INTERNATIONAL STANDARD

ISO 19119

First edition
2005-02-15

Geographic information — Services

Information géographique — Services

Reference number
ISO 19119:2005(E)

PDF disclaimer

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

Adobe is a trademark of Adobe Systems Incorporated.

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

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

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

ISO 19119 was prepared by Technical Committee ISO/TC 211, *Geographic information/Geomatics*.

Introduction

The widespread application of computers and use of geographic information systems (GIS) have led to the increased analysis of geographic data within multiple disciplines. Based on advances in information technology, society's reliance on such data is growing. Geographic datasets are increasingly being shared, exchanged, and used for purposes other than their producers' intended ones. GIS, remote sensing, automated mapping and facilities management (AM/FM), traffic analysis, geopositioning systems, and other technologies for Geographic Information (GI) are entering a period of radical integration.

This International Standard provides a framework for developers to create software that enables users to access and process geographic data from a variety of sources across a generic computing interface within an open information technology environment.

— "a framework for developers" means that this International Standard is based on a comprehensive, common (i.e. formed by consensus for general use) plan for interoperable geoprocessing;

— "access and process" means that geodata users can query remote databases and control remote processing resources, and also take advantage of other distributed computing technologies, such as software delivered to the user's local environment from a remote environment for temporary use;

— "from a variety of sources" means that users will have access to data acquired in a variety of ways and stored in a wide variety of relational and non-relational databases;

— "across a generic computing interface" means that ISO 19119 interfaces provide reliable communication between otherwise disparate software resources that are equipped to use these interfaces;

— "within an open information technology environment" means that this International Standard enables geoprocessing to take place outside of the closed environment of monolithic GIS, remote sensing, and AM/FM systems that control and restrict database, user interface, network and data manipulation functions.

Geographic information — Services

1 Scope

The scope of this International Standard is as follows:

Identification and definition of the architecture patterns for service interfaces used for geographic information and definition of the relationships to the Open Systems Environment model.

This International Standard presents a geographic services taxonomy and a list of example geographic services placed in the services taxonomy.

This International Standard prescribes how to create a platform-neutral service specification, and how to derive platform-specific service specifications that are conformant with this.

This International Standard provides guidelines for the selection and specification of geographic services from both platform-neutral and platform-specific perspectives.

2 Conformance

Any product claiming conformance with this International Standard shall pass all the requirements described in the abstract test suite given in Annex A.

NOTE The definition of an abstract test suite appears in ISO 19105.

3 Normative references

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

ISO/IEC 10746-1:1998, *Information technology — Open Distributed Processing — Reference model: Overview — Part 1*

ISO/IEC 10746-2:1996, *Information technology — Open Distributed Processing — Reference model: Foundations*

ISO/IEC TR 14252:1996, *Information technology — Guide to the POSIX Open System Environment (OSE)*

ISO/TS 19103: —[1], *Geographic information — Conceptual schema language*

ISO 19115:2003, *Geographic information — Metadata*

1) To be published.

4 Terms and definitions

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

4.1
service
distinct part of the functionality that is provided by an entity through **interfaces** (4.2)

[adapted from ISO/IEC TR 14252]

NOTE See 7.2 for a discussion of service.

4.2
interface
named set of **operations** (4.3) that characterize the behaviour of an entity

NOTE See 7.2 for a discussion of interface.

4.3
operation
specification of a transformation or query that an object may be called to execute

NOTE 1 An operation has a name and a list of parameters.

NOTE 2 See 7.2 for a discussion of operation.

4.4
interoperability
capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

[ISO/IEC 2382-1]

4.5
service chain
sequence of **services** (4.1) where, for each adjacent pair of services, occurrence of the first action is necessary for the occurrence of the second action

4.6
workflow
automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules

4.7
viewpoint
⟨**on a system**⟩ form of abstraction achieved using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within a system

[ISO/IEC 10746-2]

4.8
enterprise viewpoint
viewpoint (4.7) on an ODP system and its environment that focuses on the purpose, scope and policies for that system

4.9
information viewpoint
viewpoint (4.7) on an ODP system and its environment that focuses on the semantics of information and information processing

4.10
computational viewpoint
viewpoint (4.7) on a system and its environment that enables distribution through functional decomposition of the system into objects which interact at **interfaces** (4.2)

4.11
engineering viewpoint
viewpoint (4.7) on an ODP system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system

4.12
technology viewpoint
viewpoint (4.7) on an ODP system and its environment that focuses on the choice of technology in that system

4.13
distribution transparency
property of hiding from a particular user the potential behaviour of some parts of a distributed system

[ISO/IEC 10746-2]

NOTE Distribution transparencies enable complexities associated with system distribution to be hidden from applications where they are irrelevant to their purpose.

5 Abbreviated terms

ADO	ActiveX Data Objects
API	Application Programming Interface
CCM	Client Configuration Manager
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CICS	Customer Information Control System
DAG	Directed Acyclic Graph
DCOM	Distributed Component Object Model
DCP	Distributed Computing Platform
DEM	Digital Elevation Model
DNA	Distributed interNet Applications
EDOC	Enterprise Distributed Object Computing
DTD	Document type definitions
EJB	Enterprise Java Beans
EOSE	Extended Open Systems Environment Model
ERP	Enterprise Resource Planning
GIOP	General Inter-ORB Protocol
GUI	Graphic User Interface
HIS	Information Technology Human Interaction Service
HTI	Human Technology Interface
HTML	Hypertext Markup language
HTTP	Hypertext Transfer Protocol
IDL	Interface Definition Language
IIOP	Internet Inter-ORB Protocol
IIS	Internet Information Server
IT	Information Technology

J2EE	Java 2 Enterprise Edition with EJB
JDBC	Java Data Base Connectivity
JSP	Java Server Pages
JINI	Sun's open architecture that enables developers to create network-centric services
JNDI	Java Naming and Directory Interface
JTA	Java Connector Architecture
JTS	Java Transaction Service
MAPI	Messaging Application Programming Interface
MS MTS	Microsoft Transaction Server
MSMQ	Microsoft Message Queuing
MTS	Microsoft Transaction Server
OCL	Object Constraint Language
ODBC	Open Database Connectivity
ODMG	Object Database Management Group
ODP	Open Distributed Processing (see RM-ODP)
OGC	Open GIS Consortium
OMG	Object Management Group
OODB	Object-oriented database
ORB	Object Request Broker
OSE	Open Systems Environment
RMI	Remote Method Invocation
RM-ODP	Reference Model of Open Distributed Processing (ISO/IEC 10746)
RPC	Remote Procedure Call
SDAI	Standard Data Access Interface (ISO 10303-22)
SOAP	Simple Object Access Protocol
SOF	Service Organizer Folder
SQL	Structured Query Language
UML	Unified Modelling Language
URI	Uniform Resource Identifier
XML	Extensible Markup Language
XML RDF	XML Resource Description Framework
XSLT	XML Stylesheet Language Transformations

6 Overview of geographic services architecture

6.1 Purpose and justification

The definition of service includes a variety of applications with different levels of functionality to access and use geographic information. While specialized services will appropriately remain an area for proprietary products, standardization of the interfaces to those services allows interoperability between proprietary products. Geographic information system and software developers will use these standards to provide general and specialized services that can be used for all geographic information. The approach of this International Standard is integrated with the approaches being developed within the more general world of information technology.

The geographic services architecture specified in this International Standard has been developed to meet the following purposes:

— provide an abstract framework to allow coordinated development of specific services;

— enable interoperable data services through interface standardization;

- support development of a service catalogue through the definition of service metadata;
- allow separation of data instances and service instances;
- enable use of one provider's service on another provider's data;
- define an abstract framework which can be implemented in multiple ways.

This International Standard extends the architectural reference model defined in ISO 19101, in which an Extended Open Systems Environment (EOSE) model for geographic services is defined.

6.2 Interoperability reference model based on ISO RM-ODP

This International Standard is developed based on a system architecture approach to system design known as the Reference Model of Open Distributed Processing; see ISO/IEC 10746. Architecture is defined as a set of components, connections and topologies defined through a series of views. The geographic infrastructure enabled by this International Standard will have multiple users, developers, operators and reviewers. Each group will view the system from their own perspective. The purpose of architecture is to provide a description of the system from multiple viewpoints. Furthermore, architecture helps to ensure that each view will be consistent with the requirements and with the other views.

Table 1 shows how the RM-ODP viewpoints are utilized in this International Standard.

Table 1 — Use of RM-ODP viewpoints in this International Standard

Viewpoint Name	Definition of RM-ODP Viewpoint (ISO/IEC 10746-1:1998)	How viewpoint is addressed in this International Standard
enterprise viewpoint	a viewpoint on an ODP system and its environment that focuses on the purpose, scope and policies for that system	This is available in other parts of the ISO 19100 series of standards, e.g., reference model (ISO 19101).
computational viewpoint	a viewpoint on an ODP system and its environment that enables distribution through functional decomposition of the system into objects which interact at interfaces	See Clause 7, computational viewpoint.
information viewpoint	a viewpoint on an ODP system and its environment that focuses on the semantics of information and information processing	See Clause 8, information viewpoint.
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	See Clause 9, engineering viewpoint.
technology viewpoint	a viewpoint on an ODP system and its environment that focuses on the choice of technology in that system	See Clause 10 technology viewpoint; also to be addressed by platform-specific service specifications.

The enterprise viewpoint is concerned with the purpose, scope and policies of an enterprise or business and how they relate to the specified system or service. An enterprise specification of a service is a model of that service and the environment with which the service interacts. It covers the role of the service in the business and the human-user roles and business policies related to the service.

The computational viewpoint is concerned with the interaction patterns between the components (services) of the system, described through their interfaces. A computational specification of a service is a model of the service interface seen from a client, and the potential set of other services that this service requires to have available, with the interacting services described as sources and sinks of information.

The information viewpoint is concerned with the semantics of information and information processing. An information specification of an ODP system is a model of the information that it holds and of the information processing that it carries out.

The engineering viewpoint is concerned with the design of distribution-oriented aspects, i.e., the infrastructure required to support distribution. An engineering specification of an ODP system defines a networked computing infrastructure that supports the system structure defined in the computational specification and provides the distribution transparencies that it defines. ODP defines the following distribution transparencies: access, failure, location, migration, relocation, replication, persistence and transaction. Security may also be a mechanism.

The technology viewpoint describes the implementation of the ODP system in terms of a configuration of technology objects representing the hardware and software components of the implementation. It is constrained by cost and availability of technology objects (hardware and software products) that would satisfy this specification. These may conform to platform-specific standards that are effectively templates for technology objects.

In the computational and information viewpoint clauses of this International Standard, specific approaches that shall be followed for defining geographic information services are provided. For the engineering and technology viewpoints, this International Standard defines how a particular service shall be mapped on to an implementation technology, such as SQL-3/ODBC, ODMG, CORBA, DCOM/OLE, Internet or similar technology.

6.3 Service abstraction

This International Standard defines the approach to defining services that shall be used in the ISO 19100 series of standards. Figure 1 defines the relationship between the various types of service specifications. SV_ServiceSpecification defines services without reference to the type of specification or to its implementation. A SV_PlatformNeutralServiceSpecification provides the abstract definition of a specific type of service but does not specify the implementation of the service. Service types are given in the geographic service taxonomy in 8.3. SV_PlatformSpecificServiceSpecification defines the implementation of a specific type of service. There may be multiple platform-specific specifications for a single platform-neutral specification. SV_Service is an implementation of a service. The requirements for these specifications are addressed in this International Standard, in particular in Clause 10.

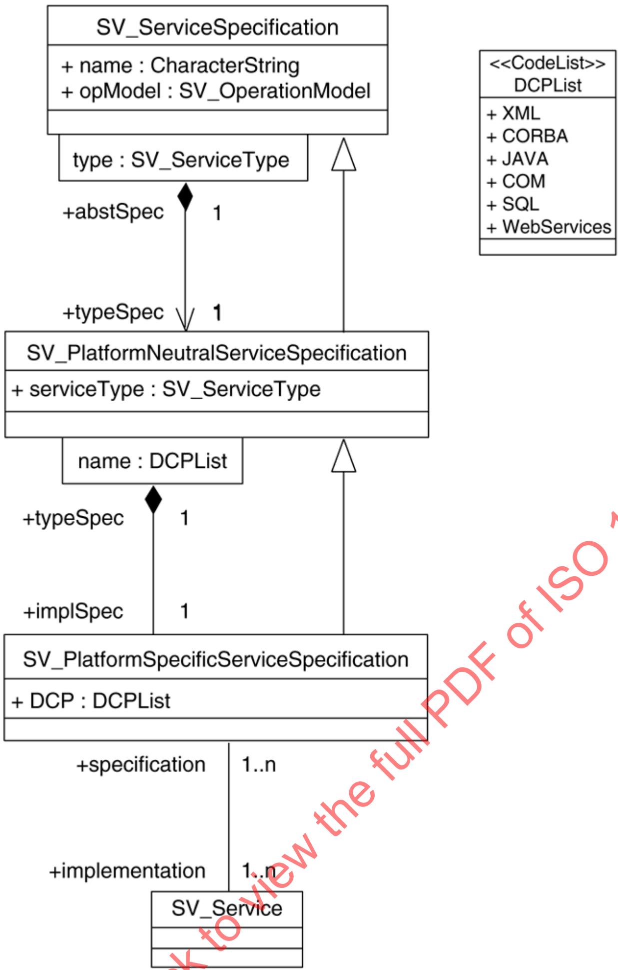

Figure 1 — Abstract and implementation service specifications

6.4 Interoperability

Interoperability is the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.

Two components X and Y (see Figure 2) can interoperate (are interoperable) if X can send requests R for services to Y, based on a mutual understanding of R by X and Y, and if Y can similarly return mutually understandable responses S to X.

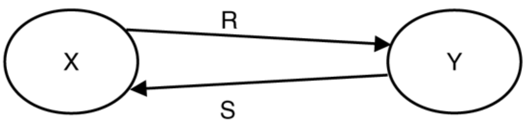

Figure 2 — Interoperability

This means that two interoperable systems can interact jointly to execute tasks. For the geographic domain, the following description of the term “geographic interoperability” is applicable:

“Geographic interoperability” is the ability of information systems to 1) freely exchange all kinds of spatial information about the Earth and about the objects and phenomena on, above, and below the Earth’s surface; and 2) cooperatively, over networks, run software capable of manipulating such information.

The ODP viewpoint abstraction provides a framework for describing a system at several abstraction levels. In this International Standard, interoperability is viewed in terms of the different abstraction levels provided by RM-ODP. This International Standard focuses, from different viewpoints, on how semantic and syntactic interoperability of geographic metadata and geographic data can be supported.

When two different organizations have independently developed distributed systems, each can be described according to the RM-ODP viewpoints, and interoperability between the systems can be discussed with respect to each of the five RM-ODP viewpoints.

For each interoperability aspect, a distinction is made between syntactical interoperability and semantic interoperability. Syntactical interoperability assures that there is a technical connection, i.e. that the data can be transferred between systems. Semantic interoperability assures that the content is understood in the same way in both systems, including by those humans interacting with the systems in a given context.

6.5 Use of other geographic information standards in service specifications

A service specification shall include relevant information models from the appropriate geographic information standards in the ISO 19100 series. The corresponding UML models shall be used in the definition of the service interfaces as appropriate.

6.6 Architecture patterns

An architecture pattern expresses a fundamental structural organization or schema for software services. It identifies a set of services, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them. Services, implemented by classes and objects, may use design patterns but this level of detail is outside the scope of this International Standard.

The Table 2 provides a listing of the elements of a pattern. When specific architecture patterns are defined in this International Standard, these elements shall be used.

Table 2 — Elements of a pattern

Element of a pattern	Description of element
Name	The name is a word or short meaningful phrase that describes the pattern. The name is extremely important, since it is used to reduce communication overhead. Nicknames or synonyms may be provided.
Problem	This is a statement of the problem which describes its intent, goals and objectives it wants to reach within the given context and forces. Often the forces oppose these objectives as well as each other.
Context	Context defines the preconditions under which the problem and its solution seem to recur, and for which the solution is desirable. This defines the pattern's applicability. It can be thought of as the initial configuration of the system before the pattern is applied.
Forces	The forces are considerations that must be weighed to reach the best solution. Forces define the kinds of trade-offs that must be considered in the presence of the tension or dissonance they create. The forces answer the question: “Why is this a hard problem?”
Structure	Structure defines the static relationships and dynamic rules describing how to realize the desired outcome. The structure description is accomplished through a collaboration diagram.

7 Computational viewpoint: A basis for service chaining

7.1 Component and service interoperability and the computational viewpoint

The computational viewpoint is concerned with describing the entities of a distributed system independent of implementation and semantic content. It describes the interaction patterns between the entities and their interfaces. To be able to interoperate from the computational viewpoints, two systems must be *interface-and-services-interoperable*. Two systems are interface-and-services-interoperable if they agree on the set of services offered by the entities of the two systems and the interfaces to these entities. If standardized interfaces are defined, the entities of one system will be able to request services from entities in another system.

The computational viewpoint clause provides the following:

— defines the concepts of services, interfaces and operations and the relations between these concepts;

— provides an approach to physical distribution of services using an n-tier architecture;

— defines a model for combining services in a dependent series to achieve larger tasks, e.g. service chaining;

— defines a service metadata model to support service discovery through a service catalogue.

7.2 Services, interfaces and operations

Definitions and relationships of several terms are provided in this subclause. These terms are used extensively in this International Standard:

— service: distinct part of the functionality that is provided by an entity through interfaces;

— interface: named set of operations that characterize the behaviour of an entity;

— operation: specification of a transformation or query that an object may be called to execute. It has a name and a list of parameters.

These terms are related to each other as depicted in Figure 3, which shows that services are specified by a set of interfaces that are a set of operations. Interfaces are implemented as ports that make services available to users.

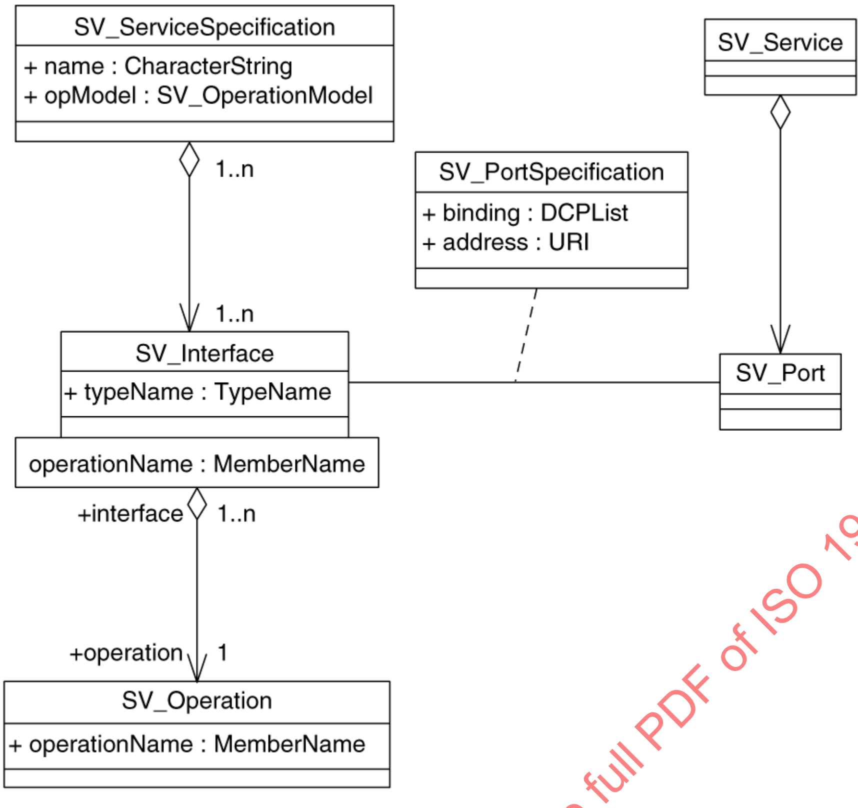

Figure 3 — Service definition relationships

The aggregation of interfaces in a service shall be for the purpose of defining functionality of value to the users. Users in this context are either software agents or human users. A service provides functionality that adds value. The value is apparent to the user who invoked the service.

The aggregation of operations in an interface, and the definition of interface, shall be for the purpose of software reusability. Interfaces shall be defined in order to be reusable for multiple service types. The syntax of an interface may be reused with multiple services with different semantics.

Services of multiple types may be aggregated. The service types shall be defined consistent with the service taxonomy of 8.3. When a service provides functionally beyond that of a single category in the service taxonomy, it shall be an aggregate service. Services' chaining results in aggregate services as defined in 7.3.5.

Interfaces are abstract specifications separate from the concrete deployment or data format bindings. The specification of an interface shall include a static portion that includes definition of the operations. The specification of an interface shall include a dynamic portion that includes any restrictions on the order of invoking the operations.

An implementation of an interface is a port. The implementation includes implementation of the platform-specific specification and a method to identify the service, e.g. an address.

An implementation of a service may be associated with a specific dataset or it may be a service that can be used to operate on multiple, unspecified datasets. The first case is referred to as a tightly coupled service. The second case is referred to as a loosely coupled service (see 7.4.1).

Interfaces are defined through operations. An operation specifies a transformation on the state of the target object or a query that returns a value to the caller of the operation. An operation shall be an abstract description of an action supported by the interface. Operations contain parameters.

7.3 Service chaining

7.3.1 Introduction to service chaining

Subclause 7.3 defines a model for combining services in a dependent series to achieve larger tasks. Subclause 7.3 addresses the syntactic issues of service chaining, e.g. the data structure of a chain. Semantic issues associated with service chaining are addressed in 8.6, e.g.: does a specific chain produce a valid result? Examples of service chaining are provided in Annex B.

This International Standard enables users to combine data and services in ways that are not pre-defined by the data or service providers. This level of data/service interoperability will be achieved in stages. At first, service catalogues will hold entries with tight data/service binding. Eventually the infrastructure will be available for a user to determine which data can be acted on by a loosely coupled service. This capability will be enabled by the infrastructure of the larger domain of IT.

Based on the ODP definition of chain, a service chain is defined as a sequence of services where, for each adjacent pair of services, occurrence of the first action is necessary for the occurrence of the second action.

7.3.2 Anatomy of a service chain

7.3.2.1 Chains as directed graphs

The action of making the input of one service dependent upon another service leads to treating service chains as directed graphs, where each service is a node in the graph and references to service interactions form the edges. In some cases, the directed graph structure is implicit. In other cases, it is necessary to make the notion of a processing graph explicit and allow such graphs to be considered as entities in their own right. Explicit representation of a service chain allows the chain to be visually represented and passed to a chain-execution service, e.g. workflow service.

A node in the directed graph is a representation of the service. When explicitly formed into a data structure, a service node contains two types of information: parameters and sources. Parameters in a service node provide the configurations of the service for the particular chain in which the service class is being used. Sources in a service node indicate the sources of input data to the node.

The arcs of a directed graph representing a service chain can be of several types. These types of arcs are detailed below as service interactions.

A key aspect of a directed graph is whether it is cyclic or acyclic. The case of directed graphs without loops, i.e. acyclic, is simpler. In some applications, an iterative approach is needed; therefore, the chain will be cyclic with conditions in the control function to address convergence.

Chains can be considered templates or as immutable graphs. A template is a directed graph that defines the chain based on abstract classes, including identification of each service type. A template can be instantiated as an immutable graph, at which time the service instances are fixed.

The following are additional elements characterizing a service chain:

— parallel vs. serial chains: Does the directed graph have parallel paths based on branches or are only serial chains permitted? Potential branch types include: if/else, merge, switch and trigger.

— iteration: Does a node in the directed graph operate as an iteration, e.g. while and count loops?

— data transport types: Does the directed graph allow variations in the links between nodes reflecting different methods for transporting data or invoking the service?

— parameters in nodes: Do the description of nodes in the directed graph contain parameters that can be changed?

— variations in control design pattern: Pull processing vs. push processing.

7.3.2.2 UML modelling of a chain

Figure 4 provides a UML model of a chain.

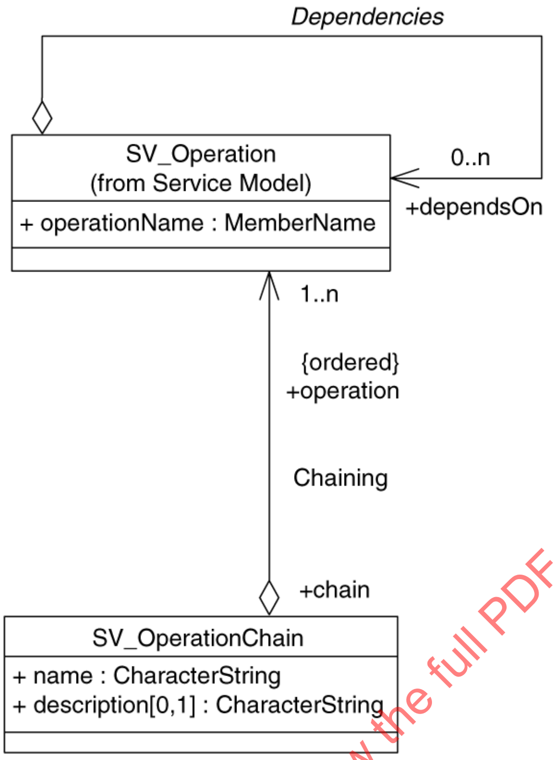

Figure 4 — Chain

In accordance with ISO/TS 19103, the modelling of directed graphs for service chains shall be done using the Unified Modeling Language (UML) activity graphs. An activity graph represents the states of executing a service chain. Table 3 identifies the elements of an activity graph.

Table 3 — UML Activity Graph entities

Element of an UML activity graph	Description for service chaining
Activity State	a state that represents the execution of a service, typically triggered by the invocation of an operation
Transition	a relationship between two states. A transition indicates that specified Actions are performed in the first state and the second state is entered when a specified Event occurs and specified Guard Conditions are satisfied.
Branch/Merge	the beginning/end of alternative threads in an activity graph
Fork/Join	the beginning/end of concurrent threads in an activity graph
Signal	an asynchronous communication between services intended to trigger transitions in an activity graph

7.3.3 Services organizer folder

Services are of many types as indicated in 8.3. Only a subset of available services is applicable to a specific situation, e.g. image analysis. A service organizer folder (SOF) is an aid for users in finding services applicable to their situation. A user may construct an SOF and then make that SOF available to other users performing tasks in a similar situation.

A services organizer folder is a data structure that shall contain references to a set of services that are applicable to a given situation. The SOF need not contain service chains but may contain just individual services.

7.3.4 Services to enable service chaining

Table 4 provides a list of services that are needed to enable service chaining. These services are placed in the extended OSE framework. Details on the services can be found in Clause 8. Some of the services are generic to all IT domains. Other services are specific to geographic data and the large size of geographic data sets.

Table 4 — Services to enable service chaining

Services needed to enable service chaining		
OSE categories	**Generic IT services**	**Geographic services**
human interaction services	▪ service-centric service for defining, controlling and statusing service chains ▪ catalogue-centric service that views and browses metadata about services	▪ catalogue-centric service that locates, browses, and manages metadata about spatial data ▪ spatial-centric service for editing, displaying, querying, and analyzing map data ▪ calculation-centric service allowing viewing and manipulation of geographic data using a spreadsheet format
workflow/task services	▪ workflow enactment service to define, invoke, status and control service chaining (interaction with other workflow services, optional) ▪ service chain validation service ▪ resource reservation and co-allocation mechanism for both storage system and other resources such as networks, to support the end-to-end performance guarantee required for predictable transfer	—
processing services	—	▪ geographic processing services (see Clause 8)
model/Information management services	▪ service instance metadata catalogue, with discovery and management sub-services ▪ service type registry, with discovery and management sub-services ▪ brokering ▪ mediation	▪ geographic dataset instance ▪ geographic metadata catalogue with discovery, access and management sub-services
system management service	▪ authorization and authentication ▪ payment methods ▪ privacy of client ▪ performance measurement and estimation techniques for key resources involved in data grid operation, including storage systems, networks, and computer ▪ instrumentation services that enable the end-to-end instrumentation of storage transfer and other operation	—
communication services	▪ messaging mechanisms ▪ large data object transfer ▪ remote file and executable management: provides access to secondary storage as if it were local ▪ format conversions	▪ geographic format conversions

7.3.5 Architecture patterns for service chaining

7.3.5.1 Introduction

The remainder of this subclause provides architecture patterns for service chaining. The architecture patterns use the template provided in 6.6.

There are many options for the allocation of service chaining services to components. Different allocation approaches reflect different priorities for different applications: user in the loop vs. user supervision. To demonstrate the breadth of the trade space defined by this variation, three design patterns are offered that vary the allocation of the control function:

— user defined (transparent) chaining: the human user manages the workflow;

— workflow-managed (translucent) chaining: in which the human user invokes a workflow management service that controls the chain and the user is aware of the individual services;

— aggregate service (opaque): in which the user invokes a service that carries out the chain, with the user having no awareness of the individual services.

In addition to the difference in visibility of the services to the user, a key distinction between these patterns is the difference in control. In transparent chaining, the control is exclusively with the user. In translucent, a workflow service is present which controls the chain execution, perhaps with oversight by the human. In the aggregate pattern, the aggregate service exclusively performs the control function with no visibility by the user.

7.3.5.2 User-defined (transparent) chaining

7.3.5.2.1 Name

As the name implies, the user defines and controls the order of execution of the individual services. Details of the services are not hidden from the user; hence, the alias for this pattern is Transparent Chaining.

7.3.5.2.2 Problem

In this pattern, the user is knowledgeable about how services can be combined. The user discovers and evaluates the available services, determines their fitness to the need, determines a valid sequence of services and controls the chaining. This pattern presupposes a knowledgeable user. The user is provided information sufficient to make the control decisions.

7.3.5.2.3 Context

The user refers to a service catalogue in order to discover services of interest. A specific chain does not exist before the user begins. The user has the ability to define a valid chain and/or be able to modify the implied chain if there are failures in execution.

7.3.5.2.4 Forces

The user must be able to design an efficient chain that will execute. The inputs and outputs of the individual services must be compatible, or an intervening service must be added, e.g. format translation. These patterns assume that each service has sufficient resources to run efficiently, but the user may need to choose services based on network considerations, e.g. network bandwidth, security, authorization. The semantic correctness of the chain is judged by the user; issues such as when data is re-gridded in a chain will affect the validity of the results. A user may iterate a chain until an acceptable result is achieved, resulting in a chain that can be saved and used by others, perhaps using the workflow-chaining pattern.

7.3.5.2.5 Structure

The user-defined chaining architecture pattern is shown in Figure 5.

NOTE The unique feature of the transparent pattern is that the chain is defined and controlled by the user. In the figure, the user discovers an available service through a catalogue service. Alternatives for the user to select services are part of this pattern. For example, a service organizer folder could be substituted for the catalogue.

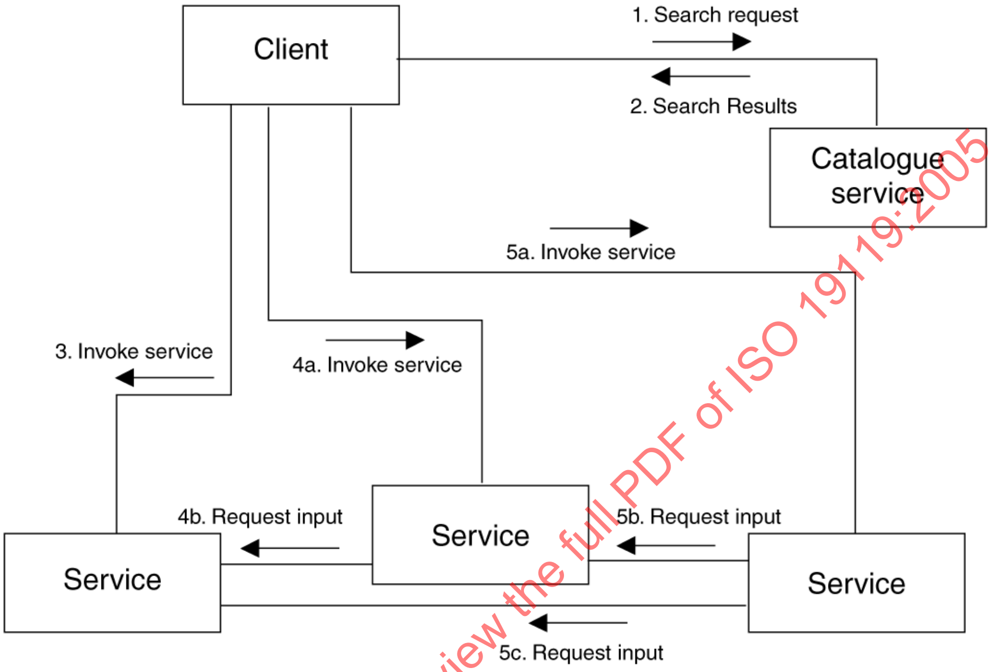

Figure 5 — Transparent chaining — user-defined chaining architecture pattern

Table 5 — Description of steps in Figure 5

Step 1. Search request	A human uses a client to send a search request (or series of searches) to a catalogue service. The catalogue service provides queries on service metadata.
Step 2. Search results	Catalogue service returns metadata about services of interest to the user. For this example, the user has found three services that will be chained.
Step 3. Invoke service	User invokes a service using the client, causing a result to be available for a subsequent service.
Step 4a. Invoke service step 4b. Request input	User invokes a second service using the client. The request includes a reference to the results from the previous step. The service creates a result that is available for the next service.
Step 5a. Invoke service Step 5b. Request input Step 5c. Request input	User invokes a third service using the client. The request includes references to the two previous services. This third service returns a result to the client.

7.3.5.3 Workflow-managed (translucent) chaining

7.3.5.3.1 Name

As the name implies, in this pattern the execution of the chain is managed by a workflow service (or multiple workflow services). The user's involvement in the steps of the chain is mostly one of watching the chain execute the individual services that are apparent to the user, hence the alias of translucent chaining. A key distinction for this pattern is the existence of a defined chain prior to the user executing the pattern.

7.3.5.3.2 Problem

In this pattern, the user relies on a workflow service to execute a predefined chain of services. The user has determined an existing chain assumed to produce results of interest to the user. The user may need to provide parameters particular to specific instance, but relies on the workflow service to carry out the chain.

7.3.5.3.3 Context

The user knows of a workflow service and has selected a chain of interest. The user interacts with the workflow service to execute the chain including providing parameters specific to the data instances of interest to the user.

7.3.5.3.4 Forces

To reduce the user's workload, the workflow service handles details of the distributed computing aspects of executing the chain. Although the predefined chain is assumed to have a degree of semantic validity, by evaluating interim results, the user can evaluate the semantic validity of the specific instance of this processing. For example for a service that includes an iterative algorithm, the user may need to judge if convergence to a sufficient degree of accuracy has been achieved.

7.3.5.3.5 Structure

The workflow-managed (translucent) chaining architecture pattern is shown in Figure 6.

NOTE There may be multiple workflow services. If there is more than one, the workflow services must coordinate to carry out the predefined chain. In the extreme case, each service in the chain contains a workflow service and the chain is passed along with the service results. The unique features of the translucent pattern are the existence of a predefined chain and the user's awareness of the chain.

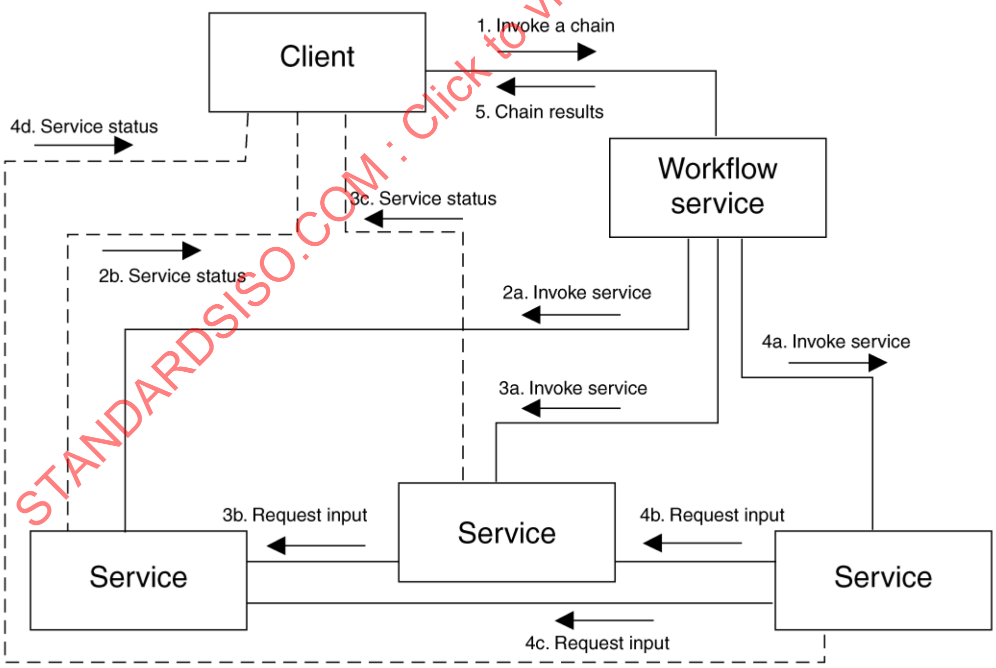

Figure 6 — Translucent chaining — Workflow-managed chaining architecture pattern

Table 6 — Description of steps in Figure 6

Step 1. Invoke a chain	A human uses a client to request that a workflow service execute a chain. The user may be allowed to modify some aspects of the chain prior to execution.
Step 2a. Invoke service Step 2b. Service status	The workflow service determines the services in the chain and invokes the first service. The service informs the workflow service of the completion of the task. Status of the service may be provided directly to the client. The client may stop the workflow.
Step 3a. Invoke service Step 3b. Request input Step 3c. Service status	Upon notification of completion of the first service, the workflow service determines the next service in the chain and invokes it. The second service requests results from the first service. The service informs the workflow service of the completion of the task. Status of the service may be provided directly to the client. The client may stop the workflow.
Step 4a. Invoke service Step 4b. Request input Step 4c. Request input Step 4d. Service status	Upon notification of completion of the second service, the workflow service determines the next service in the chain and invokes it. The third service requests results from the first and second services. The service informs the workflow service of the completion of the task. Status of the service may be provided directly to the client. The client may stop the workflow.
Step 5. Chain results	Upon notification of completion of the last service, the workflow service informs the client of the completion of the chain.

7.3.5.4 Aggregate service (opaque-chaining)

7.3.5.4.1 Name

As the name implies, in this pattern the services appear as a single service that handles all coordination of the individual services behind the aggregate service. The user has no awareness that there is a set of services behind the aggregate, hence the alias of opaque chaining.

7.3.5.4.2 Problem

In this pattern, the user relies on an aggregate service to execute a predefined chain of services. The user has discovered the aggregate service and may have no knowledge of how the aggregate accomplishes the service. The user may need to provide parameters particular to the specific instance, but relies on the aggregate service to carry out the chain.

7.3.5.4.3 Context

The user knows of an aggregate service, perhaps not knowing that a chain of services implements the aggregate. The user interacts with the aggregate service to execute the chain including providing parameters specific to the data instances of interest to the user.

7.3.5.4.4 Forces

To reduce the user's workload, the aggregate service handles all details of the multi-service aspects of executing the chain. Although the aggregate service chain is assumed to have a degree of semantic validity, by evaluating interim results, the user can evaluate the semantic validity of the specific instance of this processing. For example, for a service that includes an iterative algorithm, the user may need to judge if a sufficient degree of convergence has been achieved. These intermediate results need not reveal the underlying services.

7.3.5.4.5 Structure

The aggregate service (opaque-chaining) architecture pattern is shown in Figure 7.

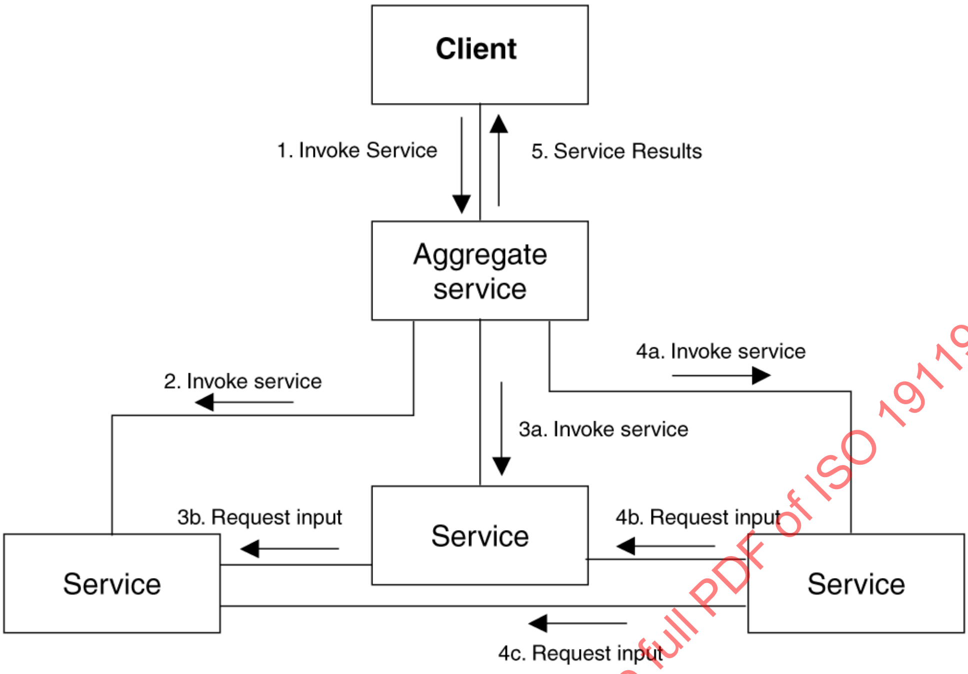

Figure 7 — Opaque chaining

Table 7 — Description of steps in Figure 7

Step 1. Invoke a service	A human uses a client to request that an aggregate service execute a chain. The user may have no knowledge that the service is implemented using a chain of services.
Step 2. Invoke service	The aggregate service determines the services in the chain and invokes the first service. The service informs the aggregate service of the completion of the task.
Step 3a. Invoke service Step 3b. Request input	Upon notification of completion of the first service, the aggregate service determines the next service in the chain and invokes it. The second service requests results from the first service. The service informs the aggregate service of the completion of the task.
Step 4a. Invoke service Step 4b. Request input Step 4c. Request input	Upon notification of completion of the second service, the aggregate service determines the next service in the chain and invokes it. The third service requests results from the first and second services. The service informs the aggregate service of the completion of the task.
Step 5. Chain results	Upon notification of completion of the last service, the aggregate service informs the client of the completion of the chain.

7.3.6 Variations on chaining patterns

The three chaining patterns discussed above could be combined in a variety of ways.

Each of the lowest level services shown in the pattern diagrams could in turn implement a chain. This is recursive composition of services supported by the opaque pattern. A service chain can become a new service. The ability to define recursive composition of services provides scalability and support for top-down progressive refinement as well as for bottom-up aggregation.

The patterns could be used to define how a library of chains is constructed. A knowledgeable user could build chains using the transparent pattern. Through iterative use of the transparent pattern, a chain is constructed that produces valid results. Chains are then made available for wider use following the translucent pattern. Certain chains may become routinely used and an aggregate service is built as an interface.

An example need for a translucent or opaque chaining pattern occurs in decision support. The decision-maker is an individual using decision-support aids to help make a decision. An example of a decision-support aid is a service chain. The decision-support aid developer is an individual who “integrates” chains of services into decision-support aids.

Another type of service interaction can be considered as chaining, where a user makes a request of a lead service and the lead service then invokes a secondary service, which invokes a tertiary service. Each of the services responds to the request when it has sufficient information from the underlying services. In this way, there is no explicit chain but rather a chain is implied.

7.4 Service metadata

7.4.1 Introduction

Service metadata records can be managed and searched using a catalogue service as is done for dataset metadata. In order to provide a catalogue for discovering services, a schema for describing a service is needed. This section defines a metadata model for service instances. The metadata elements for a service provide sufficient information to allow a client to invoke the service based on the metadata record. In order to place the service metadata in context, three types of entities need to be described:

— service instance: a service instance is the service itself, hosted on a specific set of hardware and accessible over a network;

— service metadata: a service metadata record describes a service instance, including a description of the services operations and an “address” to access the specific service instance;

— service type: in some cases a service metadata record will describe a service instance which is of a “well-known type”. By well-known type, it is meant that the service conforms to a published definition of a service type, i.e. a platform-specific service specification. Some clients will be able to access only services of well-known type. A user could search the service metadata catalogue to find instances of a specific well-known service type. A service registry is defined to be the service that provides details on service types.

A service instance may be tightly coupled with a dataset instance, or it may be un-associated with specific data instances, i.e. loosely coupled. Loosely coupled services may have an association with data types through the service type definition. In the tightly coupled case, the service metadata shall describe both the service and the geographic dataset, the latter being defined in accordance with ISO 19115. Also for the tightly coupled case, the permitted values in the SV_Operations shall be constrained by the values defined by the datasets associated with the service. For the loosely coupled case, dataset metadata need not be provided in the service metadata.

Figure 8 compares the structure of service specification with service metadata. A service specification describing a service type includes an aggregation of interfaces. Interfaces are an aggregation of operations. The aggregation of operations in an interface, and the definition of interface, is for the purpose of software reusability. To describe a service instance, the aggregation associated with an interface is not needed. A service instance is described by SV_ServiceIdentification and SV_OperationMetadata. SV_ServiceIdentification describes the functionality provided by the service. SV_OperationMetadata describes the operations by which the service instance can be invoked.

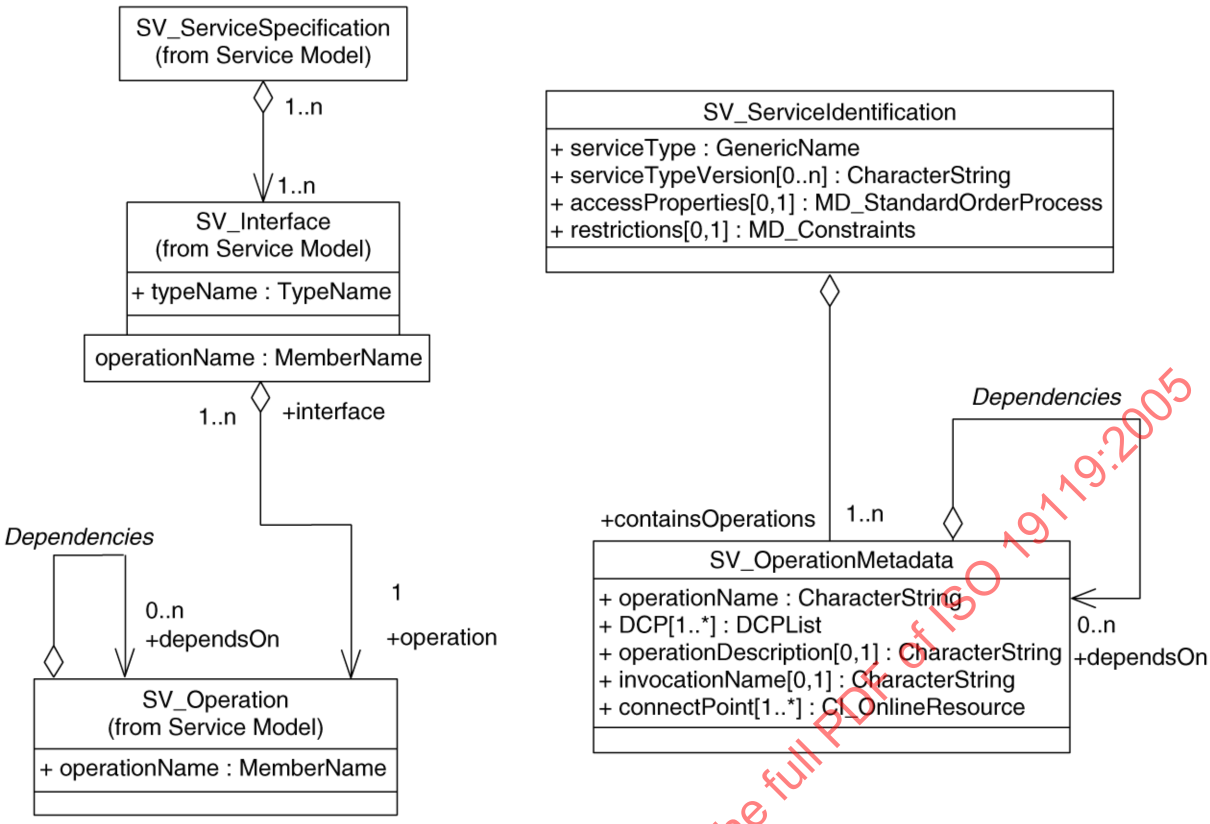

Figure 8 — Services and service metadata

7.4.2 Service metadata main class diagram

Figure 9 provides a UML static model of service metadata for a service instance.

The structure of service metadata includes three major classes: a section of basic **service** metadata (SV_ServiceIdentification class) that provides a general description of the service and two sections that describe the **operations** (SV_OperationMetadata) and **data** (MD_DataIdentification) available from a particular service. The service metadata schema is defined using elements of ISO 19115. The "Records and Class Metadata" package is described in ISO/TS 19103, Conceptual Schema Language.

The classes shown in Figure 9 are described in Annex C.

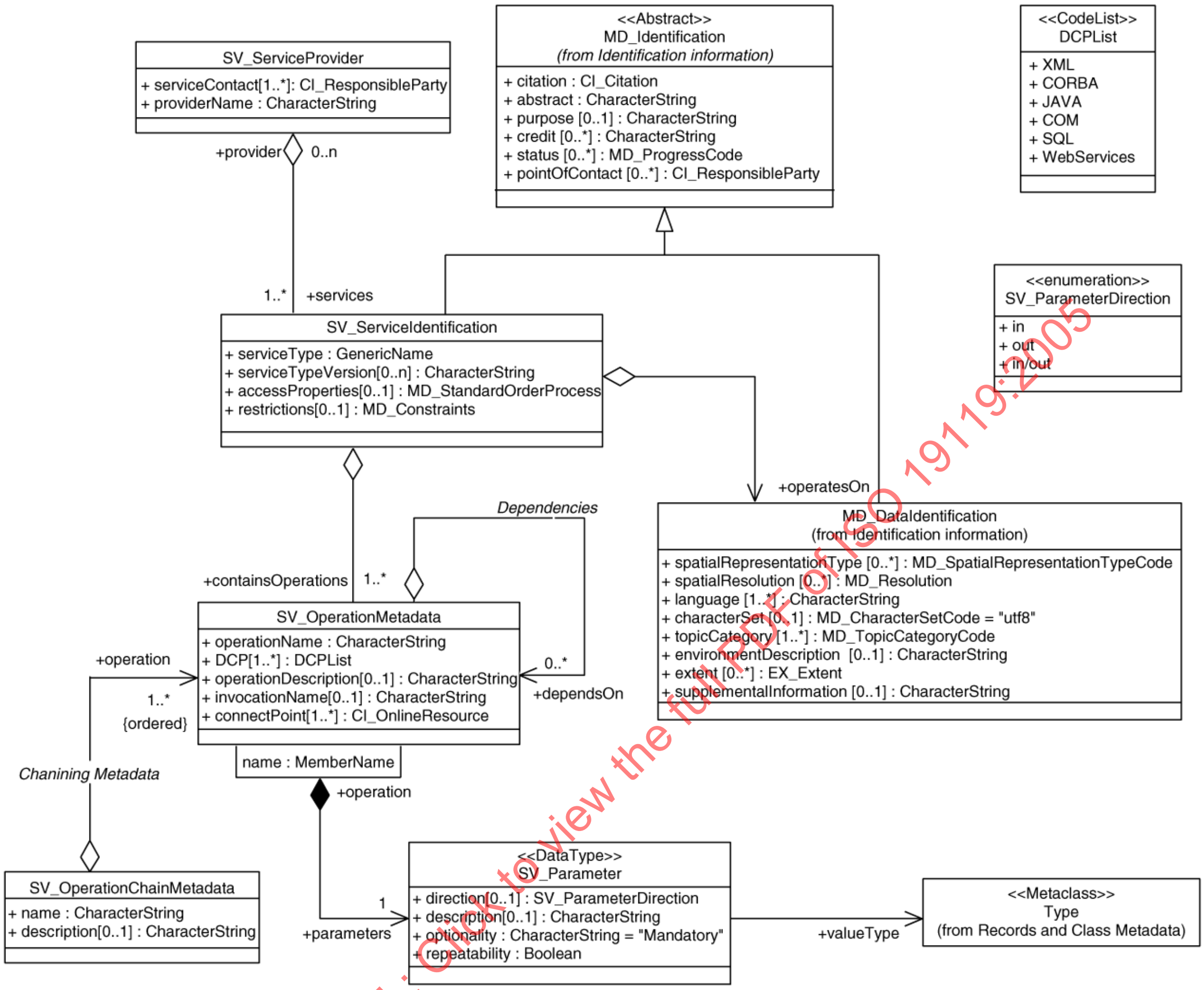

Figure 9 — Service metadata class diagram

7.4.3 Data dictionary for geographic service metadata

A data dictionary for geographic service metadata is provided in Annex C. The entities in Figure 9 are defined in the data dictionary.

7.5 Service instance of unknown type

One approach for a service catalogue is to assume that the metadata alone provide sufficient information for a client to invoke the service. An alternate approach is to require that the metadata record for the instance make reference to a service type definition. For example, a metadata record for a service instance of coordinate transformation would reference the OGC implementation specification. Table 8 describes the implications of assuming that the service is of a well-known type or is not. A key aspect to dealing with services of unknown type is to assume the presence of intelligence in the client, e.g. a human in the loop. As a human is able to interpret the semantics intended by the service metadata where a software client cannot, the table shows what is feasible.

Table 8 — Options for well known type in service metadata

	Services in the Catalogue are of a known type	Services in the Catalogue may be of an unknown type
human is in the loop with the client	feasible	Conceivably feasible assuming that the client interaction is done with opaque messages, so there can be a generalized client that can create marked-up messages. The human fills in the message based on the operations descriptions retrieved from the catalogue.
machine-to-machine; no human as the client	This is feasible if the client implements a metamodel for the known service, in addition to the syntax for the service.	questions if this is feasible

7.6 Simple service architecture

The following simplifying assumptions should be considered when implementing a message-based architecture to support service chaining. Systems compliant with this set of guidelines shall be referred to as instances of simple service architectures. Systems should comply with the simple service architecture.

— Message-operations. For simplicity, it is desirable to model the operations as messages. A message operation shall consist of a request and response. Requests and responses contain parameters as the payload, which is transferred in a uniform manner independent of content. Simple applications are characterized by message exchange patterns such as one-way (or event), and two-way (or synchronous) request response interactions. A service specification should make such simple exchange applications as easy as possible to create and to use.

— Separation of control and data. A client controlling a service may not want the full results of the service. For example, the user may have no need for the potentially voluminous intermediate products in a service chain. Only the final result of a service chain may be needed by the client. Therefore, operations of an interface should separate the control of the service from the access to the data resulting from a service. A client should have the option of receiving just the status of an operation and, separately, the data should be accessible through a separate operation.

— Stateful vs. stateless service. For simplicity it is desired that a service be stateless, i.e. that a service invocation be composed of a single request-response pair with no dependence on past or future interactions. This will not always be possible. For some services, preconditions must be set and iteration may be required; then it will be necessary to model the service with a state diagram having multiple states. Transitions between the states are triggered by operations.

— Known service type. All service instances are of specific service types and the client knows the type prior to runtime. Clients shall contain software for accessing the service type prior to encountering service instances of the type in an implemented architecture. The assumption is that the client knows the service types.

— Adequate hardware. The services described in this International Standard are software implementations running on hardware hosts. This International Standard assumes that the issues of hardware hosting of the software are transparent to the user. It is assumed that the service has adequate hardware, i.e. hardware assignment is transparent to user.

8 Information viewpoint: A basis for semantic interoperability

8.1 Information model interoperability and the information viewpoint

Achieving *information model interoperability* is one of the main goals of the ISO 19100 series of International Standards. Many of the other International Standards in this series, i.e. ISO 19107, ISO 19115 and others, are primarily focusing on defining the content of the information that is being processed by the services and exchanged between services. The information viewpoint is defined in the ODP to include a static information model and a dynamic information model. The semantics of service interactions, e.g. what services make sense to chain, are developed in 8.6.

To be able to interoperate in the information viewpoint, two systems must be *information model interoperable*. To achieve information model interoperability the two systems must be both *syntactically interoperable and semantically interoperable*:

— syntactically interoperable: two systems are syntactically interoperable if they use the same structure for the information that flows between the systems and is processed by the systems;

— semantically interoperable: two systems are semantically interoperable if they have a common understanding of the semantics of the information that flows between the systems and is processed by the systems.

The common structural models, being defined by the information models, address syntactic interoperability. In the ISO 19100 series of International Standards the models are based on a generic-feature model, which allows for representation of various types of features, all having the same structure. To achieve semantic interoperability for feature types, it is, in addition, necessary to match or make mappings between feature-type definitions from feature-type catalogues. The issues of such information model interoperability will not be addressed in this International Standard.

The information viewpoint in ISO RM-ODP describes the information that flows in a system and is processed by a system. It focuses on the structuring of semantic information, typically the information that will be stored in a database and communicated among the components of a system. An information model is used to describe the information viewpoint. This information model defines the structure and semantics of the information used in system by defining objects, their properties and their relationships.

The information viewpoint is also concerned with the semantics of the information processing. Each particular service will need to define its syntactical interfaces through operations and its semantics through description of the meaning of the operations and their legal sequencing. The latter can be done through pre- and post-conditions and invariants in OCL, and by UML state diagrams.

This section contains a description of a taxonomy of various services. There exist multiple possible taxonomies for services, based on various classification dimensions. The one that is used here is based on the extended OSE model. The purpose of defining a taxonomy in this International Standard is to have one way of identifying geographic extensions to various existing service types. It is not intended to be the only taxonomy to be used in the context of geographic services.

The identification of a required service shall specify whether a needed service is GIS-specific or more IT general, and to which of the six service domains it belongs. The processing-services domain will typically contain a variety of application-area (GI) specific services.

The next sections describe typical IT services and then show some possible geographic services in each of the six service domains.

8.2 Extended open systems environment for geographic services

The model for the information viewpoint is provided by ISO 19101, which defines the Extended Open Systems Environment (EOSE) model for geographic information. The EOSE defines classes of services based on the semantic type of computation that they provide. EOSE provides the functional decomposition of the services for the geographic domain by extending the more general Open System Environment model; see ISO/IEC TR 14252].

Consistent with ISO 19101, this subclause defines six classes of information technology services that shall be used to categorize geographic services.

— Human interaction services are services for the management of user interfaces, graphics, multimedia, and for presentation of compound documents.

— Model/Information management services are services for the management of the development, manipulation, and storage of metadata, conceptual schemas, and datasets.

— Workflow/Task services are services for the support of specific tasks or work-related activities conducted by humans. These services support use of resources and development of products involving a sequence of activities or steps that may be conducted by different persons.

— Processing services are services that perform large-scale computations involving substantial amounts of data. Examples include services for providing the time of day, spelling checkers, and services that perform coordinate transformations, e.g. that accept a set of coordinates expressed using one reference system and convert them to a set of coordinates in a different reference system. A processing service does not include capabilities for providing persistent storage of data or transfer of data over networks.

— Communication services are services for encoding and transfer of data across communications networks.

— System management services are services for the management of system components, applications and networks. These services also include management of user accounts and user access privileges.

Not every information-technology service needs to be changed or specialized to be useful for processing geographic information. A separation between geographic services and IT services is made in ISO 19101. This separation is emphasized because it is essential to identify and make use of general IT services whenever they exist.

8.3 Geographic services taxonomy

8.3.1 Geographic services taxonomy requirements

The following subclauses define the geographic services taxonomy. The taxonomy consists of the titles of the categories (Table 9) and the definitions for the categories. Definitions of the categories are provided in 8.2. Systems compliant to this International Standard shall use the geographic services taxonomy to organize their services. A specific service shall be categorized in one and only one category, unless it is an aggregate service that may perform services from more than one category.

The following subclauses provide examples of geographic services within the geographic services taxonomy. It is not required that a system provide any service listed in these subclauses. It is required that if a system provides a service named in these subclauses that the service shall be categorized as defined in these subclauses. A service catalogue compliant with this International Standard shall categorize service metadata instances in the categories of the geographic service taxonomy.

If a service uses the name of an example service, the service shall provide the functionality that is defined in these subclauses. For example, if a service entitled “catalogue viewer” is provided, it shall perform the services defined for the catalogue viewer in the geographic human interaction services category. Systems providing services should name services as found in the service examples.

Table 9 — Geographic services taxonomy

— Geographic human interaction services
— Geographic model/information management services
— Geographic workflow/task management services
— Geographic processing services
— Geographic processing services — spatial
— Geographic processing services — thematic
— Geographic processing services — temporal
— Geographic processing services — metadata
— Geographic communication services

8.3.2 Geographic human interaction services

Geographic human interaction services shall be a category in the geographic service taxonomy. The following are examples of human interaction services for working with geographic data and services:

— Catalogue viewer: Client service that allows a user to interact with a catalogue to locate, browse, and manage metadata about geographic data or geographic services;

— Geographic viewer: Client service that allows a user to view one or more feature collections or coverages. This viewer allows a user to interact with map data, e.g. displaying, overlaying and querying. An example is the viewer client generator defined in ISO 19128;

— Geographic viewer — animation: Geographic viewer that allows a human to sequence views of the same geographic location at different times;

— Geographic viewer — mosaicing: Geographic viewer that allows combination of views of geographic data for adjacent areas into a single view;

— Geographic viewer — perspective: A perspective geographic viewer allows the viewpoint to be changed; for example, to specify how high off the ground, what direction, and from what angle a viewpoint is seeing a scene;

— Geographic viewer — imagery: Geographic viewer that visualizes coverage data including the mapping of sample dimensions in the coverage to colours in the display;

— Geographic spreadsheet viewer: Client service that allows a user to interact with multiple data objects and to request calculations similar to an arithmetic spreadsheet, but extended to geographic data;

— Service editor: Client service that allows a user to control geographic processing services. Views include understanding a service, composing/scripting service chains, invoking a service, statusing a service, scheduling services for peak performance times, and invoking a service chain;

— Chain definition editor: Provides user interaction with a chain definition service;

— Workflow enactment manager: Provides user interaction with a workflow enactment service;

— Geographic feature editor; Geographic viewer that allows a user to interact with feature data, e.g. displaying, querying; supports feature annotation. The user controls view orientation, perspective, depth cueing, hidden-line/surface, light-sources, transparency, and texture mapping onto the objects. Objects in view can be picked or drawn on to generate new objects in the model.

— Geographic symbol editor: Client service that allows a human to select and manage symbol libraries. ISO 19117 is relevant to symbol libraries;

— Feature generalization editor: Client service that allows a user to modify the cartographic characteristics of a feature or feature collection by simplifying its visualization, while maintaining its salient elements — the spatial equivalent of simplification;

— Geographic data-structure viewer: Client service that allows a user to access part of a dataset to see its internal structure; allows user to request creation of new objects from parts of an object being browsed; allows user to request a check of an object, e.g. type checking.

8.3.3 Geographic model/information management services

The following are examples of model/information management services for working with geographic data and services:

— Feature access service: Service that provides a client access to and management of a feature store. An access service may include a query that filters the data returned to the client. ISO 19125-1 is relevant to feature access;

— Map access service: Service that provides a client access to a geographic graphics, i.e. pictures of geographic data. ISO 19128 is relevant to map access;

— Coverage access service: Service that provides a client access to and management of a coverage store. An access service may include a query that filters the data returned to the client. ISO 19123 and ISO 19111 are relevant to coverage access;

— Coverage access service — sensor: Service that provides access to a coverage where the source of the coverage data is a real-time sensor, i.e. not a persistent store;

— Sensor description service; Service that provides the description of a coverage sensor, including sensor location and orientation, as well as the sensor's geometric, dynamic and radiometric characteristics for geoprocessing purposes;

— Product access service: Service that provides access to and management of a geographic product store. A product can be a predefined feature collection and metadata with known boundaries and content, corresponding to a paper map or report. A product can alternately be a previously defined set of coverages with associated metadata;

— Feature type service: Service that provides a client access to and management of a store of feature type definitions. The static and dynamic information models for a feature type catalogue are provided in ISO 19110;

— Catalogue service: Service that provides discovery and management services on a store of metadata about instances. The metadata may be for dataset instances, e.g. dataset catalogue, or may contain service metadata, e.g. service catalogue. ISO 19115 is relevant to catalogue service for dataset metadata. Annex C of this International Standard is relevant to catalogue service for service metadata.

— Registry service: Service that provides access to store of metadata about types. Types are vocabularies that can be organized and related to each other. Example registries are information community registries, type dictionaries, service registries and schema registries;

— Gazetteer service: Service that provides access to a directory of instances of a class or classes of real-world phenomena containing some information regarding position. An information model for a gazetteer is provided by ISO 19112;

— Order-handling service: Service that provides a client with the ability to order products from a provider, including the formulation of quotes on orders, selection of geographic processing options, submission of an order, statusing of orders and billing and accounting of users' orders;

— Standing order service: Order-handling service that allows a user to request that a product over a geographic area be disseminated when it becomes available. Such dissemination includes receiving, preparing (i.e., reformatting, compressing, decompressing, etc.), prioritizing, and transmitting the geographic information requested through standing queries or profiles.

8.3.4 Geographic workflow/task management services

The following are examples of workflow/task management services for working with geographic data and services:

— Chain definition service: Service to define a chain and to enable it to be executed by the workflow enactment service. This includes information about its starting and completion conditions, constituent activities and rules for navigating between them, user tasks to be undertaken, references to applications which may be invoked, definition of any workflow relevant data which may need to be referenced, etc. Chain definition service may also provide a chain validation service;

— Workflow enactment service: The workflow enactment service interprets a chain and controls the instantiation of services and sequencing of activities. This is done through one or more co-operating workflow management engines, which manage the execution of individual instances of the various services. A workflow enactment service maintains control data either centralized or distributed across a set of workflow engines. Workflow control data include the internal state information associated with the various services under execution and may also include check-pointing and recovery/restart information used by the workflow engines to co-ordinate and recover from failure conditions.

— Subscription service: Service to allow clients to register for notification about events. Events are defined by a service that performs an activity resulting in the event. Events are catalogued by the subscription service. Clients identify events of interest, e.g. receipt of data with a specific geographic extent. When an event occurs, the subscription service sends notification to all clients who have registered an interest in the event. Once an event occurs, a subscription service may cause an activity to occur, e.g. delivery of a product.

8.3.5 Geographic processing services

8.3.5.1 Relation of geographic processing services to general feature model

The taxonomy within the processing services category is based on the General Feature Model as presented in ISO 19109. Processing services modify the properties of Features; therefore processing services categories are based on the property types for features given by the General Feature Model. The processing services category of EOSE is subdivided into the categories for geographic processing services shown in Table 10.

Table 10 — Geographic processing services taxonomy

— Geographic processing services — spatial
— Geographic processing services — thematic
— Geographic processing services — temporal
— Geographic processing services — metadata

8.3.5.2 Geographic processing services — spatial

The following is a non-exhaustive listing of geographic processing services — spatial.

- Coordinate conversion service: Service to change coordinates from one coordinate system to another coordinate system that is related to the same datum. In a coordinate conversion, the parameters' values are exact. Coordinate conversion services include map projection services. ISO 19111 is relevant to coordinate conversion;
- Coordinate transformation service: Service to change coordinates from a coordinate reference system based on one datum to a coordinate reference system based on a second datum. A coordinate transformation differs from a coordinate conversion in that the coordinate transformation parameter values are derived empirically; therefore, there may be several different estimations (or realizations). ISO 19111 is relevant to coordinate transformation;
- Coverage/vector conversion service: Service to change the spatial representation from a coverage schema to a vector schema, or vice versa. A standard relevant to vector schema definition is ISO 19107. A standard relevant to coverage schema definition is ISO 19123;
- Image coordinate conversion service: Coordinate transformation or coordinate conversion service to change the coordinate reference system for an image. A standard relevant to image coordinates is ISO 19123; standardization relevant to image coordinates is also discussed in ISO/TR 19121;
- Rectification service: Service that projects a tilted or oblique image onto a selected plane or other surface. The plane is often horizontal, but can be tilted to achieve some desired condition, such as to better fit the local surface of the earth;
- Orthorectification service: Rectification service that removes image displacement due to variation in terrain elevation. Orthorectification requires use of digital elevation data, usually in grid form;
- Sensor geometry model adjustment service: Service that adjusts sensor geometry models to improve the match of the image with other images and/or known ground positions;
- Image geometry model conversion service: Service that converts sensor geometry models into a different but equivalent sensor geometry model;
- Subsetting service: Service that extracts data from an input in a continuous spatial region either by geographic location or by grid coordinates;
- Sampling service: Service that extracts data from an input using a consistent sampling scheme either by geographic location or by grid coordinates;
- Tiling change service: Service that changes the tiling of geographic data;
- Dimension measurement service: Service to compute dimensions of objects visible in an image or other geodata. An alternative name for this service is "image mensuration services";
- Feature manipulation services: Register one feature to another, an image, or another data set or coordinate set; correcting for relative translation shifts, rotational differences, scale differences, and perspective differences; verifying that all features in the Feature Collection are topologically consistent according to the topology rules of the Feature Collection, and identifying and/or correcting any inconsistencies that are discovered;
- Feature matching service: Service that determines which features and portions of features represent the same real-world entity from multiple data sources, e.g. edge matching and limited conflation;
- Feature generalization service — spatial: Service that reduces spatial variation in a feature collection to increase the effectiveness of communication by counteracting the undesirable effects of scale reduction;

— Route determination service: Service to determine the optimal path between two specified points based on the input parameters and properties contained in the Feature Collection; may also determine the measured distance between two points along a specified path based on the properties supported in the Feature Collection further, may determine the length of time it takes to follow a route through the geographic data in the Feature Collection;

— Positioning service: Service provided by a position-providing device to use, obtain and unambiguously interpret position information and determine whether the results meet the requirements of the use. A standard relevant to position services is ISO 19116;

— Proximity analysis service: Given a position or geographic feature, finds all objects with a given set of attributes that are located within a user-specified distance of the position or feature.

8.3.5.3 Geographic processing services — thematic

The following is a non-exhaustive listing of geographic processing services — thematic.

— Geoparameter calculation service: Service to derive application-oriented quantitative results that are not available from the raw data themselves;

— Thematic classification service: Service to classify regions of geographic data based on thematic attributes. Classification of coverages (including images) subdivides a coverage into regions based on attribute values. Classification of features, sorts features into groups based on attribute values or feature associations;

— Feature generalization service — thematic: Service that generalizes feature types in a feature collection to increase the effectiveness of communication by counteracting the undesirable effects of data reduction;

— Subsetting service: Service that extracts features or coverage elements from a larger set based on thematic characteristics;

— Spatial counting service: Service that counts geographic features of a given type within a specified area;

— Geographic information extraction services: Services supporting the extraction of feature and terrain information from remotely sensed and scanned images;

— Image processing service: Service to change the values of thematic attributes of an image using a mathematical function. Example functions include convolution, data compression, feature extraction, frequency filters, geometric operations, non-linear filters and spatial filters;

— Reduced resolution generation service: Service that reduces the resolution of an image;

— Image manipulation services: Services for manipulating data values in images: changing colour and contrast values, applying various filters, manipulating image resolution, noise removal, “striping”, systematic-radiometric corrections, atmospheric attenuation, changes in scene illumination, etc.;

— Image understanding services: Services that provide automated image change detection, registered image differencing, significance-of-difference analysis and display and area-based and model-based differencing;

— Image synthesis services: Services for creating or transforming images using computer-based spatial models, perspective transformations, and manipulations of image characteristics to improve visibility, sharpen resolution, and/or reduce the effects of cloud cover or haze;

— Multi-band image manipulation: Services that modify an image using the multiple bands of the image. Examples include ratioing, principal components transformation, intensity-hue-saturation colour space transformation, de-correlation-stretching;

- Object detection service: Service to detect real-world objects in an image;
- Geoparsing service: Service to scan text documents for location-based references, such as a place names, addresses, postal codes, etc., in preparation for passage to a geocoding service;
- Geocoding service: Service to augment location-based text references with geographic coordinates (or some other spatial reference).

8.3.5.4 Geographic processing services — temporal

The following is a non-exhaustive listing of geographic processing services — temporal.

- Change detection services: Service to find differences between two data sets that represent the same geographical area at different times;
- Temporal reference system transformation service: Service to change the values of temporal instances from one temporal reference system to another temporal reference system. ISO 19108 is relevant to temporal reference systems. Using the terminology of ISO19108, a temporal reference system transformation service replaces the TM_Position value of a given TM_Instant with an equivalent TM_Position value associated with a different temporal reference system;
- Subsetting service: Service that extracts data from an input in a continuous interval based on temporal position values;
- Sampling service: Service that extracts data from an input using a consistent sampling scheme based on temporal position values;
- Temporal proximity analysis service: Service that, given a temporal interval or event, find all objects with a given set of attributes that are located within a user-specified interval from the interval or event.

8.3.5.5 Geographic processing services — metadata

The following is a non-exhaustive listing of geographic processing services — metadata.

- Statistical calculation service: Service to calculate the statistics of a data set, e.g. mean, median, mode and standard deviation; histogram statistics and histogram calculation; minimum and maximum of an image; multi-band cross-correlation matrix; spectral statistics; spatial statistics; other statistical calculations;
- Geographic annotation services: Services to add ancillary information to an image or a feature in a Feature Collection (e.g. by way of a label, a hot link, or an entry of a property for a feature into a database) that augments or provides a more complete description.

8.3.6 Geographic communication services

Examples of communications services for working with geographic data and services:

- Encoding service: Service that provides implementation of an encoding rule and provides an interface to encoding and decoding functionality. A standard relevant to encoding is ISO 19118;
- Transfer service: Service that provides implementation of one or more transfer protocols, which allows data transfer between distributed information systems over off-line or on-line communication media. To successfully transfer data between two systems, the sender and receiver need to agree on the transfer protocol to be used. A standard relevant to transfer is ISO 19118. For some geographic data sets, large data-object transfer is required;
- Geographic compression service: Service that converts spatial portions of a feature collection to and from compressed form;

— Geographic format conversion service: Service that converts from one geographic data format to another;

— Messaging service: Service that allows multiple users to simultaneously view, comment about, and request edits of feature collections. This service allows collaboration involving geographic data;

— Remote file and executable management: Service that provides access to secondary storage of geographic features as if it were local to the client.

8.3.7 Geographic system management services

NOTE No geographic-specific system management services have been identified.

8.4 ISO 19100 series of International Standards in geographic service taxonomy

This subclause summarizes the references to geographic information standards in the ISO 19100 series that are listed in the geographic service taxonomy of 8.3. Table 11 provides a mapping of the referenced ISO 19100 series of International Standards to the extended OSE service categories.

Table 11 — Mapping the ISO 19100 series of International Standards to extended OSE service categories

Extended OSE service category	Relevant ISO 19100 series standard	
Geographic human interaction services	ISO 19117	Geographic information — Portrayal
	ISO 19128 interface	Geographic information — Web Map server
Geographic model/information management services	ISO 19107	Geographic information — Spatial schema
	ISO 19110	Geographic information — Methodology for feature cataloguing
	ISO 19111	Geographic information — Spatial referencing by coordinates
	ISO 19112	Geographic information — Spatial referencing by geographic identifiers
	ISO 19115	Geographic information — Metadata
	ISO 19123	Geographic information — Schema for coverage geometry and functions
	ISO 19125-1	Geographic information — Simple feature access - Part 1: Common architecture
	ISO 19128 interface	Geographic information — Web Map server
Geographic workflow/task management services		(no relevant ISO 19100 series International Standards)
Geographic processing service	ISO 19107	Geographic information — Spatial schema
	ISO 19108	Geographic Information — Temporal schema
	ISO 19109	Geographic information — Rules for application schema
	ISO 19111	Geographic information — Spatial referencing by coordinates
	ISO 19116	Geographic information — Positioning services
	ISO 19123	Geographic information — Schema for coverage geometry and functions
	ISO 19118	Geographic information — Encoding
Geographic communication services		(no relevant ISO 19100 series International Standards)
Geographic system management services		(no relevant ISO 19100 series International Standard)

8.5 Geographic service chaining validity

A service chain combines services to produce results that the individual services could not produce alone. The early parts of this clause have defined type of services that could be chained. Clause 7 defines the syntactic issues of service chaining, e.g. data structure of a chain, architecture patterns for chaining. Clause 7 defines how to construct chains; it does not address whether the results of a chain are semantically valid. The human user that constructed a new chain or invoked an existing chain of services should determine semantic validity of the results of a service chain. (Here it is assumed that the chain is syntactically correct, i.e. the input and output types internal to a chain match and the chain produces a result.) Some factors to consider in the semantic evaluation of a chain result are listed below:

— Appropriateness of starting data: Are the based datasets suited to the subsequent processing? For example, are the accuracy and resolution of the data and thematic values relevant?

— Effect of services on data: How do the individual services affect the data, e.g. error sources and propagation?

— Sequence of the services: How does the order of the chain affect the results? For example, should a spatial operation, e.g. orthorectification, be performed before or after a thematic operation, e.g. resampling the attribute values?

The evaluations depend upon understanding the services, e.g. through review of the service metadata, but also rely upon the users' understanding of the combinations of the services.

8.6 Services organizer folder (SOF)

8.6.1 Introduction

Services organizer folders (SOFs) were introduced in 7.3 as groupings of services that are used for a specific task. Clause 7 introduced SOF as a structure. The remainder of this clause provides example SOFs based on the service taxonomy defined earlier in Clause 8.

8.6.2 Image exploitation SOF

Image exploitation services are required to support most aspects of image exploitation, including precision measurement of ground positions and of object dimensions. For example, a variety of services are needed for extracting features from images, or digital elevations from stereoscopic images. Image exploitation services are widely implemented and used in photogrammetric systems, currently using custom interfaces.

Table 12 provides an example Image exploitation SOF.

Table 12 — Image Exploitation SOF

Geographic Service Taxonomy	Image Exploitation Services
Human interaction services	Geographic viewer — mosaicing
Model/information management services	Coverage access service Feature access service Catalogue service
Workflow/task services	Chain definition service Workflow enactment service
Processing services	
Processing services — spatial	Coordinate conversion service Coordinate transformation service Image coordinate conversion service Orthorectification service
Processing services — thematic	Geoparameter calculation service Thematic classification service Image processing service
Processing services — temporal	Subsetting service Subsampling service
Processing services — metadata	Statistical calculation service
Communication services	Geographic format conversion service
System management services	—

8.6.3 Geographic data fusion SOF

Geographic data fusion is a framework of services for the synthesis of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of greater quality will depend upon the application. Geographic fusion is accomplished through organizing, relating and linking disparate sources of location-based information. Geographic fusion relates to the fusion of two or more geographic elements and the fusion of geographic elements with other sources of structured and unstructured data from distributed sources.

Table 13 provides an example Geographic fusion SOF.

Table 13 — Geographic data fusion SOF

Geographic Service Taxonomy	Image Exploitation Services
Human interaction services	Geographic viewer
Model/information management services	Map access service Geature access service Catalogue service Gazetteer service Feature type service
Workflow/task services	—
Processing services	
Processing services — spatial	Feature matching service
Processing services — thematic	Object detection service Change detection services Geoparsing service Geocoding service
Processing services — temporal	—
Processing services — metadata	—
Communication services	Encoding service Messaging service
System management services	—

9 Engineering viewpoint — A basis for distribution

9.1 Distribution transparencies and the engineering viewpoint

The engineering viewpoint focuses on mechanisms for distribution, distribution transparencies, and support services such as security and persistence. Distribution transparencies enable complexities associated with system distribution to be hidden from applications where they are irrelevant to their purpose.

Location transparency masks the need for an application to have information about location in order to invoke a service. Location transparency is handled by name-servers that transparently map logical names to physical server addresses. The underlying mechanisms used may differ between physical communication solutions. For example, if CORBA is used, the CORBA naming service can be used as a name service. If socket communication is used, a proprietary name server is used.

Replication transparency masks the fact that multiple copies of a service may be provided in order to provide reliability and availability. Replication transparency is handled to the extent that different calling semantics can be transparently implemented in the framework, dependent on the policies desired. The supported semantics are a shallow copy (copy of one object without its associated objects), a partial copy (copy of one object with its direct associated objects) and a deep copy (copy of one object and transitively a copy of all its associated objects until no more objects are reachable).

There are other distribution transparencies that are also important to relate to, such as the following:

— failure transparency: masks failures and recoveries of objects;

— federation transparency: masks interworking across multiple administrations;

— group transparency: masks the use of a group of objects to provide an interface;

— migration transparency: masks the relocation of an object;

— resource transparency: masks passivation and reactivation;

— persistence transparency: hides the actual activation and deactivation of objects from a persistent store, and the actual storage mechanisms and representation format used;

— transaction transparency: hides coordination for achieving the transactional properties;

— security transparency: hides the mechanisms that are being used for authentication and authorization.

In order to achieve interoperability between different platforms, it is necessary to have mappings between the platforms' support for these transparencies. This can be done through a higher abstraction layer that maps to the implementation and representation of these services in various platforms.

Some of these transparencies are being addressed in the current market-development around application server technologies and underlying infrastructures, such as Enterprise Java Beans, Microsoft Transaction Server (MS MTS) and CORBA Components. Once these are established, it is a natural step to specify an abstraction layer for addressing distribution transparencies. This is currently in development at OMG in the context of "UML profile and human readable language for EDOC — Enterprise Distributed Object Computing."

9.2 Distributing components using a multi-tier architecture model

To support flexible deployment, IT architectures are structured as multi-tiered distributed architectures. As a reference model, a logical four-tier architecture is presented with discussion on variations in different physical architectures. The logical architecture is the arrangement of services and associated interfaces that are present in the system (see Figure 10). The physical architecture is the arrangement of components and associated interfaces that implement the services. The components are hosted on hardware computing resources or nodes.

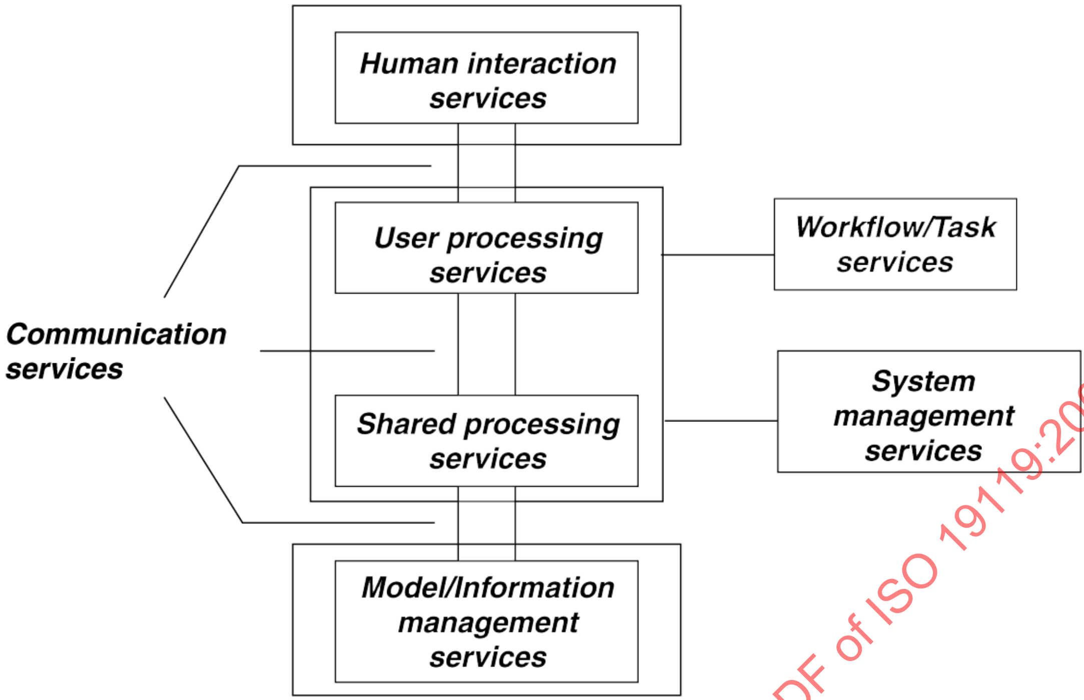

Figure 10 — Logical multi-tiered architecture

The OSE model, as defined in ISO 19101, structures the types of services of an IT system. Each tier can contain both IT-general services and GIS-extended services for that tier.

— The human interaction services tier is responsible for physical interaction with the user, through display and input media and an appropriate dialogue. This might be separated into a presentation tier and a dialogue tier.

— The user processing services tier is a part of the processing services responsible for the functionality required by the user.

— The shared processing services tier is part of the processing services responsible for common services (both domain specific and general) that can be used by multiple users.

— The model/information management services tier is responsible for physical data storage and data management.

— The workflow/task services are a set of services that can be viewed as a specialized processing service.

— The communication services are responsible for connecting the various tiers together. (The communication services are present as the connections between the other service tiers.)

— The system management services are orthogonal to the multi-tiered architecture and might be introduced in multiple tiers.

The logical architecture can be mapped to multiple physical architectures. All tiers could be mapped into one monolithic application or could be mapped using different physical client-server architectures. Figures 12 and 13 show mappings to various physical architectures.

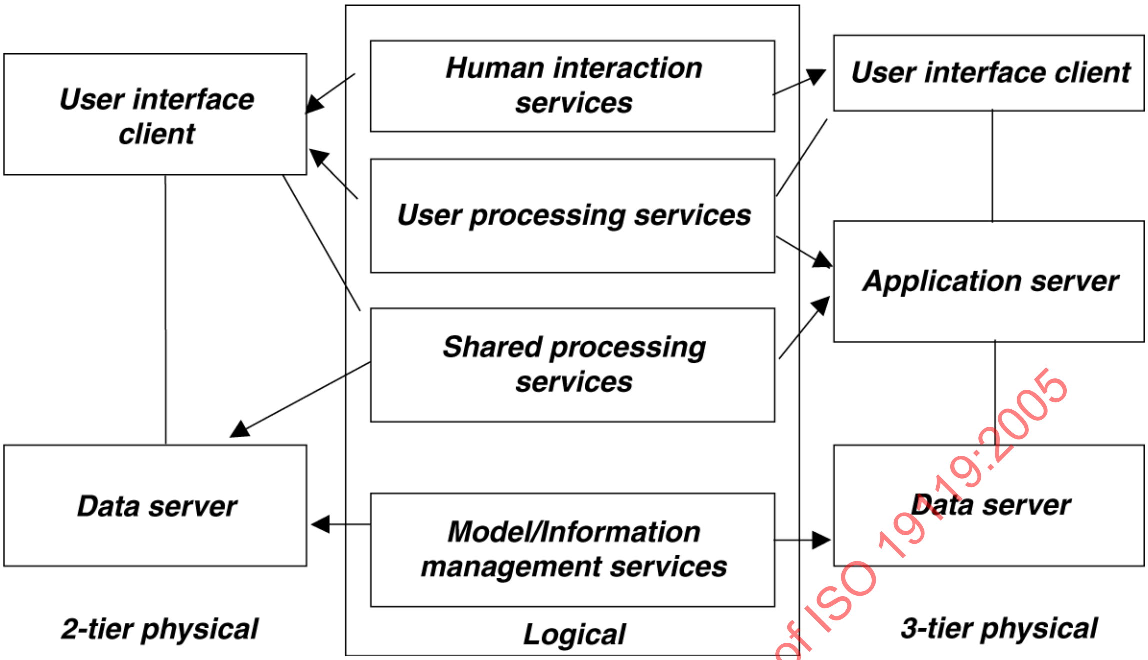

Figure 11 — From logical four-tier to physical two-tier or three-tier architecture

In Figure 11, a data server contains the logic that interfaces either with a data storage system or with some other type of external data source, such as a data feed or an external application system. The data server provides model/information management services.

An application server contains components that are responsible for processing services. An application server may provide both user processing services and shared processing services.

A user interface client provides interaction services, contains the logic that presents information to an external source and obtains input from that source. In most cases, the external source is a human end-user working at their own computer, although the external source might also be process-oriented. The client logic generally provides menus of options to allow the user to navigate through the different parts of the application, and it manipulates the input and output fields on the display device. Frequently, the presentation component also performs a limited amount of input data validation.

As shown in Figure 11, a two-tier physical architecture typically consists of a user interface client interacting directly with a data server. User services are normally executed in the user interface client, while the data server provides shared processing services.

As shown in Figure 11, a three-tier physical architecture introduces an intermediate application server that is responsible for the execution of shared processing services, sometimes also for user services. A major advantage of using a three-tier distributed information system is that the user can choose how to combine components to perform tasks. In an interoperable component environment, the user can select components that perform similar tasks and combine the chosen components to best produce the information that is needed for application.

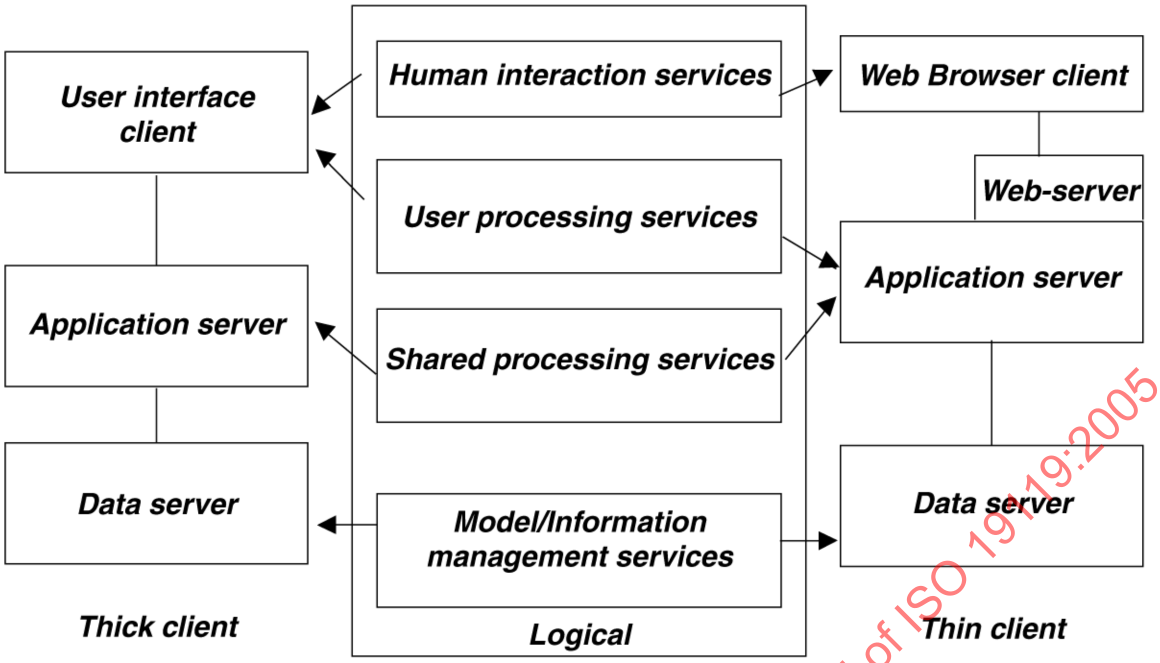

Figure 12 — Mapping logical four-tier to thick and thin clients

As shown in Figure 12, a thick user interface client architecture will typically contain a larger part of the functionality in the user service. A thin user interface client (typically a web browser) will mostly contain user dialogue and presentation code. A web browser client is a user interface client that interacts with a web server, using the internet HTTP protocol with content represented in HTML and/or XML.

A platform-independent abstract specification can include specification of both user interface (UI) plus service and data/information aspects. This means that a large specification can be broken into different parts, each addressing a different part of a total system specification (see Figure 13). The various parts can typically be mapped into different kind of specific technologies.

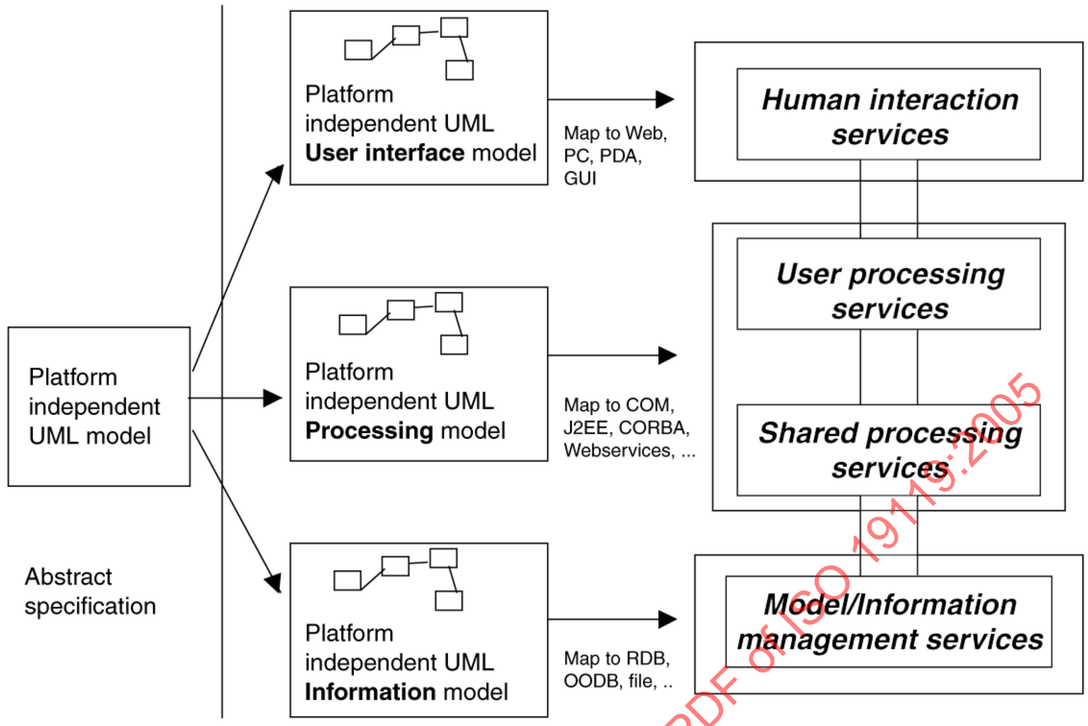

Figure 13 — Mapping from platform independent UML models

10 Technology viewpoint — A basis for cross platform interoperability

10.1 Infrastructure interoperability and the technology viewpoint

The technology viewpoint of ISO RM-ODP is concerned with the underlying infrastructure in a distributed system. It describes the hardware and software components used in a distributed system. To achieve interoperability in the technology viewpoint, an infrastructure that allows the components of a distributed system to interoperate is needed. This infrastructure, which may be provided by a distributed computing platform (DCP), allows objects to interoperate across computer networks, hardware platforms, operating systems and programming languages.

The communication between the objects in a DCP is handled by a "*communication service*", for instance, such as realized by the *Object Request Broker* (ORB) in the CORBA environment.

The communication service is responsible for routing service requests from a client object to a server object. A request from a client object to a server object is called a *remote method invocation*. To perform this remote method invocation on server objects, the communication service needs to collect all the request parameters from the client, package these in a neutral, platform independent structure, and send it to the server object. The result from the request is then collected and packaged and returned to the calling client.

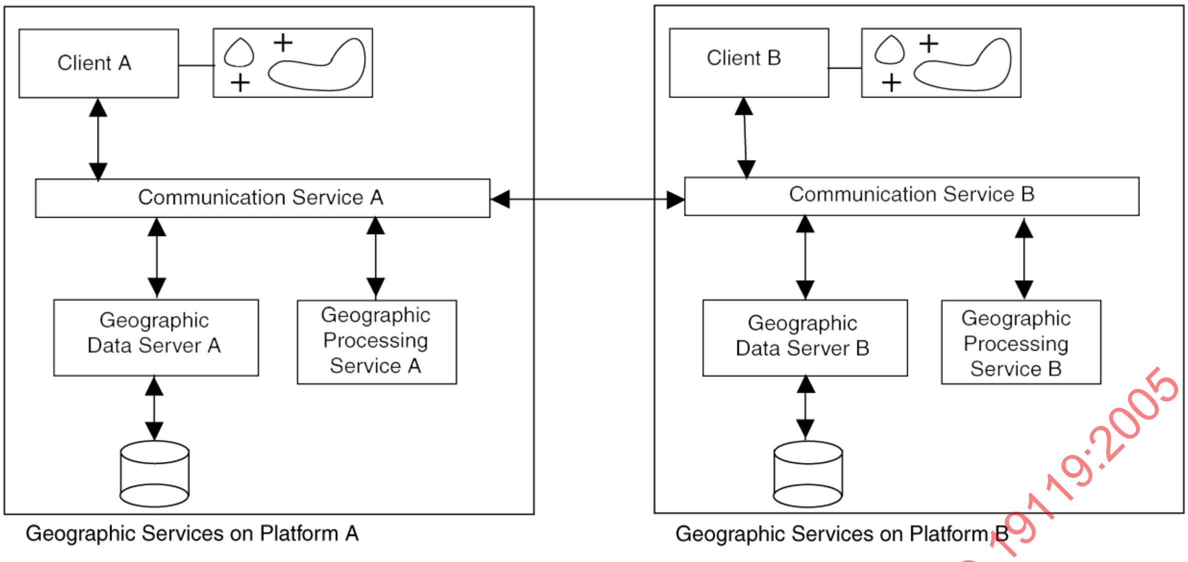

Figure 14 — Technology viewpoint model of the interoperability reference model

The communication service allows the components in a distributed system to interoperate. In the two systems depicted in Figure 14, communication service A allows the components of system A (client A, geodata server A and GIS service component A) to interoperate, while communication service B allows the components of system B to interoperate. To allow system A to interoperate with system B, the components of system A must be able to request services from the components of system B and vice versa. There are two cases to consider:

— **The two systems use the same DCP:**

OMG has defined the GIOP (general inter-ORB protocol) and IIOP (internet inter-ORB protocol) standards that allow two systems that use CORBA to interoperate. These standards define how objects in system A can invoke services from objects in system B. DCOM and Java RMI also have support for allowing objects in one system to invoke services from objects in another system.

— **The two systems use two different DCPs:**

If the two systems use different DCPs, they can interoperate through the use of special “bridging” tools. These bridging tools allow objects that use one DCP to interoperate with objects that use another DCP.

10.2 Need for multiple platform-specific specifications

It is assumed that one platform-neutral service specification will be the basis for multiple platform-specific service specifications. The OGC service specifications for simple feature access is an example of a set of platform-specific service specifications for SQL, COM/OLE and CORBA, and has demonstrated the need for at least three different implementation specifications of the same platform-neutral specification.

Multiple platform-specific specifications are necessary because of the variety of DCP's and the differences in the way in which they support the functional requirements. One platform-neutral service specification is needed to support interoperability of multiple platform-specific specifications.

10.3 Conformance between platform-neutral and platform-specific service specifications

Platform-neutral models shall be described in UML in accordance with the rules and guidelines in ISO/TS 19103. Platform-neutral service specifications shall define both static and dynamic models. Static models define objects including the attributes and operations for each object. Dynamic models capture the interaction patterns between objects and state modelling.

Platform-specific models may be described in UML, together with a description of their mapping to the corresponding platform-neutral models. It is allowed to describe the platform-specific models directly in a platform-specific language such as SQL, CORBA/IDL, web services description language, etc., as long as the mapping to the corresponding platform independent model is well defined.

Development of service specifications may proceed from platform-specific to platform-neutral or from platform-neutral to platform-specific. In either case, a service specification shall not be considered complete until it has a platform-neutral model and at least one platform-specific model; see Figure 15.

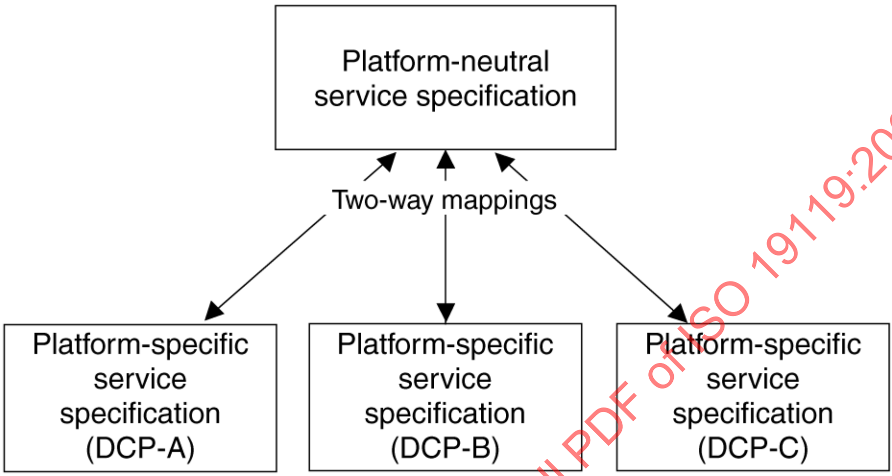

Figure 15 — From platform-neutral abstract specifications to multiple platform-specific implementations

A platform-neutral service specification shall be defined using the relevant parts of the UML information models defined in other standards in the ISO 19100 series. A platform-specific service specification shall show how it meets the intentions of the platform-neutral specification, by showing how the various parts of the platform-neutral service specification are mapped onto the platform-specific service implementation. In order to support interoperability between different implementation specifications, the reverse mapping back to the concepts in the platform-neutral model should be defined. As relevant, a platform-specific specification should include the encoding of information according to ISO 19118.

10.4 From platform-neutral to platform-specific specifications

Any ISO 19100 series specification that addresses a particular platform, i.e. SQL, CORBA or other, shall include a detailed mapping specification from the basic data types used in the platform-neutral UML models to corresponding types in the platform. For the ISO 19100 series of International Standards, basic data types for use in platform-neutral UML models are defined in ISO/TS 19103.

Implementation specifications for a particular platform may be described through the use of the UML. In particular, if platform-specific aspects have been introduced through the use of platform-specific stereotypes, tagged values and constraints. UML profiles with platform-specific types exist for various platforms.

Annex D provides mappings of the platform-neutral services of the computational viewpoint to a variety of DCPs.

Annex A
(normative)

Conformance

A.1 Conformance requirements

This International Standard defines two classes of conformance: service architecture test module and service specification test module.

The framework, concepts and methodology for conformance and testing for ISO 19100 series standards are defined in ISO 19105.

A.2 Service architecture test module

A.2.1 Geographic service types definition

The test for conformance of the geographic service type is as follows:

— test purpose: Verify that a service has been categorized in accordance with the geographic services taxonomy.

— test method: Determine whether the service has been categorized in accordance with this International Standard. If so, determine whether the categorization of the service is correct. If the service has a title identical with an example service in 8.3, determine whether the service provides the functionality that is defined in the example.

If the service provides the same functionality as an example service in 8.3, the services should use the name of the example.

— reference: this International Standard, 8.3;

— test type: basic.

A.2.2 Service chain

The test for service chain is as follows:

— test purpose: Verify that the implementation of a service chain is conformant with this International Standard.

— test method: Review the data structure for a service chain and determine whether it is a directed graph. Determine whether the nodes of the service chains are services.

An architecture should provide a means to make a chain extant and transferable between users and for users to evaluate the validity of a chain.

NOTE 1 Guidelines for evaluating chain validity are provided in 8.5.

NOTE 2 An extant service chain is required for the translucent chaining pattern.

— reference: this International Standard, 7.3;

— test type: basic.

A.2.3 Service chaining patterns

The test for service chaining patterns is as follows:

- test purpose: Verify that an architecture correctly uses the service chaining architecture patterns.
- test method: If an architecture claims to implement the transparent chaining pattern, confirm that a human user is controlling the execution of the chain. If an architecture claims to implement the translucent chaining pattern, confirm that a service chain can be made extant separate from human users and that control of the chain execution is accomplished separate from human users.

  For the transparent chaining pattern, the architecture should provide the human user with mechanisms to determine services of value, e.g. a service catalogue or service organizer folder.
- reference: this International Standard, 7.3;
- test type: basic.

A.2.4 Simple service architecture

The test for simple service architecture is as follows:

Implementing a simple service architecture is not required by this International Standard. An architecture should implement the simple service architecture. Systems that claim conformance to the simple service architecture shall comply with this subclause.

- test purpose: Verify that an architecture correctly implements the simple service architecture.
- test method: Verify that the services in the architecture are implemented with message-operations. Evaluate if the architecture provides for the separation of control operations from data access operations. Evaluate if the services are defined as stateless services, when possible. Verify that clients in the architecture utilize pre-defined service types when encountering service instances. Verify that the utilization of hardware resources is transparent to clients in the architecture.
- reference: this International Standard, 7.6;
- test type: basic.

A.3 Service specification test module

A.3.1 Introduction

Any specification, including a profile or functional standard that claims conformance with this International Standard, shall pass all of the corresponding requirements described in the following abstract test module. This distinguishes between three different conformance classes.

- platform-neutral service specifications: a base standard that specifies a specific geographic service in a platform-neutral way;
- platform-specific service specifications: an environment or language-specific specification that has been derived from a platform-neutral service specification, and is claimed to be a profile standard of this;
- platform-specific service implementations: an actual implementation of a service that can be conformance tested against the abstract test suite of a platform-specific service specification.

A.3.2 Platform-neutral service specifications

The test for each platform-neutral service specification is to check that it has been described according to the service definition rules of this International Standard and the modelling conventions of ISO/TS 19103.

— test purpose: Verify that a geographic service standard is defined in a platform-neutral way, i.e., the usage of UML has been done according to the rules. Verify that the usage of the following elements has been done according to ISO/TS 19103:

  — general usage of UML,
  — classes,
  — attributes,
  — basic data types,
  — associations,
  — operations,
  — stereotypes and tagged values,
  — packages,
  — constraints;

  Interfaces are abstract specifications separate from the concrete deployment or data format bindings. The specification of an interface shall include a static portion that includes definition of the operations. The specification of an interface shall include a dynamic portion that includes any restrictions on the order of invoking the operations.

  Verify that the dynamic aspects of the service are defined, e.g. use cases or interaction diagrams are provided and described. Evaluate whether the service could be used in a service chain.

— test method: Inspect the diagrams, use the checking/validation capabilities of the UML tools that are being used.

— reference: this International Standard, Clause 10; ISO/TS 19103;

— test type: basic.

The platform-neutral service specification shall not use constructs that are special for one specific implementation environment.

The platform-neutral service specification shall contain a platform-neutral abstract test suite for corresponding platform-specific service specifications and implementations of these.

The platform-neutral service specification shall contain a use-case oriented scenario for the usage of the service, as illustrated by the example in the Annex B.

A.3.3 Platform-specific service specifications

The test for each platform-specific service specification is to check that it is related to the corresponding platform-neutral service specification.

— test purpose: Verify that the platform-specific service specification is conformant with the corresponding platform-neutral service specification.

— test method: For each part of the platform-neutral service specification, verify that there is a description of how this part is mapped into and handled by the platform-specific service specification.

It is recommended to show the mapping from the platform-neutral UML model to a platform-specific UML model for the platform-specific service specification. The mapping from the platform-neutral specification to the platform-specific specification shall be described in the language of the implementation environment (e.g. IDL, SQL, etc.), either directly from the platform-neutral specification or through the intermediary UML model.

The platform-specific service specification shall contain a platform-specific test suite to be executed to check conformance of corresponding actual implementations. The test suite shall be based on the corresponding test suite from the platform-neutral service specification.

— Reference: this International Standard, Clause 10.

A.3.4 Platform-specific service implementations

Actual implementations claiming conformance to a platform-specific service specification shall be conformance checked against the test suite specified for the platform-specific service specification.

— test purpose: Verify that an implementation is conformant with the corresponding platform-specific service specification.

— test method: For each part of the platform-specific service specification, verify that implementation conforms.

— reference: this International Standard, Clause 10.

Annex B
(informative)

Example user scenarios

B.1 Example 1 — Service chaining for remote sensed data

B.1.1 Summary

Working in a distributed environment, a user of geographic data first locates several data sets and services, then chains the services to produce the resulting geographic products of interest.

A user wants to determine the land cover classification of a geographic area, e.g. wetlands, urban, etc., using remote sensed coverage data. The user wants a graphical image of the analysis that can be used in a presentation or document. The user wants to include an existing base map in the image.

This scenario is an example of the transparent type of service chaining. The functionality could also be achieved using translucent or opaque service chaining. In the case of opaque chaining, the user would interact with an aggregate service.

B.1.2 Preconditions

The user accesses various geographic information services using "thin" client software located on the user's hardware. The user has access to a network over which the services are accessed. Financial transactions and user authorization issues are ignored in this scenario. The user (or the user's client software) knows the network location of a catalogue service for geographic data and services. The user is familiar with coverage data and classification services.

B.1.3 Detailed steps

The detailed steps are as follows.

a) The user accesses a catalogue service to search for coverage data with a geographic extent of interest to the user. The user identifies several coverages with data over a specific geographic area of interest. Also using the catalogue service, the user discovers a digital elevation model covering the same geographic extent.

b) The user accesses a catalogue service to search for geographic services of interest to the user. The user identifies several orthorectification services and several classification services. The user reviews the service metadata for the services and determines the best orthorectification and classification services for use on the coverages that were selected in the prior step. The user also identifies a portrayal service using the catalogue service.

c) The user examines the coverages by providing a reference to the coverages to the portrayal service. Based on a visual examination, the user estimates whether the coverages would be appropriate for this use.

d) The user accesses the orthorectification service. The access operation includes references to the coverage and DEM. The orthorectification service provides the user with a reference to an orthorectified coverage.

e) The user accesses the classification service. The access operation includes a reference to the orthorectified coverage. The user interacts with the semi-automated classification service, identifying training areas of known land cover type. Statistical measures and visual techniques are provided to the user to determine if the land cover classification has been precise. The classification service result is a feature collection consisting of polygons identifying the classified regions over the geographic extent of interest. (Note that the order of services could be switched, first performing the classification and then the orthorectification. The user needs to make this type of decision based on knowledge of the services.)

f) The user combines the orthorectified coverage and the feature collection into a viewable image using the portrayal service.

g) The user accesses the catalogue service to locate a feature collection of transportation data over the same geographic extent.

h) The user combines the orthorectified coverage, the classification feature collection, and the transportation feature collection into a viewable image using the portrayal service. The user chooses an alternative symbology set for the transportation features and requests the portrayal service to recreate the composite image.

B.1.4 Post-conditions

Image file that can be viewed, printed or integrated into other desktop applications is on the user's computer.

B.1.5 Service chain directed acyclic graph

The scenario was presented above using the “transparent” design pattern described in 7.3.5. With the transparent design pattern, the human user controls the flow of the service chaining and the service chain is not explicitly materialized. When moving to one of the other design patterns, translucent or opaque, the service chain is an explicit DAG as described in 7.3. Figure B.1 shows the DAG for this scenario.

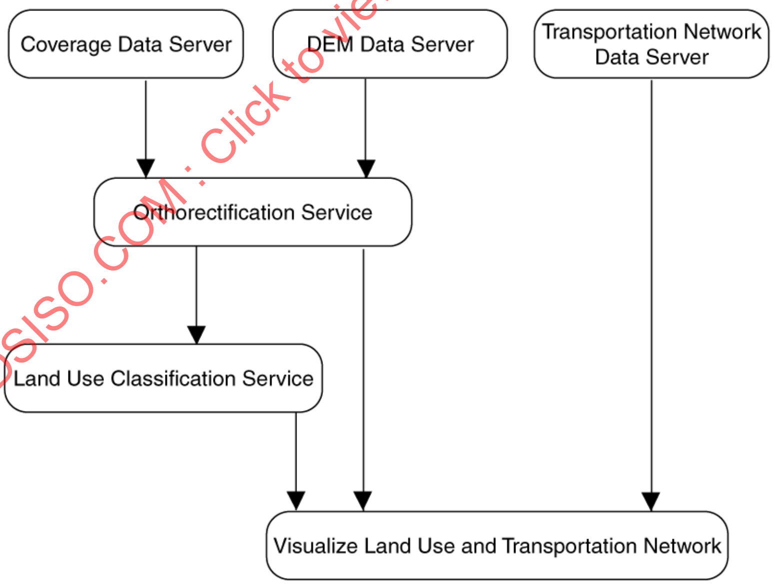

Figure B.1 — DAG for service chaining scenario example

B.2 Example 2 — Roadside services

B.2.1 Summary

The roadside services scenario illustrates the various future network-resident services that might come into play when a motorist tells his/her “personal information appliance” that he/she wants to find a service, e.g. a pizza restaurant, near the highway on which he/she is travelling.

B.2.2 Precondition

The motorist has a personal information appliance that has access to network services while travelling on the highway.

B.2.3 Detailed steps

The detailed steps are as follows.

a) Voice recognition software interprets a spoken request as a query for the nearest restaurant.

b) The broker submits the request along with coordinates and direction from the onboard positioning service to a roadside services database that reports the three nearest restaurant locations to a mapping service.

c) The mapping service queries a road map database and a set of preferences describing how the user prefers to have the data presented: map, voice directions, etc.

d) The mapping service presents route information and preferences to the presentation service, which packages information for delivery to the user.

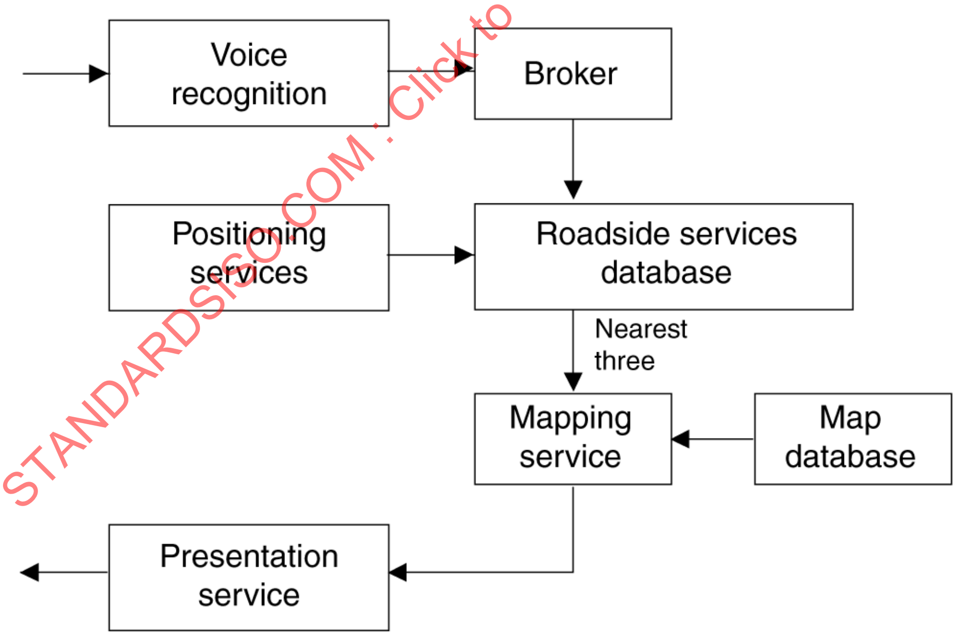

Figure B.2 — Roadside service scenario

Annex C
(normative)

Data dictionary for geographic service metadata

C.1 Data dictionary overview

C.1.1 Introduction

This data dictionary describes the characteristics of the service metadata defined in the UML model of 7.4, Figure 9. UML classes of Figure 9 are mapped into metadata entities in the data dictionary and the attributes of the UML classes are mapped into metadata elements. Within the data dictionary, entities and elements are defined by attributes. These attributes are listed below and are based on those specified in ISO/IEC 11179-3 for the description of data element concepts, i.e. data elements without representation.

C.1.2 Attribute name/role name

This is a label assigned to a metadata entity or to a metadata element. Metadata entity names start with an upper case letter. Spaces do not appear in a metadata entity name. Instead, multiple words are concatenated, with each new sub-word starting with a capital letter (example: XnnnYmmm). Metadata entity names are unique within the entire data dictionary of this International Standard. Metadata element names are unique within a metadata entity, not the entire data dictionary of this International Standard. Role names are used to identify the targets of metadata abstract model associations and are preceded by "Role name:" to distinguish them from other metadata elements. Names and role names may be in a language other than that used in this International Standard.

C.1.3 Definition

This is the metadata entity/element description.

C.1.4 Obligation/condition

C.1.4.1 General

This is a descriptor indicating whether a value for the metadata entity or metadata element shall always be provided or sometimes be provided [i.e. contains value(s)]. This descriptor may have the following values: M (mandatory) or O (optional).

C.1.4.2 Mandatory (M)

A value for the metadata entity or metadata element shall be provided.

C.1.4.3 Optional (O)

A value for the metadata entity or the metadata element may be provided or might not be provided. Optional metadata entities and optional metadata elements have been defined to provide a guide to those looking to fully document their data. (Use of this common set of defined elements will help promote interoperability among geographic data users and producers world-wide.) If an optional entity is not used, the elements contained within that entity (including mandatory elements) will also not be used. Optional entities may have mandatory elements; those elements only become mandatory if the optional entity is used.

C.1.5 Maximum occurrence

This specifies the maximum number of instances the metadata entity or the metadata element may have. Single occurrences are shown by “1”; repeating occurrences are represented by “N”. Fixed number occurrences other than one are allowed, and will be represented by the corresponding number (i.e. “2”, “3”, etc.).

C.1.6 Data type

This specifies a set of distinct values for representing the metadata elements, for example, integer, real, string, DateTime and Boolean.

NOTE Data types are defined in ISO 19118—[1]), 8.2.2.

C.1.7 Attribute class or target class of role

When the element described is an attribute, the value of this column is the class used by the attribute. When the element described is a role, the value of this column is the target class of the role.

C.1.8 Prefixes to names

The domain of elements and entities listed in these tables is indicated by a two-letter prefix for the name. Items defined in this International Standard have the prefix “SV_”. Several elements and entities listed in the data dictionary have prefixes other than SV, indicating they are defined in other standards as follows:

— SV Services (this International Standard);

— MD Metadata (ISO 19115);

— CI Citation (ISO 19115);

— EX Extent (ISO 19115).

C.2 Metadata data dictionaries

C.2.1 Data dictionary for MD_Identification

The data dictionary for MD_Identification is contained in ISO 19115.

C.2.2 Data dictionary for SV_ServiceIdentification

SV_ServiceIdentification provides descriptive data for a service instance sufficient to allow a client to invoke the service.

SV_ServiceIdentification has an aggregation relationship with multiple instances SV_OperationMetadata. While it is known that the service structure may be more complicated than this aggregation, the additional detail of services-aggregating services is not needed in a service metadata record.

The data dictionary for SV_ServiceIdentification is provided in Table C.1.

Table C.1 — Data dictionary for SV_ServiceIdentification

No.	Attribute name/ Role name	Definition	Obligation/ Condition[a]	Maximum occurrence[b]	Attribute class or target class of role
1	serviceType	a service type name from a registry of services. For example, the values of the nameSpace and name attributes of GeneralName may be “OGC” and “catalogue”.	M	1	GenericName
2	serviceTypeVersion	provides for searching based on the version of serviceType. For example, we may only be interested in OGC Catalogue V1.1 services. If version is maintained as a separate attribute, users can easily search for all services of a type regardless of the version.	O	N	CharacterString
3	accessProperties	information about the availability of the service, including, — fees — planned available date and time — ordering instructions — turnaround	O	1	MD_StandardOrderProcess
4	restrictions	legal and security constraints on accessing the service and distributing data generated by the service	O	1	MD_DataConstraints
8	*Role name:* containsOperations	provides information about the operations that comprise the service	M	N	SV_OperationMetadata
9	*Role name:* operatesOn	provides information on the datasets that the service operates on	O	N	MD_DataIdentification

a M = mandatory, O = optional.

b N = repeating occurrences.

C.2.3 Data dictionary for SV_OperationMetadata

SV_OperationMetadata describes the signature of one and only one method provided by the service.

The data dictionary for SV_OperationMetadata is provided in Table C.2.

Table C.2 — Data dictionary for SV_OperationMetadata

No.	Attribute name/ Role name	Definition	Obligation/ Condition[a]	Maximum occurrence[b]	Attribute class or target class of role
1	operationName	a unique identifier for this interface	M	1	CharacterString
2	DCP	distributed computing platforms on which the operation has been implemented	M	N	DCPlist
3	operationDescription	free text description of the intent of the operation and the results of the operation	O	1	CharacterString
4	invocationName	the name used to invoke this interface within the context of the DCP. The name is identical for all DCPs.	O	1	CharacterString
5	parameters	the parameters that are required for this interface	O	1	sequence(SV_Parameter)
6	connectPoint	handle for accessing the service interface	M	N	CL_OnlineResource
7	dependsOn	list of operations that must be completed immediately before current operation is invoked, structured as a list for capturing alternate predecessor paths and sets for capturing parallel predecessor paths	O	1	set{sequence{operationName} | set(operationName}

[a] M = mandatory, C = conditional, O = optional.
[b] N = repeating occurrences.

C.2.4 Data dictionary for SV_ServiceProvider

SV_ServiceProvider describes an organization that provides services.

The data dictionary for SV_ServiceProvider is provided in Table C.3.

Table C.3 — Data dictionary for SV_ServiceProvider

No.	Attribute name/ Role name	Definition	Obligation/ Condition[a]	Maximum occurrence[b]	Attribute class or target class of role
1	providerName	a unique identifier for this organization	M	1	CharacterString
2	serviceContact	information for contacting the service provider	M	N	CL_ResponsibleParty

[a] M = mandatory.
[b] N = repeating occurrences.

C.2.5 Data dictionary for MD_DataIdentification

The data dictionary for MD_DataIdentification is contained in ISO 19115.

C.2.6 Data dictionary for SV_OperationChainMetadata

The data dictionary for SV_OperationChainMetadata is provided in Table C.4.

Table C.4 — Data dictionary for SV_OperationChainMetadata

No.	Attribute name/ Role name	Definition	Obligation/ Condition[a]	Maximum occurrence[b]	Attribute class or target class of role
1	name	the name, as used by the service for this chain	M	1	CharacterString
2	description	a narrative explanation of the services in the chain and resulting output	O	1	CharacterString

[a] M = mandatory, O = optional.
[b] N = repeating occurrences.

C.2.7 Data dictionary for SV_Parameter

The data dictionary for SV_Parameter is provided in Table C.5.

Table C.5 — Data dictionary for SV_Parameter

No.	Attribute name/ Role name	Definition	Obligation/ Condition[a]	Maximum occurrence[b]	Attribute class or target class of role
1	name	the name, as used by the service for this parameter	M	1	MemberName
2	direction	indication if the parameter is an input to the service, an output or both	O	1	ParameterDirection
3	description	a narrative explanation of the role of the parameter	O	1	CharacterString
4	optionality	indication if the parameter is required	M	1	CharacterString
5	repeatability	indication if more than one value of the parameter may be provided	M	1	Boolean

[a] M = mandatory, O = optional.
[b] N = repeating occurrences.