
**Industrial automation systems and
integration — Integration of industrial
data for exchange, access and sharing —**

**Part 2:
Integration and mapping methodology**

*Systèmes d'automatisation industrielle et intégration — Intégration des
données industrielles pour l'échange, l'accès et le partage —*

Partie 2: Méthodologie d'intégration et de «mapping»



Reference number
ISO/TS 18876-2:2003(E)

© ISO 2003

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO 2003

All rights reserved. Unless otherwise specified, 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 either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents	Page
1 Scope	1
2 Normative references	2
3 Terms, definitions, and abbreviations	2
3.1 Terms and definitions	2
3.2 Abbreviations	6
4 Usage scenarios	6
4.1 Integrating application models	6
4.2 Integrating application models with an integration model	7
4.3 Defining an application model and its mapping to an integration model	9
4.4 Integrating an application model with two or more integration models	10
4.5 Improving an integration model	11
5 Methods for integrating application models	12
5.1 Analysis of the requirements	13
5.1.1 Pre-conditions	13
5.1.2 Method description	13
5.1.3 Post-conditions	15
5.2 Defining and extending the integration model	15
5.2.1 Pre-conditions	16
5.2.2 Method description	16
5.2.3 Post-conditions	18
5.3 Identifying a subset of the integration model	19
5.4 Mapping between the application model and the identified integration model subset	19
5.4.1 Pre-conditions	20
5.4.2 Method description	21
5.4.3 Post-conditions	22
Annex A (normative) Information object registration	23
Annex B (informative) Description of the integration process	24
Annex C (informative) Checklist for integration and mapping processes	32
Annex D (informative) Technical discussions	38
Bibliography	41
Index	43

Figures

Figure 1 — Creating an integration model that integrates two application models	7
Figure 2 — Integrating an application model with an existing integration model	8
Figure 3 — Creating an application model and its mapping to an integration model	9
Figure 4 — Integrating an application model with more than one integration model	10
Figure 5 — Improving an integration model	11
Figure 6 — Alternative mappings to an improved integration model	12
Figure 7 — Analysis of application model	14
Figure 8 — Creating a new integration model	17
Figure 9 — Extending the integration model	17
Figure 10 — Identifying a subset of the integration model	19
Figure 11 — Mapping between the application model and the integration model subset	21
Figure B-1 — Integrate application model (A-0)	24

Figure B-2 — Integrate application model (A0)	26
Figure B-3 — Analyze requirements (A1)	27
Figure B-4 — Create/extend integration model (A2)	29
Figure B-5 — Map application model to integration model subset (A4)	30
Figure D-1 — Relationship between a model and its subject	38

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of normative document:

- an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50 % of the members of the parent committee casting a vote;
- an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/PAS or ISO/TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/TS 18876-2 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

This Technical Specification is organized as a series of parts, each published separately. The structure of this Technical Specification is described in ISO/TS 18876-1.

0 Introduction

0.1 Overview of ISO 18876

This Technical Specification establishes an architecture, a methodology, and other specifications for integrating industrial data for exchange, access and sharing. It supports:

- data sharing and data integration;
- specification of mappings between models;
- data transformation.

ISO/TS 18876-1 provides an overview of the architecture and methodology of this Technical Specification.

0.2 Organization of this part of ISO 18876

The organization of this part of ISO 18876 is as follows:

- clause 1 specifies the scope and field of application of this part of ISO 18876;
- clause 2 identifies additional standards that, through references in this part of ISO 18876, constitute provisions of this part of ISO 18876;
- clause 3 defines terms used in this part of ISO 18876;
- clause 4 describes a number of usage scenarios for the application of the methods defined in this part of ISO 18876;
- clause 5 specifies the methods for integrating application models, and is supported by a detailed activity model presented in Annex B.

The methods specified in clause 5 are independent of modelling languages, mapping languages, and particular integration models. Annex C provides a checklist that can be used to ensure that all required stages in the integration and mapping process have been followed.

0.3 Target Audiences

The target audience for this document is modellers, analysts, systems integrators, and systems developers with a need to integrate application models across a range of systems and/or enterprise functions. The target audience for the introduction to this document is technical managers responsible for integration projects with a need to assess the applicability of this standard.

0.4 Conventions

This part of ISO 18876 includes provisions that indicate requirements strictly to be followed in order to conform to the standard. Such provisions are indicated through the use of the words “shall” and “shall not”. This part of ISO 18876 also includes provisions that indicate that among several possibilities one is recommended as particularly suitable. Such provisions are indicated through the use of the words “should” and “should not”. Additional material that illustrates the provisions of this part of ISO 18876 is presented in the form of notes, examples, and in the informative annexes B, C, and D.

Industrial automation systems and integration — Integration of industrial data for exchange, access and sharing —

Part 2: Integration and mapping methodology

1 Scope

This Technical Specification establishes an architecture, a methodology, and other specifications for integrating industrial data for exchange, access and sharing. Together these support the following activities:

- integrating data which may be:
 - from different sources or with different model contexts,
 - described by different models, or
 - defined in different modelling languages;
- sharing data among applications through systems integration architectures;
- resolving conflict between models developed with different objectives;
- translating data between different encodings;
- translating models between different modelling languages.

This part of ISO 18876 specifies methods for the following:

- creating and extending integration models;
- evaluating and selecting an integration model that can integrate two or more application models;
- creating an application model that is a constrained subset of an integration model to support particular application domain requirements for exchange, sharing, or both;
- creating a mapping specification between an application model and an integration model.

The following are within the scope of this part of ISO 18876:

- modelling language independent methods for creating and extending an integration model;
- methods for integrating an application model with an integration model;
- mapping language independent methods for mapping an application model to an integration model;
- criteria for the selecting modelling languages and mapping languages that can be used within the specified methods for integration and mapping.

The following are outside the scope of this part of ISO 18876:

- the structure and content of particular integration models;
- methods for creating and extending particular integration models;
- methods for mapping application models to particular integration models.

NOTE The specific methods that apply to mappings between particular application models and integration models depend on the modelling paradigm(s) applied and on the structure and content of the models.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8824-1:1998, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles*

ISO/TS 18876-1:2003, *Industrial automation systems and integration — Integration of industrial data for exchange, access and sharing — Part 1: Architecture overview and description*

3 Terms, definitions, and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply; those taken or adapted from ISO 10303-1 and ISO/TS 18876-1 are repeated below for convenience.

NOTE 1 Definitions copied verbatim from other standards are followed by a reference to the standard in brackets, such as “[ISO 10303-1]”. In these cases the definition in the referenced document is normative; its repetition here is informative and in the case of any discrepancy the definition in the referenced document has precedence. An explanatory note follows definitions that have been adapted from other standards. In these cases the definition given here is normative for the purposes of this part of ISO 18876.

3.1.1

application model (AM)

model that represents information used for some particular purpose

NOTE 1 Some application models are also integration models (see 3.1.15).

NOTE 2 An application model is not necessarily a data model, but may be a model of some other sort, such as a logic based model.

[ISO/TS 18876-1]

3.1.2

class

category or division of things

NOTE There are a number of ways that class can be defined. This definition is intended to be as broad as possible, and is broader than that used in ISO 15926-2.

EXAMPLE Pump, power station, engineer, and fictional space vehicle are examples of classes.

[ISO/TS 18876-1]

3.1.3

concept

internal conception of some thing; general notion or idea of some thing

[ISO/TS 18876-1]

3.1.4

construct

representation of a concept in some formal notation system

NOTE A construct may be a part or the whole of a data model.

3.1.5

data

representation of information in a formal manner suitable for communication, interpretation, or processing by computers and possibly human beings

[ISO 10303-1]

3.1.6

data model

set of constructs that provides the definition, structure, and format of data, whether physical or abstract in the sense of being bound to some recording medium

[ISO/TS 18876-1]

3.1.7

derived concept

concept in an integration model that is wholly defined in terms of primitive concepts

[ISO/TS 18876-1]

3.1.8

encoding transformation

transformation of the way data elements are represented for computer processing

EXAMPLE Conversion of data governed by an EXPRESS schema from an ISO 10303-21 file to an XML document is an example of an encoding transformation.

[ISO/TS 18876-1]

3.1.9

extension

process or result of adding concepts to an integration model in order to increase the model scope, without changing the concepts already represented

[ISO/TS 18876-1]

3.1.10

foundation concept

primitive concept that determines the underlying world viewpoint of an integration model

NOTE There can be a number of integration models. Each will have its own modelling paradigm which is characterised by the primitive concepts that it contains.

EXAMPLE The concepts of class and individual are foundation concepts for a general integration model.

[ISO/TS 18876-1]

3.1.11

general concept

primitive concept that has very wide applicability, but is a specialization of some foundation concept

NOTE A concept may be considered to be a foundation concept by one community, while it is considered to be a general concept by another..

[ISO/TS 18876-1]

3.1.12

individual

thing that exists in space and time

NOTE This includes things that actually exist, or have existed, and things that possibly exist in the past, present or future.

EXAMPLE The pump with serial number ABC123, Battersea Power Station, Sir Joseph Whitworth, and the Starship "Enterprise" are examples of individuals.

[ISO/TS 18876-1]

3.1.13

information

facts, concepts, or instructions

[ISO 10303-1]

3.1.14

integration

activity that creates, modifies, or extends an integration model

[ISO/TS 18876-1]

3.1.15

integration model (IM)

application model that can represent the information that is represented by two or more application models

NOTE Being an integration model is about the role one model plays with respect to one or more application models.

[ISO/TS 18876-1]

3.1.16 mapping

correspondence between instances of one model and instances of another model that represent the same meaning

NOTE 1 A mapping can be uni-directional or bi-directional.

NOTE 2 A mapping is the result of applying a mapping specification to particular models.

[ISO/TS 18876-1]

3.1.17 mapping specification

specification of the transformations necessary to take information governed by one model and represent it by information governed by another model with the same meaning

NOTE 1 A mapping specification can include data structure transformations, data value transformations, data encoding transformations, and terminology transformations.

NOTE 2 Mapping specifications can be procedural, or declarative, or a combination of these.

[ISO/TS 18876-1]

3.1.18 model

limited information representation of something suitable for some purpose

NOTE A model can be data, or a data model, or take some other form. See Annex D for further discussion of the relationship between models, data, and data models.

[ISO/TS 18876-1]

3.1.19 model context

sum of constraints that limit the possibility of adding to a model without changing any existing declarations

NOTE 1 The model context is therefore the class of all possible extensions to a model.

NOTE 2 This term is more general than application context as defined in ISO 10303-1.

[ISO/TS 18876-1]

3.1.20 model scope

range of information that an application model can describe

[ISO/TS 18876-1]

3.1.21 primitive concept

concept in an integration model that is not wholly defined in terms of other concepts

[ISO/TS 18876-1]

3.1.22 structural transformation

type of mapping specification that is a transformation to the structure of data

NOTE The change in structure could be to the rearranging of attributes, the splitting of attributes across entity data types, or the creation of new attributes.

[ISO/TS 18876-1]

3.1.23

terminology transformation

transformation of the term used to refer to a thing

NOTE This could be between synonyms in one language, or between different languages.

[ISO/TS 18876-1]

3.1.24

transformation

change of form

[ISO/TS 18876-1]

3.1.25

view

model that is a constrained subset of another model

[ISO/TS 18876-1]

3.2 Abbreviations

For the purposes of this document, the following abbreviations apply.

AM application model

NOTE In ISO 10303 the abbreviation AM is used for Application Module. An Application Module is not the same as an Application Model.

IM integration model

4 Usage scenarios

The methodology defined in this part of ISO 18876 is designed to meet requirements typified by the following usage scenarios:

- integrating two or more existing application models (see 4.1);
- integrating one or more existing application models with an existing integration model (see 4.2);
- defining an application model and its mapping to an integration model (see 4.3);
- integrating an application model with more than one integration model (see 4.4);
- improving an integration model (see 4.5).

4.1 Integrating application models

This component of the methodology integrates two or more application models by creating an integration model that is capable of representing all of the concepts and constraints of the application models. Such an integration model supports the communication of data between applications and users that operate using the application models.

This requirement and its solution are illustrated in Figure 1 below.

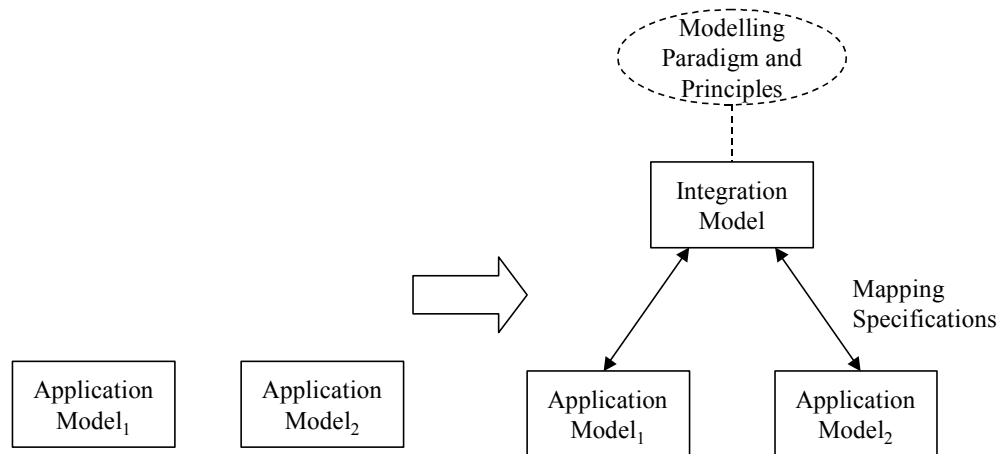


Figure 1 — Creating an integration model that integrates two application models

The inputs to the activity are as follows:

- two or more application models;
- the modelling paradigm and principles chosen for the creation of integration models.

NOTE The use of such principles is an important criterion in determining the future extensibility and reuse of an integration model.

The outputs of this activity are as follows:

- an integration model that represents the concepts and constraints of the input application models;
- mapping specifications that represent the relationships between constructs of each application model and their corresponding subsets of the integration model.

The mapping specifications include any constraints that apply for the application models. The subset of the integration model that corresponds to each application model includes concepts that are not explicit in the application model and have been discovered during the integration process.

4.2 Integrating application models with an integration model

This component of the methodology integrates one or more application models and an existing integration model. The purpose of this activity may be either or both of the following:

- integrating the application models;
- improving the quality of the application models by representing their concepts and constraints in a more consistent and extensible form and structure;
- To extend the model scope of the integration model.

The nature and results of this activity depend on the nature of the integration model and the other application models that are to be integrated. This subclause describes the scenario in which the model context of the application models to be integrated is a subset of the model context of the integration model.

NOTE 1 See 4.5 for a description of the scenario in which the model context of the application models to be integrated is not a subset of the model context of the integration model. The relationship between scope and context of different models is explained in ISO/TS 18876-1:2003, 5.1.2.

This requirement and its solution are illustrated in Figure 2 below.

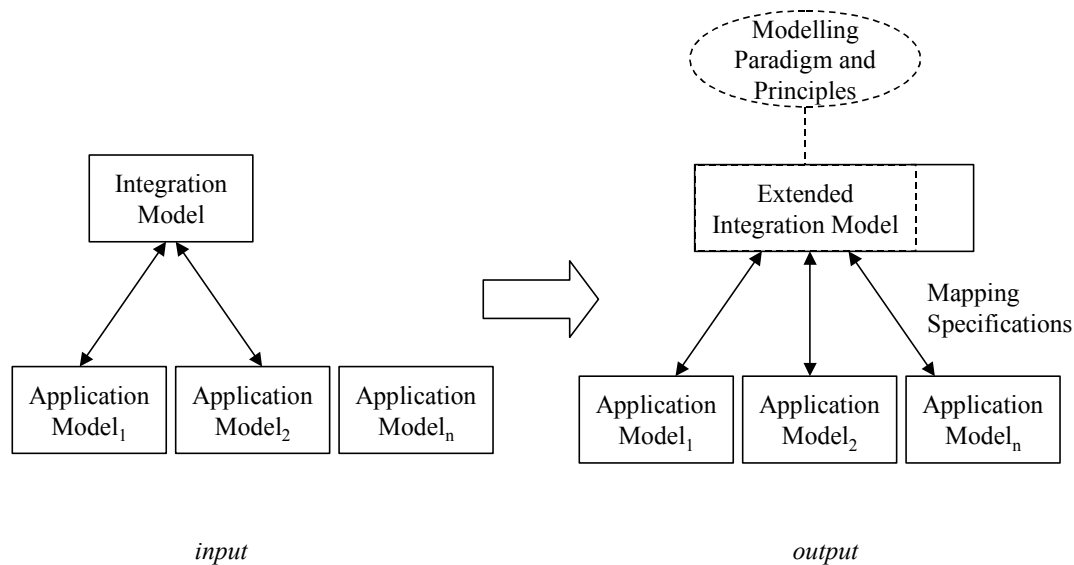


Figure 2 — Integrating an application model with an existing integration model

The inputs to the activity are as follows:

- an existing integration model;
- one or more application models that are integrated with the integration model, including the mapping specifications that relate each application model to its corresponding subset of the integration model;
- the modelling paradigm and principles used to create the integration model – these also determine how the integration model is extended to meet the requirements of the application model.

The outputs of the activity are as follows:

- an extended integration model (if the input integration model does not precisely satisfy the requirements of the application models);

NOTE 2 Extensions to an integration model do not make any change to the objects and relationships in the input integration model. The model scope of the integration model is increased. The context of a model cannot be increased except by removing constraints. This would be a change to the existing integration model, so the change would not be an extension.

- a mapping specification that represents the relationships between constructs of the application models and the corresponding subsets of the extended integration model;
- improvements to the application model (if appropriate).

4.3 Defining an application model and its mapping to an integration model

This component of the methodology creates an application model together with its mapping to an integration model. The purpose of this activity is to enable use of the integration model in order to achieve integration amongst applications or a community of users that have common information requirements.

This requirement and its solution are illustrated in Figure 3 below.

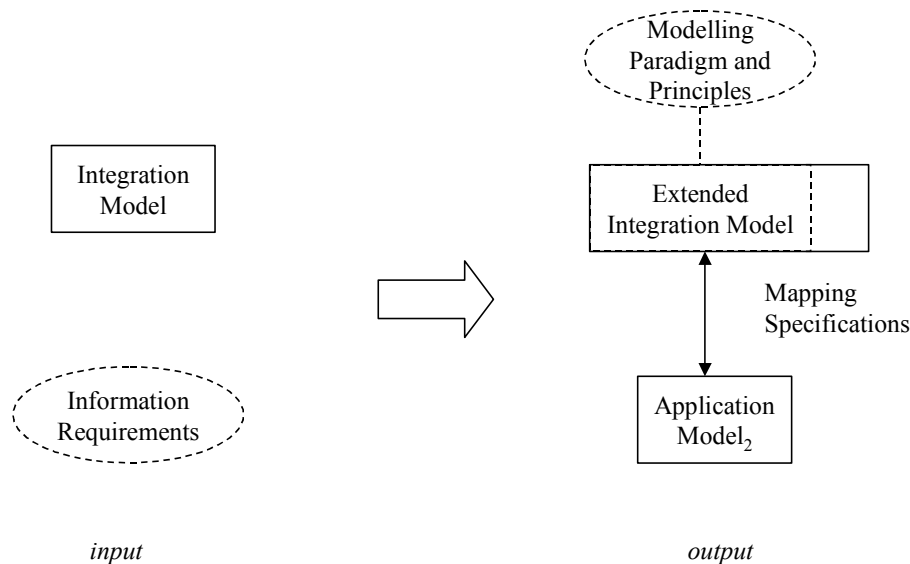


Figure 3 — Creating an application model and its mapping to an integration model

The inputs to the activity are as follows:

- information requirements, described by one or more of the following:
 - existing data;
 - existing models;
 - usage scenarios;
 - paper documents and forms;
 - results of interviews with users;
 - existing applications.
- an existing integration model;
- the modelling paradigm and principles used to create the integration model – these also determine how the integration model is extended to meet the stated information requirements.

The outputs of the activity are as follows:

- an extended integration model (if the input integration model does not precisely satisfy the stated information requirements);

- an application model that satisfies the stated information requirements and is mapped to the integration model;

NOTE The application model may use the structure and/or the terminology of the integration model. If it uses both, then it is identical to a subset of the integration model and the mapping between them is trivial.

- a mapping specification that represents the relationships between constructs of the application model and the corresponding subset of the extended integration model.

4.4 Integrating an application model with two or more integration models

An application model need not be integrated with only one integration model. As shown in Figure 4 below, an application model can be integrated with two or more different integration models.

The inputs to this activity are as follows:

- one or more application models;
- two or more integration models with which the application models are to be integrated;
- the modelling paradigm and principles used to create the integration models.

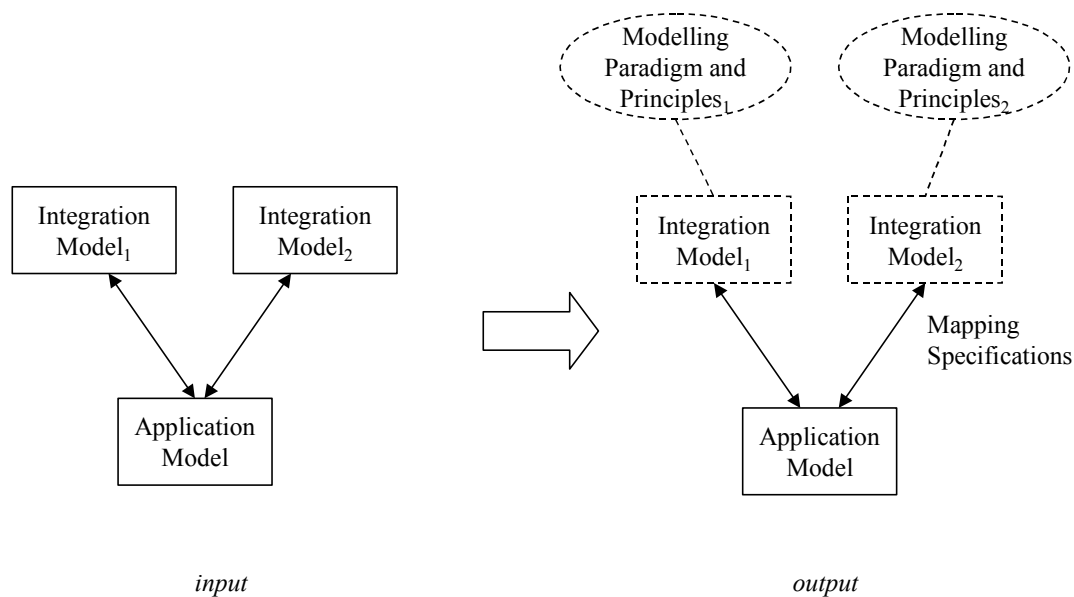


Figure 4 — Integrating an application model with more than one integration model

For an application model to be integrated with two or more integration models, the following conditions shall be satisfied:

- the model context of the application model must be a subset of the model context of each integration model;
- the model scope of the application model must be a subset of the model scope of each integration model, which may be extended to meet this requirement.

Integration with each integration model depends on the modelling principles and paradigm that apply to the development and extension of that integration model.

The outputs of the activity are as follows:

- extended integration models (if one or more of the input integration models do not precisely satisfy the requirements of the application model);
- mapping specifications that represent the relationships between constructs of the application model and the corresponding subsets of the each of the integration models;
- improvements to the application model (if appropriate).

4.5 Improving an integration model

This aspect of the methodology creates a new integration model in the case that an existing integration model does not satisfy the requirements of an application model, and cannot be extended to do so. This situation arises when the model context of the integration model is insufficiently broad to meet the needs of the application models that are to be integrated. This insufficiency may be observed as constraints in the structure of the integration model that do not apply for one or more application models that are to be integrated.

This requirement and its solution are illustrated in Figure 5 below.

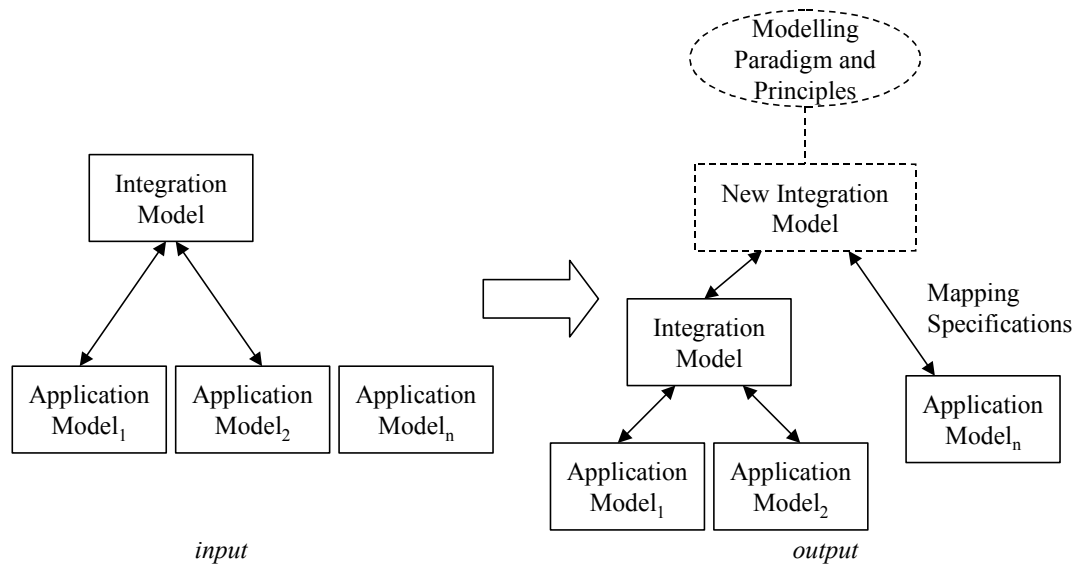


Figure 5 — Improving an integration model

The inputs to this activity are as follows:

- an existing integration model;
- one or more application models that are integrated with the integration model, including the mapping specifications that relate each application model to its corresponding subset of the integration model;
- one or more further application models that are to be integrated but whose requirements cannot be satisfied using the existing integration model.

The outputs of the activity are as follows:

- a new integration model that has a broader model context than that of the original integration model;

NOTE This new integration model may have to be developed on the basis of a more general modelling paradigm than that used for the original integration model.

EXAMPLE If an integration model has a snapshot (current state) view of the world, a change in modelling paradigm will be necessary if it is to support statements about the past and/or the future.

- mapping specifications that represent the relationships between constructs of each application model and their corresponding subsets of the new integration model.

The scenario depicted in Figure 5 results in the creation of a mapping specification from the initial integration model to the new integration model. This implies that in order to migrate data from application model AM_1 to application model AM_n it is necessary to consider three mappings: that from AM_1 to the original integration model, that from the original integration model to the new integration model, and that from AM_n to the new integration model. Technical or economic considerations may result in an alternative approach, as shown in Figure 6 below. Here mapping specifications are defined between the initial integrated application models AM_1 and AM_2 and the new integration model. This reduces the number of mapping specifications to be considered for pairwise combinations of application models, at the expense of additional analysis and mapping effort.

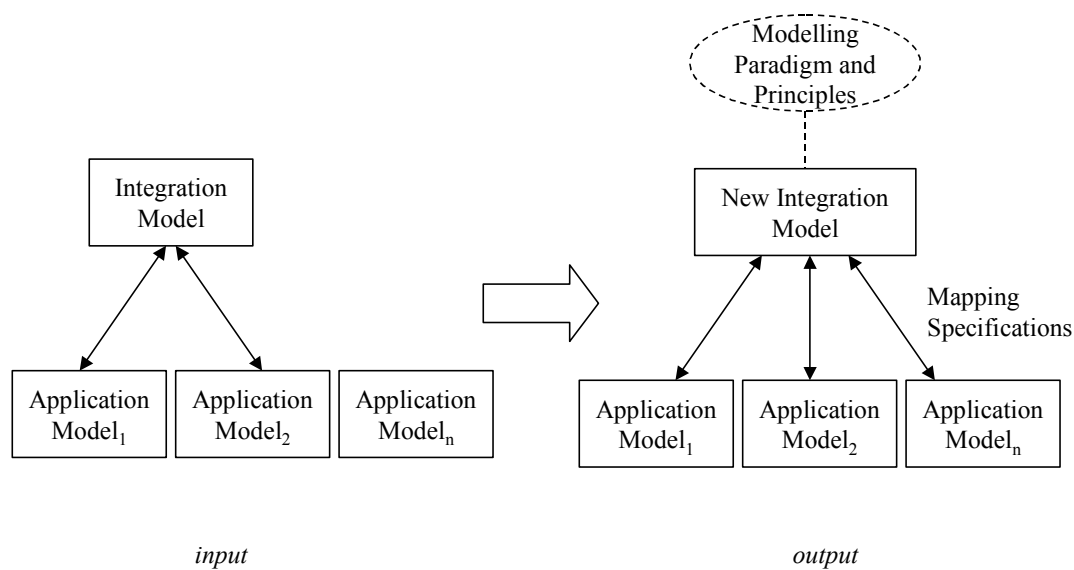


Figure 6 — Alternative mappings to an improved integration model

5 Methods for integrating application models

This clause presents the methods that shall be applied in order to integrate application models. These are divided into four stages:

- analysis of the application model and other information requirements;
- extension of the integration model (if required);
- identification of the subset of the integration model that corresponds to the application model;
- definition of a mapping between the application model and the identified subset of the integration model.

Each stage is characterized by the necessary pre-conditions, a description of the methods to be applied, and the post-conditions that determine the successful application of the methods.

NOTE An activity model for the integration process is shown in Annex B.

5.1 Analysis of the requirements

The purpose of this activity is to identify an integration model that is suitable for integrating the application model(s) of interest, and to determine how the concepts that are represented by the application model(s) are related to the concepts of the chosen integration model.

5.1.1 Pre-conditions

5.1.1.1 Application model

The following pre-conditions apply to analysis of application models:

- definitions, diagrams, and other specifications that describe the application model shall be available;
- example data populations that illustrate usage of the application model shall be available;
- application domain experts shall be available to provide information about the use of the model in different implementations.

NOTE Additional information supplied by application domain experts is often needed to fully understand the model context of the application model and to ensure that this implicit information is explicitly represented in the integration model or in the application model's mapping specification.

5.1.1.2 Modelling languages

No specific pre-conditions apply to the modelling languages used to describe application models. The documentation that describes an application model should state any assumptions, usage practices, or other supporting information that characterizes the way in which a modelling language represents the model.

NOTE 1 Although desirable, this information may not be available for some application models. Analysts should therefore be familiar with different usage practices for modelling languages.

NOTE 2 In the absence of explicit documentation of the usage practices applied, analysts should not assume that a modelling language has been applied consistently within the description of an application model.

5.1.2 Method description

The methods that apply to analysis of requirements are as follows:

- selection of an integration model;
- analysis of application model concepts.

5.1.2.1 Selection of an integration model

Although creation of an integration model that solely integrates two or more integration models is possible (see 4.1) a more common requirement is to integrate one or more application models with an existing integration model (see 4.2). The following criteria apply to the selection of a suitable integration model.

- The integration model chosen should have a model context that includes the model context of the application model(s) being integrated.

NOTE 1 If the model context of an application model is not wholly contained in the model context of an integration model, then the integration model cannot integrate all the concepts represented in the application model(s) without change to the integration model.

NOTE 2 See Annex D.2 for a discussion of the techniques that can be used to determine the model context of an application model.

- If more than one integration model is available with a suitable model context, the integration model chosen should have a model scope that is closest to or overlaps with that of the application model(s) to be integrated. If more than one application model is to be integrated the model scope to be considered is the union of the model scopes of the application models.

NOTE 3 Choice of an integration model whose model scope overlaps with that of the application model(s) to be integrated is likely to simplify the integration process and to reduce the need for extension of the integration model.

NOTE 4 Analysis of application models that have already been integrated with a candidate integration model, and their associated mapping specifications, is good practice for determining overlaps of model scope. Similarly, analysis of the derived concepts that are explicitly represented in an integration model is good practice for determining overlaps of model scope.

5.1.2.2 Analysis of application model concepts

This analysis is illustrated in Figure 7 below.

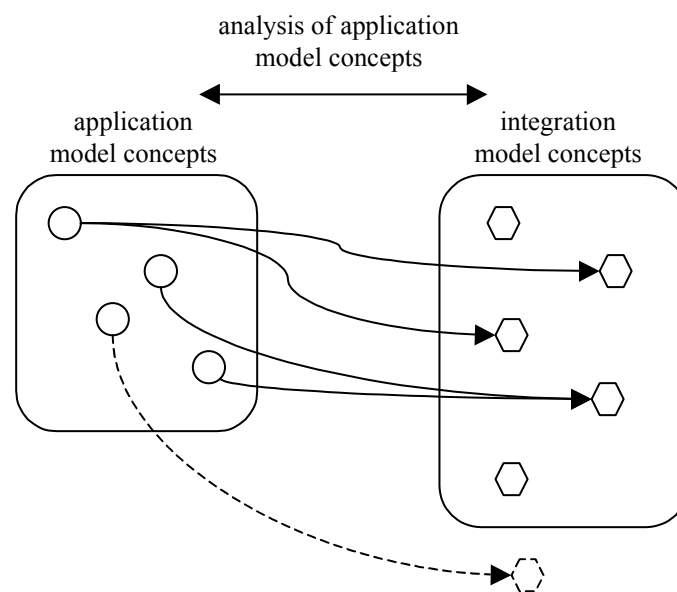


Figure 7 — Analysis of application model

The nature of these relationships varies considerably, depending on the nature of the models involved. Some or all of the following relationship types may be recognized during this analysis:

- synonyms: the same concept exists in the application model(s) and the integration model, but one or more of the models uses a different name for the concept;
- homonyms: concepts exist within one or more of the application model(s) and the integration model that have the same name but have different meanings;
- identical concepts: the same concept exists in the application model and the integration model, has precisely the same meaning, and has the same structure and constraints in both cases;
- compatible concepts: the same concept exists in the application model and the integration model, has the same meaning, but has different structure or constraints that are not contradictory;
- incompatible concepts: the same concept exists in the application model and the integration model, has the same meaning, but has different structure or constraints that are contradictory;

- complex concepts (in the application model): a group of one or more concepts in the application model corresponds to one or more concepts in the integration model;

EXAMPLE 1 An application model includes the concept “red car” and is mapped to an integration model that represents as separate concepts “red (thing)” and “car”. The application model concept is represented by a combination of the two integration model concepts.

EXAMPLE 2 An application model includes an entity data type called **product** which can represent either individual manufactured items (identified by serial numbers) or classes of manufactured items (identified by part numbers). If these are recognized as separate concepts in the integration model as **physical_object** and **class_of_physical_object**, then instances of **product** in the application model can correspond to instances of both concepts in the integration model.

- partitioned concepts (in the application model): two or more concepts in the application model correspond to a single general concept in the integration model.

EXAMPLE 3 An application model contains entity data types called **customer** and **supplier**. These may both correspond to a general entity data type in the integration model called **organization**.

Lastly, it is possible that a concept within the application model has no equivalent within the integration model. In this case it is necessary to develop extensions to the integration model so that a subset of the integration model with precise semantic equivalence to the application model can be identified.

5.1.3 Post-conditions

On completion of the process of analysing the application model requirements the following post-conditions shall be satisfied.

- Either an existing integration model has been selected or it has been determined that no existing integration model can meet the requirements stated in the application models.
- If an existing application model has been selected, an initial evaluation of this model has determined whether it can:
 - be used without modification,
 - be used with suitable extensions, or
 - be used as the basis for a new integration model.
- In the second and third cases above, voids that have been discovered in the integration model have been identified.
- The semantic relationships that have been discovered between the concepts of the application model and the integration model have been documented.

5.2 Defining and extending the integration model

If the analysis activity described in 5.1 discovers that one or more of the concepts represented in the application model have no precise equivalent in the integration model, the latter shall be extended before the subsetting and mapping activities can be completed.

NOTE The diagrams in this subclause depict only one application model and its relationship to an integration model. This is done for reasons of clarity only; it is assumed that the integration model exists and already integrates two or more other application models.

5.2.1 Pre-conditions

5.2.1.1 Integration model

The characteristics of integration models are described in ISO/TS 18876-1:2003, 5.1.2.

The following pre-conditions apply to a selected integration model.

- definitions, diagrams, and other specifications that describe the integration model shall be available;
- examples of application models that have been integrated with the integration model, and their associated mapping specifications, shall be available;
- the analysts undertaking the integration shall be familiar with the integration model and with its use to integrate application models.
- computer tools that represent the integration model and support creation of mapping specifications should be available.

5.2.1.2 Modelling languages

The following requirements on modelling languages apply.

- The modelling language should have capabilities to represent both classes and instances within a single model.

NOTE The EXPRESS language (ISO 10303-11) primarily represents classes (as entity data types). Integration models that are defined using the EXPRESS language therefore need to use a complementary specification to represent instances, such as encoding mechanisms (ISO 10303-21, ISO/TS 10303-28), instance representations (ISO 10303-12), or EXPRESS constants.

- The modelling language should completely support the modelling principles and paradigm used for the integration model.

EXAMPLE If the integration model includes the concept of classes being members of other classes, the modelling language should include a capability to represent this relationship between the constructs that represent classes.

- The modelling language should include capabilities to define mapping specifications, or should have a companion mapping language.

5.2.2 Method description

This component of the methodology can apply to creation of a new integration model or to extension of an existing integration model.

5.2.2.1 Creating a new integration model

This activity is illustrated in Figure 8 below.

The method applied to creating an integration model may depend on the following:

- the modelling paradigm and principles applied in its development and extension;
- the modelling language in which it is defined;
- the model scope and model context of the application models that it is intended to integrate.

An integration model that is defined using an entity-relationship language and paradigm may include entity data types that represent classes that occur in the application models that are being integrated, and may also include

entity instances that represent classes or individuals that occur in the application models that are being integrated.

An integration model that is defined using a first order logic-based language and paradigm shall include constructs that represent classes and individuals that occur in the applications models that are being integrated.

NOTE Entity relationship models and first order logic based languages are not the only possibilities for defining an integration model.

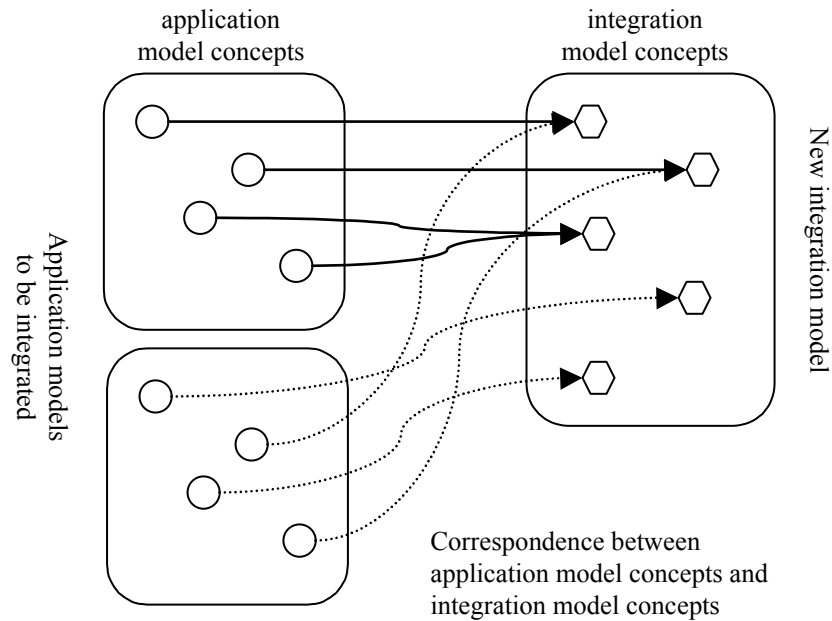


Figure 8 — Creating a new integration model

5.2.2.2 Extending an existing integration model

This activity is illustrated in Figure 9 below.

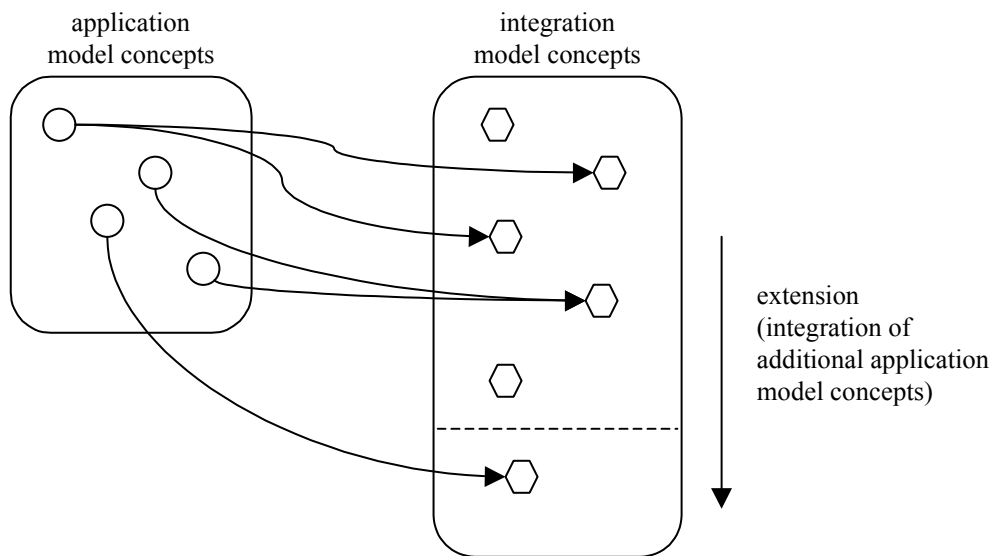


Figure 9 — Extending the integration model

The method applied to extending an integration model may depend on the following:

- the structure and semantics of the existing integration model;
- the modelling paradigm and principles applied in its development and extension;
- the modelling language in which it is defined.

Subject to these dependencies, extension to an integration model may have one or more of the following characteristics. For an entity-relationship model these are:

- creation of additional entity types as specializations of existing entity types in the integration model;
- creation of additional relationships in the integration model;
- creation of additional reference instances within the integration model.

For a logic-based model these are:

- creation of additional constructs within the integration model.

Such extensions shall not modify existing classes in the integration model since to do so can invalidate existing maps between the integration model and other application models.

If it is necessary to develop a new integration model to satisfy the requirements of the application model(s) to be integrated, then a mapping specification between the original integration model and the new integration model shall be developed.

5.2.2.3 Creating a new integration model from an existing integration model

When an existing integration model can not meet the requirements of the application model(s) that are to be integrated, a new integration model can be created by modification of the existing integration model.

NOTE Modifying an integration model is distinguished from extending an integration model.

The methods described in 5.2.2.1 and 5.2.2.2 apply with the following additional constraints:

- the initial integration model shall be regarded as an additional application model to be mapped to the new integration model, or
- each application model that is integrated using the initial integration model shall be integrated with the new integration model.

5.2.3 Post-conditions

On completion of the process of defining or extending an integration model, the following post-conditions shall be satisfied.

- The model scope of the integration model shall completely cover the model scope(s) of the application model(s) to be integrated.
- All voids in the integration model during the requirements analysis phase (see 5.1) shall be filled by extensions to the integration model.
- The semantic relationships that have been discovered between the concepts of the application model(s) and the integration model shall be documented.

NOTE It may not be possible to verify that these post-conditions have been satisfied until the mapping specification(s) for the application model(s) being integrated have been created.

5.3 Identifying a subset of the integration model

The purpose of this activity is to identify the subset of the integration model that precisely corresponds to the application model. In order for mappings between the integration model and multiple application models to be managed, the identity of each application model-specific subset shall be managed within the integration environment. The integration environment shall therefore include capabilities that allow subset identifications to be recorded.

EXAMPLE If the integration model and application models of interest are defined in EXPRESS, the integration model subset can be declared using interface statements in the declaration of an EXPRESS-X view or map that defines the mapping specification to the application model.

This activity is illustrated in Figure 10 below.

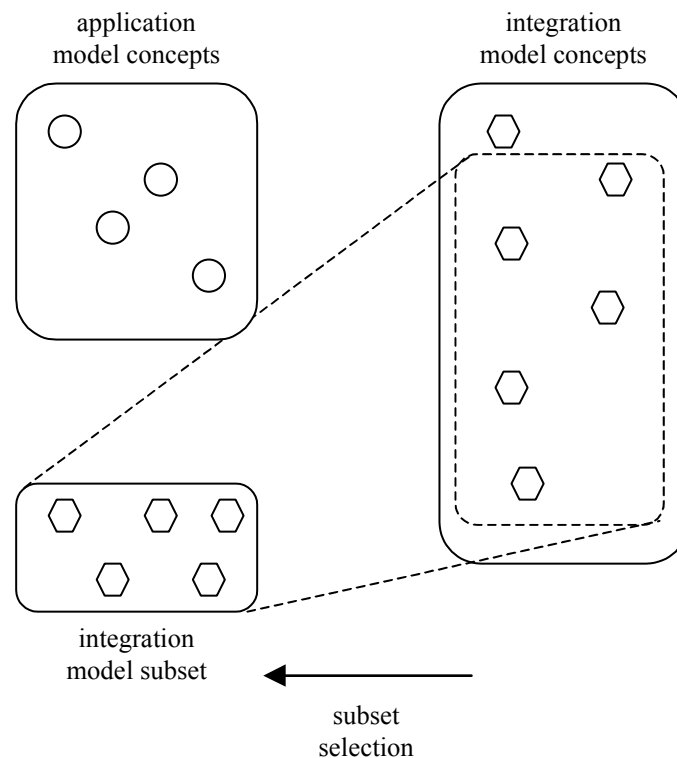


Figure 10 — Identifying a subset of the integration model

In many cases the subsetting and mapping activities are iterative, in that the extent of the integration model subset may not be fully determined until all the constructs of the application model have been mapped.

5.4 Mapping between the application model and the identified integration model subset

The purpose of this activity is to document, in an unambiguous and computer-interpretable form, all the relationships between constructs of an application model and the corresponding subset of an integration model.

Mapping specifications specify the transformations that determine how the instances of one model can be represented as instances of another model. Mapping specifications are used in two ways, as follows.

The mapping specification can describe the mapping transformations between a subset of an integration model and a pre-existing application model that governs data that is separate from that governed by the integration

model. In this case the mapping specification describes the transformations that define a pattern of instances of one model that have the same meaning as a pattern of instances of another model.

The mapping specification can describe the mapping transformations between a subset of an integration model and an application model that is used as an application view. In this case the mapping specification describes how instances in the application view are created from instances in the integration model.

When mapping specifications are created during the integration process. New concepts or constraints shall not be introduced in the mapping specification; mapping specifications are limited to transformations of structure, terminology, and encoding.

NOTE A mapping specification that is defined declaratively represents the mapping in both directions. A mapping specification that is defining procedurally or partly procedurally represents each direction of the mapping separately.

5.4.1 Pre-conditions

5.4.1.1 Mapping languages

Choice of a mapping language depends on the nature of the mapping specifications to be created. A mapping can be described by several specifications each addressing only one part or stage of the mapping; different mapping languages can be used for these different parts of the overall mapping specification.

NOTE 1 Inappropriate choice of mapping language can lead to limitations on the models that can be mapped.

Mapping languages should provide the following capabilities:

- specifying mappings based on data types;

NOTE 2 A mapping specification that relates one data type to another is a shorthand for stating that each instance of one data type is mapped to a corresponding instance of another data type.

- specifying mappings from one data type to another data type, specifying mappings from one data type to a combination of data types, specifying mappings from a combination of data types to one data type, and specifying mappings from one combination of data types to another;

NOTE 3 Mapping specifications that involve a combination of data types often require a query statement or expression to define the extent of instances for which the mapping is valid.

- identifying sets of instances for which a mapping is specified;

NOTE 4 Sets of instances can be specified based on data types, attribute values or value ranges, relationships, reference paths, or any combination of these.

- specifying mappings that associate a data type in one application model with a specified instance or set of instances in another application model;

EXAMPLE 1 An entity data type **pump** in an application model may map to an instance of a more general entity data type **class_of_physical_object** in an integration model, where the name of the instance is constrained to be 'pump'.

- specifying transformations between data values;

EXAMPLE 2 An entity data type **person** with a single-valued attribute **name** may map to an entity data type with two attributes **first_name** and **last_name**. The transformation from the second model to the first is concatenation of attribute values; the transformation in the other direction requires parsing of the source attribute value.

EXAMPLE 3 Different models may use different units of measure for representation of physical quantities such as length or mass.

- specification of mappings that are based on repeated patterns that may be parameterized.

EXAMPLE 4 In EXAMPLE 1 above, the association of a name with a class in the integration model may involve related instances of several different entity data types. By identifying this combination as a template for other mappings, the mapping specification can be simplified and made more consistent.

- definition of variables within the name space of each model that is part of the mapping.

5.4.2 Method description

The activities of analysis (see 5.1) and subsetting (see 5.2.3) establish a data specification (an integration model subset) that is semantically equivalent to the application model. This integration model subset will, however, in the general case differ from the application model in its structure and/or its terminology. The mapping activity therefore establishes and describes the transformations that are applied to data that conforms to the integration model subset such that it conforms to each of the application models (and vice versa). The mapping activity is illustrated in Figure 11 below.

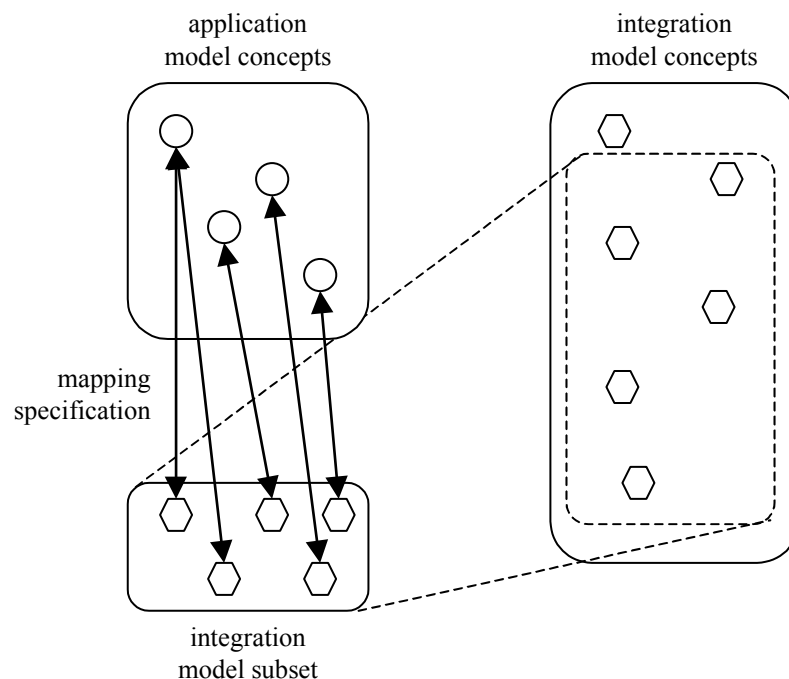


Figure 11 — Mapping between the application model and the integration model subset

The result of this activity is a mapping that characterizes the following:

- the correspondence of each construct in the application model with constructs in the identified subset in the integration model subset;
- the correspondence of each construct in the integration model subset with constructs in the identified subset in the application model.

The following constraints apply to these components of the mapping:

- every construct in the application model shall have a defined correspondence with one or more constructs of the integration model subset;
- every constraint in the application model shall have a defined correspondence with one or constraints in the integration model or in the mapping.

The mapping between the integration model and each application model that is integrated with it shall be recorded as part of the integration environment.

5.4.3 Post-conditions

The overall activity of integrating an application model is complete when the following specifications have been completed and validated:

- the subset of the integration model that precisely corresponds to the semantics of the application model;
- the mapping specification that defines the transformations between the application model and the identified integration model subset.

Annex A (normative)

Information object registration

To provide for unambiguous identification of an information object in an open system, the object identifier

```
{iso standard 18876 part{2} version {1}}
```

is assigned to this part of ISO 18876. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex B (informative)

Description of the integration process

This annex describes the integration process using an activity model. The activity model uses the IDEF0 technique [10] and is presented as a set of figures that contain the activity diagrams and a set of definitions of the activities and the flows between them.

NOTE For ease of presentation the abbreviations AM and IM are used in these IDEF0 diagrams to stand for application model and integration model respectively.

NOTE No precedence should be assumed from the ordering of the arrows in the IDEF0 diagrams.

B.1 Integrate application models (A-0)

The context diagram for this description of the integration process is shown in Figure B-1.

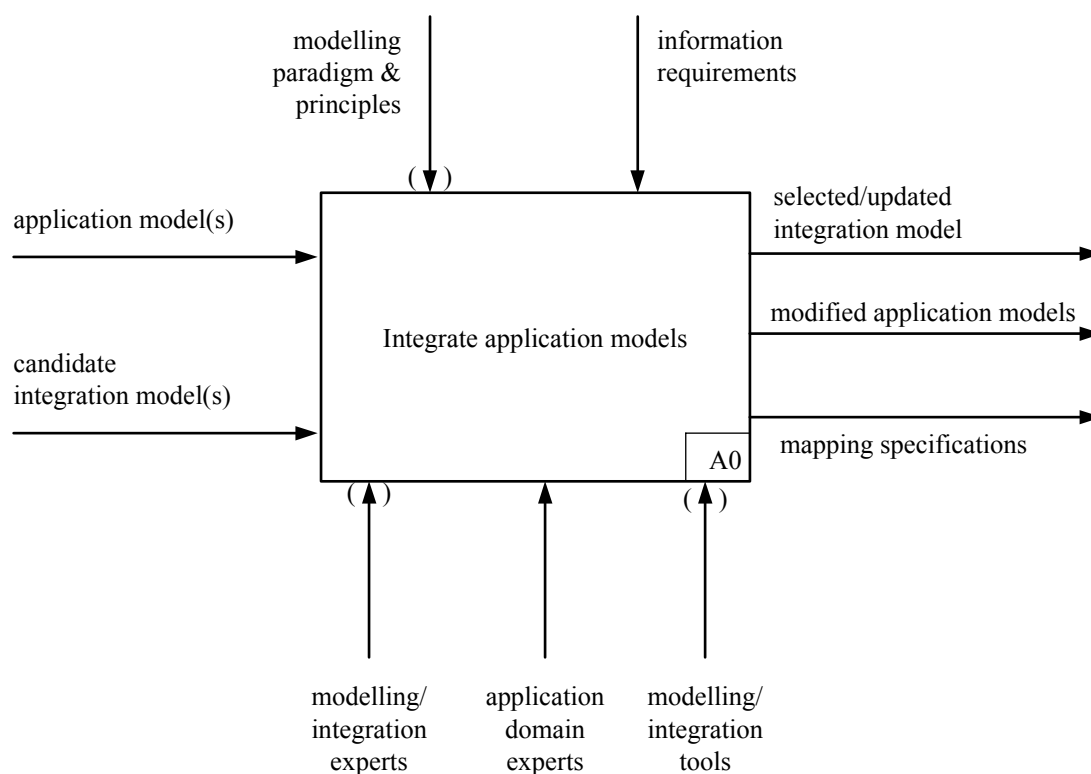


Figure B-1 — Integrate application model (A-0)

The activity and flows shown in Figure B-1 are defined as follows.

B.1.1 Application domain experts

people with knowledge of the processes and information supported by an application model (mechanism)

B.1.2 Application model(s)

see 3.1.1 (input)

NOTE An application model should have a statement of intended scope.

B.1.3 Candidate integration model(s)

integration model(s) whose model context covers that of the application model(s) to be integrated (input)

NOTE An integration model should have a statement of intended scope.

B.1.4 Information requirements

models, documents, sample data, diagrams, and other descriptions of the information used within a specific application domain (control)

B.1.5 Integrate application models

apply the methods specified in this part of ISO 18876 in order to represent the concepts, structure, terminology, and constraints of an application model in terms of an integration model (A0)

B.1.6 Mapping specification

see 3.1.17 (output)

B.1.7 Modelling paradigm and principles

rules, guidelines, and practices that apply to the integration and modelling activities (control)

B.1.8 Modelling/integration experts

people with knowledge and experience of developing and using integration models and creating mapping specifications between application models and integration model (mechanism)

B.1.9 Modelling/integration tools

software and other tools used to create, validate, and publish models, mapping specifications, and other specifications (mechanism)

B.1.10 Modified application model

revised representation of the information used within a specific application domain (output)

NOTE Some scenarios preclude modification of the application model. For example, if the application model is in use by one or more application systems then it may be inappropriate to change it in the integration process.

B.1.11 Selected/updated integration model

integration model that has been selected as suitable to integrate one or more application models; the integration model may be extended as part of the integration process (output)

B.2 Integrate application model (A0)

The decomposition of this activity is shown in Figure B-2.

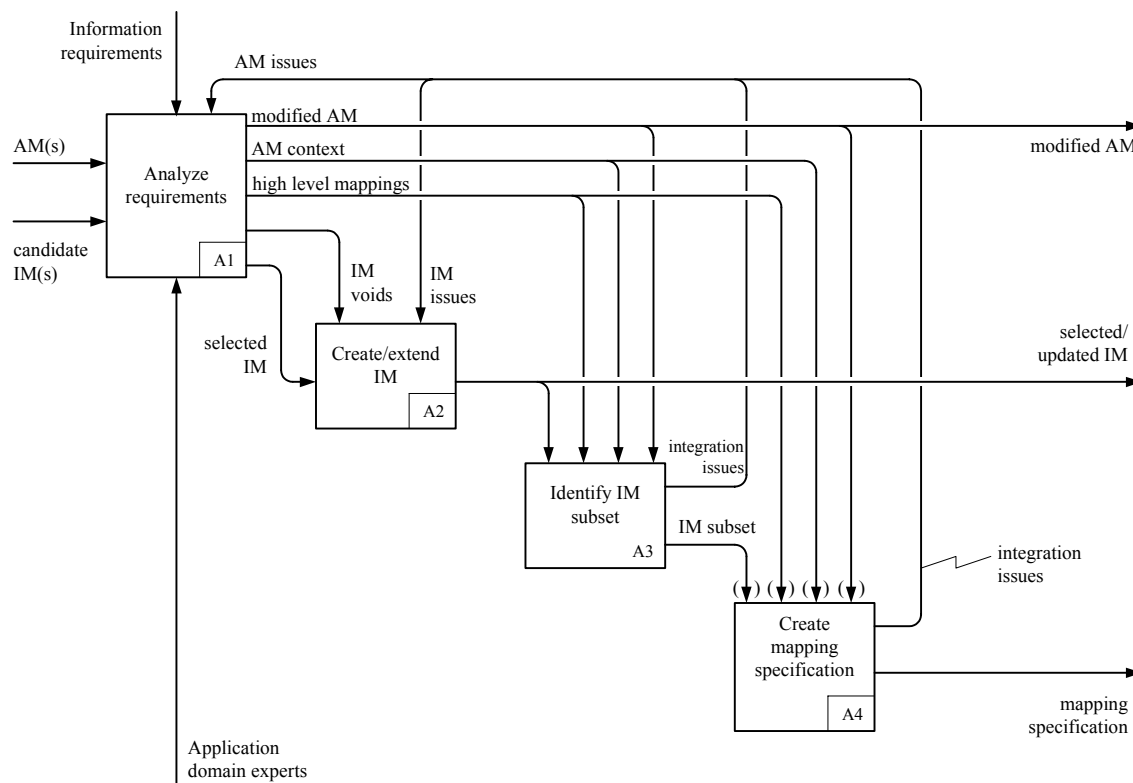


Figure B-2 — Integrate application model (A0)

B.2.1 Analyze requirements

review the application model and/or other sources of information about the requirements of one or more application domains in order to select or create a suitable integration model that covers the required model context and model scope (A1)

B.2.2 Application domain experts

see B.1.1 (mechanism for A1)

B.2.3 Application model context

the model context (see 3.1.19) of an application model (output from A1; control on A3 and A4)

B.2.4 Application model(s)

see 3.1.1 (input to A1)

B.2.5 Candidate integration model(s)

see B.1.3 (input to A1)

B.2.6 Create/extend integration model

develop an integration model that covers the model scope and model context of the application model(s) to be integrated, or extend an existing integration model to meet these requirements (A2)

B.2.7 High level mappings

initial mappings of concepts in the application model(s) and concepts in the integration model (output from A1, control on A2 and A3)

B.2.8 Identify integration model subset

select the constructs of the integration model that satisfy the concepts and constraints of the application model (A3)

B.2.9 Information requirements

see B.1.4 (control on A1)

B.2.10 Integration issues

issues against the integration model resulting from the integration and mapping processes (output from A3 and A4; control on A2)

B.2.11 Integration model voids

concepts that are present in an application model but are missing from an integration model with which the application model is to be integrated (output from A1, control on A2)

B.2.12 Create mapping specification

define how instances of each model are mapped to instances of the other (A4)

B.2.13 Mapping specification

see 3.1.17 (output from A4)

B.2.14 Modified application model

see B.1.10 (output from A1, control on A3 and A4)

B.2.15 Selected integration model

integration model that has been chosen in order to integrate one or more application models, or is capable of extension to do so (output from A1, input to A2)

B.2.16 Selected/updated integration model

see B.1.10 (output from A2, control on A3)

B.3 Analyze requirements (A1)

The decomposition of this activity is shown in Figure B-3.

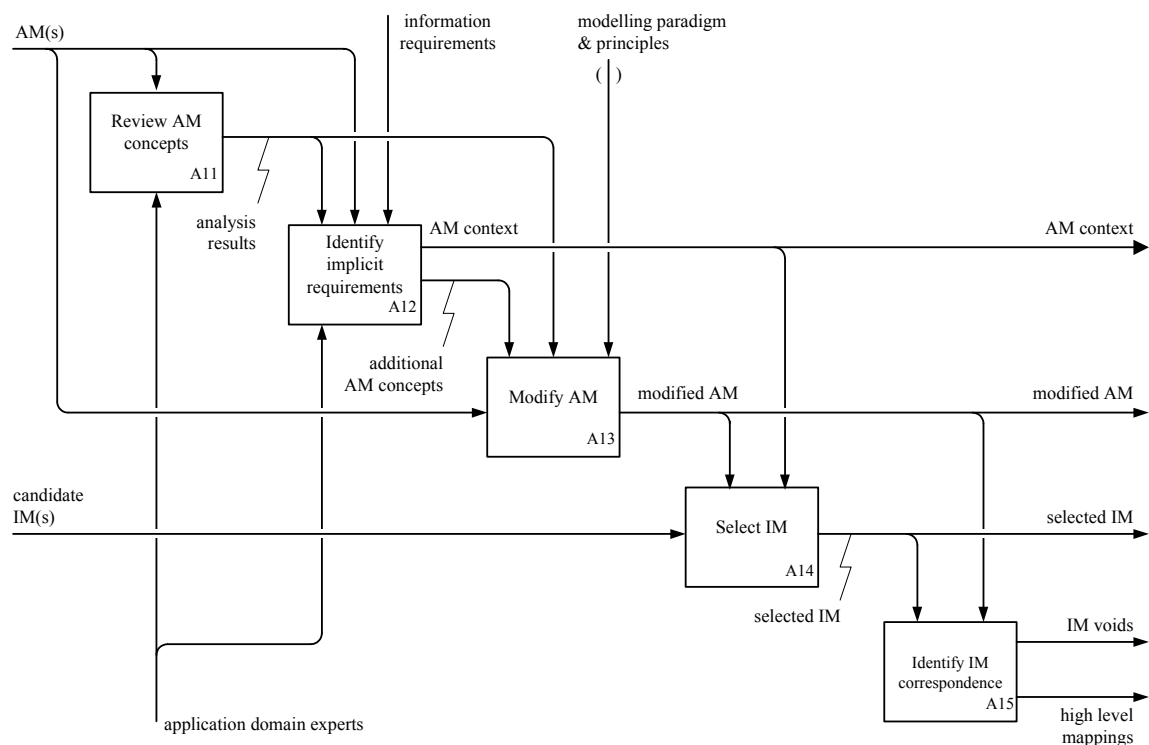


Figure B-3 — Analyze requirements (A1)

B.3.1 Additional application model concepts

concepts within the model scope and model context of the application model that are discovered by analysis of the existing application model and additional information requirements (output from A12, control on A13)

B.3.2 Analysis results

understanding of the concepts represented by the application model based on knowledge of other application models and existing integration models (output from A11, control on A12 and A13)

B.3.3 Application domain experts

see B.1.1 (mechanism for A11 and A12)

B.3.4 Application model context

see B.2.3 (output from A12, control on A14)

B.3.5 Application model(s)

see B.1.2 (input to A11)

B.3.6 Candidate integration model(s)

see B.1.3 (input to A14)

B.3.7 High level mappings

see B.2.7 (output from A15)

B.3.8 Identify implicit requirements

discover and document requirements within the model scope and model context of the application model that are not explicitly represented by the constructs of the application model (A12)

B.3.9 Identify integration model correspondence

establish the high level mappings between the concepts of the application model and those of the selected integration model (A15)

B.3.10 Information requirements

see B.1.4 (control on A12)

B.3.11 Integration model voids

see B.2.11 (output from A15)

B.3.12 Modelling paradigm and principles

see B.1.7 (control on A13)

B.3.13 Modify application model

revise the structure and content of the application model in order to increase its quality and consistency (A13)

B.3.14 Review application model concepts

assess the concepts represented by the application model in order to fully understand the problem domain for the integration process (A11)

B.3.15 Select integration model

identify an integration model whose model context covers that of the application model(s) to be integrated (A14)

NOTE In order to ensure reusability of integration models, this activity should consider a wider model context than just that defined by the application model(s) to be integrated.

B.3.16 Selected integration model

see B.2.14 (output from A14, control on A15)

B.3.17 Modified application model

see B.1.10 (output from A13, control on A14 and A15)

B.4 Create/extend integration model (A2)

The decomposition of this activity is shown in Figure B-4.

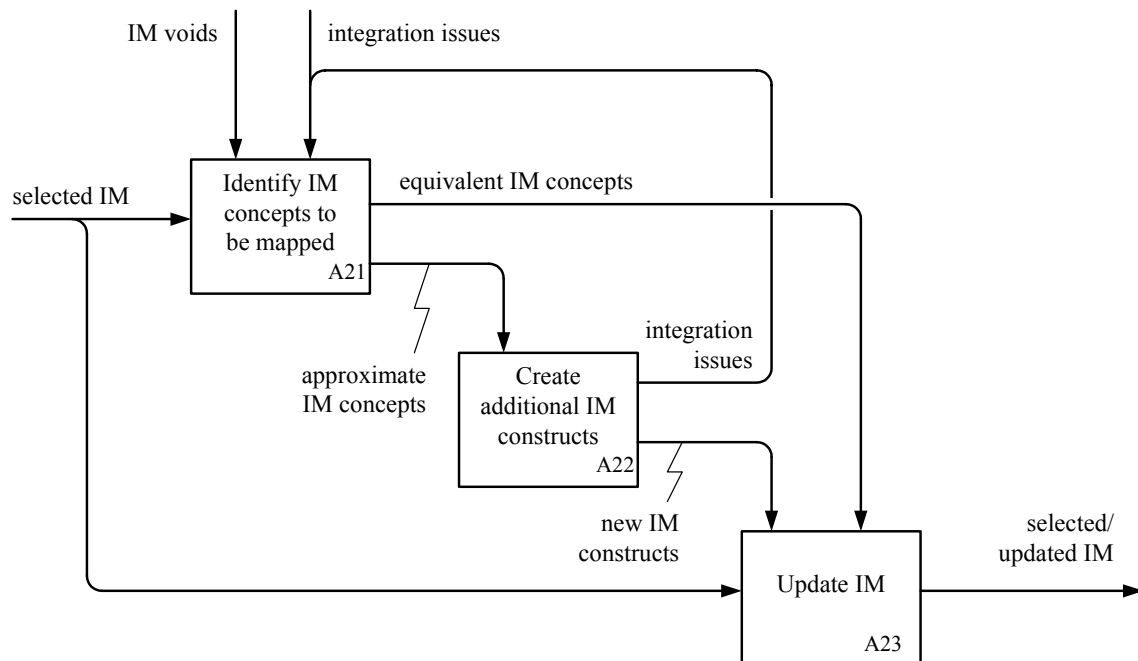


Figure B-4 — Create/extend integration model (A2)

B.4.1 Approximate integration model concepts

concepts that are represented by constructs of the selected integration model whose meaning corresponds to, but is not identical to, the voids identified (output from A21, control on A22)

EXAMPLE An integration model construct that represents a generalization of a concept required by a specific application model is an approximate integration model concept

B.4.2 Create additional integration model constructs

develop new constructs that extend the integration model to fill the identified voids (A22)

B.4.3 Equivalent integration model concepts

existing integration model concepts whose meaning and constraints precisely match those previously identified as voids (output from A21, control on A23)

NOTE Such equivalence may be discovered as a result of resolving differences in model representation or terminology.

B.4.4 Identify integration model concepts to be mapped

discover integration model concepts whose meaning and constraints cover those of the identified voids (A21)

B.4.5 Integration issues

see B.2.10 (control on A21, output from A23)

B.4.6 Integration model voids

see B.2.11 (control on A21)

B.4.7 New integration model constructs

extensions to the integration model (output from A23, control on A24)

B.4.8 Selected integration model

see B.2.14 (input to A21 and A24)

B.4.9 Selected/updated integration model

see B.1.10 (output from A24)

B.4.10 Update integration model

revise the definitions, diagrams, and other specifications that describe the integration model following appropriate documentation and quality methods

B.5 Map application model to integration model subset (A4)

The decomposition of this activity is shown in Figure B-5.

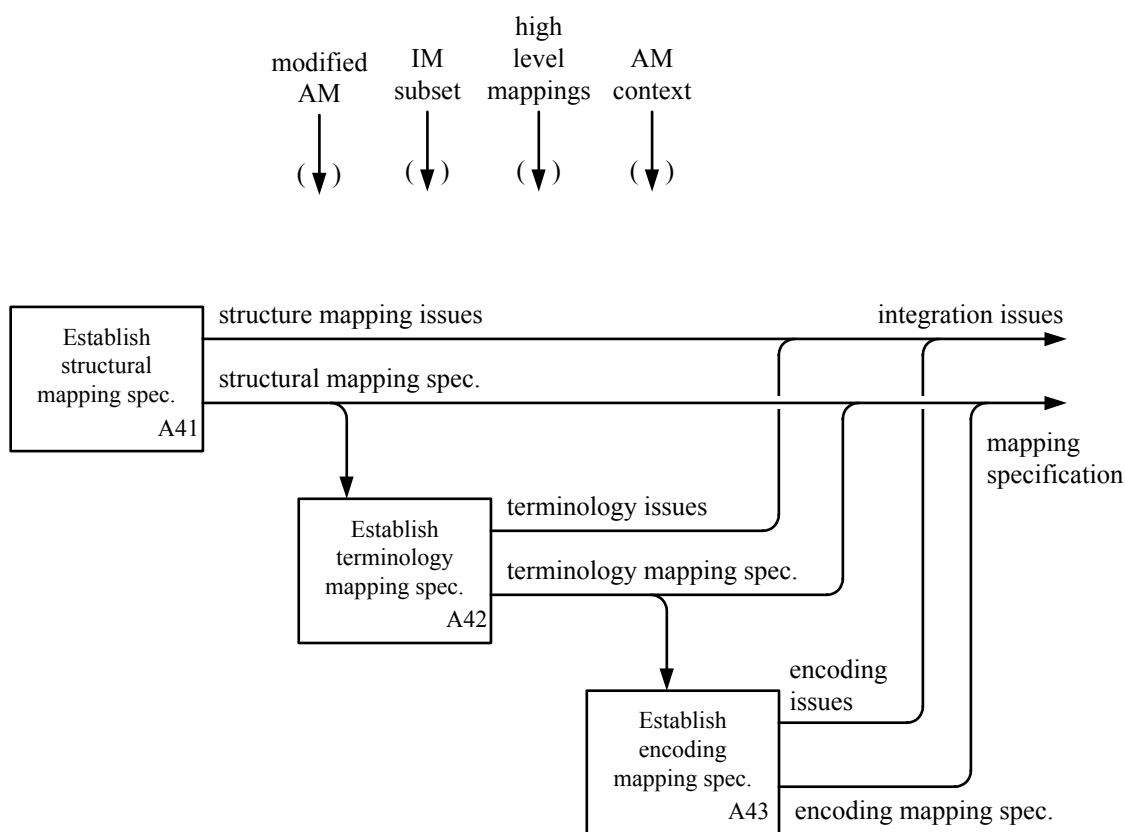


Figure B-5 — Map application model to integration model subset (A4)

B.5.1 Encoding issues

integration issues arising from the assessment of encoding transformations (output from A43)

B.5.2 Encoding mapping specification

mappings specification between the encodings of the application model and those of the integration model subset (output from A43)

B.5.3 Establish encoding mapping specification

identify mapping specification between the encodings of the application model and those of the integration model subset (A43)

B.5.4 Establish structural mapping specification

identify mapping specifications between the structure of the application model and that of the integration model subset (A41)

B.5.5 Establish terminology mapping specification

identify mapping specifications between the terminology of the application model and that of the integration model subset (A42)

B.5.6 Integration issues

see B.2.10 (output from A41, A42, and A43)

B.5.7 Mapping specification

see B.1.6 (output from A41, A42, and A43)

B.5.8 Structural mapping specification

mapping specifications between the structure of the application model and that of the integration model subset (output from A41)

B.5.9 Structure mapping issues

integration issues arising from the assessment of structural transformations (output from A41)

B.5.10 Terminology issues

integration issues arising from the assessment of terminology transformations (output from A42)

B.5.11 Terminology mapping specification

mapping specifications between the terminology of the application model and that of the integration model subset (output from A42)

Annex C (informative)

Checklist for integration and mapping processes

This checklist can be used to ensure that all required stages in the integration and mapping process have been followed.

C.1 Analysis of the requirements

C.1.1 Pre-conditions

Application model (see 5.1.1.1)

- | | Yes | N | N/A | |
|---|--------------------------|--------------------------|--------------------------|---|
| | o | | | |
| 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Definitions, diagrams, and other specifications that describe the application model are available. |
| 2 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Example data populations that illustrate usage of the model are available and have been used |
| 3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | Application domain experts have been consulted to provide information about the use of the application model in different implementations |

Modelling language used for the application model (see 5.1.1.2)

- | | | | | |
|---|--------------------------|--------------------------|--------------------------|--|
| 4 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | The documentation that describes the application model states the assumptions, usage practices, and other supporting information that characterizes the way in which the modelling language used represents the model. |
|---|--------------------------|--------------------------|--------------------------|--|

C.1.2 Method application

Selection of an integration model (see 5.1.2.1)

- | | | | | |
|---|--------------------------|--------------------------|--------------------------|--|
| 5 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | The model context (see 3.1.19) of the selected integration model includes the model context of the application model(s) that are being integrated. |
| 6 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | If more than one possible integration model is available, the selected integration model has a model scope (see 3.1.20) that is closest to or overlaps with the model scope of the application model(s) that are being integrated. |

Analysis of application model concepts (see 5.1.2.2)

- | | | | | |
|---|--------------------------|--------------------------|--------------------------|---|
| 7 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | All relevant implicit concepts been discovered or documented. |
|---|--------------------------|--------------------------|--------------------------|---|

The following questions should be answered for each concept that is represented in the application model.

- 8 There is a synonymous concept in the integration model (same meaning, different name).
- 9 There is an homonymous concept in the integration model (same name, different meaning).
- 10 There is an identical concept in the integration model (same concept with the same structure and constraints)
- 11 There is a compatible concept in the integration model (same concept with the different but non-conflicting structure and constraints)
- 12 There is an incompatible concept in the integration model (same concept with the different and conflicting structure and constraints)
- 13 The application model concept is complex (corresponds to a group of two or more concepts in the integration model)
- 14 The application model concept is partitioned (a group of two or more application model concepts correspond to one concept in the integration model)
- 15 The application model concept has no equivalent in the integration model (which therefore will be extended or changed)

C.1.3 Post-conditions

Output of the analysis activity (see 5.1.3)

- 16 An integration model has been selected
- 17 The selected integration model will require extension
- 18 The selected integration model will form the basis for a new integration model
- 19 There is no available integration model that meets the requirements of the application model(s) to be integrated.
- 20 The semantic relationships between the concepts of the application model and the chosen integration model have been documented.
- 21 Voids discovered in the chosen integration model have been documented.

C.2 Defining and extending an integration model

C.2.1 Pre-conditions

Integration model (see 5.2.1.1)

- 22 The integration model has the characteristics defined in ISO/TS 18876-1:2003, 5.1.2.
- 23 Definitions, diagrams, and other specifications that describe the integration model are available
- 24 Examples of application models that have been integrated with the integration model, and their associated mapping specifications, are available

- 25 The analysts undertaking the integration are familiar with the integration and its use to integrate application models
- 26 Computer tools are available that represent the integration model and support the creation of mapping specifications

Modelling language (see 5.2.1.2)

- 27 The modelling language used to represent the integration model can represent classes and individuals within a single model.
- 28 The modelling language used to represent the integration model can only represent classes but has a complementary instance representation language that can represent individuals as part of a single model
- 29 The modelling language completely supports the modelling principles and paradigm used for the integration model.
- 30 The modelling language has capabilities to define mapping specifications
- 31 The modelling language has a companion mapping language.

C.2.2 Method application

Creating a new integration model (see 5.2.2.1)

- 32 The integration model is consistent with the modelling paradigm and principles applied in its development.
- 33 The integration model satisfies the requirements of the application model(s) that it integrates
- 34 The model scope and model context of the integration model are understood and documented

For an entity-relationship integration model:

- 35 Integrated concepts are specified as classes in the integration model or as reference instances within the integration model

For a logic-based integration model:

- 36 Integrated concepts are specified as constructs within the integration model

Extending an existing integration model (see 5.2.2.2)

- 37 The integration model is consistent with the modelling paradigm and principles applied in its development.
- 38 The integration model satisfies the requirements of the application model(s) that it integrates
- 39 The extensions to the integration model are consistent with its initial content and are accurately related to its initial content
- 40 The initial integration model is a subset of the extended integration model

For an entity-relationship integration model:

- 41 Extensions are specified as specializations of existing classes in the integration model or as additional reference instances within the integration model

For a logic-based integration model:

- 42 Extensions are specified as additional constructs within the integration model

Creating an integration model from an existing integration model (see 5.2.2.3)

- 43 The integration model is consistent with the modelling paradigm and principles applied in its development.

- 44 The integration model satisfies the requirements of the application model(s) that it integrates

- 45 The initial integration model mapped to the new integration model

- 46 Each application model that is mapped to the initial integration model is mapped to the new integration model

For an entity-relationship integration model:

- 47 Integrated concepts are specified as classes in the integration model or as reference instances within the integration model

For a logic-based integration model:

- 48 Integrated concepts are specified as constructs within the integration model

C.2.3 Post-conditions

None.

C.3 Identifying a subset of the integration model

C.3.1 Pre-conditions

None.

C.3.2 Method application

None.

C.3.3 Post-conditions

Results of the subset identification activity (see 5.3).

- 49 The subset of the integration model that precisely corresponds to the application model has been identified

- 50 The identity of the integration model subset has been recorded as part of the integration model

C.4 Mapping between the application model and the identified integration model subset

C.4.1 Pre-conditions

Mapping language (see 5.4.1.1)

- 51 The mapping language chosen can represent mappings based on data types
- 52 The mapping language chosen can represent mappings between different combinations of data types
- 53 The mapping language chosen can identify sets of instances for which a mapping is specified
- 54 The mapping language chosen can represent a mapping between a data type and a set of instances
- 55 The mapping language chosen can represent transformations between data values
- 56 The mapping language chosen can represent parameterized mappings based on repeated patterns
- 57 The mapping language chosen supports variables within the name spaces of the models being mapped

C.4.2 Method application

Results of the mapping activity (see 5.4.2)

- 58 The mapping specification characterizes the correspondence of each construct in the application model with constructs in the integration model
- 59 The mapping specification characterizes the correspondence of each construct in the integration model subset with constructs in the application model
- 60 Each construct in the application model has a defined correspondence with one or more constructs in the integration model
- 61 Each constraint in the application model has a defined correspondence with one or more constraints in the integration model or in the mapping specification
- 62 The mapping between the application model and the integration model is recorded as part of the integration environment

C.4.3 Post-conditions

Outputs of the mapping activity (see 5.4.3)

- 63 The subset of the integration model that precisely corresponds to the semantics of the integration model has been identified and validated

- 64 The mapping that defines the transformations between the application model and the integration model subset has been specified and validated

Annex D (informative)

Technical discussions

D.1 Models, data, and data models

Discussion of models (including data models) needs to differentiate between the contents of the model and what the model represents. In the case of physical models this is clear: there is no problem in distinguishing a 1:1000 plastic model of the Eiffel Tower from the actual Eiffel Tower in Paris. There are clear differences in the characteristics of the model and what it represents.

In the case of data models, however, this distinction may be less clear since the model and its subject share many characteristics. Figure D-1 illustrates the relationships between a data model and its subject.

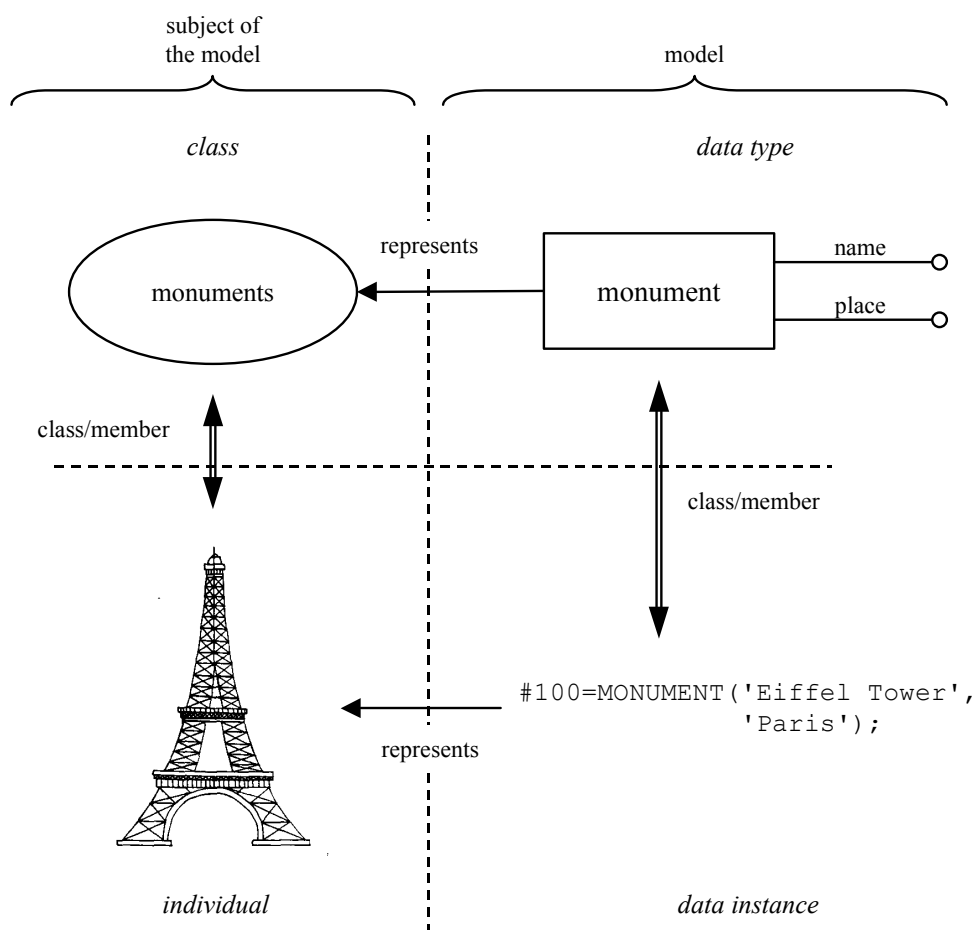


Figure D-1 — Relationship between a model and its subject

Figure D-1 illustrates a very simple data model whose subject is monuments. The objects of interest for this model (i.e. its model scope or universe of discourse) are individual things such as the Eiffel Tower, the Empire

State Building, the Pyramids of Giza, and the Albert Memorial that are members of class called “monuments”¹. The model consists of a declaration of a data type in the EXPRESS language (ISO 10303-11); the identifier of this data type is **monument**. This entity data type represents or stands for the class called “monument”, and allows computer systems to store or interchange information (as data values) about monuments. In this case, the information of interest is represented by the attributes of the data type, i.e., the name of the monument and its location.

As well as representing the class “monument”, the entity data type **monument** is itself a class, whose members are data values or instances that conform to the structure defined by the entity data type declaration. Instances of data types defined in EXPRESS can be encoded according to ISO 10303-21 – the sequence of characters introduced by #100 in Figure D-1 is such an encoded instance. This is both a member of class defined by the data type monument (the data structure conforms to that declared in the entity data type and the data values conform to the data types and constraints declared for its attributes), and also represents an individual (the Eiffel Tower) that is a member of the class “monument” represented by its data type. This second link can in part be inferred from the following statements:

- #100 is a member of the class defined by **monument** (instance of the data type);
- the data type **monument** represents or stands for the class “monument”;
- the Eiffel Tower is a member of the class “monument”.

The last part of the link, that #100 represents the Eiffel Tower, and not some other member of the class “monument”, depends on the ability of an interpreter of the data values (human or computer) to relate the character sequences ‘Eiffel Tower’ and ‘Paris’ to the actual structure in the capital city of France.

NOTE In the situation illustrated by Figure D-1, data values/instances can also represent classes if the data model contains data types that represent classes whose members are themselves classes. For example, an EXPRESS entity data type **class** can be defined to represent the class whose members are (all) other classes. If the data type is declared as follows:

```
ENTITY class;
  name : STRING;
END_ENTITY;
```

then an instance encoded as:

```
#101=CLASS('monument');
```

represents a class called “monument” – possibly the same class as represented by the entity data type **monument** in the model fragment shown in Figure D-1.

In this International Standard, the term “concept” is used to refer to classes and other ideas that are the subjects of (data) models; the term “construct” is used to refer to elements of (data) models that stand for or represent concepts.

D.2 Model contexts of application models

The model context of a model is the sum of constraints that limit the ability to extend the model scope without changing any existing declarations. These constraints include:

- activities that produce or use the data;
- organizations that produce or use the data;

¹ The following typographical convention is used in this Annex: names of classes are enclosed in quotation marks “...”; names of data types are set in **bold font**.

- concepts that the model is “about” but are not explicitly represented in the model;
- constraints inherent in the structure of the model.

The following techniques can be used to discover the model context of an application model.

- The activities that produce or use data can be determined through analysis of one or more suitable activity models. The purpose and viewpoint of each activity model should be taken into account, particularly in the case of data flows that are inputs or outputs of the primary activity described by the activity model.

EXAMPLE 1 An activity model whose primary activity is “process order” has an input named “purchase order”. The viewpoint of the activity model is that of a manager of a parts-supply enterprise. Neither the activity model nor the application data model that is to be integrated states explicitly that the purchase order is addressed to the parts-supply enterprise. This is a constraint on the application model, as it cannot handle purchase orders addressed to any other organization.

- The organizational responsibility for production and control of each element of data in the application model should be determined and taken into account.

EXAMPLE 2 Identifiers, names, and descriptions of items are often unique to an organization but may be repeated across multiple organizations. An application model may state that an attribute such as “part number” is unique; analysis of the application model should determine the organization(s) that define the scope of this uniqueness.

- Other entities that are external to the application model being analysed but which influence the validity or utility of the data should be determined and taken into account.

EXAMPLE 3 Examples of such external entities include physical location, time zone, and registration authority.

Bibliography

- [1] ISO 8879:1986, *Information processing – Text and office systems – Standard Generalized Markup Language*
- [2] ISO/TR 9007:1987, *Information processing systems — Concepts and terminology for the conceptual schema and the information base*
- [3] ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*
- [4] ISO/TR 10303-12:1997, *Industrial automation systems and integration — Product data representation and exchange — Part 12: Description methods: The EXPRESS-I language reference manual*
- [5] ISO 10303-14:—²⁾, *Industrial automation systems and integration — Product data representation and exchange — Part 14: Description methods: The EXPRESS-X language reference manual*
- [6] ISO 10303-21:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*
- [7] ISO/TS 10303-28:2003, *Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data*
- [8] ISO 15926-2:—²⁾, *Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 2: Data model*
- [9] ISO/IEC 19501-1:—²⁾, *Information technology — Unified Modeling Language (UML) — Part 1: Specification*
- [10] IEEE Std 1320.1-1998, *Standard for Functional Modeling Language – Syntax and Semantics for IDEF0*
- [11] *Extensible Markup Language (XML) 1.0 (Second Edition)*. W3C Recommendation 6 October 2000 [cited 2001-05-16]. Available from the Internet: <<http://www.w3.org/TR/REC-xml>>
- [12] BARWISE, Jon; SELIGMAN, Jerry. *Information flow: the logic of distributed systems*. Cambridge: Cambridge University Press, 1997
- [13] DANNER, William F.; YANG, Yuhwei. *STEP (Standard for the Exchange of Product Model Data) Development Methods: Specification of Semantics for Information Sharing*. ISO TC184/SC4/WG5 N30, 1992-01-25
- [14] FOWLER, Julian. *Industry requirements for SC4 standards*. ISO TC184/SC4/WG10 N173, 1998 [cited 2001-05-09]. Available from the Internet: <http://www.nist.gov/sc4/wg_qc/wg10/current/n173/wg10n173.htm>
- [15] FULTON, James. *Semantic plug and play*. Paper presented at an ISO/IEC JTC1 Joint Workshop on Standards for the Use of Models that Define the Data and Processes of Information Systems, Seattle WA, September 1996 [cited 2001-05-09]. Available from the Internet: <<http://www.mel.nist.gov/workshop/jtc1-96/papfultn.htm>>

²⁾To be published.

- [16] GUARINO, Nicola. *Some Organizing Principles for a Unified Top Level Ontology*. Proceedings of AAAI Spring Symposium on Ontological Engineering, Stanford, CA: AAAI Press, 1997
- [17] GUARINO, Nicola. *Some Ontological Principles for Designing Upper Level Lexical Resources*. Padua: 1998 [cited 2001-05-09], available from the Internet:
<<http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/LREC98.pdf>>
- [18] PARTRIDGE, Chris. *Business objects: re-engineering for reuse*. Oxford: Butterworth Heinemann, 1996
- [19] SOWA, John F. *Knowledge representation: logical, philosophical, and computational foundations*. Pacific Grove CA: Brooks/Cole, 2000
- [20] WEST, Matthew; FOWLER, Julian. *Developing High Quality Data Models*. Version 2.0, Issue 2.1. EPISTLE, 1996 [cited 2001-05-09]. Available from the Internet:
<<http://www.matthew-west.org.uk/Documents/princ03.pdf>>

Index

analysis results.....	28
analyze requirements.....	26
application domain expert.....	24
application model.....	2, 13
additional concepts.....	28
analysis.....	14
concepts.....	29
creating as a view on an integration model.....	9
integrate.....	25
integrating.....	7, 24
modified.....	25
modify.....	29
application model context.....	26
class.....	3
concept.....	3
construct.....	3
data.....	3
data model.....	3
derived concept.....	3
encoding transformation.....	3, 4
foundation concept.....	4
general concept.....	4
high level mapping.....	26
implicit requirements.....	28
individual.....	4
information.....	4
information object registration.....	23
information requirements.....	25
Integrate application models	
integrate.....	25
Integrating application models with an integration model.....	7
integration.....	4
integration issues.....	27
integration model.....	4, 16
candidate.....	25
create/extend.....	26
defining and extending.....	15
select.....	29
selected.....	27
selection.....	13
subset.....	19, 26
voids.....	27
mapping.....	5
mapping language.....	21
mapping specification.....	5, 20, 21
creating.....	27
model.....	5
model context.....	5
model scope.....	5
modelling language.....	13, 16
modelling paradigm and principles.....	25
modelling/integration expert.....	25
modelling/integration tool.....	25
primitive concept.....	5
requirements analysis.....	13
structural transformation.....	6
terminology transformation.....	6

transformation.....	6
view	6

.....

ICS 25.040.40; 35.240.50

Price based on 44 pages