INTERNATIONAL STANDARD

ISO 10303-49

First edition
1998-06-15

Industrial automation systems and integration — Product data representation and exchange —

Part 49:
Integrated generic resources: Process structure and properties

Systèmes d'automatisation industrielle et intégration — Représentation et échange de données de produits —

Partie 49: Ressources génériques intégrées: Structure et propriétés du procédé

Reference number
ISO 10303-49:1998(E)

Contents

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet central@iso.ch
X.400 c=ch; a=400net; p=iso; o=isocs; s=central

Printed in Switzerland

Annexes

Figures

Tables

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.

Draft International Standards adopted by 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.

International Standard ISO 10303-49 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data.*

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration - Product data representation and exchange*:

— Part 1, Overview and fundamental principles;

— Part 11, Description methods: The EXPRESS language reference manual;

— Part 12, Description method: The EXPRESS-I language reference manual;

— Part 21, Implementation methods: Clear text encoding of the exchange structure;

— Part 22, Implementation method: Standard data access interface specification;

— Part 23, Implementation method: C++ language binding to the standard data access interface;

— Part 24, Implementation method: C language binding to the standard data access interface;

— Part 26, Implementation method: Interface definition language binding to the standard data access;

— Part 31, Conformance testing methodology and framework: General concepts;

— Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;

— Part 34, Conformance testing methodology and framework: Abstract test methods;

— Part 35, Conformance testing methodology and framework: Abstract test methods

— Part 41, Integrated generic resource: Fundamentals of product description and support;

— Part 42, Integrated generic resources: Geometric and topological representation;

— Part 43, Integrated generic resource: Representation structures;

— Part 44, Integrated generic resource: Product structure configuration;

— Part 45, Integrated generic resource: Materials;

— Part 46, Integrated generic resources: Visual presentation;

— Part 47, Integrated generic resource: Shape variation tolerances;

— Part 49, Integrated generic resource: Process structure and properties;

— Part 101, Integrated application resource: Draughting;

— Part 104, Integrated application resource: Finite element analysis;

— Part 105, Integrated application resource: Kinematics;

— Part 106, Integrated application resource: Building construction core model;

— Part 107, Integrated application resources: Engineering analysis core application reference model (EA C-ARM);

— Part 201, Application protocol: Explicit draughting;

— Part 202, Application protocol: Associative draughting;

— Part 203, Application protocol: Configuration controlled design;

— Part 204, Application protocol: Mechanical design using boundary representation;

— Part 205, Application protocol: Mechanical design using surface representation;

— Part 207, Application protocol: Sheet metal die planning and design;

— Part 208, Application protocol: Life cycle management - Change process;

— Part 209, Application protocol: Composite and metallic structural analysis and related design;

— Part 210, Application protocol: Electronic assembly, interconnet, and packaging design;

— Part 212, Application protocol: Electrotechnical design and installation

— Part 213, Application protocol: Numerical control process plans for machined parts;

— Part 214, Application protocol: Core data for automotive design;

— Part 215, Application protocol: Ship arrangement;

— Part 216, Application protocol: Ship moulded forms;

— Part 217, Application protocol: Ship piping;

— Part 218, Application protocol: Ship structures;

— Part 221, Application protocol: Functional data and their schematic representation for process plant;

— Part 222, Application protocol: Exchange of product data for composite structures;

— Part 223, Application protocol: Exchange of design and manufacturing product information for casting parts;

— Part 224, Application protocol: Mechanical product definition for process plans using machining features;

— Part 225, Application protocol: Building elements using explicit shape representation;

— Part 226, Application protocol: Ship mechanical systems;

— Part 227, Application protocol: Plant spatial configuration;

— Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;

— Part 230, Application protocol: Building structural frame: Steelwork;

— Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;

— Part 232, Application protocol: Technical data packaging core information and exchange;

— Part 301, Abstract test suite: Explicit draughting;

— Part 302, Abstract test suite: Associative draughting;

— Part 303, Abstract test suite: Configuration controlled design;

— Part 304, Abstract test suite: Mechanical design using boundary representation;

— Part 305, Abstract test suite: Mechanical design using surface representation;

— Part 307, Abstract test suite: Sheet metal die planning and design;

— Part 308, Abstract test suite: Life cycle management - Change process;

— Part 309, Abstract test suite: Composite and metallic structural analysis and related design;

— Part 310, Abstract test suite: Electronic assembly, interconnect, and packaging design;

— Part 312, Abstract test suite: Electrotechnical design and installation;

— Part 313, Abstract test suite: Numerical control process plans for machined parts;

— Part 314, Abstract test suite: Core data for automotive mechanical design;

— Part 315, Abstract test suite: Ship arrangement;

— Part 316, Abstract test suite: Ship moulded forms;

— Part 317, Abstract test suite: Ship piping;

— Part 318, Abstract test suite: Ship structures;

— Part 321, Abstract test suite: Functional data and their schematic representation for process plant;

— Part 322, Abstract test suite: Exchange of product data for composite structures;

— Part 323, Abstract test suite: Exchange of design and manufacturing product information for casting parts;

— Part 324, Abstract test suite: Mechanical product definition for process plans using machining features;

— Part 325, Abstract test suite: Building elements using explicit shape representation;

— Part 326, Abstract test suite: Ship mechanical systems;

— Part 327, Abstract test suite: Plant spatial configuration;

— Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;

— Part 330, Abstract test suite: Building structural frame: Steelwork;

— Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;

— Part 332, Abstract test suite: Technical data packaging core information and exchange;

— Part 501, Application interpreted construct: Edge-based wireframe;

— Part 502, Application interpreted construct: Shell-based wireframe;

— Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;

— Part 504, Application interpreted construct: Draughting annotation;

— Part 505, Application interpreted construct: Drawing structure and administration;

— Part 506, Application interpreted construct: Draughting elements;

— Part 507, Application interpreted construct: Geometrically bounded surface;

— Part 508, Application interpreted construct: Non-manifold surface;

— Part 509, Application interpreted construct: Manifold surface;

— Part 510, Application interpreted construct: Geometrically bounded wireframe;

— Part 511, Application interpreted construct: Topologically bounded surface;

— Part 512, Application interpreted construct: Faceted boundary representation;

— Part 513, Application interpreted construct: Elementary boundary representation;

— Part 514, Application interpreted construct: Advanced boundary representation;

— Part 515, Application interpreted construct: Constructive solid geometry;

— Part 517, Application interpreted construct: Mechanical design geometric presentation;

— Part 518, Application interpreted construct: Mechanical design shaded presentation.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of the International Standard reflects its structure:

— Parts 11 to 12 specify the description methods,

— Parts 21 to 26 specify the implementation methods,

— Parts 31 to 35 specify the conformance testing methodology and framework,

— Parts 41 to 49 specify the integrated generic resources,

— Parts 101 to 106 specify the integrated application resources,

— Parts 201 to 232 specify the application protocols,

— Parts 301 to 332 specify the abstract test suites, and

— Parts 501 to 518 specify the application interpreted constructs.

Should further parts be published, they will follow the same numbering pattern.

Annexes A and B form an integral part of this part of ISO 10303. Annexes C, D, and E are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the integrated resources series. Major subdivisions of this part of ISO 10303 are:

— method definition;

— process property;

— process property representation.

This part of ISO 10303 specifies the information necessary to specify the actions or potential actions to realize a process. This includes the relationships between the actions or potential actions in the process and the relationships between the processes that are used to realize a product. A process plan is the specification of instructions to realize a product. This part of ISO 10303 does not specify any particular process, but defines the elements to exchange process information. This part of ISO 10303 is applicable to all types of process definitions that can be represented in a discrete manner.

Industrial automation systems and integration — Product data representation and exchange — Part 49: Integrated generic resource: Process structure and properties

1 Scope

This part of ISO 10303 specifies the resource constructs to specify the actions or potential actions to effect a process. The integrated resource constructs within this part of ISO 10303 define the structure for specifying: relationships between processes, the effectivity of a process, the properties of a process, the resources required for the process, the properties of the resource, the representation of the process, the representation of the resource, and the relationship of the process to the product. When these integrated resource constructs are utilized in the context of an application resource or an application protocol, the integrated resource constructs can be assembled into a structure to represent a process plan.

The following are within the scope of this part of ISO 10303:

— specification of a process;

— specification of the relationships between processes;

— specification of the effectivity of a process;

— specification of the relationship between a process and a product;

— specification of the steps for a process plan to realize a product;

NOTES

1 - This part of ISO 10303 does not make any distinction between a process plan that has been executed and one that has not been executed.

2 - This part of ISO 10303 does not address administrative information within a process plan. Administrative information may include organization, date, or person responsible for developing a process plan.

— specification of alternative process plan definitions;

— specification of the method for realizing a process plan;

— specification of a resource;

— process requirements of the resource;

— identification of resources used by a process;

— specification of the properties of a resource.

The following are outside the scope of this part of ISO 10303:

— definition of specific processes or specific process plans;

NOTE 3 - This part of ISO 10303 does not address any specific industrial process.

— process specifications for continuous processes;

NOTE 4 - This part of ISO 10303 has the capability to represent some aspects of continuous processes. This part of ISO 10303 does not have the capability to represent continuous processes that require interrupts, alarms, and other messaging service requirements.

— scheduling;

— process planning;

NOTE 5 - Process planning is the action of developing a process plan. The action of creating a process plan is outside the scope of this part of ISO 10303. This part of ISO 10303 provides a representation for the output of process planning.

— peer-to-peer communication mechanisms, including synchronization;

— process control.

NOTE 6 - Capabilities for process control and peer-to-peer communication are defined by other ISO standards such as ISO/IEC 9506 (Industrial Automation Systems - Manufacturing Message Specification). These other ISO/IEC standards are concerned with the communication and interworking of programmable devices of industrial process control systems utilized in the process industries. An application resource or an application protocol may specify a relationship between this part of ISO 10303 and ISO/IEC 9506.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8824-1:1995, *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 10303-11:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual.*

ISO 10303-41:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 41: Integrated generic resources: Fundamentals of product description and support.*

ISO 10303-43:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 43: Integrated generic resources: Representation structures.*

ISO 10303-45:1998, *Industrial automation systems and integration - Product data representation and exchange - Part 45: Integrated generic resource: Materials.*

3 Definitions and Abbreviations

3.1 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1:

— application;

— application protocol;

— application resource;

— context;

— component;

— data;

— information;

— interpretation;

— integrated resources;

— product;

— product data;

— product information;

— resource construct;

— structure.

3.2 Terms defined in ISO 10303-45

This part of ISO 10303 makes use of the following terms defined in ISO 10303-45:

— material;

— material property.

3.3 Other definitions

For the purposes of this International Standard, the following definitions apply.

3.3.1 process: a particular procedure for doing something involving one or more steps or operations. The process may produce a product, a property of a product, or an aspect of a product.

3.3.2 process plan: the sequence of processes required to realize or produce a given product.

3.3.3 resource: something that may be described in terms of a behavior, a capability, or a performance measure that is pertinent to the process.

3.4 Abbreviations

URL - Universal Reference Locator

4 Method definition

The following EXPRESS declaration begins the **method_definition_schema** and identifies the necessary external references.

EXPRESS specification:

```
*)
SCHEMA method_definition_schema;

REFERENCE FROM action_schema
    (action_method,
     action_method_relationship,
     action_relationship);

REFERENCE FROM document_schema
    (document,
     document_usage_constraint);

REFERENCE FROM effectivity_schema
    (effectivity);

REFERENCE FROM measure_schema
    (count_measure);

REFERENCE FROM support_resource_schema
    (label,
     text);

REFERENCE FROM process_property_schema
    (product_definition_process,
     property_process);
(*
```

NOTES

1 - The schemas referenced above can be found in the following parts of ISO 10303:

action_schema	ISO 10303-41
document_schema	ISO 10303-41
effectivity_schema	ISO 10303-41
measure_schema	ISO 10303-41
support_resource_schema	ISO 10303-41
process_property_schema	Clause 5 of this part of ISO 10303

2 - See annex D, figure D.1, for a graphical presentation of this schema.

4.1 Introduction

The subject of the **method_definition_schema** is the specification of the instructions required to perform a process. This part of ISO 10303 is applicable to all types of process definitions that can be represented in a discrete manner. This clause provides:

— composition structure of a process, based on a series of actions or potential actions;

— control structure for defining the order of execution of processes;

— method for identifying a document that defines a process;

— method for identification of process effectivity;

— structure for defining conditions that may alter order of completion of the process.

NOTE - See ISO 10303-41 for further information related to **action_method** and **action_method_relationship**.

The **method_definition_schema** represents the data in a process plan, but not the process and data that are required to develop the process plan. The **method_definition_schema** may be used in many contexts for process representation. A context is defined by an application resource or an application protocol.

4.2 Fundamental concepts and assumptions

The **method_definition_schema** provides structures for specifying the decomposition and ordering of actions or potential actions that effect a process. This schema supports the identification of resources that are employed by processes. The domain is not restricted to manufacturing.

An **action_method** defines a potential means of satisfying an **action**. An **action_method_to_select_from**, an **action_method_with_associated_documents**, and an **action_method_associated_documents_constrained**, specify different types of **action_methods** that may be used to satisfy an action.

A **concurrent_action_method**, a **serial_action_method**, and a **sequential_action_method** specify different types of **action_method_relationships** that may be used to create a process. The **action_method**, **action_method_to_select_from**, **action_method_with_associated_documents**, and the **action_method_with_associated_documents_constrained** may be combined by means of the **action_method_relationship** or the different types of **action_method_relationships** defined in this part of ISO 10303 to specify process information.

An **action_method**, or different types of **action_methods** defined in this part of ISO 10303, may be a parent in multiple parent to child relationships that are defined by means of **action_method_relationship** or the different types of **action_method_relationships** defined in this part of ISO 10303. Combination of **action_method_relationship**, or the different types of **action_method_relationships** defined in this part of ISO 10303, through parent to child and peer relationships specify the decomposition and ordering of **action_methods** that effect a process. An application protocol or an application resource specify how the **action_method_relationships** are combined into collections to specify process information.

4.3 method_definition_schema type definitions

4.3.1 relationship_with_condition

A **relationship_with_condition** type is the means for specifying items that define a relationship and have a condition on the relationship or a constraint on the relationship.

EXPRESS specification:

```
*)
TYPE relationship_with_condition = SELECT
  (action_method_relationship,
   action_relationship,
   context_dependent_action_method_relationship,
   context_dependent_action_relationship);
END_TYPE;
(*
```

4.3.2 process_or process_relationship

A **process_or_process_relationship** type is the means for specifying items that identify a process, a relationship between processes, or a relationship between actions or potential actions that effect a process.

EXPRESS specification:

```
*)
TYPE process_or_process_relationship = SELECT
  (product_definition_process,
   property_process,
   relationship_with_condition);
END_TYPE;
(*
```

4.4 method_definition_schema entity definitions

4.4.1 action_method_with_associated_documents

An **action_method_with_associated_documents** is a type of **action_method** that specifies one or more documents that define the **action_method**.

EXAMPLE 1 - A process specification document that an engineer would call out is "12B23" for a heat treat process. "12B23" is the document identification for the heat treat process. The **action_method_with_-associated_documents** has a **documents** attribute with the value of "12B23".

EXPRESS specification:

```
*)
ENTITY action_method_with_associated_documents
  SUBTYPE OF (action_method);
  documents : SET [1:?] of document;
END_ENTITY;
(*
```

Attribute definitions:

documents: the set of one or more documents that identifies the **action_method**.

4.4.2 action_method_with_associated_documents_constrained

An **action_method_with_associated_documents_constrained** is a type of **action_method_with_associated_documents** that identifies the portions of the document that are applicable.

NOTE - The **action_method_with_associated_documents_constrained** may be used to specify either the parts of a document that specify a process or the entire document with applied constraints.

EXPRESS specification:

```
*)
ENTITY action_method_with_associated_documents_constrained
  SUBTYPE OF (action_method_with_associated_documents);
  usage_constraints : SET [1:?] OF document_usage_constraint;
WHERE
  WR1: SIZEOF (QUERY (item <* usage_constraints |
       NOT (item.source
       IN SELF\action_method_with_associated_documents.documents))) = 0;
END_ENTITY;
(*
```

Attribute definitions:

usage_constraints: the specific usage or constraints on the usage of the document for the definition of the **action_method**.

EXAMPLE 2 - An **action_method_with_associated_documents_constrained** references a process specification identified by "12B23". Of the 40 sections in the specification reference, only "section 36" is applicable. The **usage_constraints** is defined by "section 36".

Formal propositions:

WR1: The **usage_constraints** shall be applicable to the documents defined by the **action_method_with_associated_documents**.

4.4.3 action_method_to_select_from

The **action_method_to_select_from** specifies the number of **action_methods** to be selected from all potential **action_methods** in the collection. The collection is defined through **action_method_relationships** where the **action_method_to_select_from** is the parent of the parent to child relationship. The parent is defined as the **action_method_relationship.relating_method**.

NOTES

1 - See annex E for an extended example using collections for **action_method_to_select_from**.

2 - The context for the collection is defined by an application protocol or an application resource.

EXAMPLE 3 - The context defined by the application protocol for the collection of **action_method_to_select_from** is alternatives.

EXPRESS specification:

```
*)
ENTITY action_method_to_select_from
  SUBTYPE OF (action_method);
  number_of_elements : count_measure;
WHERE
  WR1: number_of_elements >=1;
  WR2: SIZEOF (USEDIN (SELF,
       'ACTION_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')) >=
       number_of_elements;
END_ENTITY;
(*
```

Attribute definitions:

number_of_elements: the exact number of **action_methods** to be selected from the collection of constituent **action_methods**.

Formal propositions:

WR1: The **number_of_elements** shall be greater than or equal to one.

WR2: The value of the **number_of_elements** shall not be greater than the number of **action_methods** that are contained within the collection defined by the **action_method_to_select_from**.

4.4.4 process_or_process_relationship_effectivity

Process_or_process_relationship_effectivity is the identification of the valid use of a specific **process_or_process_relationship**.

NOTES

1 - The interpretation of different uses of this entity is not defined in this part of ISO 10303. Application resources and application protocols can provide further distinctions.

2 - The application resource or application protocol defines the context in which the effectivity for the process_or_process_relationship is valid through an **effectivity_assignment** from ISO 10303-41.

EXPRESS specification:

```
*)
ENTITY process_or_process_relationship_effectivity
  SUBTYPE OF (effectivity);
  effective_process_or_process_relationship : process_or_process_relationship;
END_ENTITY;
(*
```

Attribute definitions:

effective_process_or_process_relationship: the identified **process_or_process_relationship** for which the **effectivity** is valid.

4.4.5 serial_action_method

A **serial_action_method** is an **action_method_relationship** that is specified as part of a process. The **serial_action_method** establishes a collection of **action_methods** where individual **action_methods** shall be complete when the collection of **action_methods** is complete. The **action_methods** in the collection are completed in a manner whereby one **action_method** shall be completed before the next **action_method** is initiated.

A **serial_action_method** may be used to define either a peer relationship or a parent to child relationship between two **action_methods**. For a parent to child relationship, the parent is defined as the relating **action_method** that defines the collection. For a peer relationship, the definition of the collection and the distinction between related and relating is defined by the application resource or the application protocol.

NOTES

1 - The interpretation of different uses of this entity is not defined in this part of ISO 10303. Application resources and application protocols make the distinction of use between parent to child or a peer relationship.

2 - The definition of a collection of **serial_action_methods** and order of execution of **serial_action_methods** is defined by application resources and application protocols.

3 - The capability for sequential ordering of parent to child relationships is given by **sequential_method**.

4 - See annex E for examples of parent to child and peer relationships.

EXAMPLE 3 - A peer relationship **serial_action_method** has two **action_methods** that define the process of turning on a light or turning off a light. An **action_method** is turning on the light. An **action_method** is turning off the light. The **serial_action_method** specifies that one **action_method** has to be complete before the other **action_method** can be initiated.

EXPRESS specification:

```
*)
ENTITY serial_action_method
  SUBTYPE OF (action_method_relationship);
END_ENTITY;
(*
```

Informal propositions:

IP1: Individual **action_methods** in a collection shall be completed so that one **action_method** is completed before the next **action_method** is initiated.

4.4.6 sequential_method

A **sequential_method** is a **serial_action_method** where each of the **action_methods** is completed in a specified order. The sequence is such that the **sequential_method** with a lower index is completed before the **sequential_methods** with a higher index.

NOTE 1 - See annex E for an extended example using indexes for **sequential_method**.

EXPRESS specification:

```
*)
ENTITY sequential_method
  SUBTYPE OF (serial_action_method);
  sequence_position : count_measure;
END_ENTITY;
(*
```

Attribute definitions:

sequence_position: the position of the **sequential_method** within the ordered collection of **action_methods**.

Informal propositions:

IP1: There shall be only one **sequential_method** with the same **sequence_position** within a given context or collection.

NOTES

2 - This proposition means that for any instance of **sequential_method** in a given context or collection, the **sequence_position** is unique.

3 - If the **sequential_method** is defined within the context of parent to child relationships, then the context may be the parent **action_method**.

4 - The context may be specified by an application resource or an application protocol.

4.4.7 concurrent_action_method

A **concurrent_action_method** is an **action_method_relationship** that is specified as part of a process. The **concurrent_action_method** establishes a collection of **action_methods** where individual **action_methods** are complete when the collection of **action_methods** is complete. The individual **action_methods** in this collection shall be initiated and completed during completion of the **action_method** with the greatest duration in the collection.

The **concurrent_action_method** may be used to define either a peer relationship or a parent to child relationship between two **action_methods**. For a parent to child relationship, the parent is defined as the relating **action_method** for the collection. For a peer relationship, the distinction between related and relating is not significant.

NOTES

1 - The interpretation of different uses of this entity is not defined in this part of ISO 10303. Application resources and application protocols make the distinction of use between parent to child or a peer relationship.

2 - The duration may be specified by the **action_method** that defines the collection or an **action_method** in the collection. Application resources and application protocols specify the means for determination of the duration.

3 - See annex E for an extended example using parent to child relationships and collections for **concurrent_action_method**.

4 - The requirement is that there exits one **action_method** in the context or collection defined by the **concurrent_action_method** which all **action_methods** shall be initiated and completed. The other **action_methods** in the collection may be sequential, concurrent, or other.

EXPRESS specification:

```
*)
ENTITY concurrent_action_method
  SUBTYPE OF (action_method_relationship);
END_ENTITY;
(*
```

Informal propositions:

IP1: The individual **action_methods** in this collection shall be completed during completion of the **action_method** with the greatest duration in the collection.

4.4.8 context_dependent_action_method_relationship

A **context_dependent_action_method_relationship** is the association between two **action_method_relationships** that specifies a context for the completion of the related **action_methods**. The context is defined by a parent to child relationship between the **action_methods** that are related by the **action_method_relationships** that are given a context by the **context_dependent_action_method_relationship**. The **context_dependent_action_method_relationship** may specify a choice between alternate **action_methods**, depending on some condition.

NOTES

1 - Conditions for the context may be defined by a **relationship_condition** or a **process_or_process_relationship_effectivity.** The evaluation of the condition may determine the **action_method_relationship** to apply. The definition of conditions for the context is defined by an application protocol or an application resource.

2 - A **relationship_condition** on a **action_method_relationship** may be applied through **relationship_with_condition** or a **context_dependent_action_method_relationship.**

The **relating_relationship** in the **context_dependent_action_method_relationship** shall be an **action_method_relationship** that is a parent to child relationship between two **action_methods**. The **related_relationship** in the **context_dependent_action_method_relationship** shall be an **action_method_relationship** that is a parent to child relationship between two **action_methods**. The **relating_action_method** for the **relating_relationship** and the **related_relationship** shall be the parent in both of the **action_method_relationships**.

EXPRESS specification:

```
*)
ENTITY context_dependent_action_method_relationship;
  name                   : label;
  relating_relationship  : action_method_relationship;
  related_relationship   : action_method_relationship;
UNIQUE
  UR1: relating_relationship, related_relationship;
WHERE
  WR1: relating_relationship.relating_method :=:
       related_relationship.relating_method;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **context_dependent_action_method_relationship** is referred to.

relating_relationship: the **action_method_relationship** that is associated with the related **action_method_relationship**. If precedence is important in the association, the **relating_relationship** is the independent **action_method_relationship**.

related_relationship: the **action_method_relationship** that is associated with the relating **action_method_relationship**. If precedence is important in the association, the **related_relationship** is the dependent or alternate **action_method_relationship**.

Formal propositions:

UR1: The combination of the **related_relationship** and the **relating_relationship** shall be unique.

WR1: The **relating_method** for the **action_method_relationship** that is the **relating_relationship** in the **context_dependent_action_method_relationship** shall be instance equal to the **relating_method** for the **action_method_relationship** that is the **related_relationship** in the **context_dependent_action_method_relationship.**

NOTE 3 - WR1 states that the **relating_action_methods** in the **action_method_relationships** in the **context_dependent_action_method_relationship** define the parent of the context for the **context_dependent_action_method_relationship.**

Informal propositions:

IP1: If a **relationship_condition** has a relationship through its **applicable_relationships** to a **context_dependent_action_method_relationship**, through the **relationship_with_condition**, then the **relationship_condition.condition_description** shall specify the context or shall specify a constraint for the **context_dependent_action_method_relationship**. The **relating_action_method** in each **action_method_relationship** is the context for the **action_method**.

NOTE 4 - Context definition may be defined by the constraints that are applied. Definition of a context is through application protocols and application resources utilization of this part of ISO 10303.

IP2: If a **process_or_process_relationship_effectivity** has a relationship to the **context_dependent_action_method_relationship** through the **relationship_with_condition** and the **process_or_process_relationship**, then the effectivity defines a condition.

4.4.9 context_dependent_action_relationship

A **context_dependent_action_relationship** is the association between two **action_relationship**s that specifies a context for the completion of the related **action**s. The context is defined by a parent to child relationship between the **action**s that are related by the **action_relationship**s that are given a context by the **context_dependent_action_relationship**.

The **relating_relationship** in the **context_dependent_action_relationship** shall be an **action_relationship** that is a parent to child relationship between two **action**s. The **related_relationship** in the **context_dependent_action_relationship** shall be an **action_relationship** that is a parent to child relationship between two **action**s. The **relating_action** for the **relating_relationship** and the **related_relationship** shall be the parent in both of the **action_relationship**s.

NOTES

1 - The **context_dependent_action_relationship** may specify a choice between alternate **action**s, depending on applied conditions.

2 - Conditions for the context may be defined by a **relationship_condition** or a **process_or_process_relationship_effectivity**. The evaluation of the condition may determine the **action_relationship** to apply. The determination of conditions for the context is defined by an application protocol or an application resource.

3 - A **relationship_condition** on a **context_dependent_action_relationship** may be applied through **relationship_with_condition** or a **context_dependent_action_relationship**.

4 - See annex E for an extended example using conditions applied to **context_dependent_action_relationship**.

EXPRESS specification:

```
*)
ENTITY context_dependent_action_relationship;
  name                   : label;
  relating_relationship  : action_relationship;
  related_relationship   : action_relationship;
UNIQUE
  UR1: relating_relationship, related_relationship;
WHERE
  WR1: relating_relationship.relating_action :=:
       related_relationship.relating_action;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **context_dependent_action_relationship** is referred to.

relating_relationship: the **action_relationship** that is associated with the related **action_relationship**. If precedence is important in the association, the **relating_relationship** is the independent **action_relationship**.

related_relationship: the **action_relationship** that is associated with the relating **action_relationship.** If precedence is important in the association, the **related_relationship** is the dependent or alternate **action_relationship**.

Formal propositions:

UR1: The combination of the **relating_relationship** and the **related_relationship** shall be unique.

WR1: The **relating_action** for the **action_relationship** that is the **relating_relationship** in the **context_dependent_action_relationship** shall be instance equal to the **relating_action** for the **action_relationship** that is the **related_relationship** in the **context_dependent_action_relationship**.

NOTE 5 - WR1 states that the **relating_actions** in the **action_relationships** in the **context_dependent_action_relationship** define the parent of the context for the **context_dependent_action_relationship.**

Informal propositions:

IP1: If a **relationship_condition** has a relationship through it **applicable_relationships** to a **context_dependent_action_relationship**, then the **relationship_condition.condition_description** shall specify the context or shall specify a constraint for the **context_dependent_action_relationship**. The **relating_action** in each **action_relationship** is the context for the action.

NOTE 6 - Context definition may be defined by constraints that are applied. Definition of a context is through application protocols and application resources utilization of this part of ISO 10303.

IP2: If a **process_or_process_relationship_effectivity** has a relationship to the **context_dependent_action_relationship** through the **relationship_with_condition** and the **process_or_process_relationship**, then the effectivity defines a condition.

4.4.10 relationship_condition

A **relationship_condition** is the specification of a constraint that is to be applied to one or more relationships. The constraint applied to the relationship may be dependent on the context of the relationship. The context may be part of the definition of the constraint.

The relationship that the conditions is applied to is specified through the **action_relationship, action_method_relationship, context_dependent_action_relationship**, or the **context_dependent_action_method_relationship**. The relationship that the condition is to may be the context in which the relationship is used, exists, or is defined. The context for each relationship may be different.

EXPRESS specification:

```
*)
ENTITY relationship_condition;
  name                       : label;
  applicable_relationships : SET [1:?] OF relationship_with_condition;
  condition_description     : text;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **relationship_condition** is referred to.

applicable_relationships: the set of one or more relationships to which the condition applies.

condition_description: the description of the condition to be met.

> NOTE - The determination of the applicable values for **condition_description** shall be determined by an application protocol or an application resource.

```
*)
END_SCHEMA;  -- method_definition_schema
(*
```

5 Process property

The following EXPRESS declaration begins the **process_property_schema** and identifies the necessary external references.

EXPRESS specification:

```
*)
SCHEMA process_property_schema;

REFERENCE FROM action_schema
    (action,
     action_method,
     action_method_relationship,
     action_relationship,
     action_resource,
     action_resource_relationship,
     acyclic_action_relationship);

REFERENCE FROM product_property_definition_schema
    (characterized_product_definition,
     property_definition,
     shape_definition);

REFERENCE FROM support_resource_schema
    (identifier,
     label,
     text);
(*
```

NOTES

1 - The schemas referenced above can be found in the following parts of ISO 10303:

action_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
support_resource_schema	ISO 10303-41

2 - See annex D, figure D.2, for a graphical presentation of this schema.

5.1 Introduction

The subject of the **process_property_schema** is the definition of:

- the properties of the actions that effect the process;
- the properties of the action_methods that effect the process;
- the properties of the resources to be used in the execution of the process;
- the association between the properties of the product and the process.

5.2 Fundamental concepts and assumptions

The primary components of a process are:

- the actions that compose the process;
- the resources used within the process.

The results of a process are:

- the product or the intermediate product that will result from the process;
- the product properties that result from the process.

Process properties are the properties of the actions, resources, and products that effect or are part of the process.

Process properties include the description of the properties of the action that effect the process. Properties of the action are the characteristics that describe or specify operational information of the action that effect the process; the property is a description of the action.

Processes that establish product properties either specify a process for some aspect of the product definition or associate a process with the relevant property of a product.

Process properties may assign a resource to perform a particular action that effects a process. Resources can be assigned to an action at different levels of abstraction. More abstract definitions provide a means of deferring the specification of an actual instance of a resource until process time. Less abstract definitions provide an actual resource instance.

NOTES

1 - Several representations of a property exist such as numeric values, parametric or fundamental equations, graphical values, and non-numeric values. A numeric value of a property makes use of representation.

2 - See ISO 10303-41 for further information related to **action**.

The **process_property_schema** includes definitions which enable the identification of acceptable substitutions for a process.

5.3 process_property_schema type definitions

5.3.1 characterized_action_definition

A **characterized_action_definition** identifies either an **action,** an **action_method,** an **action_method_relationship**, or an **action_relationship**.

EXPRESS specification:

```
*)
TYPE characterized_action_definition = SELECT
   (action,
   action_method,
   action_method_relationship,
   action_relationship);
END_TYPE;
(*
```

5.3.2 characterized_resource_definition

A **characterized_resource_definition** identifies either an **action_resource**, an **action_resource_-requirement,** an **action_resource_requirement_relationship,** or an **action_resource_relationship**.

NOTE - An **action_ resource** is a specification of an resource, while an **action_resource_requirement** identifies requirements that an **action_resource** shall satisfy.

EXPRESS specification:

```
*)
TYPE characterized_resource_definition = SELECT
  (action_resource,
   action_resource_relationship,
   action_resource_requirement,
   action_resource_requirement_relationship);
END_TYPE;
(*
```

EXAMPLES

4 - An **action_resource** is a 'personal computer'. The **requirement_for_action_resource** is '4MB of memory'.

5 - An **action_resource_requirement** is a 'machine capable of lifting 500 kg'.

5.3.3 property_or_shape_select

A **property_or_shape_select** identifies either a **property_definition** or a **shape_definition**.

EXPRESS specification:

```
*)
TYPE property_or_shape_select = SELECT
     (property_definition,
      shape_definition);
END_TYPE;
(*
```

5.4 process_property_schema entity definitions

5.4.1 action_property

An **action_property** is the description of the behavior, capabilities, or performance measures that are pertinent to a process, an action to effect a process, or a potential action to effect a process. An **action_property** is a characteristic of either an **action** or an **action_method**.

EXAMPLES

6 - If the heat treatment of a part is an **action**, an **action_property** is the temperature, the time, or some other characteristic of the **action** that is required to describe the heat treatment.

7 - The movement of an object from point A to point B is an **action**. An **action_property** of the movement is the speed at which the object moves.

EXPRESS specification:

```
*)
ENTITY action_property;
  name         : label;
  description  : text;
  definition   : characterized_action_definition;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **action_property** is referred to.

description: a description of the property.

definition: the **action, action_method**, **action_method_relationship**, or **action_relationship** that is to be characterized.

5.4.2 product_definition_process

A **product_definition_process** is an **action** that defines or is part of the definition of a **characterized_product_definition**. The **product_definition_process** establishes either the product definition of a product or an aspect of the product definition of a product.

NOTES

1 - The aspect of the **product_definition** of a product may be an input or an output to the **product_-definition_process**. The **process_product_association** may specify the role of the **product_definition_-process**.

2 - The product definition may be a single product or may be the process that brings several products together to form another product.

EXAMPLE 8 - The assembly of components is specified by the process that produces the final product.

EXPRESS specification:

```
*)
ENTITY product_definition_process
  SUBTYPE OF (action);
  identification : identifier;
INVERSE
  product_definitions : SET [1:?] OF process_product_association FOR process;
END_ENTITY;
(*
```

Attribute definitions:

identification: a descriptive identifier of the **product_definition_process**.

product_definitions: the set of one or more **process_product_associations** that are associated with each **product_definition_process**.

5.4.3 process_product_association

A **process_product_association** is an association that specifies the product definition characteristic of interest to the **product_definition_process**.

NOTES

1 - The product definition may be a single product or may be the process that brings several products together to form another product.

2 - A **process_product_association** may specify a process for a specific **characterized_product_definition**. Another **process_product_association** may specify the same **characterized_product_definition** for another process. The two **characterized_product_definitions** are the same, but the **process_product_associations** are different.

3 - The product definition characteristic may be an input or an output to the **product_definition_process**.

EXPRESS specification:

```
*)
ENTITY process_product_association;
  name            : label;
  description     : text;
  defined_product : characterized_product_definition;
  process         : product_definition_process;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **process_product_association** is referred to.

description: a description of the process.

defined_product: the product definition or product definition relationship that the **process** defines or is an aspect of.

process: the process associated with the **characterized_product_definition**.

5.4.4 property_process

A **property_process** is an **action** that produces or results in a specific aspect of a product. The process may define a **product_definition**, but need not be specific to a particular product.

NOTE - An aspect of a **product_definition** may be an input or an output to the **property_process**. The **process_property_association** may specify the role of the **property_process**.

EXAMPLE 9 - The **property_process** of "drill hole" is defined. The **property_process** has the **action.chosen_method** of "drill hole". This process is for any product that a drill operation is applied to.

EXPRESS specification:

```
*)
ENTITY property_process
  SUBTYPE OF (action);
  identification : identifier;
INVERSE
  properties : SET [1:?] OF process_property_association FOR process;
END_ENTITY;
(*
```

Attribute definitions:

identification: a descriptive identifier of the **property_process**.

properties: the set of one or more **process_property_associations** that are associated with each **property_process**.

5.4.5 process_property_association

A **process_property_association** associates a specific **property_process** to a given **property_definition** or **shape_definition**.

NOTES

1 - The **process_property_association** allows for the specification of a process that results in individual properties of a product, but the specific product definition of the product that possesses the property need not be identified.

2 - The product definition characteristic may be an input or an output to the **process_property_association**.

EXAMPLES

10 - A **property_process** is a process that produces a geometric form feature that is a slot in a machined part. The action to create the slot is a process that defines shape properties for a product, but it is not specific to a particular product. The product could be made from any of a number of types of materials. A **process_property_association** associates the machining action with the slot feature.

11 - A **property_process** is a process that applies a heat treatment to a part. A **process_property_association** relates the heat treatment process to the material property through the **property_definition** and the **material_property** relationships. The heat treatment process is not specific to a particular part.

EXPRESS specification:

```
*)
ENTITY process_property_association;
  name              : label;
  description       : text;
  process           : property_process;
  property_or_shape : property_or_shape_select;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **process_property_association** is referred to.

description: a description of the relationship between the **property_process** and the **property_definition** or the **shape_definition**.

process: the process to be associated with the property.

property_or_shape: a characteristic that the **process** defines or is an aspect of.

5.4.6 replacement_relationship

A **replacement_relationship** is an **action_relationship** that specifies that a **related_action** may replace the specified **relating_action**. This replacement may occur within or depend upon a context. The context may be defined by a condition.

EXAMPLE 12 - A manufacturing process plan is designated as a functional replacement for the specified manufacturing process plan that is to be used to manufacture a product.

EXPRESS specification:

```
*)
ENTITY replacement_relationship
  SUBTYPE OF (action_relationship);
WHERE
  WR1: acyclic_action_relationship (SELF,
       [SELF\action_relationship.related_action],
       'PROCESS_PROPERTY_SCHEMA.REPLACEMENT_RELATIONSHIP');
END_ENTITY;
(*
```

Formal propositions:

WR1: An **action_relationship** shall not replace itself, either directly or indirectly.

5.4.7 resource_property

A **resource_property** is a characteristic of a resource. The characteristic may be the behavior, capability, or performance measure that is pertinent to a process or the actions to effect a process for which the resource is used.

NOTE - The **resource_property** is not the representation of the property. See clause 6, the **process_-property_representation_schema**.

EXAMPLE 13 - A **resource_property** for a milling machine might be: the size of the machine bed, the maximum rotational speed of the tool head, or the accuracy of the machine.

EXPRESS specification:

```
*)
ENTITY resource_property;
  name        : label;
  description : text;
  resource    : characterized_resource_definition;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **resource_property** is referred to.

description: a description of the property.

resource: the **action_resource, action_resource_relationship, action_resource_requirement,** or **action_resource_requirement_relationship** that has or is described by the property.

5.4.8 action_resource_requirement

An **action_resource_requirement** is a requirement of a resource for the performance of an **action** or **action_method**. The **action_resource_requirement** specifies either a particular type of resource or a characteristic possessed by the resource.

EXPRESS specification:

```
*)
ENTITY action_resource_requirement;
  name        : label;
  description : text;
  kind        : resource_requirement_type;
  operations  : SET [1:?] OF characterized_action_definition;
END_ENTITY;
(*
```

Attribute definitions:

name: the words or group of words by which the **action_resource_requirement** is referred to.

description: a description of the requirement.

kind: the type of **action_resource_requirement** that is being required.

operations: the set of one or more **action, action_method_relationship**s, **action_method**s, or **action_relationship**s that have the requirement.

5.4.9 action_property_relationship

An **action_property_relationship** is a relationship between two different **action_propertys**. The **action_property_relationship** identifies a relationship that may specify that an **action_property** is an alternative for another **action_property** or may identify a dependency between two **action_propertys**.

NOTE - The interpretation of different uses of this entity is not defined in this part of ISO 10303. Application resources and application protocols provide further distinctions.

EXAMPLE 14 - A milling operation is an **action_method**. Duration of the milling operation is an **action_property**. Temperature of the atmosphere for the milling operation is an **action_property**. The **action_property_relationship** identifies the relationship between the duration and temperature for the milling operation.

EXPRESS specification:

```
*)
ENTITY action_property_relationship;
  name                        : label;
  description                 : text;
  relating_action_property : action_property;
  related_action_property  : action_property;
WHERE
  WR1: relating_action_property :<>: related_action_property;
END_ENTITY;
(*
```

Attribute definitions:

name: the word of group of words by which the **action_property_relationship** is referred to.

description: a description of the relationship.

relating_action_property: the **action_property** that is associated with the related **action_property**. If precedence is important in the association, the **relating_action_property** is the independent **action_property**.

related_action_property: the **action_property** that is associated with the relating **action_property.** If precedence is important in the association, the **related_action_property** is the dependent or alternate **action_property**.

Formal propositions:

WR1: The **relating_action_property** and the **related_action_property** shall be different **action_propertys**.

5.4.10 requirement_for_action_resource

A **requirement_for_action_resource** is an **action_resource_requirement** that specifies one or more **action_resources** that will satisfy the requirement.

EXAMPLES

15 - A milling operation is the **action_method**. A **requirement_for_action_resource** specifies an **action_resource** of "milling machine" and a description of the milling machine.

16 - A machine that can lift 500 kilograms is needed to complete a process. The **action_resource_requirement** states the requirement, "machine with the ability to lift 500 kilograms". A specific lifting machine is an **action_resource**. If the specific lifting machine can satisfy the requirement, "machine with the ability to lift 500 kilograms", then the **requirement_for_action_resource** identifies the specific lifting machine.

EXPRESS specification:

```
*)
ENTITY requirement_for_action_resource
  SUBTYPE OF (action_resource_requirement);
  resources : SET [1:?] OF action_resource;
END_ENTITY;
(*
```

Attribute definitions:

resources: the set of one or more **action_resources** that satisfy the **action_resource_requirement**.

5.4.11 resource_property_relationship

A **resource_property_relationship** is a relationship between two **resource_propertys**.

NOTE - The interpretation of different uses of this entity is not defined in this part of ISO 10303. Application resources and application protocols provide further distinctions.

EXAMPLE 17 - An **action_resource** is a milling machine. The machine bed size and the machine milling head (e.g., 5 axis milling machine) are **resource_propertys**. A **resource_property_relationship** is the relationship between the machine milling head the machine bed size.

EXPRESS specification:

```
*)
ENTITY resource_property_relationship;
  name                         : label;
  description                  : text;
  relating_resource_property : resource_property;
  related_resource_property  : resource_property;
WHERE
  WR1: relating_resource_property :<>: related_resource_property;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **resource_property_relationship** is referred to.

description: a description of the relationship.

relating_resource_property: the **resource_property** that is associated with the related **resource_property**. If precedence is important in the association, the **relating_resource_property** is the independent **resource_property**.

related_resource_property: the **resource_property** that is associated with the relating **resource_property.** If precedence is important in the association, the **related_resource_property** is the dependent or alternate **resource_property**.

Formal propositions:

WR1: The **relating_resource_property** and the **related_resource_property** shall be different **resource_property**s.

5.4.12 action_resource_requirement_relationship

An **action_resource_requirement_relationship** is a relationship between two **action_resource_requirement**s.

NOTE - The interpretation of different uses of this entity is not defined in this part of ISO 10303. Application resources and application protocols can provide further distinctions.

EXAMPLE 18 - If the **action_resource** were for a drill, then a milling machine could be specified as an alternative for the requirement for a hand drill, since the milling machine can perform drilling operations.

EXPRESS specification:

```
*)
ENTITY action_resource_requirement_relationship;
  name                                : label;
  description                         : text;
  relating_action_resource_requirement : action_resource_requirement;
  related_action_resource_requirement  : action_resource_requirement;
WHERE
  WR1: relating_action_resource_requirement :<>:
       related_action_resource_requirement;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **action_resource_requirement_relationship** is referred to.

description: a description of the relationship.

relating_action_resource_requirement: the **action_resource_requirement** that is associated with the related **action_resource_requirement**. If precedence is important in the association, the **relating_action_resource_requirement** is the independent **action_resource_requirement**.

related_action_resource_requirement: the **action_resource_requirement** that is associated with the relating **action_resource_requirement.** If precedence is important in the association, the **related_action_resource_requirement** is the dependent or alternate **action_resource_requirement**.

Formal propositions:

WR1: The **relating_action_resource_requirement** and the **related_action_resource_requirement** shall be different **action_resource_requirements**.

5.4.13 resource_requirement_type

A **resource_requirement_type** is a taxonomic designation that is used to distinguish requirements or characteristics of requirements as belonging to an identified group or classification.

EXPRESS specification:

```
*)
ENTITY resource_requirement_type;
  name        : label;
  description : text;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **resource_requirement_type** is referred to.

description: a description of the **resource_requirement_type**.

5.4.14 resource_requirement_type_relationship

A **resource_requirement_type_relationship** is a relationship between two **resource_requirement_types**.

EXPRESS specification:

```
*)
ENTITY resource_requirement_type_relationship;
  name                        : label;
  description                 : text;
  relating_requirement_type : resource_requirement_type;
  related_requirement_type  : resource_requirement_type;
WHERE
  WR1: relating_requirement_type :<>: related_requirement_type;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **resource_requirement_type_relationship** is referred to.

description: a description of the **resource_requirement_type_relationship**.

relating_requirement_type: the **resource_requirement_type** that is associated with the related **resource_requirement_type**. If precedence is important in the association, the **related_requirement_type** is the independent **resource_requirement_type**.

related_requirement_type: the **resource_requirement_type** that is associated with the relating **resource_requirement_type**. If precedence is important in the association, the **related_requirement_type** is the dependent **resource_requirement_type**.

Formal propositions:

WR1: The **relating_requirement_type** and the **related_requirement_type** shall be different **resource_requirement_type**s.

```
*)
END_SCHEMA; -- process_property_schema
(*
```

6 Process property representation

The following EXPRESS declaration begins the **process_property_representation_schema** and identifies the necessary external references.

EXPRESS specification:

```
*)
SCHEMA process_property_representation_schema;

REFERENCE FROM support_resource_schema
     (label,
      text);

REFERENCE FROM representation_schema
     (representation);

REFERENCE FROM process_property_schema
     (action_property,
      resource_property);
(*
```

NOTES

1 - The schemas referenced above can be found in the following parts of ISO 10303:

support_resource_schema	ISO 10303-41

representation_schema	ISO 10303-43
process_property_schema	Clause 5 of this part of ISO 10303

2 - See annex D, figure D.3, for a graphical presentation of this schema.

6.1 Introduction

The subject of the **process_property_representation_schema** is the representation of the properties required by either a resource, an action, or a potential action to affect a process.

6.2 Fundamental concepts and assumptions

This clause supports the representation of a process property. A process property representation may be one of the following:

— resource parameter values;

— action parameter values.

NOTE - Usage of parameter values is not specified in this part of ISO 10303.

A given process property may be represented in different ways.

6.3 process_property_representation_schema entity definitions

6.3.1 action_property_representation

An **action_property_representation** is the representation of the property of an **action** or the **action_method** to affect a process. The representation specifies the realization of the property that the **action** exhibits or the **action_method** exhibits to affect a process.

EXAMPLES

19 - Spindle speed is an **action_property**. The parameter value "2200 r.p.m." is the **descriptive_representation_item.description** of the property spindle speed.

20 - The value "20 days" is the **descriptive_representation_item.description** of the **action_property** "time to complete".

EXPRESS specification:

```
*)
ENTITY action_property_representation;
  name           : label;
  description    : text;
  property       : action_property;
  representation : representation;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **action_property_representation** is referred to.

description: a description of the property representation.

property: the characteristic of the **action** or the **action_method** to effect a process that is being represented.

representation: the **representation** of the **property** of the **action** or the **action_method** to effect a process.

6.3.2 resource_property_representation

A **resource_property_representation** is the representation of the property of a resource. The representation specifies the realization of the property that the resource exhibits.

NOTE - There are different ways that a given property may be equivalently represented.

EXAMPLES

21 - A laser inspection machine is an **action_resource**. The table size is the **resource_property**; the text "1 meter by 2 meters" is the value of a **resource_property_representation** of the table size.

22 - A person is an **action_resource_requirement**. The person's "typing speed" is the **resource_property**. The value of "20 words per minute" is the **resource_property_representation**. This is represented through **representation**.

EXPRESS specification:

```
*)
ENTITY resource_property_representation;
  name           : label;
  description    : text;
  property       : resource_property;
  representation : representation;
END_ENTITY;
(*
```

Attribute definitions:

name: the word or group of words by which the **resource_property_representation** is referred to.

description: a description of the representation.

property: the characteristic of the resource being represented.

representation: the **representation** of the property of the resource.

```
*)
END_SCHEMA; -- process_property_representation_schema
(*
```

Annex A
(normative)

Short names of entities

Table A.1 provides the short names of entities specified in this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table A.1 - Short names of entities

Entity names	Short names
ACTION_METHOD_TO_SELECT_FROM	AMTSF
ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS	AMWAD
ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS_CONSTRAINED	AMWADC
ACTION_PROPERTY	ACTPRP
ACTION_PROPERTY_RELATIONSHIP	ACPRRL
ACTION_PROPERTY_REPRESENTATION	ACPRRP
ACTION_RESOURCE_REQUIREMENT	ACRSRQ
ACTION_RESOURCE_REQUIREMENT_RELATIONSHIP	ARRR
CONCURRENT_ACTION_METHOD	CNACMT
CONTEXT_DEPENDENT_ACTION_METHOD_RELATIONSHIP	CDAMR
CONTEXT_DEPENDENT_ACTION_RELATIONSHIP	CDAR
PROCESS_OR_PROCESS_RELATIONSHIP_EFFECTIVITY	POPRE
PROCESS_PRODUCT_ASSOCIATION	PRPRAS
PROCESS_PROPERTY_ASSOCIATION	PRPRS
PRODUCT_DEFINITION_PROCESS	PRDFPR
PROPERTY_PROCESS	PRPPRC

Table A.1 - Short names of entities (concluded)

Entity names	Short names
RELATIONSHIP_CONDITION	RLTCND
REPLACEMENT_RELATIONSHIP	RPLRLT
REQUIREMENT_FOR_ACTION_RESOURCE	RFAR
RESOURCE_PROPERTY	RSRPRP
RESOURCE_PROPERTY_RELATIONSHIP	RSPRRL
RESOURCE_PROPERTY_REPRESENTATION	RSPRRP
RESOURCE_REQUIREMENT_TYPE	RSRQTY
RESOURCE_REQUIREMENT_TYPE_RELATIONSHIP	RRTR
SEQUENTIAL_METHOD	SQNMTH
SERIAL_ACTION_METHOD	SRACMT

Annex B
(normative)

Information object registration

B.1 Document identification

In order to provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(49) version(1) }

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

B.2 Schema identification

ISO 10303-1 further describes how ISO/IEC 8824-1 can be used to identify individual schemas. This part of ISO 10303 contains three such schemas, and each is assigned a unique object identifier to provide for unambiguous identification of the schema in an open information system.

B.2.1 method_definition_schema identification

The **method_definition_schema** (see clause 4) is assigned the object identifier

{ iso standard 10303 part(49) version(1) object(1) method-definition-schema(1) }

B.2.2 process_property_schema identification

The **process_property_schema** (see clause 5) is assigned the object identifier

{ iso standard 10303 part(49) version(1) object(1) process-property-schema(2) }

B.2.3 process_property_representation_schema identification

The **process_property_representation_schema** (see clause 6) is assigned the object identifier

{ iso standard 10303 part(49) version(1) object(1) process-property-representation-schema(3) }

Annex C
(informative)

Computer-interpretable listings

This annex provides a listing of the EXPRESS entity names and corresponding short names as specified in this part of ISO 10303. It also provides a listing of the complete EXPRESS schema specified in this part of ISO 10303 without comments or other explanatory text. This annex is available in computer-interpretable form and can be found at the following URLs:

Short names: http://www.mel.nist.gov/div826/subject/apde/snr/
EXPRESS: http://www.mel.nist.gov/step/parts/part049/is/

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

NOTE - The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex D
(informative)

EXPRESS-G figures

Figures D.1 through D.3 correspond to the EXPRESS listing given in annex C. The figures use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

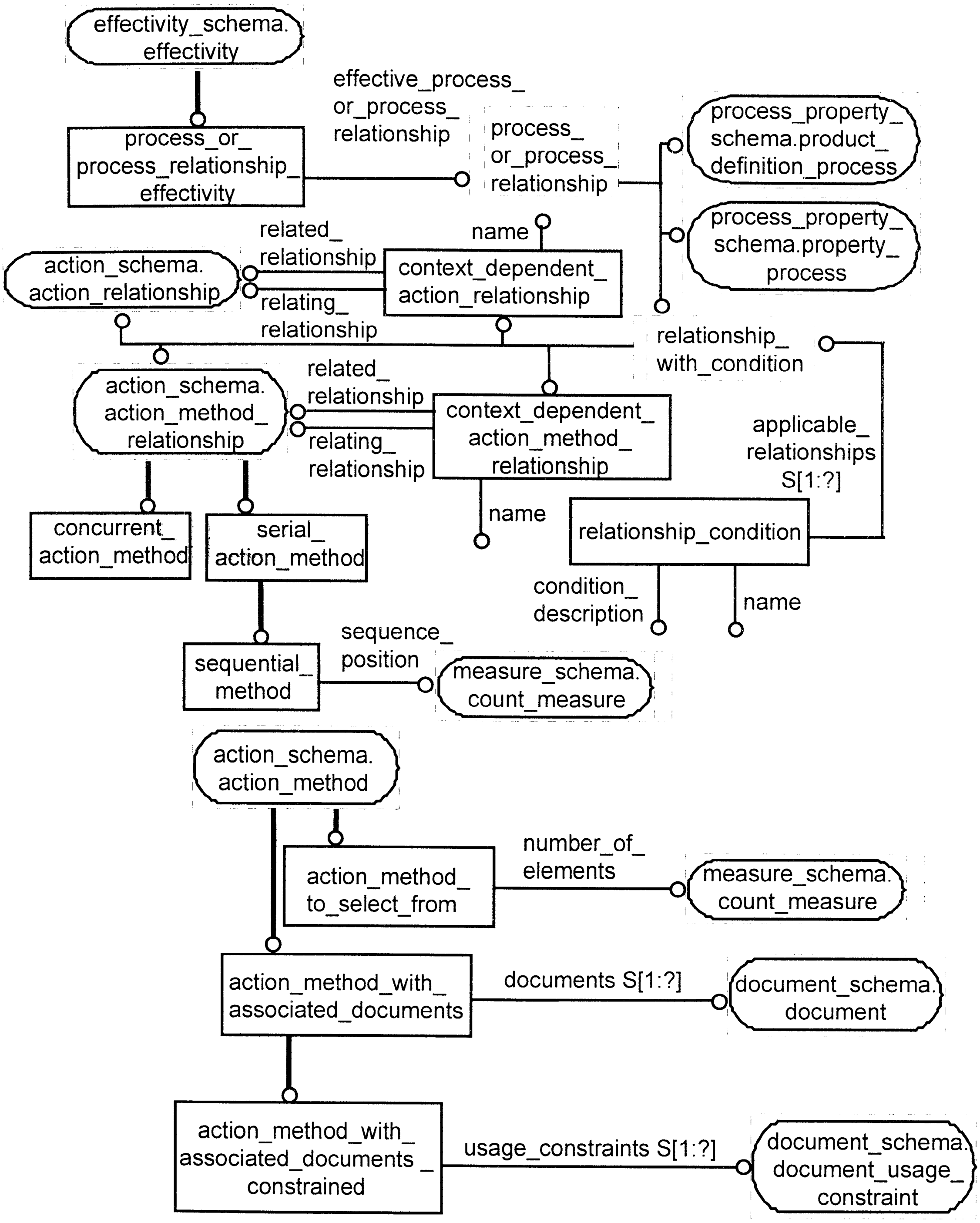

Figure D.1 - method_definition schema - EXPRESS-G diagram 1 of 1

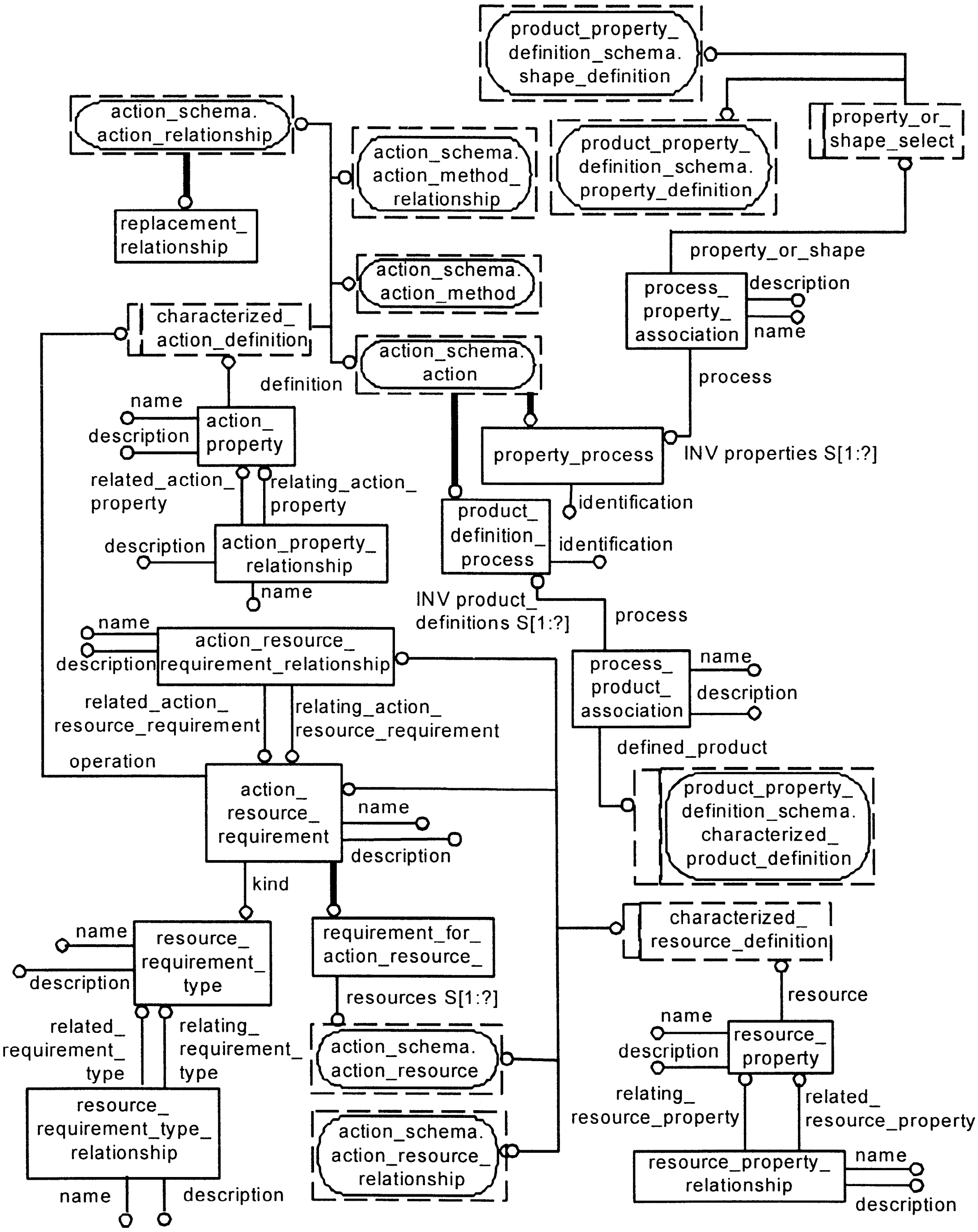

Figure D.2 - process_property schema - EXPRESS-G diagram 1 of 1

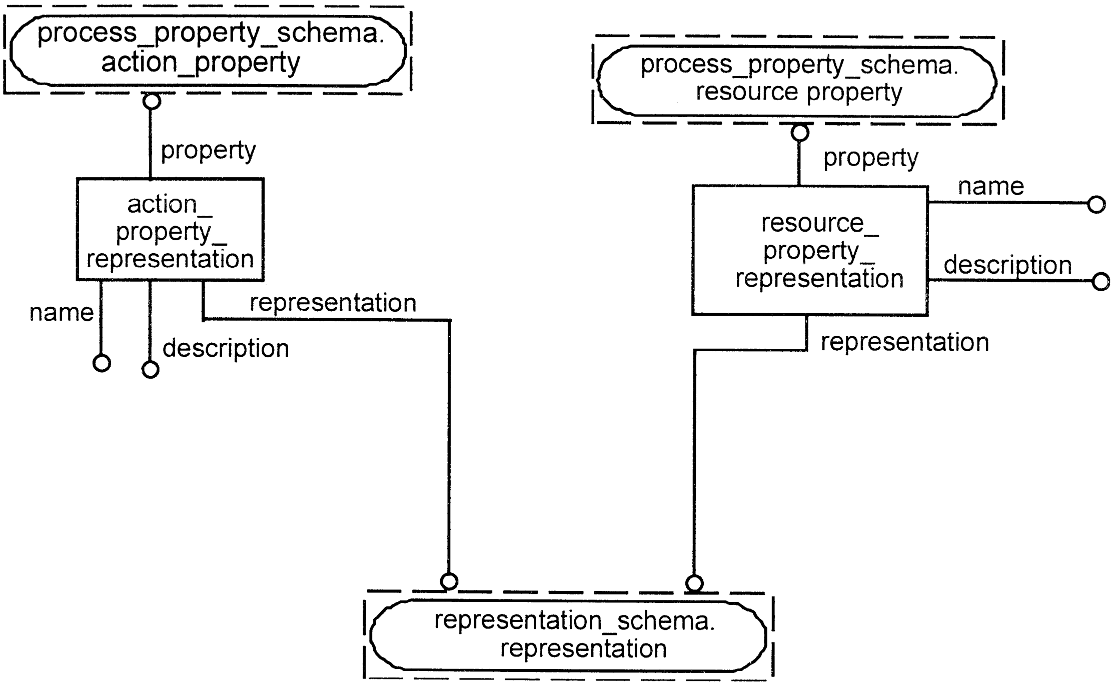

Figure D.3 - process_property_representation schema - EXPRESS-G diagram 1 of 1

Annex E
(informative)

Examples

The following examples are provided to clarify concepts that are presented within this part of ISO 10303.

23 - For a relationship with a parent to child **serial_action_method** within the context of the gear settings of an automobile automatic transmission, there are four possible gear settings that represent methods for the car to operate: drive, reverse, neutral, and park. Each gear setting is an **action_method** and is given in table E.1.

Table E.1 - Automobile gear setting as an action_method

action_method
"park"
"reverse"
"neutral"
"drive"

The context **action_method** would be the context of the capability of the automatic transmission: four possible methods for gear operation. The **serial_action_methods** are given in table E.2. The car is only in one of four gear operations at any point in time. The **action_method** of "gear operation" defines the context of the automatic transmission capabilities.

Table E.2 - Automobile gear operation as a serial_action_method

serial_action_method	
relating_action_method	**related_action_method**
"gear operation"	"drive"
"gear operation"	"reverse"
"gear operation"	"neutral"
"gear operation"	"park"

24 - For a **serial_action_method** with a peer relationship, a different set of relationships can be defined. The **serial_action_method** defines the relations between the gear operations: 1) a car has "park" next to "reverse", 2) "reverse" is next to "neutral", and 3) "neutral" is next to "drive". Table E.3 defines this set of relationships.

Table E.3 - Serial_action_method definition for gear operations

serial_action_method	
relating_action_method	**related_action_method**
"park"	"reverse"
"reverse"	"neutral"
"neutral"	"drive"

25 - Figure E.1 is a graphic of a drilling operation. For a **serial_action_method** parent to child relationship, there is a blind hole ('C') and two countersink holes ('B' and 'A'). The "drill hole" operation includes all three drill operations. Only one hole 'A', 'B' or 'C' can be drilled at one time. Therefore, the drill operations can only be executed in a serial manner. The sequence is not important, therefore 'A', 'B', or 'C' can be drilled in any order. The population is given in table E.4 and table E.5.

Figure E.1 - Drilling operation

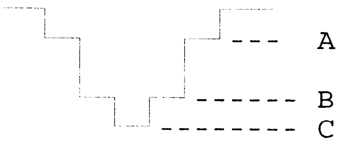

Table E.4 - Drilling operation population - 1 of 2

action_method
"drill A"
"drill B"
"drill C"
"drill hole"

Table E.5 - Drilling operation population - 2 of 2

action_method_relationship (serial_action_method)	
relating_action_method	related_action_method
"drill hole"	"drill A"
"drill hole"	"drill B"
"drill hole"	"drill C"

26 - There are several **action_methods** in the life of a person: awake, asleep, work, write document, eat, life restrictions. These are given in table E.6.

Table E.6 - Action_methods in the life of a person

action_method
"awake"
"asleep"
"work"
"write document"
"eat"
"life restrictions"

Two of these **action_methods** can not exist concurrently, asleep and awake. Therefore, a constraint is defined that restricts the life or a person as one of two methods: 1) asleep or 2) awake. The constraint is represented as a **serial_action_method** and is given in table E.7. For the second method, several things can be done concurrently: work, write document, and eat. These three activities are **action_methods** which are done during the **action_method** of awake. A **concurrent_action_method** that defines the **action_method** has parent to child relationship given in table E.8.

Table E.7 - Life restrictions as a serial_action_method

serial_action_method	
relating_action_method	**related_action_method**
"life restrictions"	"asleep"
"life restrictions"	"awake"

Table E.8 - Concurrent_action_method for awake

concurrent_action_method	
relating_action_method	**related_action_method**
"awake"	"work"
"awake"	"write document"
"awake"	"eat"

The longest **action_method** would be "awake" for the **concurrent_action_method** population. All other **action_method**s in the **concurrent_action_method** population may be completed within this timeframe if "awake" is the **concurrent_action_method**. The determination of the parent to child or peer relationships is defined by an application resource or an application protocol.

27 - In the United States, a car is driving down a street and approaches a traffic light. There are three conditions that exist for the traffic light:

1) green light;

2) yellow light;

3) red light.

The set of desired car actions for each traffic light condition is given in table E.9. The set of **action**s and the **action_relationship**s from ISO 10303-41 are given in table E.10 and table E.11.

Table E.9 - Desired car actions

condition	desired car action
"green light"	"maintain speed"
"yellow light"	"slow and stop"
"red light"	"stop"

Table E.10 - Actions for the car

action
"drive down street"
"maintain speed"
"slow and stop"
"stop"

Table E.11 - Action_relationships for the car

action_relationship	
relating_action	**related_action**
"drive down street"	"maintain speed"
"drive down street"	"slow and stop"
"drive down street"	"stop"

The set of **relationship_conditions** are defined in table E.12. The color of the light during the car approach will determine the **related_action** outcome.

Table E.12 - Relationship_conditions for the car

relationship_condition		
condition_description	**applicable_relationships = action_relationship**	
	relating_action	**related_action**
"If green light"	"drive down street"	"maintain speed"
"If yellow light"	"drive down street"	"slow and stop"
"If red light"	"drive down street"	"stop"

28 - Two factories build a part known as a widget. The **relationship_condition** contains the condition that determines in which factory a specific instance of a widget is built.

It takes three phases to build a widget: 1) figure profit, 2) build widget, 3) paint widget. In either factory figure profit is the first operation. For Factory 1, the factory paints all the components of the widget as the second operation and then assembles the widget. For Factory 2, the factory builds the widget as the second operation and then paints the assembled widget.

In terms of the widget, each phase is an **action**. The **action_relationship** from ISO 10303-41 describes the order of execution of the respective **action** as given in table E.13. The **context_dependent_action_relationship** is given in table E.14.

Table E.13 - The order of execution defined by action_relationship

action_relationship		
name	**relating_action**	**related_action**
"AR1"	"figure profit"	"build widget"
"AR2"	"build widget"	"paint widget"
"AR3"	"figure profit"	"paint widget"
"AR4"	"paint widget"	"build widget"

Table E.14 - The context_dependent_action_relationship for order of execution

context_dependent_action_relationship		
name	**relating_relationship**	**related_relationship**
"CDAR1"	"AR3"	"AR1"

The **relationship_condition** is a test if the factory is Factory 1 or Factory 2. The determination of the condition returns true or false. If the company wants to manufacture the widget in Factory 1, the condition is true and "AR3" is selected. If the company wants to build the widget in Factory 2, the condition is false and "AR1" is selected. It follows that the appropriate methods for the selected factory are completed.

29 - If a set of three activities must be completed in a specified order, then the **sequential_method** would be defined (in terms of example 28) as given in table E.15 and table E.16.

Table E.15 - Action_method for sequential_method

action_method
"figure profit"
"paint widget"
"assemble widget"
"make widget"

Table E.16 - Sequential_method for specified order

sequential_method		
action_method attribute		sequence_position
relating	related	
"make widget"	"figure profit"	1
"make widget"	"paint widget"	2
"make widget"	"assemble widget"	3

The **sequence_position** attribute defines the order that the **action_method**s are to be completed in.

30 - If a builder has three ways to mix concrete, then three **action_method**s exist as given in table E.17.

Table E.17 - Builder concrete mix methods

action_method
"mix 1"
"mix 2"
"mix 3"

If the builder can mix the concrete only two ways during a day, then the builder has created an **action_method** (that is called "daily_mix_combo") that will define the constraint for mixing concrete. The set of **action_methods** has been increased as given in table E.18.

Table E.18 - Builder concrete mix methods with restriction

action_method
"mix 1"
"mix 2"
"mix 3"
"daily_mix_combo"

The remaining aspect of the "daily_mix_combo" **action_method** constraint will be that the **action_method** is an **action_method_to_select_from** because it has an additional constraint of **number_of_elements** that is the number of mixes that the builder will mix in a day. The **number_of_elements** is defined as 2. To build the collection for the **action_method_to_select_from** there has to be a collection of the **action_method_relationships** defined in table E.19.

Table E.19 - Builder collection of action_method_relationships

action_method_relationship	
relating_action_method (parent)	**related_action_method (child)**
"daily_mix_combo"	"mix 1"
"daily_mix_combo"	"mix 2"
"daily_mix_combo"	"mix 3"

The person that acquires the builder's services through an **action** specifies the required **action_methods** to follow at the construction site. Each day the builder would select two of the three options in the "daily_mix_combo".

Index

ICS 25.040.40

Descriptors: automation, automation engineering, computer applications, industrial products, data, data representation, data exchange.

Price based on 50 pages