INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.910
(09/98)

SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

Open distributed processing

Information technology – Open distributed processing – Naming framework

ITU-T Recommendation X.910

(Previously CCITT Recommendation)

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.899
OPEN DISTRIBUTED PROCESSING	**X.900–X.999**

For further details, please refer to ITU-T List of Recommendations.

INTERNATIONAL STANDARD 14771

ITU-T RECOMMENDATION X.910

INFORMATION TECHNOLOGY – OPEN DISTRIBUTED PROCESSING – NAMING FRAMEWORK

Summary

This Recommendation | International Standard expands the naming concepts introduced in the Reference Model for Open Distributed Processing. It provides a general framework for naming in heterogeneous distributed systems, giving concepts and procedures which support fully general context-relative naming. These concepts can be applied in any ODP viewpoint. They can be applied to any function which uses naming and is subject to distribution and federation.

Source

The ITU-T Recommendation X.910 was approved on the 25th of September 1998. The identical text is also published as ISO/IEC International Standard 14771.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation the term *recognized operating agency (ROA)* includes any individual, company, corporation or governmental organization that operates a public correspondence service. The terms *Administration, ROA* and *public correspondence* are defined in the *Constitution of the ITU (Geneva, 1992).*

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Introduction

Names and naming are central concepts for the design and construction of open distributed systems. Most existing systems are built upon the assumption of a naming scheme in which only one selected naming convention applies to all the entities of concern (a global naming scheme). This assumption proves unsatisfactory in the context of large, evolving, heterogeneous distributed systems, managed by different authorities. Therefore, ITU-T Recs. X.90x series | ISO/IEC 10746, the Reference Model for Open Distributed Processing (ODP-RM), defines names to be context-relative.

This Recommendation | International Standard expands on the naming concepts introduced in the ODP-RM. It provides a general framework for naming, giving concepts and procedures which support fully general context-relative naming. These concepts can be applied in any ODP viewpoint. They can be applied to any function which uses naming and is subject to distribution or federation.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN DISTRIBUTED PROCESSING – NAMING FRAMEWORK

1 Scope

This Recommendation | International Standard:

- defines a general framework for context-relative naming, refining and elaborating on the naming concepts defined in Part 2 of the ODP-RM;
- identifies and characterizes functions necessary to handle names in the context of a federation of different naming systems; and
- clarifies the relationship between the concepts of name management (i.e. federation and naming) in distributed computing systems.

It provides a general framework for the naming of entities of interest in ODP systems, which includes naming in the infrastructure of an ODP system, naming in the applications built on the infrastructure, and naming in the enterprise the system serves.

2 Normative references

The following ITU-T 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 Telecommunications Standards Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendations X.90x series | ISO/IEC 10746, *Open Distributed Processing.*
- ITU-T Recommendations X.500 series | ISO/IEC 9594 (all parts): *The Directory.*
- ITU-T Recommendation X.650 (1996) | ISO/IEC 7498-3:1997, *Information technology – Open Systems Interconnection – Basic reference model: Naming and addressing.*
- ITU-T Recommendation X.902 (1995) | ISO/IEC 10746-2:1996, *Information technology – Open Distributed Processing – Reference Model: Foundations.*
- ITU-T Recommendation X.903 (1995) | ISO/IEC 10746-3:1996, *Information technology – Open Distributed Processing – Reference Model: Architecture.*

3 Definitions

For the purposes of this Recommendation | International Standard, the following terms and definitions apply:

3.1 Terms defined in other International Standards

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.902 | ISO/IEC 10746-2:

- ODP system;
- viewpoint;
- name;
- identifier;
- name space;
- naming context;
- naming action;
- naming domain;
- naming graph;
- name resolution;
- <X> domain.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.903 | ISO/IEC 10746-3:

- ODP function;
- ODP system;
- location transparency;
- migration transparency;
- community;
- <X> federation;
- operation.

4 Abbreviations

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

ODP	Open Distributed Processing
OSI	Open Systems Interconnection
XFN	X/Open Federated Naming

5 Overview of ODP naming

5.1 Introduction

ITU-T Rec. X.902 | ISO/IEC 10746-2 introduces the basic ODP naming concepts. This clause provides an informal discussion of the naming structures supported by those concepts.

In principle, using the ODP concepts, it would be possible to construct a single universal naming context for distributed systems – a *global naming scheme*. In such a naming scheme only one selected naming convention would apply to all the entities of concern in the ODP system. Names in such a scheme are usually referred to as *absolute names*, although they are, in fact, relative to a single agreed context from which all name resolution starts.

Such a scheme is based on the assumption that all the parties participating in distributed processing follow the same naming scheme. This implies that:

- all the parties in the ODP systems agree to use the same naming convention;
- there is a relevant name authority to administer the scheme.

These conditions can apply in some specialized environments or for systems administered by a single administration. However, a truly global scheme would require agreement on a universal name space to be used in the naming of any entities that require naming, and a universal mapping between the names from the name space and the entities. Many independently developed naming schemes have already been implemented and many more are being proposed. The ODP naming model must accommodate these different naming schemes and allow their interconnection. A single universal naming context would not support the necessary heterogeneity, and could not be managed.

Moreover, in a very large naming system, naming must remain unambiguous and the name space must therefore be equally large. Each naming action must be checked for consistency with the effects of all previous naming actions. The time required for a naming action will thus become related to the number of previous naming actions, making the availability of the naming action process unpredictable. The performance of the naming action process can only be increased at the cost of reduced consistency; some names may be used more than once (creating homonyms) and name resolution becomes ambiguous.

A context-relative naming scheme allows the federation of naming contexts and solves many of these problems.

In a context-relative naming scheme, multiple naming contexts can apply to entities in different administrative domains of the ODP system, but these naming contexts can be related to one another, so that it is possible to refer from one naming context to an entity in another naming context. In order to achieve this, in addition to associating a name with an entity, a naming action can also associate a name with another naming context. Since a naming context is something of interest, it is an entity and can be named.

Such an approach provides:

- the ability to avoid the need for unmanageably large naming systems;
- autonomy of the elements of open distributed processing systems, so that elements administered under different naming schemes continue to work on their own, but can work together;
- ease in reaching agreements and getting commitments on a naming scheme, since smaller communities are involved, which can then be brought together to form a global ODP community by federating their naming schemes;
- efficient and compact representation of names within local communities;
- the ability to incorporate different existing "global" naming schemes.

Thus, the naming process and management of consistency of names become manageable problems and existing naming contexts can be accommodated.

5.2 Structure of naming contexts

Any large distributed system is likely to comprise a number of administrative and technology domains. As a result, it is likely that the system also comprises a number of naming contexts, each relating to a name space and a set of target entities. At any given time, not all of the names from the name space, and not all of the entities from the set of target entities will be involved in a naming context.

Where a number of naming contexts exist, it may at times be necessary for an entity in one naming context to name an entity in some other naming context. Denoting an entity in another naming context requires a name for the entity, and the identification of the naming context in which the name resolves to the entity. To support such identification, a naming context can name other naming contexts. However, not all naming contexts can be named from any particular naming context, Figure 1 shows an example of how naming contexts relate names to entities.

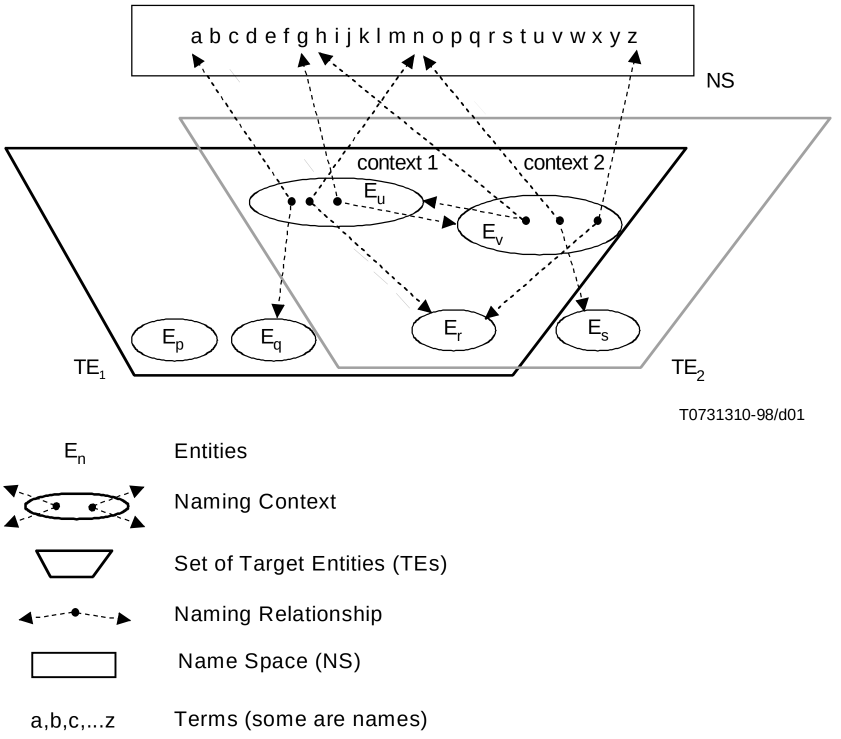


Figure 1 – Example of naming contexts

In the example in Figure 1, there are two naming contexts, each associated with:

- a set (TE$_1$, TE$_2$) of target entities. The target entities in TE$_1$ are: Ep, Eq, Er, Eu, and Ev; the target entities in TE$_2$ are Es, Er, Eu, and Ev;
- a Name Space (NS), generated by the small letters of the Roman Alphabet and shared by both contexts;
- a set of relationships between names and entities. The entity Eu is a context with relationships of: the name a with Eq, n with Er, and g with Ev. The entity Ev is a context with relationships of: the name h with Eu, n with Es, and z with Er. However, because of the relationship of a name with the naming context entity itself (g with Ev and h with Eu), context relative names of entities are possible. Thus, Eq in TE$_1$ can be named h.a from the context Ev, and Es in TE$_2$ can be named g.n from the context Eu. An entity that is in both TE$_1$ and TE$_2$, e.g. Er, can be named either directly or with a context relative name.

Table 1 shows how the names for the entities vary with the naming context in which the names are resolved. The separator syntax (the character after the "." in the example) is deternined by the naming conventions of the context in which the name being terminated is defined.

Table 1– Context relative names for entities in Figure 1

Entity	name from context 1	name from context 2
Eq	a	h.a
Er	n or g.z	h.n or z
Es	g.n	n
Eu	– (or g.h)	h
Ev	g	– (or h.g)

From both contexts, one entity (Eq as a from Ev and Es as n from Ev) is only named directly, one entity (Eq as h.a from Ev and Es as g.n from Eu) is only named indirectly via the other context, and one entity (Er) has two names, one direct and the other indirect. One entity, Ep, is left unnamed.

NOTE – Redundant names for Eq such as g.h.a from context 1 can be generated, but those can be optimized.

5.3 Processes involving names

This Recommendation | International Standard defines a naming model comprising concepts, rules and structures governing naming in ODP systems. It places constraints and gives guidance to the specifiers of ODP-compliant naming systems. The remainder of this Recommendation | International Standard defines the processes involving names in the following clauses:

- basic naming concepts (clause 6), including the management of names which involves naming and unnaming;
- resolution of names (clause 7), in which a name is interpreted in order to make it possible to interact with the entity named;
- communication of names (clause 8), which may involve the transfer of a name to an entity which interprets names in a different context from that used by the sender. In general, the communication of names will be an active process which involves transformation of the name so that, when resolved, it continues to refer to the same entity;
- comparison of names (clause 9), to determine whether two names are known to refer to the same entity (are synonyms). However, if the naming system is sufficiently complex (for example, involving loose federations), in some cases comparison may fail to identify synonyms;
- federation of naming systems (clause 10), which involves the definition of the abstract processes for name resolution, name communication, and name comparison necessary to handle names in the federation of different naming schemes.

6 Basic Naming Concepts

The basic naming concepts comprise concepts from ITU-T Rec. X.902 | ISO/IEC 10746-2 and concepts defined in this clause.

Definitions from ITU-T Rec. X.902 | ISO/IEC 10746-2 are reproduced here for completeness. In some cases these definitions are also refined and clarified.

6.1 **name**: A name is a term (a linguistic construct) which, in a given naming context, refers to an entity (see 12.1 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

NOTE 1 – Names can be used in a number of ways. These include:

- A name as a basis for interaction: a name may be required in the interaction with some entity. The name is used to distinguish the entity which is the target of an interaction from all other entities. Such a name is known as an invocation name. An invocation name conveys an opportunity (not a right) to interact with the entity denoted by that name. In the computational model, for instance, interfaces, operations and terminations are given invocation names.
- A name as an attribute: a name that cannot be used to interact with an entity may be used to denote that entity. An attributive name is used in the interaction between two entities to refer to a third entity, which is itself kept outside the interaction pattern.
- A name as an entity: a name can itself be seen as an entity of interest in a system. In the information viewpoint, for instance, a customer name is treated as a particular piece of information.

NOTE 2 – A name can be used within a predicate; a name may be used as part of a predicate. For instance, in "the papers written by Salzer", Salzer is a name. In query languages, predicates are formed to specify a set of responses. In constructing the query, names may be used as a shorthand to determine information about an entity.

NOTE 3 – Names can exist at different levels of abstraction, e.g. an address is a name, an interface reference is a name, and an identifier is a name.

6.2 **identifier**: An unambiguous name, in a given naming context (see 12.2 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

6.3 **name space**: A set of terms usable as names (see 12.3 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

Each name space is defined by a naming convention. This Recommendation | International Standard places no constraints on the nature of the symbols that may be used to construct names.

6.4 **naming convention**: The specification of the syntax to generate a set of terms to be used as names and of the algorithm to be used to parse those names.

A naming convention generates a name space. In general, the syntax comprises a part called a **handle** which can be used immediately in resolving the name and a part called a **remainder**, the resolution of which depends on the interpretation of the handle.

A naming convention influences the way in which name resolution proceeds with respect to a given name:

- left to right, as in "/usr/etc/ping";
- right to left, as in "support@iso.org.ch";
- no specific order, as in /S=X/P=SA/A=Telememo/C=AU;
- in some other fashion.

Individual naming systems may each impose a different constraint on the symbol set that may be used to construct names. Where entities in one naming system can be referred to from another naming system, a name translation might be required.

6.5 **naming context**: A relation between names and entities. Each name is drawn from a name space (a set of terms generated by a naming convention) that can be used as names in the context. Each entity is drawn from a set of target entities that can be named in the context (see 12.4 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

This Recommendation | International Standard places no constraints on what may be named. Any entity, of any type, including entities external to a system, can be named.

NOTE 1 – This definition is a refinement of the definition in 12.4 ITU-T Rec. X.902 | ISO/IEC 10746-2 that supports the definition of a naming graph.

In general, not all terms are used, and not all entities are named in any particular context.

NOTE 2 – Not all names in the name space have to be bound to an entity. Names that are unbound, but nevertheless used, are unresolveable.

Each naming action and each name resolution is relative to some context.

A naming context may specify predicates to be satisfied by all the entities which are to be named from it (such as a constraint that they be of a certain type).

A naming context may specify constraints on the number of names from that context which can be associated with a single entity. However, an entity can still be given alternative names in other contexts.

A naming context may specify constraints on the number of entities that can be associated with a particular name. The constraint that at most one entity can be associated with a single name is common. If more than one entity is associated with a single name, the context may include rules for selection of a single entity as the result of name resolution.

A naming context shall not place any constraint on the length or content of the remainder resulting from name resolution.

6.6 **synonym**: A member of a set of names which all resolve to the same entity.

6.7 **homonym**: A name which, in a particular context, applies to more than one entity.

6.8 **naming action**: An action which associates a term from a name space with a given entity.

All naming actions are relative to a naming context (see 12.5 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

NOTE – An entity is not necessarily involved when it is named, and so is not necessarily able to determine all its own names.

6.9 **unnaming action**: An action which removes the relationship between an entity and a name in a given context.

6.10 **naming domain**: A subset of a naming context such that all naming actions are performed by the controlling object of the domain (the name authority object) (see 12.6 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

NOTE 1 – The set of entities in a distributed system can be referenced from a number of domains. There are many reasons for the introduction of multiple domains. For instance, domains may be used to delineate authorities, or to distinguish areas in which different security, management, or naming policies apply. Where multiple domains arise, different relationships between domains may exist. There is no requirement for different kinds of domain to coincide with regard to their membership.

NOTE 2 – The controlling object may perform other functions, in addition to performing the naming and unnaming actions.

6.11 **name authority** A name authority is the controlling object of a naming domain.

NOTE – Name authorities can form hierarchies to reflect the authority/sub-authority relationship, a domain may be a sub-domain of another domain. The sub-domain may have further sub-domains.

6.12 **naming graph**: A directed graph where each vertex denotes a naming context, and where each edge denotes an association between:

- a name appearing in the source naming context, and
- the target naming context

(see 12.7 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

This Recommendation | International Standard places no constraints on the structure of the naming graph.

NOTE 1 – *Pathnames*. The structure of a naming graph can be described by a set of vertices, V, each denoting a context, and a set of edges, E, each denoting a link between two contexts. The edges are directed and labelled. The label represents the name that an entity in the source context (at the source of the directed edge) uses to identify a particular context. The link is with the target context (at the destination of the directed edge). A pathname is a list of names that, taken together, describe a path through the directed graph G = (E, V) from the context in which the source entity resides to the target context. In that way, the target context is singled out from amongst all possible target contexts in the graph. The graph G need not be fully connected. The connection matrix is often sparse, and in that way it specifies the constraints on the extent to which source and target entity pairs can be formed.

NOTE 2 – *Physical pathname*. A physical pathname describes a path through a physical naming graph. A physical naming graph is a naming graph that is isomorphic to the perceived physical structure of a distributed system. A distributed system consists of a number of interconnected components or nodes. The system is modelled by a directed graph G = (V, E). Each vertex in V denotes a node in the system. Each edge in E identifies a connection between two nodes in V. For example, some electronic mail systems reflect the structure of the network that connects machines in the way in which names are composed and resolved.

6.13 name resolution: The process by which, given an initial name and an initial naming context, an association between a name and the entity designated by the initial name can be found (see 12.8 of ITU-T Rec. X.902 | ISO/IEC 10746-2).

This process may require the recursive translation of a name into its handle and its optional remainder. The name resolution mechanism is described in detail in clause 7.

NOTE – If a naming scheme permits ambiguity, then there might be another process which finds all entities designated by the name. Examples of such alternative processes are homonym resolution which finds all entities designated by a homonym, and dynamic name resolution which provides dynamically selected result for each invocation.

6.14 naming system: A naming system is a naming graph and the set of naming contexts denoted by the vertices of the naming graph.

7 Name resolution model

When knowledge of an entity is to be communicated, without the presence of the entity, a name must be generated for that entity. The naming action takes place in a particular naming context. The name must be taken from the name space of that context, and the association between the name chosen and the entity becomes part of the context. The name resulting from the naming action can be transferred by the entity performing the naming action to some other entity. The receiver of a name can resolve that name with reference to one of its own naming contexts and an associated name space. However, the information transfer is only successful when the naming contexts of the sender and receiver overlap or coincide.

Names may be acquired by many kinds of interaction, and the receiver must associate a naming context with each kind of interaction it participates in. This association may be as a result of prior knowledge of the sender, or from other information conveyed by the interaction.

As a general rule, name resolution must take place in the same naming context as the one in which the naming action that allocated the name took place. If this is not the case, then there must exist some process that knows about both the namer's name space and the resolver's name space, such that the name may be suitably transformed (for instance, from ukc!acorn!ansa!xyz to xyz@ansa.co.uk).

This clause is concerned with the generic process which characterizes all name resolution. A variety of information is needed to perform name resolution, such as for example, details of naming conventions and name translations. Consistency and availability of name resolution are important issues which affect the distribution of name resolution processes. Distribution strategies for information required to perform name resolution are described in Annex B.

7.1 The name resolution step

In general, name resolution takes place in a series of steps, each of which is dependent on the context it is performed in. Each step involves the following stages (shown in Figure 2):

a) initiation of the resolution, either as a result of an interaction with some other object or of an internal action of the object concerned (starting the resolution process); this identifies the name to be resolved;

b) association of a context with the name, either implicitly from the identity of the initiating action, or as a result of other information associated with the initiating action;

c) analysis of the name, using the rules from the naming conventions associated with naming context identified in (b) and the stored internal information (i.e. state) of the object performing the resolution. The result of this analysis is a piece of information (called the handle) which can be used immediately to continue the resolution process and naming information which cannot yet be resolved (called the remainder). The handle determines the actions to be performed and any necessary parameters for that action. In many common naming conventions, the remainder is simply the residual string from a left to right or right to left parsing process, but it may, in general, be any transformation or, indeed, a complete replacement of the original name;

d) execution of the consequential action, using the handle and conveying the remainder.

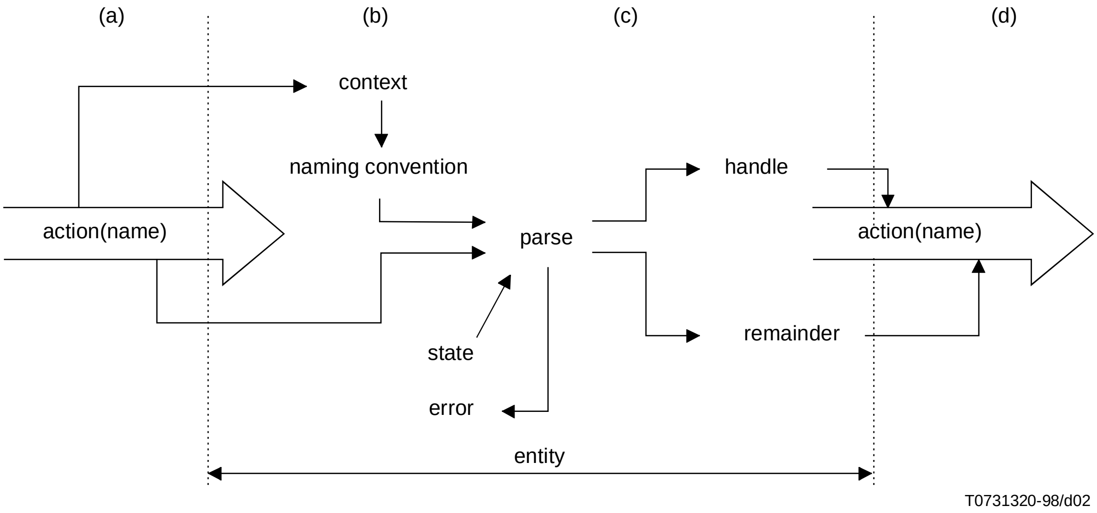


Figure 2 – The general name resolution step

Thus each name resolution step involves a transformation function which takes a context and a name and yields an action, a handle and a name. The function is partial, because not all names in the name space are resolvable. It is many-to-one, because an entity may have multiple names. Not all entities need to be named in all contexts.

Some naming contexts define homonyms; in some naming contexts, syntactic forms are defined which will match sets of names from the name space (e.g. wildcards). In either case, the resolution step can, potentially, match multiple objects. The naming conventions of the particular context may result in failure or arbitrary selection if more than one match is possible; alternatively some form of group naming may be supported. Depending on the semantics of homonyms or wildcards, the resolution process may need to respond to failures of the consequential action by attempting a sequence of other possible consequential actions until one is successful.

7.2 The complete name resolution process

The resolution step given above can be used to form a complete resolution process. In the complete resolution process, the result of any resolution step which identifies a further naming context, is passed as a resolution request to an object responsible for that context. This is shown in Figure 3.

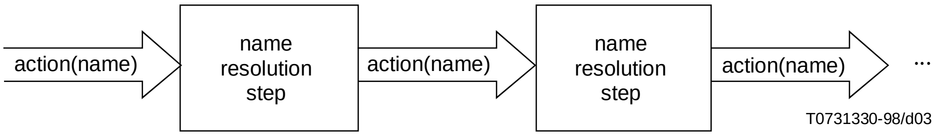


Figure 3 – The name resolution model in cascade

In detail, the consequential action is either an action on the entity named, or an action which invokes a further resolution step. The handle provides parameterization for this action. The remainder provides the name to be resolved in the next step.

The process starts with a decision to resolve a name in a known context, and ends when the consequential action is an action on the named object, rather than a further resolution step. Termination is determined entirely by the consequential action; the resolution does not necessarily terminate because the remainder is null, or continue because the remainder is non-null.

NOTE – Possible outcomes of resolution steps are, resolution successful (found or not found), resolution failed (could not complete resolution steps), and unknown.

The set of interactions resulting from all possible consequential actions and handles corresponds to the set of edges of the naming graph. Care must be taken if this graph contains cycles, since the stability of the resolution process then depends on the details of the transformation functions. Even in the presence of cycles, the process will converge if names can be ordered (e.g. by length) and the transformed name is always strictly less than the name from which it is derived, but otherwise stability analysis can become very complicated. Particular care is needed when the naming conventions include homonyms or the definition of wildcards or other mechanisms, which allow the name resolution process to return a set of entities.

8 Name communication

When two entities from two different naming contexts wish to interwork, they often need to communicate names. The sending entity needs to communicate a name that is meaningful in its own naming context to the receiver in another naming context in such a way that the meaning of the name sent is also understood by the receiver.

To preserve the meaning of a communicated name, it is not enough simply to transmit the symbols used in the name. To be able to use the name in its proper meaning, the receiver also has to have knowledge of the naming convention being used by the sender and the intended use of that name. The sender may use its own local context, the receiver's context or some transfer context. This knowledge is usually conveyed before the name is communicated.

To communicate a name, some infrastructure is needed which can mechanise the transfer of symbols from one entity to another. The technologies for this are well known and range from shared memory to networks and communication protocols.

Usually the intended use of the transferred name is clear from the communication action, i.e. by its communication context. Sometimes the intended use is included in the name, imbedded in the name structure and revealed by the naming convention. Sometimes the intended use has to be explicitly stated (e.g. a fax telephone number instead of a voice telephone number).

If there is some common understanding of what a name signifies, then the exchange of symbols can become a meaningful one. However, there are no effective mechanisms which ensure a common understanding of the significance (meaning) attached to the symbols exchanged. The context of the communication often gives the significance of the name by indicating the naming context used when transferring a name.

The way a name is communicated can be classified into three cases:

1) The simplest case is when both sender and receiver are in the same naming context, and are under the same name authority using the same naming convention. Thus, the names used in the sender's naming system can be used meaningfully in the receiver's naming system. A name's syntax is correct and there are no naming collisions i.e. the name received is not already used for some entity in the receiving system (exchange of names does not create new homonyms).

2) In the case where the sender and receiver use the same naming convention but do not belong to the same naming context, the names will still have the same syntax and resolution algorithm. However, their naming actions will be performed independently. The same name can be used to name one entity in one domain and another entity in the other domain. Exchange of names can cause a name collision, i.e. create an unintended homonym in the receiver's system, that must be resolved. The usual way to resolve such collisions is to attach the name of the sending domain to each communicated name. The "exchange protocol" determines whether the sender or the receiver of the name performs the context attachment.

3) If the sender and the receiver belong to different naming contexts using different naming conventions, then the name exchange between them is more complicated. Usually name translation is needed. The exchange of a name may require translation of the symbols used in the name in order to transfer them or to enable their usage by the receiver (this might be done by passing a name to a third entity for translation). The entities involved need, in general, some form of agreement similar to those used in federation of naming systems, but possibly on a more restricted and less permanent base. For example, when communicating a name, the sender could expose part of the naming context of the name communicated to this receiver. However, this exposure of naming context has to be known by the receiver somehow, whether implicitly or explicitly. Name resolution and name translation can then be used in the same way as in federation (see clause 10).

9 Name comparison

A frequent need is to test whether two names refer to the same entity, that is, are synonyms. In some situations, however, the equivalence of name cannot be decided, as one entity may not have a complete view of the entire naming system. Thus the signature of the equivalence testing (name comparison) function is:

name × name -> {true, false, indeterminate}

Equivalence is, in any case, only defined relative to a particular specification and level of detail, since a single entity in one descriptive system may be a set of interacting entities in another.

NOTE – In case result of comparison is indeterminate, retry action may be taken, depending on the nature of indeterminacy.

Note that equivalence of names does not necessarily imply that the names concerned are mutually substitutable.

10 Federation

Naming systems are joined together to form a larger naming system whenever a context in one of the original systems is named from within at least one context in the other. The name resolution process is still consistent with the model given in clause 7. However, if the original naming systems are managed by autonomous organizations, as is normally the case in a federation, care is needed to ensure that the resultant system evolves in a predictable way over time. This clause discusses some of the problems to be solved and introduces some particular types of context which can be used to simplify federation agreements.

The stepwise name resolution process defined in clause 7 will, in general, result in responsibility for name resolution initiated by one of the federation partners being passed to another in order to complete the process. This requires a mutual federation agreement. While an organization can unilaterally name a context in any other organization in one of its own contexts without some mutual agreement, there is no guarantee that name resolution can be completed. Such agreements may be light weight, for example, when an organization makes certain name resolution services publicly available.

As local naming and unnaming actions are performed within the various contexts which make up a federated naming system, the set of naming relationships represented by the naming system will change. One of the aims of the federation agreement is to ensure that such changes do not result in unexpected results, so that names which have been communicated to others either continue to refer to appropriate entities, or fail to resolve to anything. This will require a mutability policy (a set of rules on how name can be changed) to be associated with each context. One approach to federation is to introduce specific additional contexts to highlight mutability obligations arising from federation (see export context in 10.1).

A mutability policy will reflect the objectives of the corresponding part of the naming system. For example, names are often given to the most up to date version of some piece of information or some service, rather than being associated immutably with a single entity. Manipulation of a context managed by the service or information provider could then change the entity to which the name binds, but not the type of that entity.

10.1 Controlling the names exported – export contexts

When setting up a federation agreement, an organization may not wish to make its whole naming system visible to other partners in the federation. It can control the parts of the naming system to be made visible by defining one or more export contexts.

An export context is a context introduced by a partner in a federation in order to make explicit the set of names which can be resolved as a result of this federation. There is an obligation to cooperate in the resolution of names via the export context. There is no such obligation in the case of names which reference other contexts than export contexts defined by the organization. This is equivalent to defining a policy such that resolution requests from other organizations are always analyzed initially in an export context.

Figure 4 shows an example of the use of an export context.

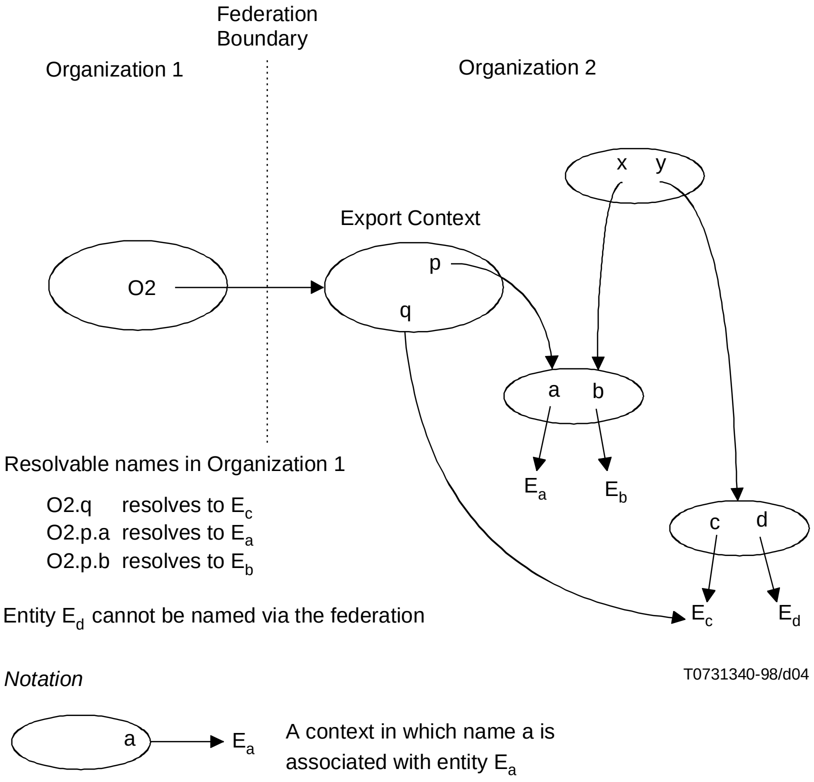


Figure 4 – An Export Context

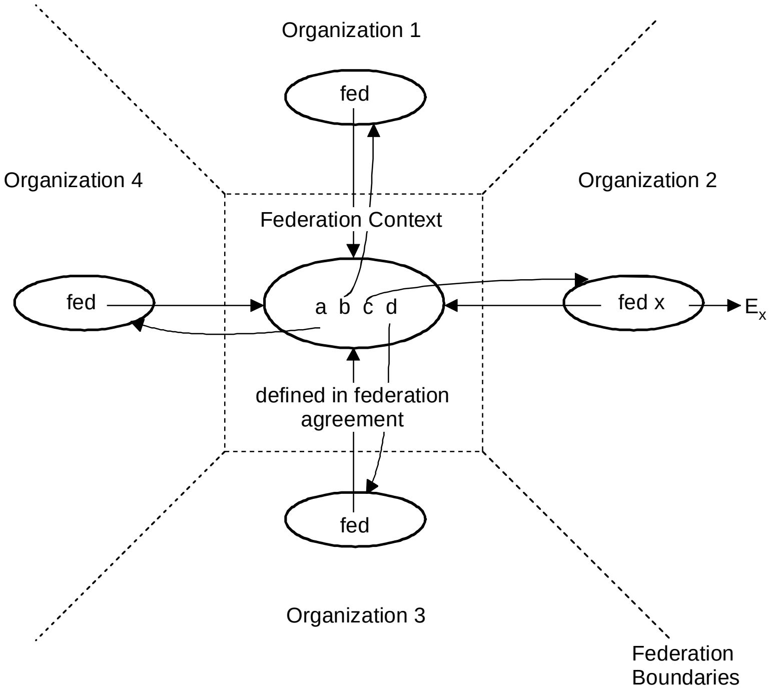
10.2 Uniform naming of federation partners – federation contexts

Each partner in a federation can choose its own names for the other partner's contexts. However, doing so may cause problems in the management of large federations, since each participant may have been given many different names by others, who will therefore not agree on a single name for it. This variation in naming can be avoided by creating a federation context as part of the federation agreement.

A federation context is a context identified within a federation agreement to provide a shared context in which the export contexts of the federation partners are named in a uniform way. A federation context will normally be subject to a particularly strict mutability policy. A federation context names export contexts of the participants. Therefore a participant can then name a federation context in its local context. If a federation context is defined, it is named by the participants in their local contexts, and names their export contexts. A federation context is only needed if uniform naming is required as part of the federation agreement; if there is no federation context, participants name each other's export contexts directly.

In principle, there may be a different federation context for each sender-receiver pair involved in name communication, but a federation agreement may specify a single federation context for multiple communication paths. The way the federation context is controlled is stated in the federation agreement; the federation context may be defined within the federation agreement itself.

Figure 5 shows an example of the use of a federation context.



NOTE – The name fed.c.x is resolvable to entity E_x by all the federation partners, including Organization 2

T0731350-98/d05

Figure 5 – A Federation Context

10.3 Providing a uniform user view of federated naming

If an organization wishes to participate in a naming federation, but already has constraints on the syntax and values of the names it uses internally, it may be difficult for it to accommodate the names resulting from the federation (which will typically consist of the name for another organization's export context together with a name using the naming conventions of that context). In such circumstances, the details of the new names introduced by the federation can be hidden by introducing suitable local name translations.

In terms of the name resolution model, this can be done by resolving a local name to generate a handle which will result in communication with the appropriate federation partner and a remainder which is the name required by the federation agreement. The syntax of this remainder is therefore not seen by users within the organization.

This style of use implies that there must be an alias and a translation for every federated name imported. This may be seen as a limitation, because a new translation must be installed before each new name defined by federation partners can be used, or it may be seen as a useful security tool, giving local access control. Whether it is seen as a regrettable necessity or a management opportunity will depend on the policies of the organization concerned.

The set of naming relationships involving translation to support federation may form a subset of an existing naming context, or a new context may be introduced for this purpose. A context introduced specially to support local translation of federated names is sometimes called an import context.

Figure 6 shows an example of the use of local name translation to support federation.

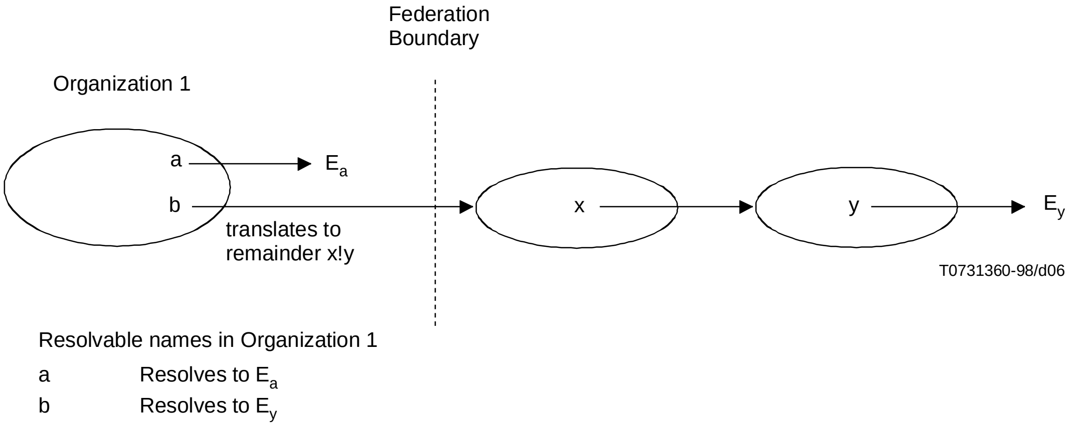


Figure 6 – Use of local name translation

10.4 Joining a federation

A federation is a community of domains. The smallest possible domain which can become part of a federation is thus a single object controlling an otherwise empty domain. The smallest possible naming system which can participate in a federation is therefore one involving a single empty context.

Creating a federation agreement involves:

a) agreeing the contractual basis for, and responsibilities implicit in, the federation; this includes agreement on the common understanding which will form the basis of communication;

b) identifying a communication mechanism capable of linking the members of the federation for naming purposes;

and optionally involves:

c) creating a set of federation contexts to be used when communicating names via this communication mechanism; this includes agreement on the naming conventions to apply to names actually being communicated;

d) creating a name in the appropriate federation context for the export context of each member of the federation; each member must create their own export context, giving access to the entities to be visible as a result of the federation.

e) each member of the federation creating any necessary local name translations, naming the federation contexts it will use in communication. The members of the federation each act separately to define any necessary translations to be applied when passing names into the federation contexts.

In simple cases, either of the export or federation contexts can be overlaid onto contexts already maintained by one or more of the federation members, provided that appropriate guarantees are made to preserve the properties of these contexts established in the federation agreement.

10.5 Evolution and optimization

In a situation where there are many naming systems, each participating in a number of overlapping federation agreements with others, very large numbers of entities are accessible by name, but the routes taken during name resolution can become exceedingly tortuous. This is particularly true if names are repeatedly communicated, passing through many contexts in the process.

To avoid unnecessary complexity, there is a need for optimizations in which some names are replaced by equivalent, more readily resolvable, names based on knowledge of the topology of the naming graph. The degree to which optimization can take place depends on the amount of knowledge held at each node in the naming graph about the topology of the graph, since optimization is limited to replacement of one name by another on the basis of knowledge that they are equivalent.

Consider, for example, three contexts shown in Figure 7. The entity named "a" in context 1 is known as "p.a" in context 2 and "q.a" in context 3. However, if an entity using context 3 communicates its name to an entity in context 2, the result would be "r.q.a". Only if there is knowledge of the topology of the naming graph within context 2 can the names "p.a" and "r.q.a" be recognized as synonyms and the obvious optimization made when, for example, creating a binding to the object.

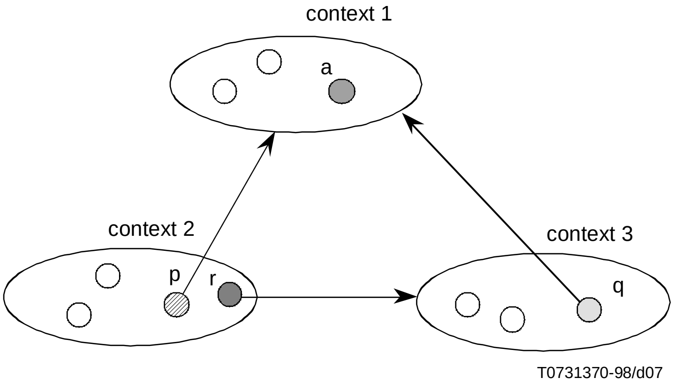


Figure 7 – Example showing synonym

However, widespread knowledge of the topology of the naming graph may not be desirable, because naming system evolution then implies more complex requirements for propagation of updates when the graph is changed. If the naming graph as actually defined becomes inconsistent with the beliefs about it held by the objects managing its nodes, chaos will result. There is therefore a need to match federation agreements to propagate knowledge about the naming graph with obligations either to keep the naming links fixed or to give warning of any changes.

If the association between a name and an entity is changed, there will be consequences for earlier users of that name. Changes in the naming structure may or may not invalidate bindings established previously using the name, depending on the specification of policies in the system using the name.

Functions are required to support maintenance of the naming graph so that:

a) objects which are about to become inaccessible or to change in such a way as to vary the actions needed to access them (e.g. to change their location) notify the controlling objects of any naming domains in which they believe they are named;

b) controlling objects of naming domains notify each other of changes to the naming graph which are likely to affect name resolution.

11 Compliance

This Recommendation | International Standard can be related to other, less abstract, specifications in two ways, described as follows:

a) Existing standards which define naming systems that have a compatible naming model and the properties necessary for them to take part in federations of the kind defined by this Recommendation | International Standard are said to be "consistent" with the ODP naming framework, even though they do not themselves reference this Recommendation | International Standard.

b) ODP standards which contain a reference to this Recommendation | International Standard in order to define the properties of their naming systems are said to be "compliant" with the ODP naming framework. Such standards are expected to define the relationship of their specification to the naming framework, as documented in the following clauses.

11.1 Basis for compliance

Naming activities seldom occur in isolation. Subclause 7.1 has described name resolution in terms of a series of name resolution actions, terminated by an action on the entity named. In many practical cases, however, the name resolution actions also have other effects, such as allocation of resources or creation of a communication binding (e.g. a connection). These other actions may be necessary to allow the communication needed to request the next resolution step, and so the two kinds of action are closely associated.

In such cases, the sequence of steps taken is determined by the name resolution process, and the standard specifying the associated action need only define what happens during a single resolution step, and is not concerned with the sequencing rules for such steps; these are provided by the name resolution procedures.

Any particular standard may be concerned with a number of naming systems, corresponding to the naming of different types of object. In such cases, each naming system must be identified, and any dependencies between naming systems, such as requirements that parts of two different systems be isomorphic, must be identified.

Examples of the close interaction between name resolution and other actions can be seen in the federation of traders or type repositories.

11.2 Compliance requirements

A Recommendation | International Standard complying with this framework shall declare:

a) what types of entities are named in the Recommendation | International Standard;

b) what behaviour defined in the Recommendation | International Standard requires naming actions or name comparison;

c) which of the information elements communicated between the objects defined in the Recommendation | International Standard are context-relative names, and what type of entities form part of the context;

d) what actions in the Recommendation | International Standard are performed in association with name resolution.

For each action identified in d) the Recommendation | International Standard shall specify the behaviour to be associated with each name resolution step. This specification shall include statements of how externally initiated actions correspond to naming contexts, and what actions should be performed to continue, or to terminate, the name resolution process.

For each naming system, the Recommendation | International Standard shall specify any necessary properties, such as immutability or notification of changes in naming relations.

Annex A

Relationship with Related Naming Services and Frameworks

(This annex does not form an integral part of this Recommendation | International Standard)

The naming framework described in this Recommendation | International Standard is sufficiently general that well-established naming systems fit in this framework. More importantly, such different Naming Systems can federate with each other within this specified framework. However, the specification of detailed mechanisms for name translations required for federation is beyond the scope of this Recommendation | International Standard.

A.1 Relationship with OMG Object Naming Service

The OMG Object Naming Service supports a context relative naming system and is consistent with the ODP naming system.

An OMG simple name consists of two parts: an identifier followed by a description. No interpretation of the name is made by the naming system. Syntactic conventions are used to partition the name space meaningfully .

A name context is an object that contains a set of name-entity associations in which each name is unique. A name graph is a directed graph of a set of name-context associations. Given a context in a naming graph, a sequence of names can reference an object. This sequence of names, called a compound name, defines a path in the resolution process. That is, a compound name is resolved by repeatedly separating the first context (the handle) from the rest of the name.

Federation of namespaces is supported in a distributed way with no assumptions about centralized functions. There is no need for a distinguished root context and existing graphs may be joined at any point. The OMG naming system can federate with other naming systems, which can be transparently encapsulated using name contexts.

The OMG Names Library allows the representation of names to evolve without affecting existing clients and hides the representation of names from clients. The names library implements names as pseudo-objects that are light-weight entities which are efficient to create, manipulate, and transmit. That is, the names library is a mechanism to provide evolution and optimisation of name usage.

The OMG Naming Service also specifies a set of operation signatures for the association, disassociation, and resolution of names; creation, deletion, and listing of context and operations for the management of its Names Library. These operations are computational manifestations consistent with the abstract processes identified in the ODP Naming Framework.

A.2 Relationship with X/Open Federated Naming API

The X/Open Federated Naming API (XFN) supports a context relative naming system and is consistent with the ODP naming system.

An XFN composite name is a name with syntactic and structural properties.

A context is defined as an object whose state is a set of name to entity associations with distinct atomic names, which is equivalent to naming context in this Recommendation | International Standard. A naming graph is introduced implicitly by the introduction of superior/subordinate relationship of naming contexts. Given a context in a naming graph, a composite name defines a path in the resolution process. Composite name resolution is achieved by combining name resolution process in each naming system across a naming system boundary.

Federation of namespaces is supported in a distributed way with no assumptions about centralized functions. There is no need for a distinguished root context and existing graphs may be joined at any point. XFN naming system can federate with other naming systems, which can be transparently encapsulated using name contexts.

XFN, however, defines a set of concrete application programming interfaces, and provides more detailed functions than those described in this Recommendation | International Standard.

XFN can be used to realize ODP Naming Framework compliant naming system. ODP Naming Framework compliant naming systems will be interoperable with XFN based system.

Concrete mapping of concepts and functions of this Recommendation | International Standard onto XFN specification including APIs is outside the scope of this annex.

NOTE – XFN uses the X/Open Directory API to access systems supporting the OSI Directory protocol.

A.3 Relationship with OSI Naming and Addressing

The OSI Naming architecture is designed to be applied to OSI, therefore federation with other naming system is outside the scope of OSI Naming and Addressing standard.

The OSI Naming standard defines several naming concepts such as primitive name, generic name, descriptive name, Directory Name and Object Identifier which are used in protocols. Other OSI Recommendations | International Standards are the global naming authorities that define Directory Names and Object Identifiers, which are used in protocols. The values used are names under the global naming authorities.

A name resolution process is also given in the OSI Naming standard, and it is consistent with the process described in this Recommendation | International Standard.

OSI Distinguished Names (used in Directory and Systems Management Standards) employ a hierarchic naming tree with a global root. This global root can be considered as its own naming context, thus a naming system compliant with ODP naming framework can interoperate with systems using OSI Distiguished Names. However, in general it is not possible for systems which require OSI Distinguished Names to interoperate with all ODP naming framework compliant systems.

Annex B

Configuring name services

(This annex does not form an integral part of this Recommendation | International Standard)

A name server is an entity which offers the name service at, at least one of its interfaces, and is also an entity which maps attributive names to invocation names. This annex identifies several configurations of name servers and examines the quality of the resulting name service.

Centralised name services will not be examined. The obvious performance penalty of a single name resolution process is not acceptable in distributed systems because it forms a bottleneck in anything but the smallest system.

> NOTE – Some of the terminology used in this annex reflects existing usage in the industry, rather than the stricter interpretation provided in this Recommendation | International Standard. Such usage is now deprecated, but is included here to clarify links to current practice.

B.1 Quality of name service concerns

Some of the ways in which the quality of a name service may be measured can be expressed as:

- How can ownership and autonomy issues be reflected?

  The name service database contains information about other services. The rules about ownership and autonomy over these services vary. At the same time it is convenient to partition the information about services in such a way as to reflect these ownership and autonomy issues. The name service is long lived and subjected to many organisational changes over its life span. It will thus be necessary to provide a flexible approach to the partitioning of the name service data structure throughout it's life time.

- How accessible is the name server, and how long does access take?

  A name service database can be very large. There will be few, if any, name service clients that will require frequent access to all of the database. It is common for several name service clients, with similar access requirements, to be clustered together. If the location of name service clients is static or does not vary too much (e.g. they remain in a single LAN), then it is possible to partition the name service database over name servers which can serve certain client communities locally. Name service clients may of course migrate and alter their usage pattern, and this again requires a flexible approach to the partitioning of the name service data structure.

- What is the effect of parts of the service failing?

  Fault isolation is an important factor in deciding how to partition or replicate the name service data structure over name servers. Each name server is considered as a separate unit of failure. By partitioning and replicating the data structure over several name servers, the failure of one name server will not lead to a total loss of service to all users.

B.2 Distributing the name service

The name service maps attributive names to invocation names. To do this its state consists of a data structure which reflects the relation between both kinds of names. Distribution of this data structure is achieved by:

– partitioning the relation to reflect ownership and autonomy issues, and to increase accessibility by clients; and

– replicating the relation to improve resilience to failure, and again to increase accessibility by clients.

There exists a conflict between the consistency and availability of the data structure. As a result, distribution criteria will vary from system to system and from application to application. The following options for distributing the name service exist:

– one name server per name space;

– one name server per service type;

– one name server per service provider or consumer;

– one name server per administration;

- one name server per application or utility, e.g. a directory server in a file system;
- one name server per failure domain;
- one name server per host node;
- any combination of the above.

The data structure reflects the form of the naming graph. Viewing the naming graph as a graph helps when reasoning about the distribution of the data structure used by name services. The graph can be carved up and parts of it replicated. Each possibly replicated partition is assigned to a name server. The name servers are then connected together to reconstruct the original naming graph.

The architecture does not place any constraints on the structure of the naming graph. Particular naming systems will do this. For instance, in OSF DCE, the Cell Directory Service (CDS) entries are organised in directories, which are structured hierarchically. The directory structure is then partitioned over CDS databases, each called a clearinghouse.

In ITU-T Recs. X.500 series | ISO/IEC 9594 (all parts) a similarly hierarchical structure gets partitioned over a set of Directory System Agents (DSA), which are able to interwork using the Directory System Protocol (DSP).

B.3 Configuring name servers

B.3.1 Basic model

A name service in a distributed system is implemented by a collection of interconnected name servers. There are two kinds of name servers: local and global. Local name servers look after a set of entities in a local domain, which is determined by the server placement policy (see B.2). The entities in the local domain can only see other entities that are also in that domain. Visibility of entities in other domains is strictly through "global" name servers.

The way in which local and global name servers are implemented and access one another determines the configuration. In all cases, a client of a name service must have access to at least a local name server, and optionally to a global name server, as in Figure B.1.

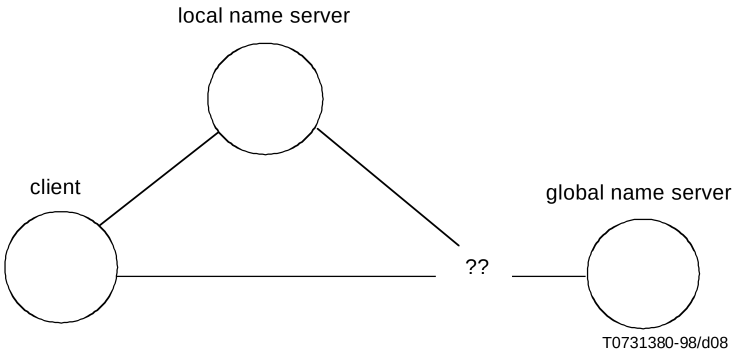


Figure B.1 – A name service client has access to a local and a global name server

Sometimes, the global name server consists of a client and a server part. In ITU-T Recs. X.500 series | ISO/IEC 9594 (all parts) for instance, the Directory User Agent and its cache form the client part of the global name server. The DUA and its cache can clearly be seen as a local name server in the model of Figure B.1.

The four basic name server configurations that can be built from these components are "direct access" or "referral", "re-registration", "chained", and "federal" configurations.

Before a client can use any name service, it needs to determine which name server should be used. Once this has been determined, the second activity is to access the selected server. Depending on the configuration, the local and global name servers take part in either or both activities.

B.3.2 Direct access (or referral) approach

In the direct access approach the name server client accesses remote name servers directly. First part of call is the client's own local name server. When a name cannot be resolved there, the global name server is consulted. The global server knows about other local name services, and holds a single level mapping from global names to local name servers. The name service client thus obtains a handle, allowing direct access to other local name servers. It is as if the global name server refers to the other name servers (hence the term referral).

The knowledge which the global name server has about other name servers has been represented by dashed lines in Figure B.2.

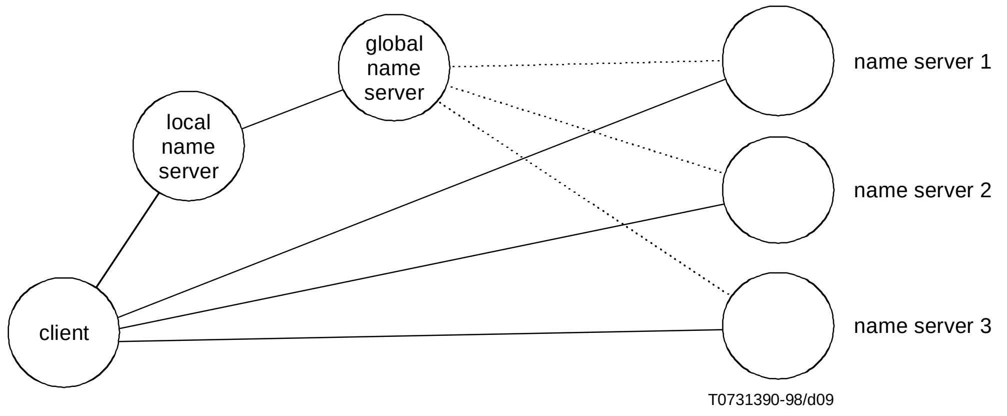


Figure B.2 – Direct access approach: Global name server hands back reference to other local servers

The name service client is aware of the different local name servers and access, location (and migration) transparency are not offered.

Referral is one of the methods by which Directory System Agents in ITU-T Recs. X.500 series | ISO/IEC 9594 (all parts) search the directory structure (chaining being the other).

B.3.2.1 Local name servers in different name spaces

Different local name servers may be in different name spaces. Before the name service client can access the local name servers, it needs to know about the name spaces in which they operate. This information could be provided by the global name server. Any functionality to deal with the differences is firmly positioned at the client end. Adding a new local name server that supports a different name space would require an update to all potential name service clients as well as the global server; a solution that clearly does not scale.

It is possible to place the functionality for the provision of name space transparency at the server end. This would require a front end to each local name server in the system. This solution scales better because adding a new name server means the addition of a single new mapping.

The transparency may also be included in the infrastructure which supports the interactions between the name service client and the local name servers. The global name server hands back a handle to a so called Naming Semantics Manager (NSM). This manager can handle the translations as part of the RPC mechanism.

B.3.2.2 Other solutions

To hide the differences between the local name servers (servers 1, 2, ..., N in Figure B.2), it is necessary to isolate the name server client from the local servers. This can be done in two ways: re-registration or chaining.

B.3.3 Re-registration approach

In the re-registration approach a name service client accesses the global name server for access to the information about services outside of the local context. No access to other local name servers is required, as these have all passed their responsibility for maintaining the mapping between attributive names to invocation names to the global name server.

They have re-registered everything that is registered locally and that needs to be made available globally at the global name server. Figure B.3 illustrates the relationships between the local name servers and the global name server.

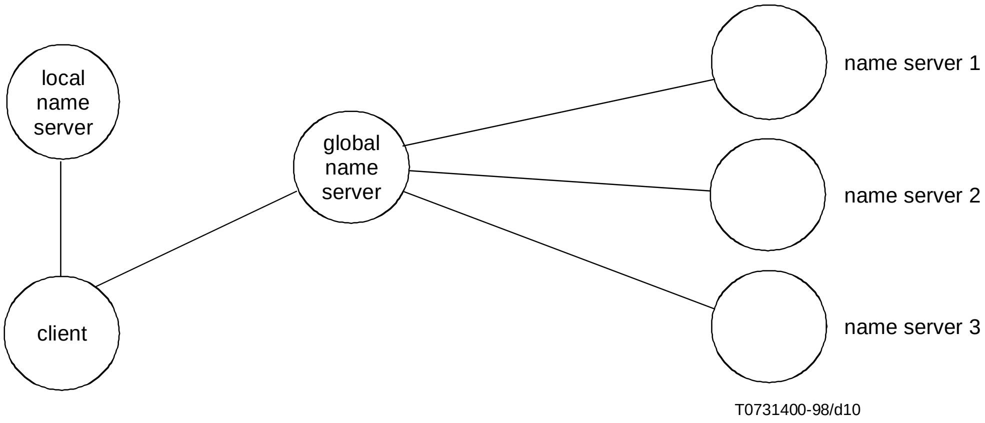


Figure B.3 – Re-registration approach

The re-registration approach imposes several constraints on the local and global name servers. The local name server is not at liberty to change any of the names of entities whose names have been re-registered. To do so would require the information held in the global name server to be changed as well. Conversely, the global name server needs to keep track of the source of the registrations, so as to disambiguate any potential homonyms. The global name server represents an administration that is hierarchically above the local name servers.

It is clear that this approach does not scale, as it relies on a single global name server that is expected to keep track of all mappings. The global name server would soon become a bottleneck.

B.3.3.1 Caching and involuntary re-registration

Copies of name server entries in the local name servers may appear in the global name server when the global name server caches information obtained in queries. The client part of the global name server, such as the Clerk in CDS and the Directory User Agent and its cache in ITU-T Recs. X.500 series | ISO/IEC 9594 (all parts), perform this function. This leads to much better scaling characteristics with respect to availability, but increases the burden on the name service as a whole to keep its data consistent.

B.3.4 Chaining name servers

Chaining yields a similar name server configuration to re-registration. The difference is that the global name server holds a single level mapping as in the direct access approach. Unlike the direct access approach, the name server client does not interact with any other local name servers. The global name server acts as an agent, on behalf of the name server client in accessing name servers. Figure B.4 illustrates the configuration.

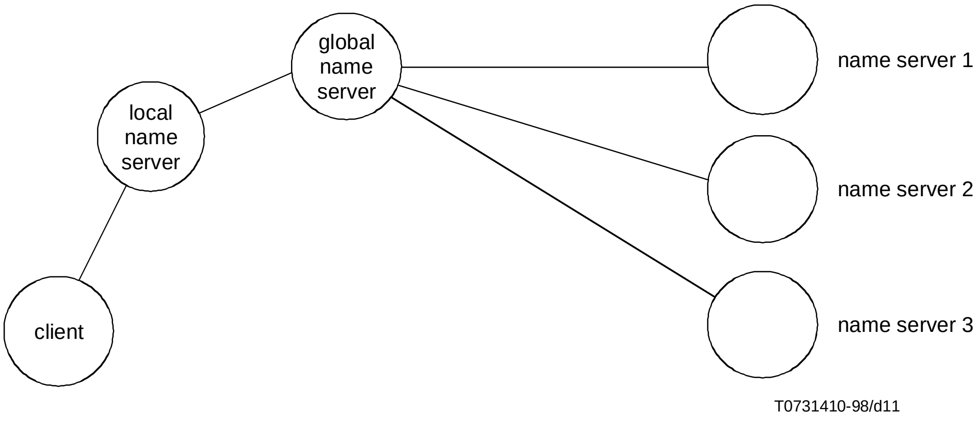


Figure B.4 – Chaining

Because the global name server accesses the name servers (1-3), then from the point of view of the client, these name servers are invisible, since the decision about which name server to access is also internal to the global name server. In that way the global server offers complete access and location transparency.

Chaining is one of the ways in which the Directory System Agents in ITU-T Recs. X.500 series | ISO/IEC 9594 (all parts) search the directory structure (direct access or referral is the other).

This scheme allows some freedom in the mapping between the global names and the local names. Since the global server maps from global names to a single name server name, the local name servers are allowed to change the bindings between local names and the entities in the local domain. The combination of global and local name servers can be made to have the effect of a level of indirection in the name translation.

B.3.4.1 Following a chain

There are two strategies in searching a number of chained name servers. The shallow search strategy will search all of the information held in a particular name server and only follow a chain if name resolution is not successful at that server. The deep search strategy will follow the chain to another server as soon as such a chain is encountered. It will return when the end of a chain has been reached to continue searching locally.

B.3.4.2 Caching

The global name server may want to cache some of the information it has already got from one or several of the local name servers. If this is done, then the problems described in the re-registration approach may return. Using out of date name mappings will cause misdirection of interactions later on. If mappings change infrequently (as is expected in ITU-T Recs. X.500 series | ISO/IEC 9594 (all parts)) and resulting misdirection errors can be dealt with efficiently, then a decision to have a cache and lazily update it (i.e. periodically or on problems only) may be acceptable. This increases availability of the name server at the cost of reducing the consistency of the name server data base.

B.3.5 Federal approach

Local name servers may be connected in a federal manner by distributing the global name server over all the name servers that take part in the federation. The resulting completely distributed "federal" name server is characterised and implemented by the protocol of interaction between the local name servers in the federation.

In any one federation, each name server is able to use the service offered by any one of the other name servers in the federation. The tasks of (1) determining which server holds the required information and (2) accessing that server are devolved over all members of the federation. Each component in a federation can be described as a set of protocols that must be adhered to as a condition of belonging to a federation.

Each server must hold a set of mappings between locally known external names it knows about and the name server that implements the mapping between attributive and invocation name.

Figure B.5 illustrates the structure of a system built using the notion of federation in terms of the components introduced in the descriptions of the other approaches to combining name servers.

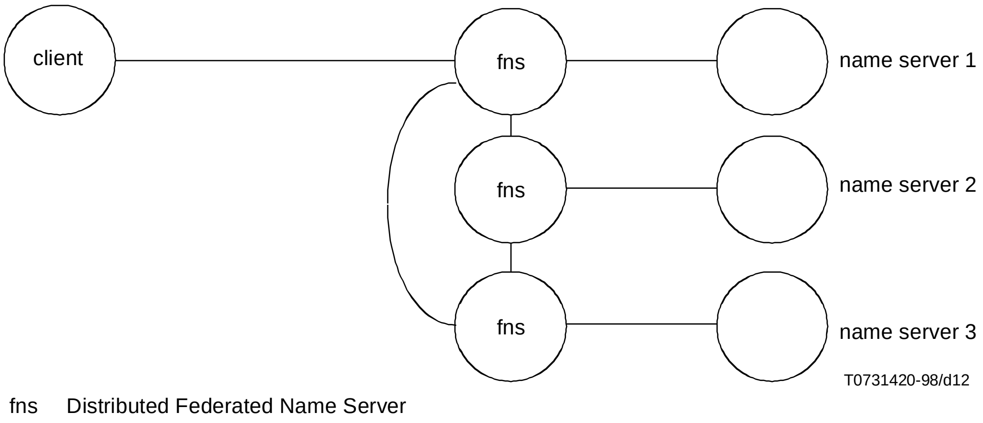


Figure B.5 – Federal structure: distributing the global name server

B.3.5.1 Scaling

Note that the federal approach scales better than any of the other approaches. The federation places a bound on the context in which a name is assumed resolvable: it is restricted to the scope of the federation. Name servers are allowed to be a member of more than one federation at the time.

Annex C

Application of Naming

(This annex does not form an integral part of this Recommendation | International Standard)

Various engineering mechanisms can be achieved by manipulation of the naming graph. In particular, they can be used for achieving transparencies.

a) Location transparent naming

Location transparent naming requires that a name for a particular entity is independent of the location of the target entity. When the path name reflects the physical structure of a system, location transparency is not provided if such names are visible to an application. A soft or symbolic link to the full source routing name can be inserted between the application and the system that resolves the name to achieve the required location transparency.

b) Migration transparent naming

Migration transparent naming requires that a name for a particular entity is independent of the change of location of the target entity. When the path name reflects the physical structure of a system, migration transparency is not provided if the changes in location of the target entity are reflected in changes in the name. To make this invisible, a transparency layer can be inserted that will map a name to the new source routing name, each time the target entity migrates. Note that, initially, the target can be named directly, and this name can be converted to a soft link (with the same representation) when the target entity migrates for the first time.

c) Version control

Naming contexts are often dynamic; that is, the set of name relations is subject to change. This change may be caused by the definition of a new name relation and/or the deletion of an existing name relation. There are two views on dynamic naming contexts. In the first, changes in the name relations cause the existing naming context to be replaced; the old context is no longer available. In the second view, changes in the name relations cause the creation of a new naming context, and the old context remains available. Both naming contexts have to be named to distinguish between them. Version management uses the latter view. When a change takes place, a new version is created, and the old version remains.

d) Consistent or uniform naming

End users frequently require that a name for a particular entity is the same from the set of contexts from which the entity is most frequently named by a user. In that way the entity will be known by a single name. In each context a soft link may be established. The binding between the soft link and the full source routing name can be created above each context separately or implemented through a new shared context (sometimes referred to as a pseudo-root).

e) Soft links

The handle resulting from name analysis in a given context may be interpreted directly to identify a new context. On the other hand, it may itself be a name which requires resolution. In this way, naming structure can be configured so that the entity identified can be modified without changing the name being held in other parts of the system. This kind of structure can be used to provide soft links, as is commonly done within filing systems.

Annex D

Bibliography

(This annex does not form an integral part of this Recommendation | International Standard)

- [ANSA Naming 93] VAN DER LINDEN, (R.J.): "The ANSA Naming Model".

  ANSA Architecture Report AR.003.01.
- ITU-T Recommendation X.901 (1997) | ISO/IEC 10746-1:1998, *Information technology – Open Distributed Processing – Reference Model: Overview.*
- ITU-T Recommendation X.904 (1997) | ISO/IEC 10746-4:1998, *Information technology – Open Distributed Processing – Reference Model: Architectural Semantics.*
- ITU-T Recommendations X.950 series | ISO/IEC 13235 (Parts 1 and 3), *Information technology – Open distributed processing – Trading function.*
- ITU-T Recommendation X.920 (1997) | ISO/IEC 14750:1999, *Information technology – Open Distributed Processing – Interface Definition Language.*
- ITU-T Recommendation X.650 (1996) | ISO/IEC 7498-3:1997, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing.*
- Naming Service Specification Clause 3, CORBAServices™, OMG®.
- X/Open Federated Naming – The Open Group.
- X/Open Directory Service – The Open Group.
- DCE – The Open Group.

Index

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks and open system communications**
Series Y	Global information infrastructure
Series Z	Programming languages