expected to be initiated in sequence. Intended starting times and projected duration/ending times may be added. Further, equipment assignments and the use of other resources may be specified by the batch schedule entry. The scheduling may happen at more detailed levels (e.g., scheduling of individual unit recipes and their assignment to equipment, and potentially scheduling individual operations or phases, their projected duration, and their consumption of resources, including shared or exclusive-use resources that constrain the schedule). Units and equipment modules are allocated or de-allocated as required by the specific control recipe.

The collection of schedule entries may be viewed in the following different ways:

- 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

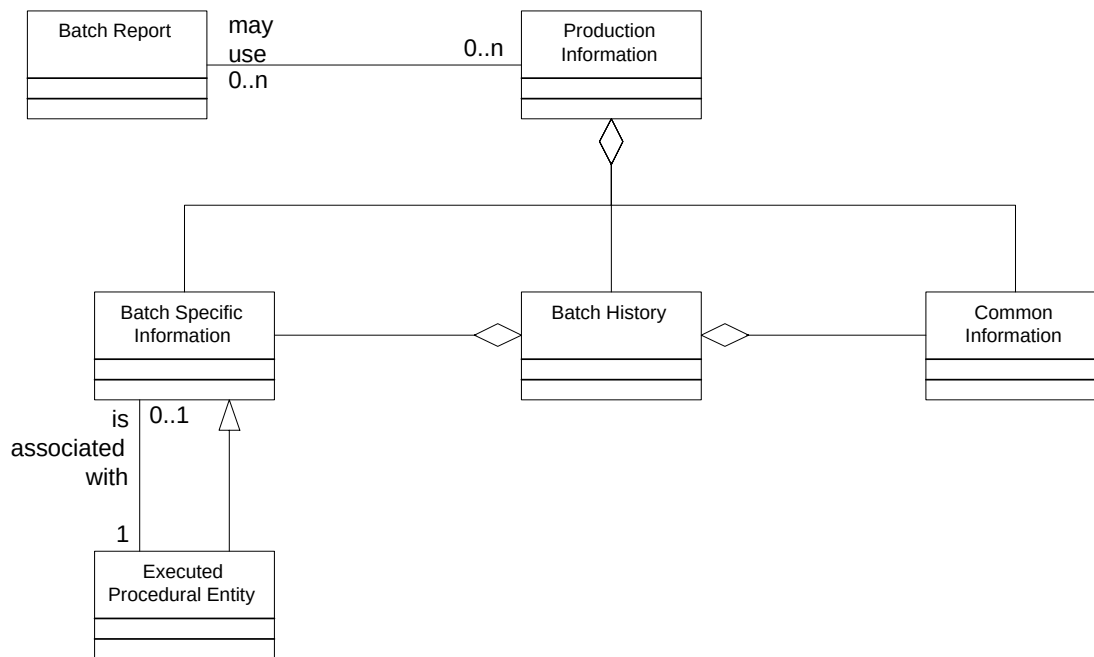
Table 22 — Schedule relation

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 through 28).

**Figure 12 — Production information****Table 23 — Production information**

NAME	PRODUCTION INFORMATION
FunctionalDescription	Information generated during the production of a batch.
ATTRIBUTES	

Table 24 — Batch specific information

NAME	BATCH SPECIFIC INFORMATION
FunctionalDescription	Data that relates to one batch history entry.
ATTRIBUTES	
BatchID	Defines the actual Batch ID.
EntryID	Provides unique identification.
NewValue	Indicates the current actual value.
EngineeringUnits	Defines the engineering units, if any, that are appropriate for the NewValue.
EquipmentID	Identifies an equipment element that may be associated with the entry.
UTC	Identifies the Universal Coordinated Time (UTC) and date of the recorded entry.
UserID	Identifies the user, if any, who is associated with the change.

Table 25 — Batch history

NAME	BATCH HISTORY
FunctionalDescription	An item of information that documents batch production.
ATTRIBUTES	
BatchID	Defines the actual Batch ID.

Table 26 — Common information

NAME	COMMON INFORMATION
FunctionalDescription	Data that relates to more than one batch history entry (e.g., cooling water temperature, atmospheric pressure, steam capacity).
ATTRIBUTES	
EntryID	Provides unique Identification.
NewValue	Indicates the current actual value.
EngineeringUnits	Defines the engineering units, if any, that are appropriate for the NewValue.
EquipmentID	Identifies an equipment element that may be associated with the entry.
UTC	Identifies the Universal Coordinated Time (UTC) and date of the recorded entry.
UserID	Identifies the user, if any, who is associated with the change.

Production information may include some of the following:

- a) A copy of the control recipe
- b) A copy of the master recipe

- c) Information about materials that are used and produced
- d) Trend information
- e) Alarms and messages
- f) Operator interactions with the batch (e.g., overwrites, acknowledgements)
- g) Late records and asynchronous records (e.g., lab sample measurements)
- h) Additional information (e.g., allocation, start/stop)

An executed procedural entity is a recording of an instance of execution of a recipe entity or an equipment procedural entity (see table 25). Data is associated with the execution instance and that data is maintained in the related batch history records.

The executed procedural entities that result from the execution of a control recipe will frequently have the same structure as the control recipe entities. The structure of executed procedural elements may, however, deviate from the control recipe in some cases. The following are some typical examples:

- a) Repeated instances of a recipe entity created through looping in the procedural logic
- b) Not executed instances due to branches or GoTos in the procedural logic
- c) Recipe entities that are inserted or repeated manually
- d) Recipe entities that are activated on the spot in units that are already associated with a batch or that are manually associated with the batch

The recipe entity relation and the rest of the recipe entity design pattern is not repeated here, because the history is about logging the facts as they were and not the intended structure.

Table 27 — Executed procedural entity

NAME	EXECUTED PROCEDURAL ENTITY
FunctionalDescription	A representation of a recipe entity (e.g., equipment procedural element) that has been executed.
ATTRIBUTES	
ExecutedProceduralEntityID	Provides a unique identification.
ProceduralEntityCounter	Uniquely identifies repeated execution of the same procedural entity.

Batch reports (see table 28) are understood to be any extraction of batch data from production information for display on screen or on paper, or for transfer to other systems.

Table 28 — Batch report

NAME	BATCH REPORT
FunctionalDescription	A component of a batch report.
ATTRIBUTES	
ReportID	Provides unique identification.

5 Relational tables for information exchange

ISA hereby grants a non-exclusive, royalty-free, limited license under ISA's copyright in the standard, to copy, display and distribute this section of this ISA standard (including software included in or defined by such section), as follows:

1. Producers of products or services intended to comply with the standard may incorporate this designated section, but only to the extent reasonably necessary to make, use, and distribute any product or service (including product documentation) that is compliant with this standard.
2. End users of a product or service made by a producer acting under the preceding license may reproduce and use the designated section, but only to the extent reasonably necessary to enjoy the intended functions of the product or service and to maintain, configure, or reconfigure systems to be compliant.
3. Persons providing education on or promotion of the standard may copy, display and distribute the designated section, but only to the extent reasonably necessary to provide information related to the standard.

Except as expressly permitted, all other reproduction and distribution without permission of ISA is prohibited. All copies of this section of the standard made or distributed under this license must cite the standard and include the following notice of copyright:

Copyright © 2001 by ISA–The Instrumentation, Systems, and Automation Society. All rights reserved. Used with permission of ISA.

5.1 Introduction

This clause defines the structure of SQL relational tables for the exchange of selected batch control related information between systems. It defines an interface specification, meeting the requirements in clause 4, for the exchange of batch information in the following selected categories:

- a) Master and control recipe information
- b) Process cell equipment information
- c) Schedule information
- d) Production information

Exchange tables shall be built using the table names, field names and relations that are defined in this clause. Not all tables need to be implemented, but all fields in an implemented table shall be included. Any such implementation shall be consistent with the table definitions presented here and the concepts of Part 1.

The format for the exchange tables only defines standard information that can be exchanged. The table definitions can be extended through the inclusion of additional attributes, additional related tables and additional enumerations. The extended structures may be used to exchange information among tools that understand the structures, but this information is outside the scope of this standard.

Examples of additional information are

- a) the addition of icons that represent different elements of the equipment hierarchy, so that the equipment may be represented consistently by different tools; and
- b) the addition of control system addresses for equipment element procedures, procedure parameters and data elements.

5.1.1 Method

The relational table structures are defined using SQL, as specified in ISO/IEC 9075:1992, *Information processing systems – Database language SQL with integrity enhancement*.

The exchange mechanism is based on a common structure for database tables. These tables are defined as a data base schema that can be defined in SQL tables. Annex B contains the SQL table definitions of the exchange tables.

Figure 13 illustrates how the exchange tables would be used to exchange batch information between different tools. Each tool will generally have its own local data stores for batch information and each tool is responsible for importing from, and exporting to, the exchange tables.

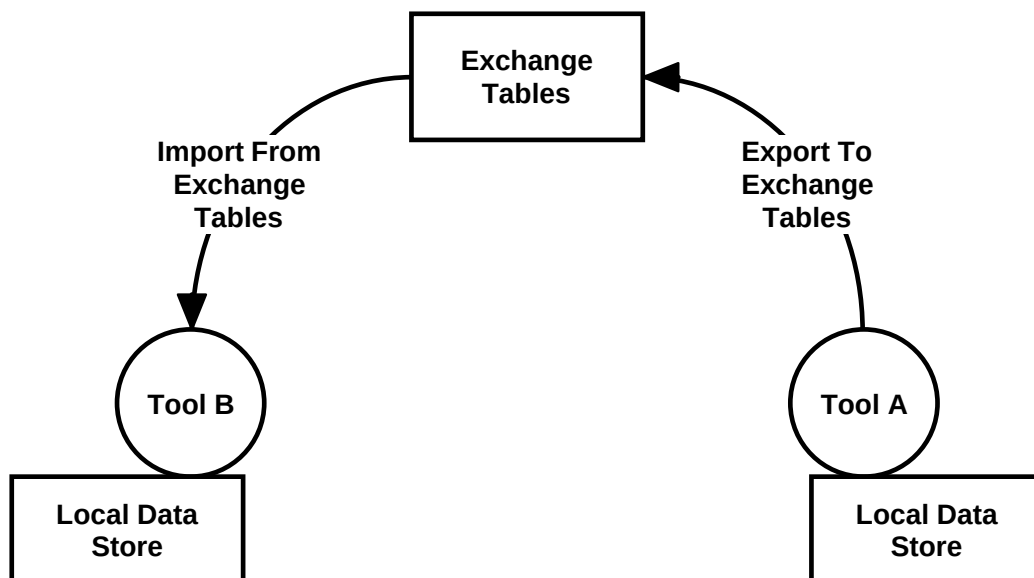


Figure 13 — Data transfer using exchange tables

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