INTERNATIONAL STANDARD

ISO/IEC 10746-3

Second edition
2009-12-15

Information technology — Open distributed processing — Reference model: Architecture

Technologies de l'information — Traitement réparti ouvert — Modèle de référence: Architecture

Reference number
ISO/IEC 10746-3:2009(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 by ISO in 2010

Published in Switzerland

CONTENTS

Page

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

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

ISO/IEC 10746-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in collaboration with ITU-T. The identical text is published as Rec. ITU-T X.903 (10/2009).

This second edition cancels and replaces the first edition (ISO/IEC 10746-3:1996), which has been technically revised.

ISO/IEC 10746 consists of the following parts, under the general title *Information technology — Open distributed processing — Reference model*:

— *Part 1: Overview*

— *Part 2: Foundations*

— *Part 3: Architecture*

— *Part 4: Architectural semantics*

Introduction

The rapid growth of distributed processing has led to a need for a coordinating framework for the standardization of open distributed processing (ODP). This reference model provides such a framework. It creates an architecture within which support of distribution, interworking and portability can be integrated.

The reference model of open distributed processing, Recommendations ITU-T X.901 | ISO/IEC 10746-1 to X.904 | ISO/IEC 10746-4, is based on precise concepts derived from current distributed processing developments and, as far as possible, on the use of formal description techniques for specification of the architecture.

The reference model consists of:

– Recommendation ITU-T X.901 | ISO/IEC 10746-1: Overview: Contains a motivational overview of ODP giving scoping, justification and explanation of key concepts, and an outline of the ODP architecture. It contains explanatory material on how this reference model is to be interpreted and applied by its users, who may include standards writers and architects of ODP systems. It also contains a categorization of required areas of standardization expressed in terms of the reference points for conformance identified in this Recommendation | International Standard. This part is not normative.

– Recommendation ITU-T X.902 | ISO/IEC 10746-2: Foundations: Contains the definition of the concepts and analytical framework for normalized description of (arbitrary) distributed processing systems. It introduces the principles of conformance to ODP standards and the way in which they are applied. This is only to a level of detail sufficient to support this Recommendation | International Standard and to establish requirements for new specification techniques. This part is normative.

– Recommendation ITU-T X.903 | ISO/IEC 10746-3: Architecture: Contains the specification of the required characteristics that qualify distributed processing as open. These are the constraints to which ODP standards must conform. It uses the descriptive techniques from Recommendation ITU-T X.902 | ISO/IEC 10746-2. This part is normative.

– Recommendation ITU-T X.904 | ISO/IEC 10746-4: Architectural semantics: Contains a formalization of the ODP modelling concepts defined in Recommendation ITU-T X.902 | ISO/IEC 10746-2, clauses 8 and 9. The formalization is achieved by interpreting each concept in terms of the constructs of the different standardized formal description techniques. This part is normative.

This Recommendation | International Standard contains two annexes:

– Annex A – Formal computational supertype/subtype rules.

– Annex B – Human-computer interactions

Annex A forms an integral part of the reference model. Annex B does not form an integral part of the reference model.

INTERNATIONAL STANDARD
RECOMMENDATION ITU-T

Information technology – Open distributed processing – Reference model: Architecture

1 Scope

This Recommendation | International Standard:

- defines how ODP systems are specified, making use of concepts in Rec. ITU-T X.902 | ISO/IEC 10746-2;
- identifies the characteristics that qualify systems as ODP systems.

It establishes a framework for coordinating the development of existing and future standards for ODP systems and is provided for reference by those standards.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- Recommendation ITU-T X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The basic model.*
- Recommendation ITU-T X.810 (1995) | ISO/IEC 10181-1:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Overview.*
- Recommendation ITU-T X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework.*
- Recommendation ITU-T X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework.*
- Recommendation ITU-T X.813 (1996) | ISO/IEC 10181-4:1997, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Non-repudiation framework.*
- Recommendation ITU-T X.814 (1995) | ISO/IEC 10181-5:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Confidentiality framework.*
- Recommendation ITU-T X.815 (1995) | ISO/IEC 10181-6:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Integrity framework.*
- Recommendation ITU-T X.816 (1995) | ISO/IEC 10181-7:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Security audit and alarms framework.*
- Recommendation ITU-T X.851 (1997) | ISO/IEC 9804:1998; *Information technology – Open Systems Interconnection – Service definition for the Commitment, Concurrency and Recovery service element.*
- Recommendation ITU-T X.902 (2009) | ISO/IEC 10746-2:2010, *Information technology – Open distributed processing – Reference model: Foundations.*
- ISO/IEC 11770-1:1996, *Information technology – Security techniques – Key management – Part 1: Framework.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications.*

  ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture.*

3 Definitions

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

3.1 Descriptive definitions

This reference model makes use of the following term defined in Rec. ITU-T X.200 | ISO/IEC 7498-1:

- transfer syntax.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.810 | ISO/IEC 10181-1:

- trusted third party.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.811 | ISO/IEC 10181-2:

- claimant;
- exchange authentication information;
- principal.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.812 | ISO/IEC 10181-3:

- access control information;
- access control decision function;
- access control enforcement function;
- initiator;
- target.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.813 | ISO/IEC 10181-4:

- evidence generator;
- evidence user;
- evidence verifier;
- (non-repudiable data) originator;
- (non-repudiable data) recipient;
- non-repudiation evidence;
- non-repudiation service requester;
- notary;
- originator;
- recipient.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.814 | ISO/IEC 10181-5:

- confidentiality-protected data;
- hide;
- reveal.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.815 | ISO/IEC 10181-6:

- integrity-protected data;
- shield;
- validate.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.816 | ISO/IEC 10181-7:

- audit recorder function;
- audit trail examiner function;
- audit trail collector function.

This Recommendation | International Standard makes use of the following terms defined in Rec. ITU-T X.851 | ISO/IEC 9804:

- atomicity;
- consistency;
- durability;
- isolation.

This Recommendation | International Standard makes use of the following terms defined in ISO/IEC 11770-1 Key Management Framework:

- certification authority;
- key distribution centre;
- key management;
- key translation centre;
- public key.

This reference model makes use of the terms defined in Rec. ITU-T X.902 | ISO/IEC 10746-2, shown in Figure 1.

3.2 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

ACID Atomicity, Consistency, Isolation and Durability

ODP Open Distributed Processing

OSI Open Systems Interconnection

4 Framework

This reference model defines a framework comprising:

- five *viewpoints*, called enterprise, information, computational, engineering and technology which provide a basis for the specification of ODP systems;
- a *viewpoint language* for each viewpoint, defining concepts and rules for specifying ODP systems from the corresponding viewpoint;
- specifications of the *functions* required to support ODP systems;
- *transparency prescriptions* showing how to use the ODP functions to achieve distribution transparency.

The architecture for ODP systems and the composition of functions is determined by the combination of the computational language, the engineering language and the transparency prescriptions.

abstraction;	fault;	persistence;
action;	factory;	policy;
action signature:	<x> group;	producer object;
activity;	identifier;	programmatic reference point;
architecture;	information;	prohibition;
atomicity;	initiating object;	Quality of Service;
behaviour;	instance;	reference point;
binding;	instantiation;	refinement;
class;	interaction;	role;
client object;	interchange reference point;	rule;
communication;	interface;	server object;
communication management;	interface signature;	spawn action;
composition;	interworking reference point;	specification;
component;	introduction;	stability;
configuration;	invariant;	state;
conformance point;	liaison;	subdomain;
consumer object;	location in space;	subtype;
contract;	location in time;	supertype;
creation;	name;	system;
data;	naming context;	<x> template;
decomposition;	naming domain;	term;
deletion;	notification;	thread;
distributed processing;	object;	trading;
distribution transparency;	obligation;	type;
<x> domain;	ODP standards;	viewpoint.
entity;	ODP system;	
environment;	open distributed processing;	
error;	perceptual reference point;	
establishing behaviour;	permission;	
failure;		

Figure 1 – Terms taken from Rec. ITU-T X.902 | ISO/IEC 10746-2

4.1 Viewpoints

4.1.1 Concepts

4.1.1.1 enterprise viewpoint: A viewpoint on an ODP system and its environment that focuses on the purpose, scope and policies for that system.

4.1.1.2 information viewpoint: A viewpoint on an ODP system and its environment that focuses on the semantics of information and information processing.

4.1.1.3 computational viewpoint: A viewpoint on an ODP system and its environment which enables distribution through functional decomposition of the system into objects which interact at interfaces.

4.1.1.4 engineering viewpoint: A viewpoint on an ODP system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system.

4.1.1.5 technology viewpoint: A viewpoint on an ODP system and its environment that focuses on the choice of technology in that system.

4.1.2 Using viewpoints

The enterprise, information, computational, engineering and technology viewpoints have been chosen as a necessary and sufficient set to meet the needs of ODP standards. Viewpoints can be applied, at an appropriate level of abstraction, to a complete ODP system, in which case the environment defines the context in which the ODP system operates. Viewpoints can also be applied to individual components of an ODP system, in which case the component's environment will include some abstraction of both the system's environment and other system components.

NOTE – The process of abstraction might be such that the system's environment and the other system components are composed into a single object.

4.2 ODP viewpoint languages

4.2.1 Concepts

4.2.1.1 <Viewpoint> language: Definitions of concepts and rules for the specification of an ODP system from the <viewpoint> viewpoint; thus: **engineering language**: definitions of concepts and rules for the specification of an ODP system from the engineering viewpoint.

4.2.2 Using viewpoint languages

This reference model defines a set of five languages, each corresponding to one of the viewpoints defined in 4.1.1. Each language is used for the specification of an ODP system from the corresponding viewpoint. These languages are:

– the enterprise language (defined in clause 5);

– the information language (defined in clause 6);

– the computational language (defined in clause 7);

– the engineering language (defined in clause 8);

– the technology language (defined in clause 9).

Each language uses concepts taken from Rec. ITU-T X.902 | ISO/IEC 10746-2, and introduces refinements of those concepts, prescriptive rules and additional viewpoint-specific concepts relevant to the nature of the specifications concerned. These additional concepts are, in turn, defined using concepts from Rec. ITU-T X.902 | ISO/IEC 10746-2.

A system specification comprises one or more viewpoint specifications. These specifications must be mutually consistent. Rules for the consistent structuring of viewpoint specifications are given in clause 10. The specifier must demonstrate by other means that terms in the specifications are used consistently. A specification of a system using several viewpoint specifications will often restrict implementations more than a specification using fewer viewpoint specifications. Objects identified in one viewpoint can be specified using the viewpoint language associated with that viewpoint or using the viewpoint languages associated with other viewpoints. It is not necessary to specify an object fully from every viewpoint in order to achieve a mutually consistent set of viewpoint specifications.

NOTE 1 – The qualification of a term from Rec. ITU-T X.902 | ISO/IEC 10746-2 by the name of a viewpoint (e.g., as in "**computational** object") is interpreted as using the term from Rec. ITU-T X.902 | ISO/IEC 10746-2, subject to whatever additional provisions are specified in the identified viewpoint language.

NOTE 2 – The unqualified use of a term from Rec. ITU-T X.902 | ISO/IEC 10746-2 in a viewpoint specification (e.g., "interface") is interpreted as if the term had been qualified by the name of the viewpoint (i.e., "computational interface"), if the associated viewpoint language places additional constraints on the term.

4.3 ODP functions

4.3.1 ODP function: A function required to support open distributed processing.

4.3.2 Using ODP functions

This reference model specifies, in clauses 11 to 15, the functions required to achieve open distributed processing.

Each ODP function description contains:

– an explanation of the use of the function for open distributed processing;

– prescriptive statements, about the structure and behaviour of the function, sufficient to ensure the overall integrity of the reference model;

– a statement of other ODP functions upon which it depends.

4.4 ODP distribution transparencies

4.4.1 Concepts

4.4.1.1 access transparency: A distribution transparency which masks differences in data representation and invocation mechanisms to enable interworking between objects.

4.4.1.2 failure transparency: A distribution transparency which masks, from an object, the failure and possible recovery of other objects (or itself), to enable fault tolerance.

4.4.1.3 location transparency: A distribution transparency which masks the use of information about location in space when identifying and binding to interfaces.

4.4.1.4 migration transparency: A distribution transparency which masks, from an object, the ability of a system to change the location of that object. Migration is often used to achieve load balancing and reduce latency.

4.4.1.5 relocation transparency: A distribution transparency which masks relocation of an interface from other interfaces bound to it.

4.4.1.6 replication transparency: A distribution transparency which masks the use of a group of mutually behaviourally compatible objects to support an interface. Replication is often used to enhance performance and availability.

4.4.1.7 persistence transparency: A distribution transparency which masks, from an object, the deactivation and reactivation of other objects (or itself). Deactivation and reactivation are often used to maintain the persistence of an object when a system is unable to provide it with processing, storage and communication functions continuously.

4.4.1.8 transaction transparency: A distribution transparency which masks coordination of activities amongst a configuration of objects to achieve consistency.

4.4.2 Using distribution transparency

Distribution transparency is an important end-user requirement in distributed systems. This reference model defines a set of distribution transparencies which make it possible to implement ODP systems which are distribution transparent from the point of view of users of those systems. Distribution transparency is selective; the reference model includes rules for selecting and combining distribution transparencies in ODP systems.

This reference model requires, for each distribution transparency defined in 4.4.1.1 to 4.4.1.8, definitions of both:

- a transparency schema for expressing requirements for the particular transparency – the schema is specified in the enterprise, information computational specifications;
- a refinement process for transforming a specification which contains requirements for the particular distribution transparency to a specification which explicitly realizes the masking implied by that transparency – the refinement process is specified in the engineering specification.

NOTE 1 – In some cases (e.g., access transparency) the schema is null; in others (e.g., transaction transparency) the schema contains one or more parameters dictating the precise form of transparency required.

NOTE 2 – The refinement process typically involves introducing additional behaviour, including the use of one or more ODP functions, into the specification.

The specifications of the refinement processes in clause 16 are prescriptive to the level required to ensure overall integrity of the reference model.

4.5 Standards derived from the framework

This reference model provides a framework for the definition of new standards and the use of existing standards as ODP standards.

ODP standards are any of:

- standards for components of ODP systems;
- standards for composing ODP system components;
- standards for modelling and specifying ODP systems.

ODP standards:

- use the enterprise language to specify policies;
- use the information language to specify consistent use and interpretation of information in, and between, standards;
- use the computational language to specify the configuration and behaviour of interfaces;
- use the engineering language to specify the infrastructures they require;
- use the technology language to specify conformance to international, private, or consensual specifications.

Standards for methodology, modelling, programming, implementation and testing of ODP systems use the framework as a whole.

ODP standards can be based on a subset of this reference model (e.g., by excluding some forms of interaction, particular functions or transparencies). Such standards can also extend this reference model, provided that the extensions they introduce do not change or contradict its provisions. Extensions will relate new terms to terms defined in this reference model: for example, by introducing new types and new type rules.

ODP standards comply with all the prescriptive statements in this reference model.

4.6 Conformance

The enterprise, information, computational and engineering languages are used to specify the conformance requirements for ODP systems. The technology language can be used to assert conformance to ODP standards in ODP systems. Each interface which is defined as a conformance point has an information specification to enable interpretation of interactions of that interface. The rules for identifying conformance points are given in the computational and engineering languages.

An ODP system conforms to an ODP standard if it satisfies the conformance requirements of that standard.

5 Enterprise language

The enterprise language comprises concepts, rules and structures for the specification of an ODP system from the enterprise viewpoint.

An enterprise specification defines the purpose, scope and policies of an ODP system.

In this reference model, prescription in the enterprise viewpoint is restricted to a basic set of concepts and rules addressing the scope and nature of enterprise specifications.

5.1 Concepts

The enterprise language contains the concepts of Rec. ITU-T X.902 | ISO/IEC 10746-2 and those defined here, subject to the rules of 5.2.

5.1.1 **community**: A configuration of objects formed to meet an objective. The objective is expressed in a contract, which expresses how this objective can be met by defining roles and interactions required, assignments of objects to the roles, and policies governing the collective behaviour.

5.1.2 **<X> federation**: A community of <x> domains.

5.2 Structuring rules

An enterprise specification defines, and the enterprise language is able to express, the purpose, scope and policies of an ODP system in terms of each of the following items:

- roles played by the system;
- activities undertaken by the system;
- policy statements about the system, including those relating to environment contracts.

In an enterprise specification, an ODP system and the environment in which it operates are represented as a community. At some level of description the ODP system is represented as an enterprise object in the community. The objectives and scope of the ODP system are defined in terms of the roles it fulfils within the community of which it is part, and policy statements about those roles. A community is defined in terms of each of the following elements:

- the enterprise objects comprising the community;
- the roles fulfilled by each of those objects;
- policies governing interactions between enterprise objects fulfilling roles;
- policies governing the creation, usage and deletion of resources by enterprise objects fulfilling roles;
- policies governing the configuration of enterprise objects and assignment of roles to enterprise objects;
- policies relating to environment contracts governing the system.

NOTE 1 – Enterprise policies may constrain some of the distribution transparencies defined in the computational specifications. For example, an enterprise policy on avoidance of single points of hardware failure might imply a requirement for migration transparency for all computational objects, irrespective of their individual constraints in the computational specification.

A role is defined in terms of the permissions, obligations, prohibitions and behaviour of the enterprise object fulfilling the role. An enterprise object can fulfil one or more roles in a community, and the roles which it can fulfil are determined by the contract on which the community is based. While it is part of one community, the enterprise object can continue to fulfil roles in other communities, subject to the provisions in the contracts of the communities involved. The enterprise object can fulfil different roles in different communities. Interactions between enterprise objects fulfilling appropriate roles within different communities can be considered as interactions between those communities.

NOTE 2 – Examples of roles include policy administrator, president, service provider, owner, manager, shareholder, consumer.

NOTE 3 – Examples of environment contracts in enterprise specifications include safety requirements, legislative requirements and codes of practice.

NOTE 4 – In an enterprise specification the term "<x> object", where <x> is a role, is interpreted as meaning "an enterprise object fulfilling an <x> role"; where an enterprise object fulfils multiple roles, the names can be concatenated, e.g., "owner driver object".

When fulfilling a role, an object becomes subject to permissions, obligations and prohibitions by delegation or transfer. In some roles, objects are permitted to change policy. There are five fundamental types of actions with respect to contractual matters:

– an object incurs an obligation to another object (it must currently be permitted to incur the obligation);

– an object fulfils an obligation to another object;

– an object waives an obligation to another object;

– an object acquires permission from another object to perform some action it was previously forbidden to perform;

– an object is forbidden to perform an action it was previously permitted to perform.

NOTE 5 – An important special case of acquisition is where the permitted action is performative, i.e., when an object in a subordinate role is enabled to issue further permissions or obligations on behalf of an object fulfilling a superior role. This leads to the notion of agency or delegation.

Obligations include accounting and charging for the use of resources. Billing and payment is modelled as the reassignment of resources between objects in accordance with the roles they fulfil.

A resource is either consumable or non-consumable. A consumable resource deletes itself after some amount of use. In <x> federation, the objective defines the resources each <x> domain in the federation shares with the other members of the federation. The objective can leave each domain with a defined degree of autonomy in the use of its own resources. The establishing behaviour for an <x> federation can allow autonomy for each participating <x> domain in deciding whether or not to become part of the federation.

5.3 Conformance and reference points

Conformance statements in the enterprise language require that the behaviour of an ODP system is conformant to a particular set of objectives and policies.

An implementor claiming conformance must identify the engineering reference points which give access to the system and the engineering, computational and information specifications which apply at them. By this act the identified reference points become conformance points. The interactions at these conformance points can then be interpreted in enterprise language terms to check that the enterprise specification is not violated.

Enterprise specifications can be applied to all four classes of reference point (programmatic, perceptual, interworking and interchange reference points) identified in Rec. ITU-T X.902 | ISO/IEC 10746-2.

6 Information language

The information language comprises concepts, rules and structures for the specification of an ODP system from the information viewpoint.

An information specification defines the semantics of information and the semantics of information processing in an ODP system.

In this reference model, prescription in the information viewpoint is restricted to a basic set of concepts and rules addressing the scope and nature of information specifications.

NOTE – The information specification includes the specification of all information held by the systems: however, it does not specify the representation of that information in the system itself, although there is, necessarily, a notation for representation of the information specification. Some representations will be determined by the nature of the platform (e.g., transfer formats); others will be to do with representation for human consumption.

6.1 Concepts

The information language contains the concepts of Rec. ITU-T X.902 | ISO/IEC 10746-2 and those defined here, subject to the rules of 6.2.

6.1.1 invariant schema: A set of predicates on one or more information objects which must always be true. The predicates constrain the possible states and state changes of the objects to which they apply.

NOTE 1 – Thus, an invariant schema is the specification of the types of one or more information objects that will always be satisfied by whatever behaviour the objects might exhibit.

NOTE 2 – An invariant schemata may constrain some of the distribution transparencies defined in the computational specifications. For example, a particular information object type may place a constraint that all the information objects that are instances of it must be accessed using transaction transparency, without the need to so qualify all such interactions in the computational specification.

6.1.2 static schema: A specification of the state of one or more information objects, at some point in time, subject to the constraints of any invariant schemata.

NOTE – Thus, a static schema is the specification of the types of one or more information objects at some particular point in time. These types are subtypes of the types specified in the invariant schema.

6.1.3 dynamic schema: A specification of the allowable state changes of one or more information objects, subject to the constraints of any invariant schemata.

NOTE 1 – Behaviour in an information system can be modelled as transitions from one static schema to another, i.e., reclassification of instances from one type to another.

NOTE 2 – In the information language, a state change involving a set of objects can be regarded as an interaction between those objects. Not all of the objects involved in the interaction need change state; some of the objects may be involved in a read-only manner.

6.2 Structuring rules

An information specification defines the semantics of information and the semantics of information processing in an ODP system in terms of a configuration of information objects, the behaviour of those objects and environment contracts for the system.

An information object template references static, invariant and dynamic schemata. The relationships between information objects can be modelled as part of the state of those information objects. Information objects are either atomic or are represented as a composition of component information objects. The state of the composite object is represented by the combined state of its component information objects. An atomic information object template represents a concept for which there is no model at a particular level of abstraction. A composite information object represents a derived concept expressed in terms of other concepts. Since object composition includes encapsulation, an information object that is a component of one composite object cannot be a component of another. Therefore, information objects resulting from the instantiation of a composite information object template only exist as part of the instantiated composite object and have no meaning outside it.

Allowable state changes specified by a dynamic schema can include the creation of new information objects and the deletion of information objects involved in the dynamic schema. Allowable state changes can be subject to ordering and temporal constraints.

NOTE 1 – The result of accessing the state of one or more information objects can be modelled as the creation of a new information object.

In an information specification, the configuration of information objects and the behaviour of those objects need not be suitable for distribution (e.g., there need not be any concept of failure or location for information interactions).

NOTE 2 – If an information notation uses the concept of interface, any interfaces defined cannot themselves be reference points; thus there is no commitment to the interfaces appearing in an implementation.

6.3 Conformance and reference points

Conformance statements in information specifications require that the behaviour of an ODP system is conformant to a particular set of invariant, static and dynamic schemata.

An implementor claiming conformance must list the relevant engineering reference points which give access to the system and the engineering and computational specifications which apply to them. By this act the identified reference points become conformance points. The interactions at these conformance points can then be interpreted in information language terms to check that they are consistent with the invariant, static and dynamic schemata.

Information specifications can be applied to all four classes of reference point (programmatic, perceptual, interworking, and interchange reference points) identified in Rec. ITU-T X.902 | ISO/IEC 10746-2.