INTERNATIONAL STANDARD

ISO 15531-42

First edition
2005-09-15

Industrial automation systems and integration — Industrial manufacturing management data —

Part 42: **Time Model**

Systèmes d'automatisation industrielle et intégration — Données de gestion de fabrication industrielle —

Partie 42: Modèle du temps

Reference number
ISO 15531-42:2005(E)

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 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

Figures

Foreword

The International Organisation for Standardisation (ISO) 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 organisations, 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 ISO/IEC Directives, Part 2.
Draft International Standards (DIS) 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.

Attention is drawn to the possibility that some of the elements of this part of ISO 15531 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

This part of ISO 15531 was prepared by the Technical committee ISO/TC 184 *Industrial automation systems and integration,* Sub-Committee 4 *Industrial data.*

A complete list of parts of ISO 15531 is available from the Internet.

http://www.tc184-sc4.org/titles

Introduction

Software applications related to factory or enterprise production, such as scheduling software, manufacturing management software, cost evaluation software, maintenance management, purchasing software, delivery software..., strongly require a reference to time related features such as point in time (date) and duration (interval of time). These references are needed to ensure the necessary time related relationships between the events dealt with by the applications.

The availability of standardised time related references is particularly important for complex applications with multi-process environments, what is an environment commonly met in manufacturing.

In most of the standards, the time features are not independent from the events and the manufacturing management data they address. This leads to some difficulties in the way to handle time related relationships between events or data that include their own time relation and representation. In some of them the time related features may depend on events or objects addressed and their representation may change depending on the context, without any simple tool to identify the relation between them. This may be crucial in an environment where various processes are performed simultaneously or where many closely related software tools are used at the same time.

Developed in compliance with the “System theory” approach this part of ISO 15531 identifies the time as a constraint of the system environment and provides time related features included in a time model fully independent from the events handled by the manufacturing system. This time model is also fully independent from any manufacturing management data used by the manufacturing applications.

Note: For further explanations on time related concepts in system theory see ISO 15531-31 annex D.

Furthermore the time domain makes use of a domain property schema that, as one dimension domain, is generic enough to be usable separately from time specific properties.

In addition, the time model provided in this standard is written in EXPRESS to ensure better compatibility with ISO 10303.

Note: The background of this standard comes from the initial work developed in the JWG8 by Professor Dangelmaier from the Hans Nixdorf Institute of the Paderborn University, Germany.

Industrial automation systems and integration — Industrial manufacturing management data —

Part 42: **Time Model**

1 Scope

This part of ISO 15531 specifies a generic framework for the description of topological properties applicable to a wide range of one dimension domains.

However, and given the needs of the other parts of ISO 15531, the developments made in this part are focused on the time domain, thus providing a universal, self consistent model, independent of any event that may occur, or has already occurred, at a given point in time.

The time model specified in this part enables any software application to provide an accurate time reference to any related event, or sequence of events, whether in the past, in the present, or in the future. This time model may be used by any application that needs to reference events, actions, or sequences of action linked to time or intervals of time simultaneously.

According to this, the scope of this part of ISO 15531 includes the following:

— the description of the topological properties of an one dimension domain;

— the description of the time model and of the related definitions;

— the EXPRESS definitions of entities, attributes and schemas as prescribed in ISO 10303-11 (the reference manual of the EXPRESS language);

— the EXPRESS-G diagrams of the model as prescribed in ISO 10303-11.

The following are out of the scope of this part of ISO 15531:

— the description of any kind of measure theory or measure method

— elements and domains of more than one dimension

— the modelling of any event as described in ISO 10303-41

EXAMPLE Since their domain dimension is greater than one, squares, rectangles, triangles,..as well as cubes and spheres are out of the scope of this part of ISO 15531.

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: *Information Technology — Abstract Syntax Notation One (ASN.1) — Specification of Basic Notation — Part 1.*

ISO 10303-11: *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.*

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

ISO 15531-1*: Industrial automation systems and integration — Industrial manufacturing management data — Part 1: General overview.*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
discrete manufacturing
production of discrete items.

[ISO 15531-1]

3.1.2
domain
collection of one dimension elements, that can be ordered and positionned applying a given measure method

NOTE Since the purpose of this document is the description of a “time model” the restriction of “one dimension” element in the definition only applies to this document that excludes domain such as rectangle, triangle, cubes and spheres. Whithout the restriction to “one dimension” element the definition may apply to any kind of domain.

EXAMPLE a domain can be specified as a list of integers such as 1, 2, 3,4….

3.1.3
domain point
point in a domain
domain element for which any defined measure of it in the domain is zero

3.1.4
element
static representation of a part of the universe of discourse that may be identified and characterised by its behaviour and attributes

NOTE A static representation is a snapshot of the part of the universe of discourse under consideration at a given time. It may include dynamic attributes as, for example, behaviour. Those attributes characterise the element as it is or as it is expected to be at a given time.

[ISO 15531-1]

3.1.5
entity
a class of information defined by common properties

[ISO 10303-11]

3.1.6
environment
part of the universe of discourse that does not belong to the system itself

EXAMPLE inputs and outputs of the systems such as raw material, final products etc., belong to the environment of the system as well as constraints that apply to it or time

3.1.7
event_occurrence
fact of an existence of a state at some point in time

NOTE the point in time of the existence may not be known in term of calendar date before the event_occurrence actually happens. One reason, why the event_occurrence cannot always be expressed as a calendar date, is that the event_occurrence is not possible to plan, for example a breakdown of machine.

EXAMPLE "Start of production", "breakdown of machine A".

[ISO 10303-41ed2 clause 16.4.7]

3.1.8
flow
motion of a set of physical or informational objects in space and time

[ISO 15531-1]

3.1.9
flow control
specific production control system that is based primarily on setting production rates and feeding work into production to meet these planned rates, then monitoring and controlling production.

NOTE That includes the act of checking and driving the flow according to a given purpose. The term may also apply to the function or service.

3.1.10
duration
interval of time
length of a period of time, measured using a given unit of time

EXAMPLE 1 the 24 hours between Monday 1.00 p.m. and Tuesday 12.00 a.m.

EXAMPLE 2 every Monday of every week between January and July.

NOTE interval of time measures the distance between two points in time. In that case it is the length of the time domain that is bounded by the two points in time under consideration.

3.1.11
measure

result of a quantitative evaluation of a given property of any physical or mathematical object

NOTE 1 the term measure may also be utilised for the operation that leads to this result.

EXAMPLE measure of length, area, volume, mass distribution, probability distribution, period of time, etc.

NOTE 2 any measure implies the definition of a method (procedure, theory) to get it, that includes the definition of the correponding unit of the measure. In particular any measure of an interval of time shall imply the definition of a unit of time and shall refer to it.

NOTE 3 From a mathematical point of view a measure is sometimes defined as a way to determine the distance between two points of the space under consideration (the length of the segment that joint these two points) as well as the result . In that sense duration is the measure of a period of time.

NOTE 4 For information the generic mathematical definition of the measure is : nonnegative function of subsets of a space completely additive in the sense that the measure of the union of a sequence of mutually disjoint sets is the sum of the measures of the sets.

3.1.12
method of measure

set of rules and intermediate steps, including the definition and the use of a unit of measure to be observed to achieve the considered measure

NOTE 1 Methods of measure often result from theories and principles that implicitly or explicitly derive from mathematical theories of the measure theories and from theoritical or experimental considerations on the property to be assessed.

NOTE 2 Mathematical theory of the measure theory is the study of measurable sets and functions, introduced by Lebesgue in order to generalize the Riemann integral.

3.1.13
point in time

location of something noticeable within a time domain

NOTE another equivallent definition may be : point in the time domain, applying definition 3.4.1 and 3.4.2 to the time domain

EXAMPLE 1 Wednesday, 15th of March, 2003.

EXAMPLE 2 9.30 a.m.

3.1.14
scheduling

act, function or result of planning occurrences of manufacturing activities

[ISO 15531-1]

3.1.15
time
feature of the enterprise universe of discourse that enables the location of noticeable things along an infinite oriented axis and allows the ordering of their succession or the identification and characterisation of this succession

EXAMPLE noticeable things may be events that have occurred, that may occur or are expected to occur. That may also be tags along the time axis.

NOTE Two main enterprise entities are related to the time. The first one is the point in time that enables the identification or assessment of location along the time axis. The second one is the interval of time that enables the determination of the distance between two points in time. See ISO 15531-31 annex D and IEC 62264-1 annex F.

3.1.16
time domain
period
set of points in time

EXAMPLE 1 the worked period within a year.

EXAMPLE 2 the maintenance period of a machine tool.

NOTE 1 A time domain may be finite or infinite. It may be bounded or not by one or two points in time.

NOTE 2 A time domain may be composed of other time domains.

3.1.17
time_interval
Identification of an intervening time

EXAMPLE "strike duration", "delay of production", "Christmas holydays"

[ISO 10303-41ed2 clause 16.4.14]

3.1.18
time model
model of the enterprise environment feature "time"

[ISO 15531-1]

3.1.19
unit of time
unit, that is implicitly or explicitly a multiple of the internationally defined second, to which a measure of time in a time domain has to refer

NOTE 1 in other words unit of time is the quantity of time chosen as a reference in terms of witch other quantity of time may be expressed. The second is the unit of time defined in the SI system of units.

NOTE 2 the multiplication ratio may be either greater or smaller than one.

3.1.20
universe of discourse
the collection of concrete or abstract things that belong to an area of the real world, selected according to its interest for the system to be modelled and for its corresponding environment

[ISO 15531-1]

3.2 Abbreviated terms

MANDATE	MANufacturing DATa Exchange
STEP	STandard for the Exchange of Product model data

4 Overview of the whole ISO 15531 scope

ISO 15531 specifies the characteristics for a representation of manufacturing management information over the entire industrial process with the necessary mechanisms and definitions to enable manufacturing management data to be shared and exchanged within the factory, with other plants or with companies.

Exchanges are made through different computer systems and environments associated with the complete industrial process. The standard is focused on discrete manufacturing but not limited to it. Nevertheless any extension to industrial processes which does not belong to discrete manufacturing is always under consideration when it does not imply any contradiction or inconsistency with the initial objective of the standard.

The following are within the scope of ISO 15531:

— the representation of production and resources information including capacity, monitoring, maintenance constraints and control;

NOTE - Maintenance constraints and relevant maintenance management data are taken into account from the point of view of their impact on the flow control.

— the exchange and sharing of production information and resources information including storing, transferring, accessing and archiving.

The following are outside the scope of ISO 15531:

— enterprise modelling;

NOTE - That means that tools, architecture and methodologies for the modelling of an enterprise in its whole are not in the scope of ISO 15531.

— product data (representation and exchange of product information);

— component data (parts library: representation and exchange of computer-interpretable parts library information);

— cutting tools (electronic representation for exchange of cutting tool data);

— technical maintenance information (technical information such as those included in devices repair, operation and maintenance manuals).

ISO 15531-1 provides a full overview of the whole ISO 15531 and of the relationship between its different parts.

The modelling language used to model the manufacturing management data addressed by ISO 15531 is EXPRESS that is defined in ISO 10303-11.

5 Domain property

5.1 Preamble

The domain property schema provides a generic structure applicable to any one dimension domain, enabling a hierarchical decomposition at the upper level of the tree, becoming then a network at the lower levels of the decomposition.

The root of the tree is made of an abstract entity called domain. This domain is considered as a set of domain points. This entity defines, in turn, the abstract supertype of three other abstract entities: complete or composite domain, discrete or continuous domain and bounded domain, characterizing the three basic categories from which the different kinds of domains are built. These four entities define the fundamental elements on which the domain property schema is built.

At the third level of the hierarchy, the three abstract entities of the second level are split into two sub-categories, leading to the entities of the fourth level. The entities belonging to the fourth level are related in between them through a network structure. This network structure comes from fundamental topological features of a domain, mentioned below.

5.2 Fundamental concepts and assumptions

Basically, the fundamental properties of a given domain are aimed at providing the possibility of covering the following range of topological structures for a set of domain points:

— complete;

— composite;

— discrete;

— continuous;

— upper-bounded;

— lower-bounded.

Combining the entities of each of the three initial branches with entities of the other branches leads to a network of combinations. the combinations are made in order to keep sense to the resulting entity.

EXAMPLE: possible combinations are:

— lower bounded composite discrete domain

— upper bounded complete continuous domain

In parallel to the definition of all possible domains, another concept enables the definition of domain points the domain is made of. This concept is necessary to bring a lower bound to a lower bounded domain, an upper bound to an upper bounded domain and to provide a reference to the origin of the complete domain entity.

5.3 Domain property schema definition

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

EXPRESS specification

```
*)
SCHEMA domain_property_schema;

REFERENCE FROM measure_schema            -- ISO 10303-41
(unit);
REFERENCE FROM support_resource_schema   -- ISO 10303-41
(label);

(*
```

NOTE 1 The schemas referred to above can be found in the following parts of ISO 10303-41:

measure_schema: clause 21
support_resource_schema: clause 20

NOTE 2 See annex C.1 for a graphical presentation of this schema using the EXPRESS-G notation.

5.4 Domain property type definitions

5.4.1 type_of_identifier_of_dom_gen_struct

A **type_of_identifier_of_dom_gen_struct** is an alphanumeric string which may be used for something to be identified. It does not need to be understandable through the natural language.

EXPRESS-specification:

```
*)
TYPE type_of_identifier_of_dom_gen_struct = STRING;
END_TYPE;
(*
```

5.4.2 type_of_domain

A **type_of_domain** is an alphanumeric string which may be used to identify the domain.

EXPRESS-specification:

```
*)
TYPE type_of_domain = STRING;
END_TYPE;
(*
```

5.5 domain_property subtype constraint definitions

5.5.1 covered_range

A **covered_range** is the subtype constraint used to define subtypes of **domain**s for which a topological structure is needed.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT covered_range FOR domain;
    ABSTRACT SUPERTYPE;
    ONEOF (time_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.2 complcompos_discrcont_bounded

A **complcompos_discrcont_bounded** is the subtype constraint used to define subtypes of **domain_generic_structure** related to the property of being either complete, or composite, or discrete, or continuous, or bounded.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT complcompos_discrcont_bounded FOR
    domain_generic_structure;
    ABSTRACT SUPERTYPE;
    ONEOF (complete_or_composite_domain, discrete_or_continuous
    _domain, bounded_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.3 complete_composite

A **complete_composite** is the subtype constraint used to define subtypes of a **complete_or_composite_domain** related to the property of being either complete or composite.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT complete_composite FOR
    complete_or_composite_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (complete_domain, composite_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.4 discrete_continuous

A **discrete_continuous** is the subtype constraint used to define subtypes of a **discrete_or_continuous_domain** related to the property of being either discrete or continuous.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT discrete_continuous FOR
    discrete_or_continuous_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (continuous_domain, discrete_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.5 lower_upper

A **lower_upper** is the subtype constraint used to define subtypes of a **bounded_domain** related to the property of being either lower bounded or upper bounded.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT lower_upper FOR bounded_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (lower_bounded_domain, upper_bounded_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.6 compl_continuous_discrete

A **compl_continuous_discrete** is the subtype constraint used to define subtypes of a **complete_domain** related to the property of being either continuous or discrete.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT compl_continuous_discrete FOR complete_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (complete_continuous_domain, complete_discrete_domain);
```

```
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.7 compos_continuous_discrete

A **compos_continuous_discrete** is the subtype constraint used to define subtypes of a **composite_domain** related to the property of being either continuous or discrete.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT compos_continuous_discrete FOR composite_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (composite_continuous_domain, composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.8 cont_complete_composite

A **cont_complete_composite** is the subtype constraint used to define subtypes of a **continuous_domain** related to the property of being either complete or composite..

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT cont_complete_composite FOR continuous_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (complete_continuous_domain, composite_continuous_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.9 disc_complete_composite

A **disc_complete_composite** is the subtype constraint used to define subtypes of a **discrete_domain** related to the property of being either complete or composite.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT disc_complete_composite FOR discrete_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (complete_discrete_domain, composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.10 both_bd_compl_compos_disc_cont

A **both_bd_compl_compos_disc_cont** is the subtype constraint used to define subtypes of a **both_sides_bounded_domain** related to the property of being either complete and continuous, or complete and discrete, or composite and continuous, or composite and discrete.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT both_bd_compl_compos_disc_cont FOR
both_sides_bounded_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (both_sides_bounded_complete_continuous_domain,
    both_sides_bounded_complete_discrete_domain,
    both_sides_bounded_composite_continuous_domain,
    both_sides_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.11 low_bd_compl_compos_disc_cont

A **low_bd_compl_compos_disc_cont** is the subtype constraint used to define subtypes of a **lower_bounded_domain** related to the property of being either complete and continuous, or complete and discrete, or composite and continuous, or composite and discrete.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT low_bd_compl_compos_disc_cont FOR
    lower_bounded_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (lower_bounded_complete_continuous_domain,
    lower_bounded_complete_discrete_domain,
    lower_bounded_composite_continuous_domain,
    lower_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;
(*
```

5.5.12 up_bd_compl_compos_disc_cont

A **up_bd_compl_compos_disc_cont** is the subtype constraint used to define subtypes of an **upper_bounded_domain** related to the property of being either complete and continuous, or complete and discrete, or composite and continuous, or composite and discrete.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT up_bd_compl_compos_disc_cont FOR
    upper_bounded_domain;
    ABSTRACT SUPERTYPE;
    ONEOF (upper_bounded_complete_continuous_domain,
    upper_bounded_complete_discrete_domain,
    upper_bounded_composite_continuous_domain,
    upper_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;
```

(*

5.5.13 compl_disc_low_both_up

A **compl_disc_low_both_up** is the subtype constraint used to define subtypes of a **complete_discrete_domain** related to the property of being either lower bounded, or upper bounded, or both sides bounded.

EXPRESS-specification:

*)

```
SUBTYPE_CONSTRAINT compl_disc_low_both_up FOR
    complete_discrete_domain;
    ONEOF (lower_bounded_complete_discrete_domain,
    both_sides_bounded_complete_discrete_domain,
    upper_bounded_complete_discrete_domain);
END_SUBTYPE_CONSTRAINT;
```

(*

5.5.14 compl_cont_low_both_up

A **compl_cont_low_both_up** is the subtype constraint used to define subtypes of a **complete_continuous_domain** related to the property of being either lower bounded, or upper bounded, or both sides bounded.

EXPRESS-specification:

*)

```
SUBTYPE_CONSTRAINT compl_cont_low_both_up FOR
    complete_continuous_domain;
    ONEOF (lower_bounded_complete_continuous_domain,
    both_sides_bounded_complete_continuous_domain,
    upper_bounded_complete_continuous_domain);
END_SUBTYPE_CONSTRAINT;
```

(*

5.5.15 compos_cont_low_both_up

A **compos_cont_low_both_up** is the subtype constraint used to define subtypes of a **composite_continuous_domain** related to the property of being either lower bounded, or upper bounded, or both sides bounded.

EXPRESS-specification:

*)

```
SUBTYPE_CONSTRAINT compos_cont_low_both_up FOR
    composite_continuous_domain;
    ONEOF (lower_bounded_composite_continuous_domain,
    both_sides_bounded_composite_continuous_domain,
    upper_bounded_composite_continuous_domain);
```

```
END_SUBTYPE_CONSTRAINT;
 (*
```

5.5.16 compos_disc_low_both_up

A **compos_disc_low_both_up** is the subtype constraint used to define subtypes of a **composite_discrete_domain** related to the property of being either lower bounded, or upper bounded, or both sides bounded.

EXPRESS-specification:

```
*)
SUBTYPE_CONSTRAINT compos_disc_low_both_up FOR
    composite_discrete_domain;
    ONEOF (lower_bounded_composite_discrete_domain,
    both_sides_bounded_composite_discrete_domain,
    upper_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;
*)
```

5.6 Domain property entity definitions

5.6.1 domain

As indicated in the clause 3.4.1 a domain is a collection of elements that can be ordered and positioned appling a given method of measure. In this document only one dimension domains and one dimension elements are considered.

EXPRESS-specification:

```
*)
ENTITY domain;
    id: type_of_domain;
UNIQUE
    UR1: id;
END_ENTITY;
(*
```

Attribute definitions:

id: allows a **domain** to be identified.

Formal propositions:

UR1: the identification of the domain shall be unique.

5.6.2 time_domain

A **time_domain** is a domain which is able to contain a sequence of points in time.

EXAMPLE 1 - A time_domain can be specified as the list of all years in an gregorian calendar:
{..., 1955, 1956, 1957,...}.

EXAMPLE 2 - A time_domain can be specified as the term of payment of salaries of workers.

EXPRESS-specification:

```
*)
ENTITY time_domain
SUBTYPE OF (domain);
    name: label;
    decomposes_into: SET [0:?] OF time_domain;
END_ENTITY;
(*
```

Attribute definitions:

name: the label by with the **time_domain** is known.
decomposes_into: the set of **time_domain**s the **time_domain** is composed of of.

5.6.3 domain_generic_structure

A **domain_generic_structure** specifies a generic structure applicable to multiple kinds of one dimension domains.

EXPRESS-specification:

```
*)
ENTITY domain_generic_structure;
    id: type_of_identifier_of_dom_gen_struct;
UNIQUE
    UR1: id;
END_ENTITY;
(*
```

Attribute definitions:

id: allows a **domain_generic_structure** to be identified.

Formal propositions:

UR1: the identification of the domain_generic_structure shall be unique.

5.6.4 complete_or_composite_domain

A **complete_or_composite_domain** is the abstract representation of a **complete_domain** or a **composite_domain**.

EXPRESS-specification:

```
*)
```

```
ENTITY complete_or_composite_domain
SUBTYPE of (domain_generic_structure);
END_ENTITY;
(*
```

5.6.5 discrete_or_continuous_domain

A **discrete_or_continuous_domain** is the abstract representation of a **discrete_domain** or a **continuous_domain**.

EXPRESS-specification:

```
*)
ENTITY discrete_or_continuous_domain
SUBTYPE of (domain_generic_structure);
END_ENTITY;
(*
```

5.6.6 bounded_domain

A **bounded_domain** is the abstract representation of a **domain** limited by bounds.

EXPRESS-specification:

```
*)
ENTITY bounded_domain
SUBTYPE of (domain_generic_structure);
END_ENTITY;
(*
```

5.6.7 complete_domain

A **complete_domain** is a **domain** without gaps, where the **domain_points** specified by the **domain** cover the whole spectrum of possibilities. There are no **domain_point**s which cannot be assigned to a **complete_domain**.

EXPRESS-specification:

```
*)
ENTITY complete_domain
SUBTYPE OF (complete_or_composite_domain);
    reference_to_unit: unit;
    reference_to_origin: domain_point;
END_ENTITY;
(*
```

Attribute definitions:

reference_to_unit: **unit** by which the duration of periods in the **complete_continuous_domain** is measured.
reference_to_origin: **domain_point** defining the origin of the **complete_continuous_domain.**

5.6.8 composite_domain

A **composite_domain** is a **domain** composed of one or more instances of **domain**s. The combined **domain**s must not overlap.

EXPRESS-specification:

```
*)
ENTITY composite_domain
SUBTYPE OF (complete_or_composite_domain);
    components: SET[0:?] OF domain;
WHERE     -- components do not overlap
    WR1 : components (domain.id [1] <>
        components (domain.id [2]));
END_ENTITY;
(*
```

Attribute definitions:

Components: the set of **domain**s the **composite_domain** is composed of.

Formal propositions:

WR1: the components of two domain**s shall not overlap.**

5.6.9 continuous_domain

A **continuous_domain** specifies a domain as an ordered, infinite and continuous set of points. The ordering expresses the sequence of the points of the domain. A **continuous_domain** allows to exactly determine the **domain_point** needed.

NOTE 1 In a **continuous_domain**, between every ordered pair of domain_points there is a third point being bigger than the first and smaller than the second point.

NOTE 2 It is impossible to identify all the points in a **continous_domain** by any enumeration. Hence a **continuous_domain** can only be represented as an interval of real numbers.

EXPRESS-specification:

```
*)
ENTITY continuous_domain
SUBTYPE OF (discrete_or_continuous_domain);
END_ENTITY;
(*
```

5.6.10 discrete_domain

A **discrete_domain** specifies a domain as a set of **domain_point**s which can be enumerated. This ordering denotes the sequence of the points within the domain.

EXPRESS-specification:

```
*)
ENTITY discrete_domain
SUBTYPE OF (discrete_or_continuous_domain);
END_ENTITY;
(*
```

5.6.11 lower_bounded_domain

A **lower_bounded_domain** is a **bounded_domain** limited by a **lower_bound**.

EXPRESS-specification:

```
*)
ENTITY lower_bounded_domain
SUBTYPE OF (bounded_domain);
    lower_bound: domain_point;
END_ENTITY;
(*
```

Attribute definitions:

lower_bound: smallest **domain_point** in the **lower_bounded_domain.**

5.6.12 upper_bounded_domain

An **upper_bounded_domain** is a **bounded_domain** limited by an **upper_bound**.

EXPRESS-specification:

```
*)
ENTITY upper_bounded_domain
SUBTYPE OF (bounded_domain);
    upper_bound: domain_point;
END_ENTITY;
(*
```

Attribute definitions:

upper_bound: biggest **domain_point** in the **upper_bounded_domain.**

5.6.13 both_sides_bounded_domain

A **both_sides_bounded_domain** is a **bounded_domain** limited by a **lower_bound** and an **upper_bound**.

EXPRESS-specification:

```
*)
ENTITY both_sides_bounded_domain;
```

```
    low_bound_assigned_to: lower_bounded_domain;
    up_bound_assigned_to: upper_bounded_domain;
END_ENTITY;
(*
```

Attribute definitions:

low_bound_assigned_to: smallest **domain_point** in the **lower_bounded_domain.**
up_bound_assigned_to: biggest **domain_point** in the **upper_bounded_domain.**

5.6.14 complete_continuous_domain

A **complete_continuous_domain** is a **complete_domain** which is also continuous.

EXPRESS-specification:

```
*)
ENTITY complete_continuous_domain
SUBTYPE OF (complete_domain, continuous_domain);
END_ENTITY;
(*
```

5.6.15 lower_bounded_complete_continuous_domain

A **lower_bounded_complete_continuous_domain** is a **complete_continuous_domain** with a lower bound.

EXPRESS-specification:

```
*)
ENTITY lower_bounded_complete_continuous_domain
SUBTYPE OF (lower_bounded_domain, complete_continuous_domain);
END_ENTITY;
(*
```

5.6.16 upper_bounded_complete_continuous_domain

An **upper_bounded_complete_continuous_domain** is a **complete_continuous_domain** with an upper bound.

EXPRESS-specification:

```
*)
ENTITY upper_bounded_complete_continuous_domain
SUBTYPE OF (upper_bounded_domain, complete_continuous_domain);
END_ENTITY;
(*
```

5.6.17 both_sides_bounded_complete_continuous_domain

A **both_sides_bounded_complete_continuous_domain** is a **complete_continuous_domain** with an upper bound and a lower bound.

EXPRESS-specification:

```
*)
ENTITY both_sides_bounded_complete_continuous_domain
SUBTYPE OF (both_sides_bounded_domain, complete_continuous_domain);
END_ENTITY;
(*
```

5.6.18 composite_continuous_domain

A **composite_continuous_domain** is a **composite_domain**, which is also continuous.

EXPRESS-specification:

```
*)
ENTITY composite_continuous_domain
SUBTYPE OF (composite_domain, continuous_domain);
END_ENTITY;
(*
```

5.6.19 lower_bounded_composite_continuous_domain

A **lower_bounded_composite_continuous_domain** is a **composite_continuous_domain** with a lower bound.

EXPRESS-specification:

```
*)
ENTITY lower_bounded_composite_continuous_domain
SUBTYPE OF (lower_bouded_domain, composite_continuous_domain);
END_ENTITY;
(*
```

5.6.20 upper_bounded_composite_continuous_domain

An **upper_bounded_composite_continuous_domain** is a **composite_continuous_domain** with an upper bound.

EXPRESS-specification:

```
*)
ENTITY upper_bounded_composite_continuous_domain
SUBTYPE OF (upper_bouded_domain, composite_continuous_domain);
END_ENTITY;
(*
```

5.6.21 both_sides_bounded_composite_continuous_domain

A **both_sides_bounded_composite_continuous_domain** is a **composite_continuous_domain** with an upper bound and a lower bound.

EXPRESS-specification:

```
*)
ENTITY both_sides_bounded_composite_continuous_domain
SUBTYPE OF (both_sides_bouded_domain, composite_continuous_domain);
END_ENTITY;
(*
```

5.6.22 complete_discrete_domain

A **complete_discrete_domain** is a **complete_domain**, which is also discrete.

EXPRESS-specification:

```
*)
ENTITY complete_discrete_domain
SUBTYPE OF (complete_domain, discrete_domain);
END_ENTITY;
(*
```

5.6.23 lower_bounded_complete_discrete_domain

A **lower_bounded_complete_discrete_domain** is a **complete_discrete_domain** with a lower bound.

EXPRESS-specification:

```
*)
ENTITY lower_bounded_complete_discrete_domain
SUBTYPE OF (lower_bouded_domain, complete_discrete_domain);
END_ENTITY;
(*
```

5.6.24 upper_bounded_complete_discrete_domain

An **upper_bounded_complete_discrete_domain** is a **complete_discrete_domain** with an upper bound.

EXPRESS-specification:

```
*)
ENTITY upper_bounded_complete_discrete_domain
SUBTYPE OF (upper_bouded_domain, complete_discrete_domain);
END_ENTITY;
(*
```

5.6.25 both_sides_bounded_complete_discrete_domain

A **both_sides_bounded_complete_discrete_domain** is a **complete_discrete_domain** with an upper bound and a lower bound.

EXPRESS-specification:

```
*)
ENTITY both_sides_bounded_complete_discrete_domain
SUBTYPE OF (both_sides_bouded_domain, complete_discrete_domain);
END_ENTITY;
(*
```

5.6.26 composite_discrete_domain

A **composite_discrete_domain** is a **composite_domain**, which is also discrete.

EXPRESS-specification:

```
*)
ENTITY composite_discrete_domain
SUBTYPE OF (composite_domain,discrete_domain);
END_ENTITY;
(*
```

5.6.27 lower_bounded_composite_discrete_domain

A **lower_bounded_composite_discrete_domain** is a **composite_discrete_domain** with a lower bound.

EXPRESS-specification:

```
*)
ENTITY lower_bounded_composite_discrete_domain
SUBTYPE OF (lower_bouded_domain, composite_discrete_domain);
END_ENTITY;
(*
```

5.6.28 upper_bounded_composite_discrete_domain

An **upper_bounded_composite_discrete_domain** is a **composite_discrete_domain** with an upper bound.

EXPRESS-specification:

```
*)
ENTITY upper_bounded_composite_discrete_domain
SUBTYPE OF (upper_bouded_domain, composite_discrete_domain);
END_ENTITY;
(*
```

5.6.29 both_sides_bounded_composite_discrete_domain

A **both_sides_bounded_composite_discrete_domain** is a **composite_discrete_domain** with an upper bound and a lower bound.

EXPRESS-specification:

```
*)
ENTITY both_sides_bounded_composite_discrete_domain
SUBTYPE OF (both_sides_bouded_domain, composite_discrete_domain);
END_ENTITY;
(*
```

5.6.30 domain_point

A **domain_point** is a domain element for which any measure of it in the domain is zero.

NOTE a **domain_point** is an identification of a something within a **domain**. The measure of the domain_point is zero but its position is measured applying the same method of measure

EXPRESS-specification:

```
*)
ENTITY domain_point;
    unit_of_measure: unit;
END_ENTITY;
(*
```

Attribute definitions:

unit_of_measure: the **unit** the **domain_point** is positioned with within the **domain**.

```
END_SCHEMA;       -- domain_property_schema
(*
```

6 Time domain

6.1 Preamble

The time schema provides the definition of concepts related to the time representation, needed by software applications mainly dealing with scheduling and manufacturing management operations. It enables multiple representations of time domains, intervals of time, points in time and time units.

For the management of industrial manufacturing systems an assignment of a point in time to an event occurrence is necessary to enable the observation and comparison of points in time characteristic of the system.

Since the time domain and all its related sub-categories only define different categories of sets of points, another entity is needed to enable the consideration of the length, the duration of a period in time: this concept is provided by the entity interval of time, whose duration is given by the type of value of point or interval of time entity.

The concept of frequency of events is an important feature of this model, since it enables the characterisation of the intervals of time separating event occurrences.

Transformation rules between two time domains are provided through the entity time domain relation. These rules can be expressed either in terms of time unit or in terms of change of origin (translation).

6.2 Fundamental concepts and assumptions

Time is defined by a time domain containing a sequence of points in time. A point in time is defined by a selected location on the time axis, through the use of a time unit. A time unit is used to measure the duration in the related time domain.

A time domain is defined as follows:

$(T,\leq)$ with T = set of points in time,
$\leq$ = complete order-relation on T.

This definition allows for both continuous and time domains considered by this standard.

EXAMPLE 1: A continuous time domain can be described by:

time (R+) = time (R+,≤). In this example the measure of the position of point in time according to the time unit defined are real positive numbers.

EXAMPLE 2: A discrete time domain can be described by:

time (N_0) = time (N_0,≤) . In this example the positions of point in time according to the time unit defined are integers.

Transformation rules are necessary to compare points in time in different time domains.

NOTE ISO DIS 15926-2 proposes in clause 4.5 space-time extensions such as states, periods of time, points in time and events. But these individuals seems to be restricted to continuous domains. In particular periods of time of ISO 15926-2 seems to be fully compatible with continuous time domains of ISO 15531-42 but not with its discrete time domains. With this restriction, if duration, that is used in clause 4.5 of ISO DIS 15926-2 but not defined, is the measure of the periods of time extension, the two concepts of point in time are fully compatible (even equivallent) in ISO 15926-2 and ISO 15531-42 in a continuous time domain. Then the time related space-time extensions proposed in ISO 15926-2 are not compatible with the full time model proposed in ISO 15531-42. They are not compatible with any of its discrete or composite time domain but only with continuous time domain (only if duration may be considered as a measure of periods of time extension). In addition there is no event defined in ISO 15531-42.

6.3 Time domain schema definition

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

EXPRESS specification

```
*)
SCHEMA time_domain_schema;

REFERENCE FROM domain_property schema      -- ISO 15531-42
(time_domain);
(*
```

NOTE 1 The schema referred to above can be found in the clause 4 of this part.

```
*)
REFERENCE FROM date_time_schema            -- ISO 10303-41
(event_occurrence, time_interval);
REFERENCE FROM measure_schema              -- ISO 10303-41
(time_unit);
REFERENCE FROM support_resource_schema     -- ISO 10303-41
(label,text);
 (*
```

NOTE 2 The schemas referred to above can be found in the following parts of ISO 10303-41:

date_time_schema:	clause 16
measure_schema:	clause 21
support_resource_schema:	clause 20

NOTE 3 See annexes D figures D.2 and D.3 for a graphical representation of this schema using the EXPRESS-G notation.

6.4 Time domain type definitions

6.4.1 type_of_description_of_tranformation_rule

A **type_of_description_of_transformation_rule** is the type which may be used to describe any rule of transformation.

EXPRESS-specification:

```
*)
TYPE type_of_description_of_transformation_rule = STRING;
END_TYPE;
(*
```

6.4.2 type_of_value_of_point_or_interval_in_time

A **type_of_value_of_point_or_interval_in_time** is the type which may be used to describe the position of a point or the value of an interval in a continuous or discrete time domain.

EXPRESS-specification:

```
*)
TYPE type_of_value_of_point_or_interval_in_time = REAL;
END_TYPE;
(*
```

6.5 Time domain entity definitions

6.5.1 interval_of_time

An **interval_of_time** refers to a length of a **time_domain**, measured in an unit of time.

EXPRESS-specification:

```
*)
ENTITY interval_of_time;
    SUBTYPE OF (time_interval);   -- generic type from 10303-41
    unit_of_measure: time_unit;
    duration_of_period_in_time:
    type_of_value_of_point_or_interval_in_time;
    of_time_domain: time_domain;
END_ENTITY;
(*
```

Attribute definitions:

unit_of_measure: the **time_unit** the **interval_of_time** is measured in within a particular **time_domain.**
duration_of_period_in_time: value of the length of the **interval_of_time** related to the unit of measure.
of_time_domain: the **time_domain** the **interval_of_time** is specified in.

6.5.2 point_in_time

A **point_in_time** is the location of something noticeable within a **time_domain**.

EXPRESS-specification:

```
*)
ENTITY point_in_time;
    unit_of_measure: time_unit;
    value_of_point_in_time:
    type_of_value_of_point_or_interval_in_time;
    of_time_domain: time_domain;
END_ENTITY;
(*
```

Attribute definitions:

unit_of_measure: the **time_unit** the position of the **point_in_time** is measured in**.**
value_of_point_in_time: position of the **point_in_time** related to the unit of measure.
of_time_domain: the **time_domain** the **point_in_time** is specified in.

6.5.3 frequency_of_event

A **frequency_of_event** defines the frequency of occurring of a given event within a given period of time.

EXPRESS-specification:

```
*)
ENTITY frequency_of_event;
    unit_of_measure: string;
    based_upon: LIST OF point_in_time_event_assignment;
    related_to: event_occurrence;

DERIVE
    frequ_event: REAL:= frequency(SELF);

END_ENTITY;
(*
```

Attribute definitions:

unit_of_measure: the list of **time_domain**s the **composite_time_domain** is composed of.
based_on **: the list of** point_in_time_event_assignment**s used to count the number of** event_occurrence**s**
related_to : the **event_occurrence** whose frequency is calculated

6.5.4 time_domain_relation

A **time_domain_relation** describes the relationship between two **time_domain**s.

EXPRESS-specification:

```
*)
ENTITY time_domain_relation;
    related_time_domain: time_domain;
    relating_time_domain: time_domain;
    origin_relation: rule_of_reference_to_origin_relation;
    reference_to_unit_relation: OPTIONAL rule_of_unit_relation;
END_ENTITY;
(*
```

Attribute definitions:

related_time_domain: one **time_domain** in the relationship.
relating_time_domain: the other **time_domain** in the relationship.
origin_relation: rule of origin transformation.
reference_to_unit_relation: rule of unit transformation.

6.5.5 rule_of_reference_to_origin_relation

A **rule_of_reference_to_origin_relation** describes the transformation rule between the origins of two **time_domain**s.

EXPRESS-specification:

```
*)
ENTITY rule_of_reference_to_origin_relation;
    contents_of_rule: type_of_description_of_transformation_rule;
END_ENTITY;
(*
```

Attribute definitions:

contents_of_rule: contents of the description of the transformation rule.

6.5.6 rule_of_unit_relation

A **rule_of_unit_relation** describes the transformation rule between two time units.

EXPRESS-specification:

```
*)
ENTITY rule_of_unit_relation;
    contents_of_rule: type_of_description_of_transformation_rule;
END_ENTITY;
(*
```

Attribute definitions:

contents_of_rule: contents of the description of the transformation rule.

6.5.7 time_unit_relation

A **time_unit_relation** describes the relationship between two time units.

EXPRESS-specification:

```
*)
ENTITY time_unit_relation;
    relating_time_unit: time_unit;
    related_time_unit: time_unit;
    reference_to_unit_relation: rule_of_unit_relation;
END_ENTITY;
(*
```

Attribute definitions:
relating_time_unit: one **time_unit** in the relationship.
related_time_unit: the other **time_unit** in the relationship.
reference_to_unit_relation: transformation rule for the unit.

6.5.8 point_in_time_event_assignment

A **point_in_time_event_assignment** defines the association of an assignment between **point_in_time** and **event_occurrence** entities.

EXPRESS-specification:

```
*)
ENTITY point_in_time_event_assignment;
    assigned_1_to: event_occurrence;
            -- event_occurrence: generic entity from 10303-41
    assigned_2_to: point_in_time;
    role: point_in_time_event_ass_role;
END_ENTITY;
(*
```

Attribute definitions:

assigned_1_to: the **event_occurrence** the **point_in_time_event_assignment** is related to.
assigned_2_to: the **point_in_time** the **point_in_time_event_assignment** is related to.
role: role defined by the **point_in_time_event_ass_role** entity.

6.5.9 point_in_time_event_assignment_role

A **point_in_time_event_assignment_role** defines the role of the **point_in_time_event_assignment** entity.

EXPRESS-specification:

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

Attribute definitions:

name: the name of the role defined by the **point_in_time_event_assignment_role** entity.
description: plain description of the role.

6.6 Time domain function definition

The **frequency** function determines the number of occurrences of a given event within a given period of time.

The function returns a REAL which is the frequency of an event.

EXPRESS-specification:

```
*)
FUNCTION frequency
    (event: date_time_schema;
    event_occurrence.id: string;
    time_meas_unit:point_in_time;
    unit_of_measure: unit;
    limit_points: SET [0:2] OF assigned_2_to):REAL;

    LOCAL
        inf: REAL;
        sup: REAL;
        time_dist; REAL;
        a:SET OF assigned_1_to:= event;
        b: SET OF assigned_2_to;
        count: INTEGER;
        num_event: INTEGER;
    END_LOCAL;

    inf := LOBOUND OF (limit_points);
    sup := HIBOUND OF (limit_points);
    time_dist := sup – inf;
    count := 0;

    REPEAT I := 1 TO HIINDEX(a);
        IF (b > inf) AND (b < sup)
        THEN count := count + 1;
        END_IF;
    END_REPEAT;

    num_event := count;
    frequ_event := num_event / time_dist;

RETURN (frequ_event);
END_FUNCTION;
(*
```

Argument definitions:

event: (input) the **event_occurrence** for which the frequency is calculated. This is an input of the function.

time_meas_unit: (input) the unit used to express the time element of the frequency. This is an input of the function.

limit_points: (input): the two **point_in_time** between which the **event_occurrence**s are counted. This is an input of the function.

EXPRESS-specification:

```
END_SCHEMA;      -- time_domain_schema
(*
```

Annex A
(normative)

Use of ASN.1 Identifiers in SC4 standards

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

```
iso standard 15531 part 42 version 1
```

is assigned to this part of ISO 15531. The meaning of this value is defined in ISO/IEC 8824-1 and is described in ISO 15531-1.

Annex B
(informative)
EXPRESS listing

This annex provides a listing of the EXPRESS specified in this Part of ISO 15531. No text or annotation is included.

This annex is provided only in computer-interpretable form.

```
(*
    JWG 8 N438                                          2005-01-16
    EXPRESS Declarations for ISO 15531-42
*)
SCHEMA domain_property_schema;

REFERENCE FROM measure_schema         -- ISO 10303-41
(unit);
REFERENCE FROM support_resource_schema     -- ISO 10303-41
(label);

TYPE type_of_identifier_of_dom_gen_struct = STRING;
END_TYPE;
TYPE type_of_domain = STRING;
END_TYPE;

SUBTYPE_CONSTRAINT covered_range FOR domain;
    ABSTRACT SUPERTYPE;
ONEOF (time_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT complcompos_discrcont_bounded FOR
domain_generic_structure;
    ABSTRACT SUPERTYPE;
ONEOF (complete_or_composite_domain, discrete_or_continuous_domain,
bounded_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT complete_composite FOR
complete_or_composite_domain;
    ABSTRACT SUPERTYPE;
ONEOF (complete_domain, composite_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT             discrete_continuous              FOR
discrete_or_continuous_domain;
    ABSTRACT SUPERTYPE;
ONEOF (continuous_domain, discrete_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT lower_upper FOR bounded_domain;
    ABSTRACT SUPERTYPE;
ONEOF (lower_bounded_domain, upper_bounded_domain);
END_SUBTYPE_CONSTRAINT;
```

```
SUBTYPE_CONSTRAINT compl_continuous_discrete FOR complete_domain;
    ABSTRACT SUPERTYPE;
ONEOF (complete_continuous_domain, complete_discrete_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT compos_continuous_discrete FOR composite_domain;
    ABSTRACT SUPERTYPE;
ONEOF (composite_continuous_domain, composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT cont_complete_composite FOR continuous_domain;
    ABSTRACT SUPERTYPE;
ONEOF (complete_continuous_domain, composite_continuous_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT disc_complete_composite FOR discrete_domain;
    ABSTRACT SUPERTYPE;
ONEOF (complete_discrete_domain, composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT both_bd_compl_compos_disc_cont FOR
both_sides_bounded_domain;
    ABSTRACT SUPERTYPE;
ONEOF (both_sides_bounded_complete_continuous_domain,
both_sides_bounded_complete_discrete_domain,
both_sides_bounded_composite_continuous_domain,
both_sides_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT         low_bd_compl_compos_disc_cont         FOR
lower_bounded_domain;
    ABSTRACT SUPERTYPE;
ONEOF (lower_bounded_complete_continuous_domain,
lower_bounded_complete_discrete_domain,
lower_bounded_composite_continuous_domain,
lower_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT         up_bd_compl_compos_disc_cont         FOR
upper_bounded_domain;
    ABSTRACT SUPERTYPE;
ONEOF (upper_bounded_complete_continuous_domain,
upper_bounded_complete_discrete_domain,
upper_bounded_composite_continuous_domain,
upper_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT            compl_disc_low_both_up            FOR
complete_discrete_domain;
ONEOF (lower_bounded_complete_discrete_domain,
both_sides_bounded_complete_discrete_domain,
upper_bounded_complete_discrete_domain);
END_SUBTYPE_CONSTRAINT;
```

```
SUBTYPE_CONSTRAINT            compl_cont_low_both_up            FOR
complete_continuous_domain;
ONEOF (lower_bounded_complete_continuous_domain,
both_sides_bounded_complete_continuous_domain,
upper_bounded_complete_continuous_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT            compos_cont_low_both_up           FOR
composite_continuous_domain;
ONEOF (lower_bounded_composite_continuous_domain,
both_sides_bounded_composite_continuous_domain,
upper_bounded_composite_continuous_domain);
END_SUBTYPE_CONSTRAINT;

SUBTYPE_CONSTRAINT            compos_disc_low_both_up           FOR
composite_discrete_domain;
ONEOF (lower_bounded_composite_discrete_domain,
both_sides_bounded_composite_discrete_domain,
upper_bounded_composite_discrete_domain);
END_SUBTYPE_CONSTRAINT;

ENTITY domain;
id: type_of_domain;
domain_structure:domain_generic_structure;
UNIQUE
UR1: id;
END_ENTITY;

ENTITY time_domain
SUBTYPE OF (domain);
name: label;
decomposes_into: SET [0:?] OF time_domain;
END_ENTITY;

ENTITY domain_generic_structure;
id: type_of_identifier_of_dom_gen_struct;
UNIQUE
UR1: id;
END_ENTITY;

ENTITY complete_or_composite_domain
SUBTYPE OF (domain_generic_structure);
END_ENTITY;

ENTITY discrete_or_continuous_domain
SUBTYPE OF (domain_generic_structure);
END_ENTITY;

ENTITY bounded_domain
SUBTYPE OF (domain_generic_structure);
END_ENTITY;

ENTITY complete_domain
```

```
SUBTYPE OF (complete_or_composite_domain);
reference_to_unit: unit;
reference_to_origin: domain_point;
END_ENTITY;

ENTITY composite_domain
SUBTYPE OF (complete_or_composite_domain);
components: SET[0:?] OF domain;
WHERE     -- components do not overlap
    WR1 : (components (domain.id [1]) <>
        components (domain.id [2]));
END_ENTITY;

ENTITY continuous_domain
SUBTYPE OF (discrete_or_continuous_domain);
END_ENTITY;

ENTITY discrete_domain
SUBTYPE OF (discrete_or_continuous_domain);
END_ENTITY;

ENTITY lower_bounded_domain
SUBTYPE OF (bounded_domain);
lower_bound: domain_point;
END_ENTITY;

ENTITY upper_bounded_domain
SUBTYPE OF (bounded_domain);
upper_bound: domain_point;
END_ENTITY;

ENTITY both_sides_bounded_domain;
    low_bound_assigned_to: lower_bounded_domain;
    up_bound_assigned_to: upper_bounded_domain;
END_ENTITY;

ENTITY complete_continuous_domain
SUBTYPE OF (complete_domain, continuous_domain);
END_ENTITY;

ENTITY lower_bounded_complete_continuous_domain
SUBTYPE OF (lower_bounded_domain, complete_continuous_domain);
END_ENTITY;

ENTITY upper_bounded_complete_continuous_domain
SUBTYPE OF (upper_bounded_domain, complete_continuous_domain);
END_ENTITY;

ENTITY both_sides_bounded_complete_continuous_domain
SUBTYPE OF (both_sides_bounded_domain, complete_continuous_domain);
END_ENTITY;

ENTITY composite_continuous_domain
SUBTYPE OF (composite_domain, continuous_domain);
```

```
END_ENTITY;

ENTITY lower_bounded_composite_continuous_domain
SUBTYPE OF (lower_bounded_domain, composite_continuous_domain);
END_ENTITY;

ENTITY upper_bounded_composite_continuous_domain
SUBTYPE OF (upper_bounded_domain, composite_continuous_domain);
END_ENTITY;

ENTITY both_sides_bounded_composite_continuous_domain
SUBTYPE OF (both_sides_bounded_domain, composite_continuous_domain);
END_ENTITY;

ENTITY complete_discrete_domain
SUBTYPE OF (complete_domain, discrete_domain);
END_ENTITY;

ENTITY lower_bounded_complete_discrete_domain
SUBTYPE OF (lower_bounded_domain, complete_discrete_domain);
END_ENTITY;

ENTITY upper_bounded_complete_discrete_domain
SUBTYPE OF (upper_bounded_domain, complete_discrete_domain);
END_ENTITY;

ENTITY both_sides_bounded_complete_discrete_domain
SUBTYPE OF (both_sides_bounded_domain, complete_discrete_domain);
END_ENTITY;

ENTITY composite_discrete_domain
SUBTYPE OF (composite_domain, discrete_domain);
END_ENTITY;

ENTITY lower_bounded_composite_discrete_domain
SUBTYPE OF (lower_bounded_domain, composite_discrete_domain);
END_ENTITY;

ENTITY upper_bounded_composite_discrete_domain
SUBTYPE OF (upper_bounded_domain, composite_discrete_domain);
END_ENTITY;

ENTITY both_sides_bounded_composite_discrete_domain
SUBTYPE OF (both_sides_bounded_domain, composite_discrete_domain);
END_ENTITY;

ENTITY domain_point;
unit_of_measure: unit;
END_ENTITY;

END_SCHEMA;      -- domain_property_schema


SCHEMA time_domain_schema;
```

```
REFERENCE FROM domain_property_schema      -- ISO 15531-42
(time_domain);
REFERENCE FROM date_time_schema            -- ISO 10303-41
(event_occurrence, time_interval);
REFERENCE FROM measure_schema          -- ISO 10303-41
(time_unit);
REFERENCE FROM support_resource_schema -- ISO 10303-41
(label,text);

TYPE type_of_description_of_transformation_rule = STRING;
END_TYPE;
TYPE type_of_value_of_point_or_interval_in_time = REAL;
END_TYPE;

ENTITY interval_of_time
    SUBTYPE OF (time_interval);   -- generic type from 10303-41
unit_of_measure: time_unit;
duration_of_period_in_time:
type_of_value_of_point_or_interval_in_time;
of_time_domain: time_domain;
END_ENTITY;

ENTITY point_in_time;
unit_of_measure: time_unit;
value_of_point_in_time: type_of_value_of_point_or_interval_in_time;
of_time_domain: time_domain;
END_ENTITY;

ENTITY frequence_of_event;
unit_of_measure : STRING;
related_to: event_occurrence;
based_upon : LIST OF point_in_time_event_assignment;
DERIVE
    frequ_event: REAL:= frequency(SELF);
END_ENTITY;

ENTITY time_domain_relation;
related_time_domain: time_domain;
relating_time_domain: time_domain;
origin_relation: rule_of_reference_to_origin_relation;
reference_to_unit_relation: OPTIONAL rule_of_unit_relation;
END_ENTITY;

ENTITY rule_of_reference_to_origin_relation;
contents_of_rule: type_of_description_of_transformation_rule;
END_ENTITY;

ENTITY rule_of_unit_relation;
contents_of_rule: type_of_description_of_transformation_rule;
END_ENTITY;

ENTITY time_unit_relation;
relating_time_unit: time_unit;
```

```
related_time_unit: time_unit;
reference_to_unit_relation: rule_of_unit_relation;
END_ENTITY;

ENTITY point_in_time_event_assignment;
assigned_2_to: point_in_time;
assigned_1_to: event_occurrence; -- event_occurrence generic entity
from 10303-41
role: point_in_time_event_assignment_role;
END_ENTITY;

ENTITY point_in_time_event_assignment_role;
name: label;
description: text;
END_ENTITY;

FUNCTION frequency
(event: event_occurrence;
id: string;
time_meas_unit:point_in_time;
unit_of_measure: unit;
limit_points: SET [0:2] OF assigned_2_to):REAL;
LOCAL
        inf: REAL;
        sup: REAL;
        time_dist: REAL;
a:SET OF assigned_1_to := event;
b:SET OF assigned_2_to ;
count: INTEGER;
num_event: INTEGER;
END_LOCAL;
inf := LOBOUND (limit_points);
sup := HIBOUND (limit_points);
time_dist := sup - inf;
count := 0;

REPEAT I := 1 TO HIINDEX(a);
    IF (b > inf) AND (b < sup)
    THEN count := count + 1;
    END_IF;
END_REPEAT;

num_event := count;
frequ_event := num_event / time_dist;

RETURN (frequ_event);
END_FUNCTION;

END_SCHEMA;      -- time_domain_schema
```

Annex C
(informative)
EXPRESS-G figures

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

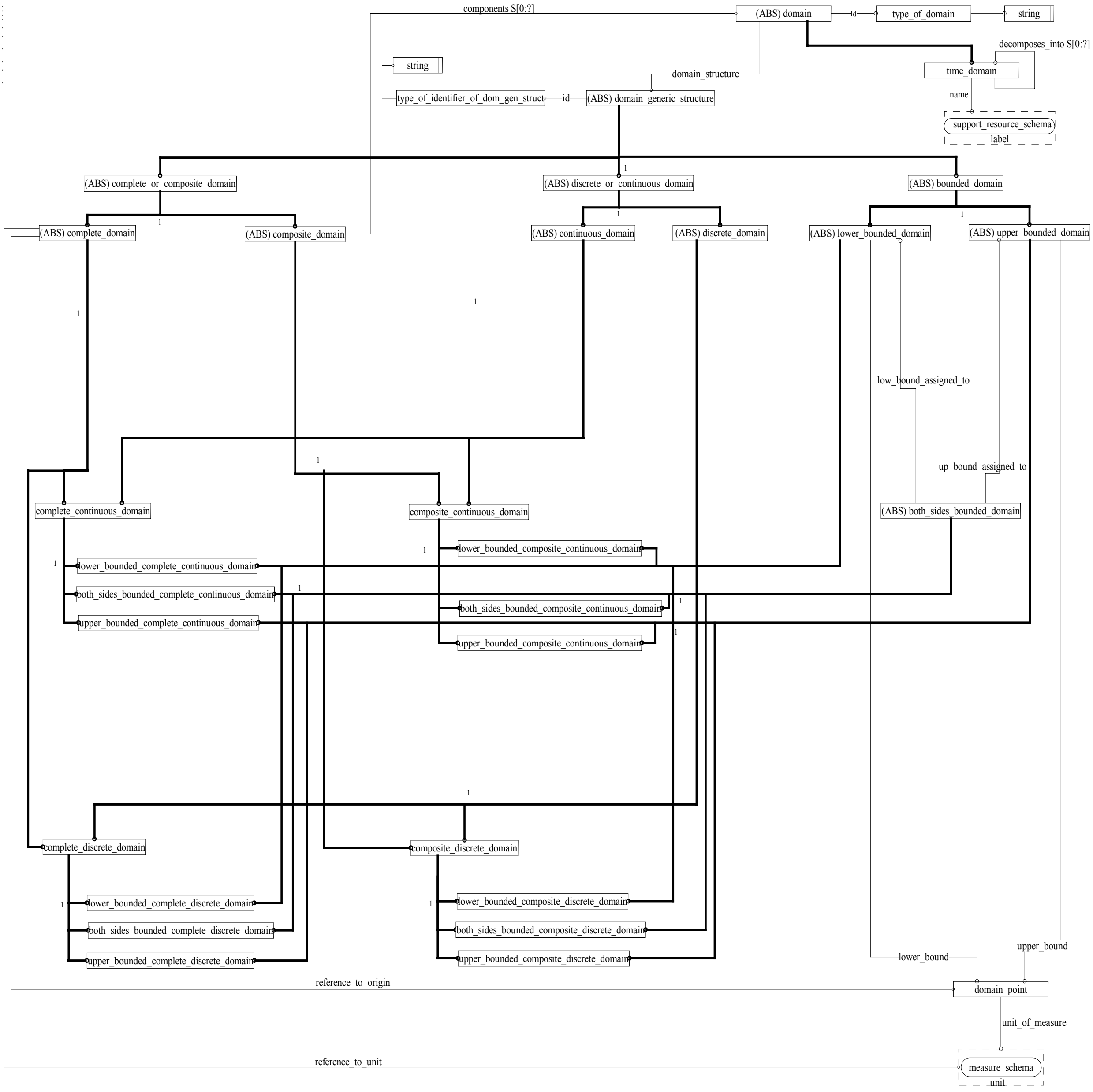
components S[0:?]
(ABS) domain
id
type_of_domain
string
decomposes_into S[0:?]
string
domain_structure
time_domain
type_of_identifier_of_dom_gen_struct
id
(ABS) domain_generic_structure
name
support_resource_schema
label
(ABS) complete_or_composite_domain
(ABS) discrete_or_continuous_domain
(ABS) bounded_domain
(ABS) complete_domain
(ABS) composite_domain
(ABS) continuous_domain
(ABS) discrete_domain
(ABS) lower_bounded_domain
(ABS) upper_bounded_domain
low_bound_assigned_to
up_bound_assigned_to
complete_continuous_domain
composite_continuous_domain
(ABS) both_sides_bounded_domain
lower_bounded_composite_continuous_domain
lower_bounded_complete_continuous_domain
both_sides_bounded_complete_continuous_domain
both_sides_bounded_composite_continuous_domain
upper_bounded_complete_continuous_domain
upper_bounded_composite_continuous_domain
complete_discrete_domain
composite_discrete_domain
lower_bounded_complete_discrete_domain
lower_bounded_composite_discrete_domain
both_sides_bounded_complete_discrete_domain
both_sides_bounded_composite_discrete_domain
upper_bounded_complete_discrete_domain
upper_bounded_composite_discrete_domain
lower_bound
upper_bound
reference_to_origin
domain_point
unit_of_measure
reference_to_unit
measure_schema
unit

Figure C.1 domain_property schema

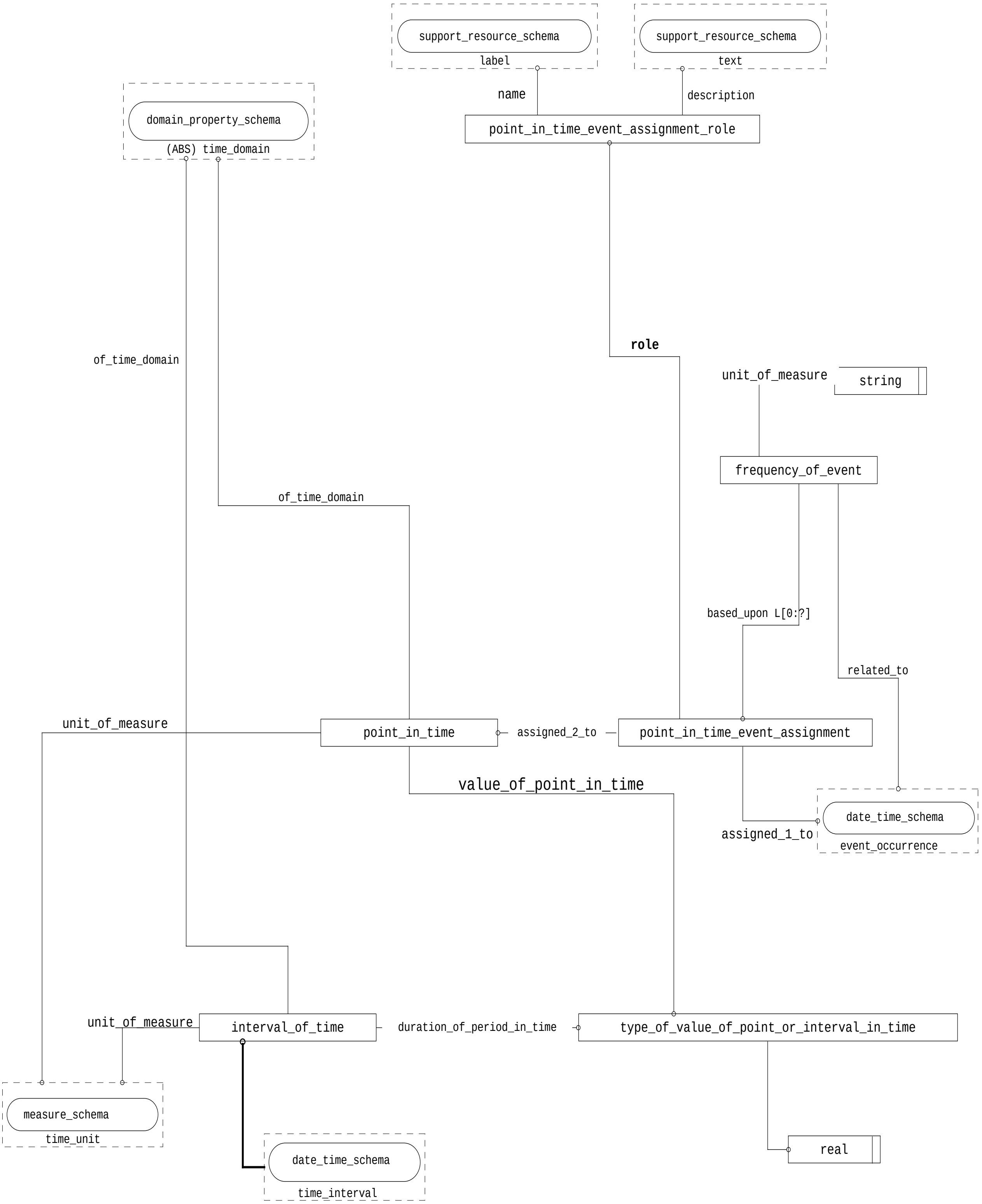

Figure C.2 - time_domain schema (1/2)

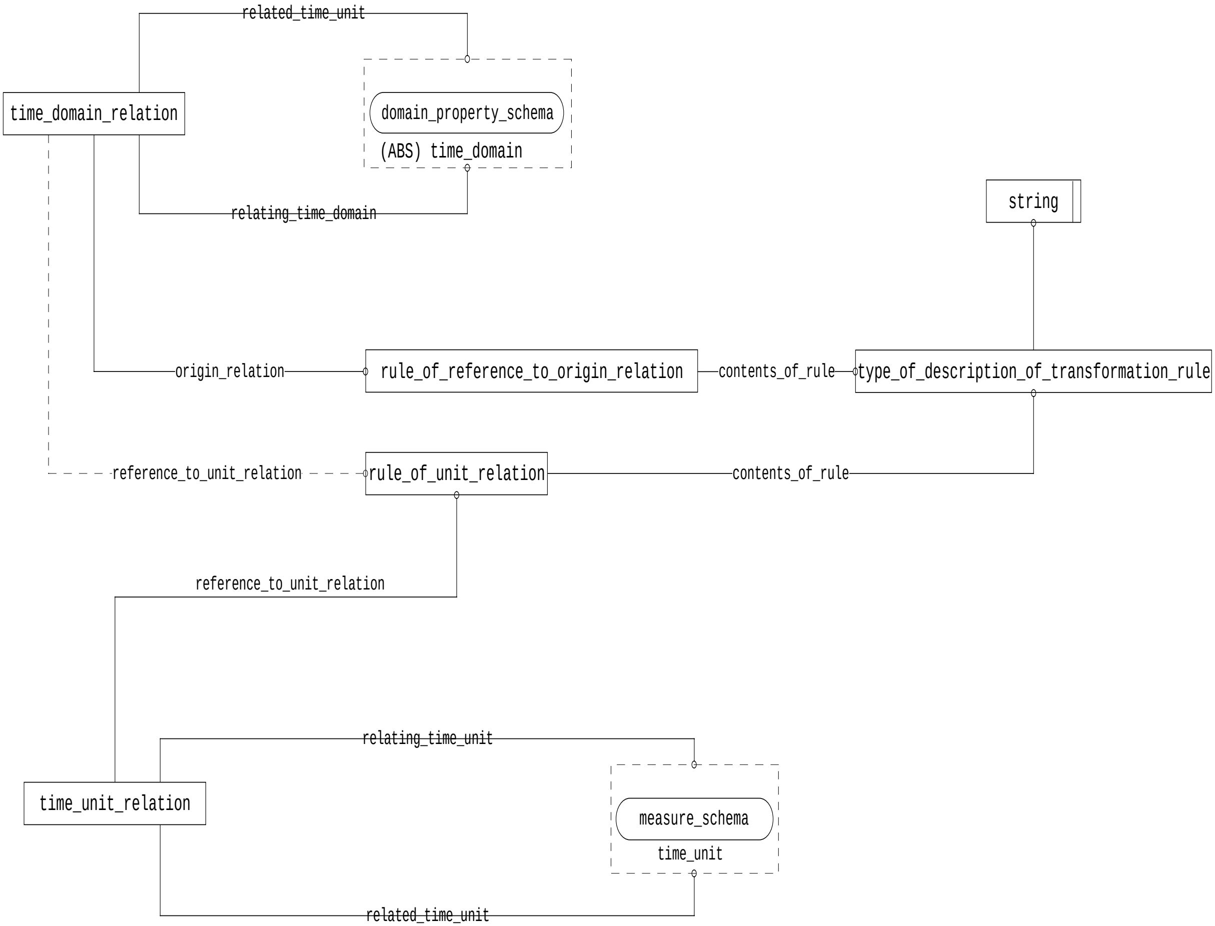

Figure C.3 time_domain schema (2/2)

Bibliography

— ISO 8879:1986, *Information processing - Text and office systems - Standard Generalized Markup Language (SGML)*

— IEC 62264-1:2003 *Enterprise-control system integration-Part 1: Models and terminology*

— ISO/IEC 2382-24:1995, *Information technology - Vocabulary - Part 24: Computer-integrated manufacturing*

— ISO 15531-31:2004, *Industrial automation systems and integration - Industrial manufacturing management data - Part 31: Resource information model*

— ISO 15926-2:2003, *Industrial automation systems and integration - Integration of life-cycle data for process plants including oil and gas facilities - Part 2: Data model*

Index

ICS 25.040.40

Price based on 43 pages