INTERNATIONAL STANDARD

ISO 12967-3

First edition
2009-08-15

Health informatics — Service architecture —

Part 3:
Computational viewpoint

Informatique de santé — Architecture de service —

Partie 3: Point de vue informatique

Reference number
ISO 12967-3:2009(E)

PDF disclaimer

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

Adobe is a trademark of Adobe Systems Incorporated.

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

COPYRIGHT PROTECTED DOCUMENT

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 12967-3 was prepared by Technical Committee ISO/TC 215, *Health informatics*, based on the European Standard EN 12967-3:2007 with minor editorial amendments.

ISO 12967 consists of the following parts, under the general title *Health informatics — Service architecture*:

— *Part 1: Enterprise viewpoint*

— *Part 2: Information viewpoint*

— *Part 3: Computational viewpoint*

Introduction

ISO 12967 is a multi-part standard that provides guidance for the description, planning and development of new systems as well as for the integration of existing information systems, both within one enterprise and across different healthcare organizations through an architecture integrating the common data and business logic into a specific architectural layer (i.e. the middleware), distinct from individual applications and accessible throughout the whole information system through services, as shown in Figure 1.

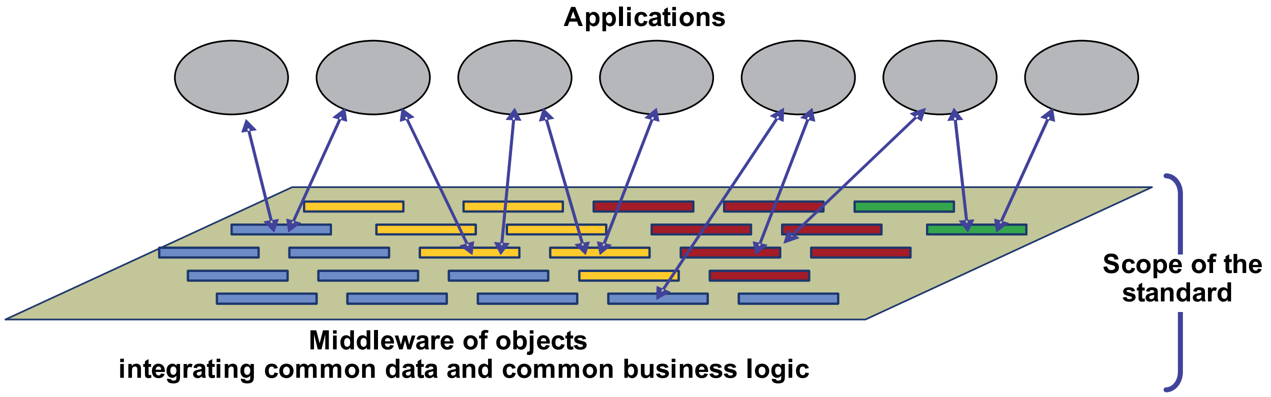

Figure 1 — Scope of this International Standard

The overall architecture is formalized according to ISO/IEC 10746 (all parts)[10] [11][12][13] and is therefore structured through the following three viewpoints.

a) Enterprise viewpoint: specifies a set of fundamental common requirements at enterprise level with respect to the organizational purposes, scopes and policies that must be supported by the information and functionality of the middleware. It also provides guidance on how one individual enterprise (e.g. a regional healthcare authority, a large hospital or any other organization where this model is applicable) can specify and document additional specific business requirements, with a view to achieving a complete specification, adequate for the characteristics of that enterprise.

   Enterprise viewpoint is specified in ISO 12967-1.

b) Information viewpoint: specifies the fundamental semantics of the information model to be implemented by the middleware to integrate the common enterprise data and to support the enterprise requirements formalized in ISO 12967-1. It also provides guidance on how one individual enterprise can extend the standard model with additional concepts needed to support local requirements in terms of information to be put in common.

   Information viewpoint is specified in ISO 12967-2.

c) Computational viewpoint: specifies the scope and characteristics of the services that must be provided by the middleware for allowing access to the common data as well as the execution of the business logic supporting the enterprise processes identified in the information viewpoint and in ISO 12967-1. It also provides guidance on how one individual enterprise can specify additional services needed to support local specific requirements in terms of common business logic to be implemented.

   Computational viewpoint is specified in this part of ISO 12967.

Health informatics — Service architecture —

Part 3: Computational viewpoint

1 Scope

HISA specifies fundamental requirements for 'information infrastructure' and healthcare specific middleware services.

This part of ISO 12967 specifies the fundamental characteristics of the computational model to be implemented by a specific architectural layer of the information system (i.e. the middleware) to provide a comprehensive and integrated interface to the common enterprise information and to support the fundamental business processes of the healthcare organization, as defined in ISO 12967-1. The computational model is specified without any explicit or implicit assumption about the physical technologies, tools or solutions to be adopted for its physical implementation in the various target scenarios. The specification is nevertheless formal, complete and non-ambiguous enough to allow implementers to derive an efficient design of the system in the specific technological environment which will be selected for the physical implementation.

The computational model provides the basis for ensuring consistency between different engineering and technology specifications (including programming languages and communication mechanisms) since they must be consistent with the same computational object model. This consistency allows open inter-working and portability of components in the resulting implementation.

This specification does not aim at representing a fixed, complete, specification of all possible interfaces that may be necessary for any requirement of any healthcare enterprise. It specifies only a set of characteristics – in terms of overall organization and individual computational objects, identified as fundamental and common to all healthcare organizations, and that are satisfied by the computational model implemented by the middleware.

Preserving consistency with the provisions of this part of ISO 12967, physical implementations shall allow extensions to the standard computational model in order to support additional and local requirements. Extensions shall include both the definition of additional properties in the objects of the standard model and the implementation of entirely new objects.

Also this standard specification shall be extendable over time according to the evolution of the applicable standardization initiatives. The specification of extensions shall be carried out according to the methodology defined in Clause 7 of ISO 12967-1:2009, which identifies a set of healthcare common information services, describing their need and the methodology through which they will be used. These are only the minimal identifiable set of services according to the needs of the healthcare enterprise, and constituting the "middleware" platform (i.e. integration platform) to serve as the basis for healthcare applications, e.g. EHR or patient administration.

2 Normative references

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

ISO 12967-1:2009, *Health informatics — Service architecture — Part 1: Enterprise viewpoint*

ISO 12967-2:2009, *Health informatics — Service architecture — Part 2: Information viewpoint*

3 Terms and definitions

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

3.1
interface
abstraction of the behaviour of an object which consists of a subset of the possible interaction mechanisms of that object, together with the set of constraints when that interaction occurs

3.2
computational object
object as seen in a computational viewpoint representing the functional decomposition of a system showing a state and behaviour as well as interactions through interfaces with other computational objects

4 Abbreviations

EHR	Electronic Health Record
HISA	Health Informatics Service Architecture
ODP	Open Distributed Processing
UML	Unified Modelling Language

5 Methodological principles

5.1 General

This part of ISO 12967 encompasses the computational viewpoint, which is concerned in answering HISA middleware design aspects through the functional decomposition of the system into a set of computational objects that interact at interfaces, also enabling distribution. The Health Informatics Service Architecture will thus be further specified in terms of computational objects, which manage information and provide services, and their interfaces, starting from the clusters of objects identified in ISO 12967-1 and further detailed in ISO 12967-2.

5.2 Clusters of objects

ISO 12967-1 has identified the scope, need for, and use of the HISA standard by both developers and end users. It has described the scope of the business objects from the organization's viewpoint, by summarising the related user activities and requirements through natural language. During this process the main healthcare common clusters of objects have been identified:

1) Subject of care objects

These objects handle the information necessary for supporting the users' activities identified in the "Subject of Care workflow" of ISO 12967-1.

2) Activity management objects

These objects handle the information necessary for supporting the users' activities identified in the "Activity Management workflow" of ISO 12967-1.

3) Clinical information objects

These objects handle the information necessary for supporting the users' activities identified in the "Clinical Information workflow" of ISO 12967-1.

4) Users and authorization objects

These objects handle the information necessary for supporting the users' activities related to the management of users and authorizations, as identified in ISO 12967-1.

5) Resources objects

These objects handle the information necessary for supporting the users' activities related to the management of resources, as identified in ISO 12967-1.

6) Classification objects

These objects handle the information necessary for supporting the users' activities related to the management of classifications, coding criteria and dictionaries, as identified in ISO 12967-1.

7) Messaging objects

These objects handle the information necessary for supporting the structuring of data and the communications with other systems through messaging mechanisms, as identified in ISO 12967-1.

ISO 12967-2 has formalized the conceptual model of the information being manipulated by the services, arising from the textual descriptions contained in ISO 12967-1. For each of the clusters of objects, an information model composed of information objects has been identified in ISO 12967-2.

This part of ISO 12967 defines the computational model, composed of computational objects, capable of meeting the requirements described in ISO 12967-1. It is necessary here to identify its relationship to the information model, and the interfaces or access mechanisms it provides to access the information handled by the system, which below are also referred to as methods or services.

The individual methods provided by the computational objects are described illustrating how they allow actual access to the information handled by the system (identifying the interfaces, the constraints, as well as which information of the underlying overall information model is accessed), and eventual parallel actions to be taken.

5.3 Computational language

This part of ISO 12967 is directly concerned with the distribution of processing but not with the interaction mechanisms that enable distribution to occur. The computational specification decomposes the system into objects performing individual functions and interacting at well-defined interfaces.

The heart of the computational language is the computational object model, which constrains the computational specification by defining:

- form of interface that an object can have;
- the way the interfaces can be bound and the forms of interaction which can take place at them;
- actions an object can perform, in particular the creation of new objects and interfaces.

5.4 The computational objects and interfaces

The computational objects provide the interfaces through which it is possible to access and manipulate the information managed by the information objects described in the information viewpoint. Each cluster itself can be seen as a computational object, providing interfaces that comprise all interfaces of the objects belonging to such cluster. The computational objects are defined at the level of the HISA object.

For each cluster of objects there will be a set of computational objects providing interfaces allowing the management of the common information and business logic relevant to the organization. Two types of computational object are foreseen per cluster:

— basic computational objects deriving directly from the corresponding information object (i.e. one computational object per information object);

— complex, higher-level computational objects providing interfaces achieving higher-level complex business logic.

Thus, the majority of the computational objects will be derived directly from the corresponding information objects. The further higher-level computational objects also envisaged provide interfaces achieving higher-level complex business logic on possibly multiple information objects within the same operation. Such more complex business logic is described in ISO 12967-1 and has to do with the main workflow processes (i.e. patient management, activity management, etc.).

NOTE The term patient is used in this specification as a synonym of subject of care as has been done in the other parts of ISO 12967.

The basic computational objects, corresponding one to one to the information objects, will be equipped with *standard* lower-level basic interfaces having the scope of *adding, updating and deleting – in short maintaining*, *listing*, and *getting one instance* of the main classes described in the information viewpoint. These basic methods allow the access to and the manipulation of each element of the underlying model and secure the openness of the system.

Figure 2 shows an example.

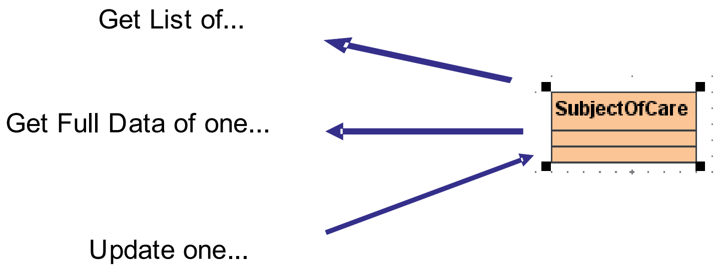

Figure 2 — Example of "basic services"

NOTE 1 The actual basic services that shall be available for HISA objects are detailed in 6.2.

The higher-level computational objects implement more complex business transactions on the objects of the information model, simplifying and ensuring consistency of developments and building common fundamental procedures of the organization.

EXAMPLES:

Patient/person area, including registering a person, patient administration, merging patient identifiers, period of care, etc.;

Activity management and life cycle, including requests, planning, booking, etc.;

Clinical and EHR, including terminologies, classifications, problem-orientation, etc.;

Resource management, including standard usages, etc.

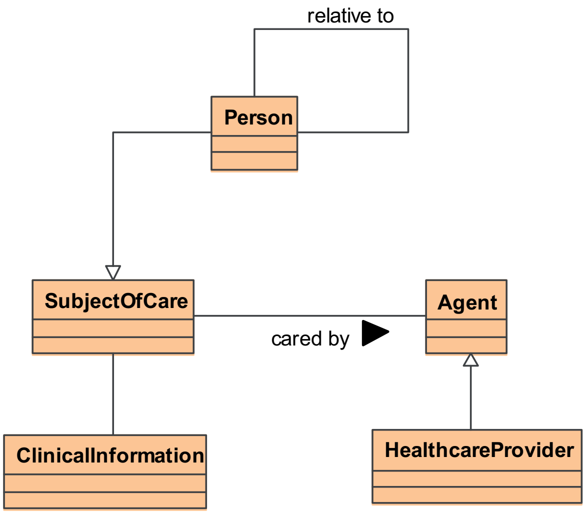

Figure 3 — Example of "complex services"

NOTE 2 The actual complex services that shall be available for HISA objects are detailed in 6.4.

The HISA middleware also provides a set of interfaces relating to functionalities of general utility for the management of the overall system, with respect to the execution of particular functionalities. These services do not pertain to any specific middleware component, and are related to general-purpose issues like session management (logging in and out of the system, setting system variables, etc.), transaction management, etc. These services will be provided by at least a further computational object equipped with appropriate methods, namely the general purpose interface.

5.5 Interaction

Three types of interaction are envisaged in ODP: signals, operations and flows. Signals are single actions conveying data from one object to another, while operations can be seen as “client-server” interactions between objects in which the server object elaborates the data provided by the client, sending back a result. Flows can be considered as a sequence of interactions (i.e. information exchanges) between objects pertaining to a specific domain.

The interaction type is part of the interface signature. In HISA the focus is on the interaction type *operation*. For this reason it will not be explicitly referred to in this specification. Such interaction type implies the need to identify for each computational object the role it plays in the client-server interaction. However, HISA prescribes the general external characteristics through which each identified computational object provides interfaces, while the interaction amongst the computational objects is not part of this part of ISO 12967. Thus, the role is always “server”.

NOTE Of the three types of interaction or operation are the ones that present a service-oriented call/return, or client-server pattern required in the service architecture. The other interaction types can, when necessary, be described as particular type of operations.

6 General characteristics of the model

6.1 The two types of computational objects for handling the information

The computational objects provide the interfaces through which it is possible to access and manipulate the information managed by the information objects described in the information viewpoint. An example of the two types of computational objects is displayed in Figure 4, and shall be referred to in the following as “basic” and “complex” computational objects according to the terminology adopted in 5.4. The methods that these will expose shall also be referred to in the following as “basic” and “complex”.

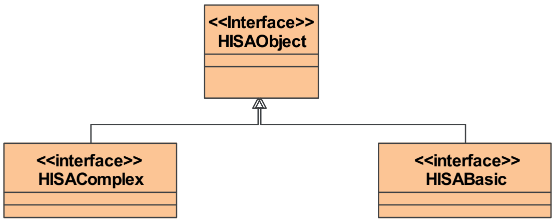

Figure 4 — types of computational object

6.2 The basic methods

6.2.1 General requirement

For each class belonging to the seven clusters of objects defined in ISO 12967-1 and specified in the information viewpoint the middleware shall be equipped with a computational object in turn equipped with a set of methods allowing to access and to manipulate every concept (i.e. objects and properties) of the class, the generic structure of which is displayed in Figure 5.

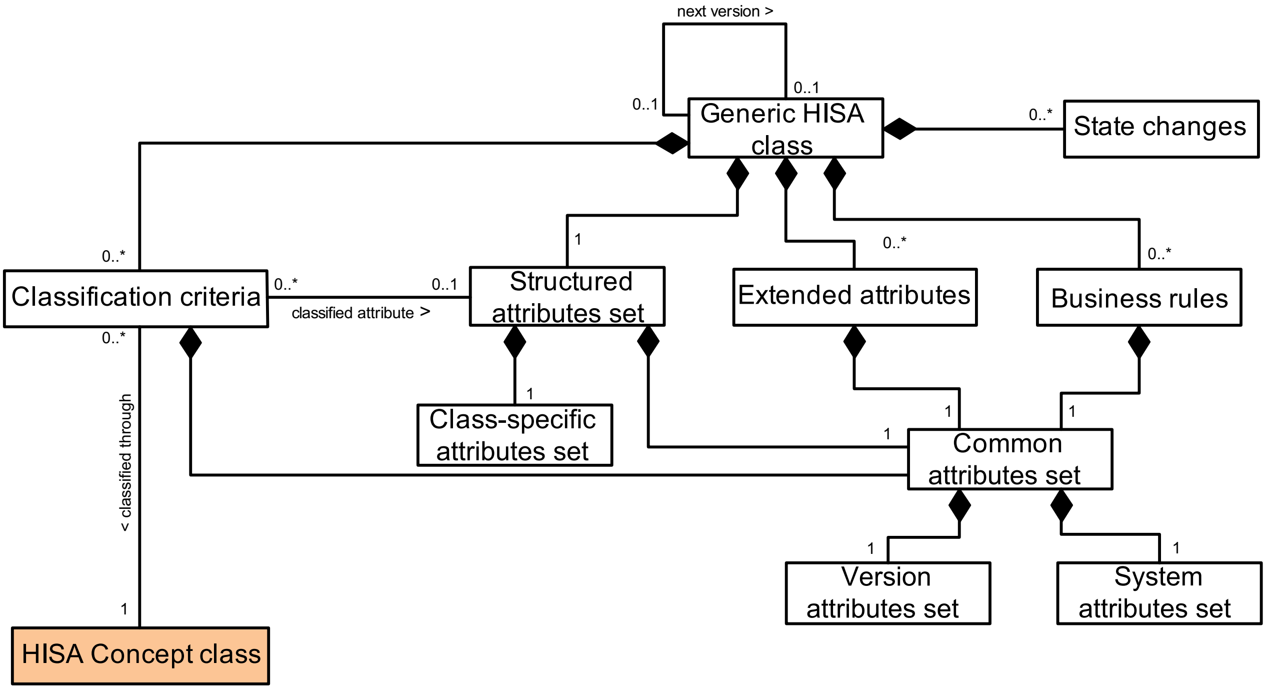

Figure 5 — Generic structure of the computational objects

The following methods shall be available in the basic computational objects. Each method has a scope and a description. Many of the method specification tables also include an example.

6.2.2 “Add” basic methods

6.2.2.1 General

The “add” methods shall allow the client of the computational object to create instances of HISA objects.

6.2.2.2 List of methods

6.2.2.2.1 Method add

Method	**add**
Scope	Shall be used to create a new instance of a HISA object.
Description	The instances that shall be added are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The instance, its class-specific attribute set, and the common system and version properties shall be created through this method.
Example	The addition of a new person in the system shall be accomplished by calling the add method of the person computational object [Person.add]. The caller will pass as input several fields belonging to the class-specific attribute set (id, name, birthTime, deceasedTime, gender, address, etc.). The method shall also allow the user to pass information to override any default value for its common system and version-related attributes.

6.2.2.2.2 Method Xadd

Method	**Xadd**
Scope	Shall be used to create a new instance of extended data and associate it to a HISA object.
Description	The specification of the extended data object that will be created is found in 6.3.7 of ISO 12967-2:2009. The instances to which the extended data shall be attached are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The extended data properties and the common system and version properties it comprises shall be created through this method.
Example	The calling of this method [Person.Xadd] shall allow doing things such as adding, among others: the digital photograph of the person to the instance of the person object, the scanned image of the signed consent to receive treatment, etc. The semantics of the extended datum shall be classified in its property type, the media type in the property mediaType, the language in the language property, etc.

6.2.2.2.3 Method Cadd

Method	**Cadd**
Scope	Shall be used to create a new instance of classification criteria data and associate it to a HISA object.
Description	The specification of the classification criteria data that will be created is found in 6.3.10 of ISO 12967-2:2009s standard. The instances to which the classification data shall be attached are the individual HISA classes specified in Clause 7 ISO 12967-2:2009. The classification data properties and its common system and version properties shall be created through this method.
Example	Persons might be classified in healthcare organizations according to their education. The calling of this method [Person.Cadd] allows classifying the person according to the criteria in use e.g. in the national, regional, or local classification in use.

6.2.2.2.4 Method BRadd

Method	BRadd
Scope	Shall be used to create a new instance of a business rule concerning the HISA object to which it is associated.
Description	The specification of the business rule data that will be created is found in 6.3.9 of ISO 12967-2:2009. The instances to which the business rule data shall be attached are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The business rule properties and its common system and version properties shall be created through this method.
Example	As happens in Denmark, experimental drugs are used in some treatments and need specific business rules to be identified for their correct usage, for example that a specific drug can be prescribed only by certain physicians for a certain class of illnesses. In this case, the method [Resource.BRadd] shall be used once the business rule is defined.

6.2.2.3 Behavioural specifications

It is up to the client of the object (i.e. the caller) to pass valid data as input for the class-specific properties and for those system and version attributes for which the default values should be overridden.

The system shall generate appropriate values for the common system and version attributes of the HISA class when the instance is created, for those parameters that are not passed by the caller. In particular the following.

— Instance identifier shall be unique within the open distributed processing environment of the system within the enterprise. A unique identifier of the HISA class may be specified by the caller of the operation. In this case the service shall verify the uniqueness of the identifier.

— System shall take care of managing state changes by adding dynamically and automatically the values that record the changes occurred in the specific properties of the class or when extended data, classification data and business rules are attached to it, thus keeping track of the life cycle of the instance during time. The state changes class is specified in 6.3.8 of ISO 12967-2:2009.

— Method shall return to the client at least the set of common attributes related to the HISA object created. Such information shall include a value indicating successful execution of the operation or an error indicating the reason for failure.

— Method shall implement all integrity checks with respect to the information passed by the client of operation, aborting it if the operation would lead to an inconsistent informational state.

6.2.3 "Update" basic methods

6.2.3.1 General

The "update" methods shall allow the client of the operation to update instances of HISA objects.

6.2.3.2 List of methods

6.2.3.2.1 Method update

Method	update
Scope	Shall be used to update an existing instance of a HISA object.
Description	The instances that shall be updated are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The instance, its class-specific attribute set, and the common system and version properties shall be updated through this method.
Example	The updating of a person in the system shall be accomplished by calling the update method of the person computational object [Person.update]. The method shall also allow the user to pass information to override any default value for its common system and version-related attributes.

6.2.3.2.2 Method Xupdate

Method	**Xupdate**
Scope	Shall be used to update an existing instance of the extended data associated to a HISA object.
Description	The extended data instances that shall be updated are the ones already attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The extended data object and its common system and version properties shall be updated through this method.
Example	The updating of the properties of the extended data of a person in the system shall be accomplished by calling the update method of the person computational object [Person.Xupdate]. The method shall also allow the user to pass information to override any default value for the extended data's common system and version-related attributes.

6.2.3.2.3 Method Cupdate

Method	**Cupdate**
Scope	Shall be used to update an existing instance of classification data associated to a HISA object.
Description	The classification data instances that shall be updated are the ones already attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The classification object and its common system and version properties shall be updated through this method.
Example	The updating of the classification criteria of a person in the system shall be accomplished by calling the update method of the person computational object [Person.Cupdate]. The method shall also allow the user to pass information to override any default value for its common system and version-related attributes.

6.2.3.2.4 Method BRupdate

Method	**BRupdate**
Scope	Shall be used to update an existing instance of a business rule associated to the HISA object.
Description	The specification of the business rule that will be created is found in 6.3.8 of ISO 12967-2:2009. The instances to which the business rule data shall be attached are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The business rule properties and its common system and version properties shall be updated through this method.
Example	Experimental drugs are often used in treatments and need specific business rules to be identified for their correct usage, e.g., that a specific drug can be prescribed only by certain physicians for a certain class of illnesses. When these business rules need to be updated, the method [Resource.BRupdate] shall be used.

6.2.3.3 Behavioural specifications

It is up to the client of the object to pass valid data as input for the class specific properties and for those system and version attributes for which the default values should be overridden.

The system shall generate and/or update the appropriate values for the common system and version attributes of the HISA class when it is updated. In particular the following.

— Operation shall require as input the common attributes necessary for the update information.

— Instance identifier shall never be updated.

— Update shall not occur if the timestamp of the instance to be updated differs from the timestamp received as input, since this indicates that the value in the HISA system has changed since the client originally read the instance from the system. The change occurred while the caller was performing the operation and the update would cause inconsistency in the common information heritage.

NOTE Such an approach in managing concurrent access to data is generally called "optimistic-locking".

- System shall take care of managing state changes when the update operation is invoked by adding dynamically and automatically the values that record the changes occurred in the specific properties of the class or when extended data, classification data and business rules attached to it are updated, thus keeping track of the life cycle of the instance during time. The state changes class is specified in 6.3.8 of ISO 12967-2:2009.

- Operations shall return to the client at least the set of common attributes related to the HISA object updated with the update operation. Such information shall include a value indicating successful execution of the operation or an error indicating the reason for failure.

- operations shall implement all integrity checks with respect to the information passed by the client of operation, aborting it if the operation would lead to an inconsistent informational state.

6.2.4 "Delete" basic methods

6.2.4.1 General

The "delete" methods shall allow the client of the operation to logically delete instances of HISA objects.

6.2.4.2 List of methods

6.2.4.2.1 Method delete

Method	**delete**
Scope	Shall be used to delete an existing instance of a HISA object.
Description	The instances that shall be deleted are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009f. The object, its class-specific attribute set, and its common system and version properties shall be deleted through this method.

6.2.4.2.2 Method Xdelete

Method	**Xdelete**
Scope	Shall be used to logically delete an existing instance of the extended data associated to a HISA object.
Description	The extended data instances that are deleted shall be the ones already attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The extended data object and its common system and version properties shall be logically deleted through this method.

6.2.4.2.3 Method Cdelete

Method	**Cdelete**
Scope	Shall be used to logically delete an existing instance of classification data associated to a HISA object.
Description	The classification data instances that shall be deleted are the ones already attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The object and its common system and version attributes shall be logically deleted through this method.

6.2.4.2.4 Method BRdelete

Method	**BRdelete**
Scope	Shall be used to logically delete an existing instance of a business rule associated to the HISA object.
Description	The specification of the business rule that shall be deleted is found in 6.3.8 of ISO 12967-2:2009. The instances to which the business rule data is attached are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The business rule properties and its common system and version properties shall be deleted through this method.

6.2.4.2.5 Method pack

Method	**Pack**
Scope	Shall be used to physically delete the logically deleted instances.
Description	The logical deletion allows keeping in the system instances for which clients have requested the deletion. This functionality shall be available mainly for system administration purposes, to physically remove elements that have been logically deleted and can safely be removed from the system.

6.2.4.3 Behavioural specifications

The system shall generate and/or update the appropriate values for the common system and version attributes of the HISA class when it is deleted. In particular the following.

- Operation shall require as input the common attributes necessary for the delete information.
- Logical deletion shall cause the “isDeleted” attribute of the object to be set to “True”, without physically removing the instance from the common information heritage of the enterprise.
- Delete operation shall not occur if the timestamp of the instance to be logically deleted differs from the timestamp received as input, since this indicates that the value has changed in the HISA system since the client originally read the value from the system. The change occurred while the caller was performing the operation and the logical delete operation would cause inconsistency in the common information heritage.
- System shall take care of managing state changes when the logical delete operation is invoked by adding, dynamically and automatically, the values that record the changes occurred in the specific properties of the class or when extended data, classification data and business rules attached to it are deleted, thus keeping track of the life cycle of the instance during time. The state changes class is specified in 6.3.8 of ISO 12967-2.2009.
- Operations shall return to the client a value indicating successful execution of the operation or an error indicating the reason for failure.
- Operations shall implement all integrity checks with respect to the information passed by the client of the operation aborting it if the operation would lead to inconsistent information.

6.2.5 “Detail” basic methods

6.2.5.1 General

The “detail” operations shall allow the client of the operation to fetch one fully detailed instance of a HISA object.

6.2.5.2 List of methods

6.2.5.2.1 Method detail

Method	**Detail**
Scope	Shall be used to get the information regarding an existing instance of the HISA object.
Description	The instances that shall be retrieved are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The object, its class-specific attribute set and its common system and version attributes shall be retrieved through this method.
Example	The retrieval of the full detail of a person’s properties in the system for displaying it through user functionality shall be accomplished by calling this method of the person computational object [Person.detail], thus accessing the person’s id, his/her birthDate, etc.

6.2.5.2.2 Method Xdetail

Method	**Xdetail**
Scope	Shall be used to retrieve an existing instance of the extended data associated to a HISA object.
Description	The extended data instances that shall be retrieved from the system are described in 6.3.7 of ISO 12967-2:2009 and are attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The extended data object and its common system and version attributes shall be extracted and presented to the caller through this method.
Example	The retrieval of the extended data properties of a person in the system, such as the person's picture or the signed consensus for receiving treatment are examples of extended data that may need to be visualized by user functionality by calling the Xdetail method of the person computational object [Person.Xdetail].

6.2.5.2.3 Method Cdetail

Method	**Cdetail**
Scope	Shall be used to retrieve an existing instance of the classification data associated to a HISA object.
Description	The classification data instances that shall be retrieved form the system are described in 6.3.10 of ISO 12967-2:2009 and attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The classification data object's common system and version attributes shall be extracted and presented to the caller through this method.
Example	The retrieval of the classification data properties of a person, such as the level of education coded according to national, regional, or local criteria shall be accomplished by calling this method of the person computational object [Person.Cdetail].

6.2.5.2.4 Method BRdetail

Method	**BRdetail**
Scope	Shall be used to retrieve an existing instance of a business rule associated to the HISA object.
Description	The specification of the business rule that shall be retrieved is found in 6.3.9 of ISO 12967-2:2009. The instances to which the business rule data is attached are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The business rule properties and its common system and version properties shall be extracted through this method and presented to the caller.

6.2.5.2.5 Method Sdetail

Method	**Sdetail**
Scope	Shall be used to retrieve an existing instance of the state change data associated to a HISA object.
Description	The state change data instances that shall be retrieved form the system are described in 6.3.8 of ISO 12967-2:2009 and attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The state change object's properties and its common system and version properties shall be extracted through this method
Example	During the life cycle of an activity, from the moment it is defined and requested till the moment it is completed, it can undergo quite a few changes in terms of status and properties. This method [Activity.Sdetail] allows retrieving the full detail of one change that the activity has undergone.

6.2.5.3 Behavioural specifications

— Operation shall require as input the selection criteria necessary for retrieving the detailed information of the HISA class involved in the operation, usually being the identifier(s) of the object.

— Operations shall also return to the client the set of common attributes related to the HISA object fetched from the common information heritage. The set of information returned shall include a value indicating successful execution of the operation or an error indicating the reason for failure.

- For operations that return the detail of text and extended data items:
  - input parameter shall allow the user to specify the maximum size (in number of characters or bytes) that are expected to be in the output of the method;
  - input parameter shall allow to specify an offset parameter to fetch remaining characters (bytes) with respect to the previous maximum number of characters (bytes) parameter;
  - output parameter shall specify the actual number of characters (bytes) fetched;
  - output parameter shall specify whether there are still characters (bytes) to fetch.

6.2.6 "List" basic methods

6.2.6.1 General

The "list" operations shall allow the client of the operation to fetch a list of instances of a HISA object.

6.2.6.2 List of methods

6.2.6.2.1 Method list

Method	**list**
Scope	Shall be used to fetch the information regarding existing instances of the HISA object that meet the selection criteria passed by the client of the operation.
Description	The instances that shall be retrieved are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The objects, a significant subset of their class-specific attribute set and some common system and version properties shall be fetched through this method.
Example	The retrieval of the list of persons meeting the example selection criteria "living in Rome and born in 1980" shall be accomplished by calling this method [Person.list].

6.2.6.2.2 Method Xlist

Method	**Xlist**
Scope	Shall be used to fetch the information regarding existing extended data instances attached to one HISA objects meeting the selection criteria passed by the client of the operation.
Description	Several instances of extended data, which is described in 6.3.7 of ISO 12967-2:2009, shall be retrieved according to the selection criteria passed by the caller. The classification data are attached to one of the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The objects, a significant subset of their class-specific attribute set and some common system and version properties shall be fetched through this method.
Example	This method [Person.Xlist] can be used to fetch information regarding all extended data available in the system for one person, identified through his/her identifier.

6.2.6.2.3 Method Clist

Method	**Clist**
Scope	Shall be used to fetch the information regarding existing classification data instances attached to one HISA object meeting the selection criteria passed by the caller.
Description	The instances that shall be retrieved are the classification data described in 6.3.10 of ISO 12967-2:2009 that are associated to the existing instances of individual HISA classes specified in Clause 7 of ISO 12967-2:2009. The objects, a significant subset of their class-specific attribute set and some common system and version properties shall be fetched through this method.
Example	The retrieval of the list of classification objects associated to one Person shall be accomplished by calling this method [Person.Clist], possible examples of which are the level of education, the employment status, etc.

6.2.6.2.4 Method BRlist

Method	**BRlist**
Scope	Shall be used to retrieve a list of existing instances of business rules meeting specific selection criteria and currently associated to a well identified HISA object.
Description	The specification of the business rules that shall be retrieved is found in 6.3.9 of ISO 12967-2:2009. The instances to which the business rules are attached are the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. A significant subset of the properties of the business rules and some of the common system and version properties shall be fetched through this method and presented to the caller.
Example	Calling this method [Resource.BRlist] shall result in the retrieval of all business rules related to the usage of the specific drug.

6.2.6.2.5 Method Slist

Method	**Slist**
Scope	Shall be used to fetch several existing instances of state change data associated to one well identified HISA object.
Description	The state change data instances that shall be retrieved form the system are described in 6.3.8 of ISO 12967-2:2009 and attached to the individual HISA classes specified in Clause 7 of ISO 12967-2:2009. A significant subset of the properties of the state change objects and some of the common system and version properties shall be fetched and presented to the caller of this method.
Example	The retrieval of the several state changes that a diagnosis, handled as clinical information in HISA, has undergone in the system, shall be accomplished by calling this method of the person computational object [ClinicalInformation.Slist].

6.2.6.3 Behavioural specifications

- Operation shall require as input the selection criteria necessary for retrieving the list of instances of the HISA class involved in the operation invocation.

- Operations shall return to the client also the set of common attributes related to the HISA object. The set of information returned shall include a value indicating successful execution of the operation or an error indicating the reason for failure.

- For these operations that return multiple elements:

  - input parameter shall allow the user to specify the maximum number of elements that are expected to be in the output of the method call;

  - input parameter shall allow the specification of an offset parameter to fetch remaining elements with respect to the previous maximum number of elements parameter, thus retrieving the elements starting from offset;

  - output parameter shall specify the actual number of elements fetched;

  - output parameter shall specify whether there are still elements to fetch.

6.3 General purpose interface

6.3.1 General

The middleware shall provide at least one computational object providing a set of interfaces for methods relating to functionalities of general utility for the access to and the management of the system in the physical implementation. These methods shall take care of managing and providing session-management information that needs to be passed to and from the computational objects for the correct functioning of the HISA middleware system, which is not a stateless system.

NOTE The general purpose interface, unlike the two presented in 6.1, does not relate to the handling of information managed by the HISA middleware, rather to the functioning of the middleware as service architecture.

6.3.2 List of methods

6.3.2.1 General

At least the following methods shall be provided by the middleware.

6.3.2.2 Method login

Method	**Login**
Scope	Shall be used to create a client session with the middleware.
Description	The method shall take as input the information regarding the user that is requesting the session connection, which is specified in 7.6.3.2 of ISO 12967-2:2009, and create a connection to the system.

6.3.2.3 Method logout

Method	**Logout**
Scope	Shall be used to close a client session with the middleware.

6.3.2.4 Method setParameter

Method	**setParameter**
Scope	Shall be used to establish a specific modality within a client session with middleware.
Description	The method shall take as input the information regarding the desired parameter and set it for the user session
Example	For certain specific queries the expert user may need to consult the logically deleted records of a HISA object. Such functionality shall be enabled by setting a session parameter through this functionality. Another example is represented by the number of elements that the client of the list operations wishes to be returned. This shall also represent a parameter that can be configured in the system.

6.3.2.5 Method getParameter

Method	**getParameter**
Scope	Shall be used to retrieve information about a specific modality within a client session with middleware.
Description	The method shall take as input the information regarding the desired parameter and fetch its value for the specific user session, passing it back to the caller.

NOTE The parameters referred to in the last two tables relate to the interaction between the user of the interface or client of the operation and the middleware providing the service. There are several different parameters referring to different behavioural modalities of the system.

6.3.3 Behavioural specifications

- It is up to the client (i.e. caller) of the operation to pass valid data as input for the operations.
- System shall populate a system log with information about the session and the operations invoked by the client during a session.
- Operation shall return to the client a value indicating successful execution of the operation or an error indicating the reason for failure.
- Operations that set the system parameters shall require valid data as input for configuring the parameter.

6.4 The complex interfaces of the workflow related computational objects

6.4.1 General

The workflows that will be specified in terms of methods in the following clauses have been introduced in Clause 9 of ISO 12967-1:2009, which also provides use cases and appropriate specification at ISO 12967-1 level.

6.4.2 Complex services managing healthcare workflows

The methods of these interfaces shall implement transactions of consistent business logic supporting the events and processes of the organization.

NOTE It is clarified that each one of such transactions of business logic could also be implemented through the basic interfaces, adding/updating information in each class one at a time, but it is useful and appropriate to group such “complex business logic” into specific functions provided by an object to provide common functionality to the whole organization, at the same time optimizing performance.

The middleware shall, at least, provide computational objects providing methods supporting:

- fundamental events of the workflows identified in ISO 12967-1:
  - subject of care;
  - activity management;
  - clinical information management;
- general purpose activities relating to the access to, and the management of the system.

6.4.3 Interfaces supporting the “Subject of care workflow”

6.4.3.1 General

The middleware shall provide at least one computational object providing interfaces to operations supporting the following fundamental business activities related to the Subject of Care Workflow, described in 9.4 of ISO 12967-1:2009, in which some example use cases shall also be found.

6.4.3.2 List of methods

6.4.3.2.1 General

At least the following methods shall be provided by the middleware.

6.4.3.2.2 Method registerSOC

Method	**registerSOC**
Scope	Shall be used to register a subject of care, together with associated information (i.e. relatives, general practitioner, etc.).
Description	The service shall allow the registering of all available/relevant information related to the person regardless of whether or not that person is or will be a patient. This information covers not only the person and subject of care class but also several other related classes, such as the patient's general practitioner (i.e. an agent or healthcare provider), the patient's relatives (i.e. handled in the person class), etc. A very important part of this process is the unique identification of the subject of care within the organization running the HISA middleware. The information managed in the person and subject of care classes and handled in this method is described in 7.2.3.2 and 7.2.3.3 of ISO 12967-2:2009. The information regarding the agent is described in 7.6.3.2 of ISO 12967-2:2009.
Example	When a first encounter is established, a health informatics system will provide such method to register the subject of care.

6.4.3.2.3 Method admit

Method	admit
Scope	Shall be used to admit one patient (i.e. a Subject of Care) to receive treatment and define a new contact if not already foreseen, or modify an already registered admission.
Description	The service shall allow the registering of all available/relevant information related to the admission of a patient in a healthcare organization by not only identifying the contact, but also by identifying the person responsible for the treatment (i.e. agent). The classes that will be involved in such transaction are subject of care described in 7.2.3.2 of ISO 12967-2:2009, Agent described in 7.6.3.2 of ISO 12967-2:2009, and contact described in 7.2.3.5 of ISO 12967-2:2009.
Example	This is the typical procedure that takes place in patient administration systems when a Contact is initialized and the patient is admitted to the healthcare centre.

6.4.3.2.4 Method discharge

Method	**Discharge**
Scope	Shall be used to discharge one patient and end the treatment process, closing the contact within the organization.
Description	The service shall allow the registering of all available/relevant information related to the discharge of a patient in a healthcare organization. The classes that shall be involved in such transaction are subject of care described in 7.2.3.2 of ISO 12967-2:2009, agent described in 7.6.3.2 of ISO 12967-2:2009 and contact described in 7.2.3.5 of ISO 12967-2:2009.
Example	During discharge, the appropriate values related to the contact need to be set, so that all clinical and administrative procedures can be finalized. In Italy, for example, at each discharge, a "Scheda Dimissione Ospedaliera" needs to be filled out and sent to the regional authorities.

6.4.3.2.5 Method transfer

Method	**Transfer**
Scope	Shall be used to transfer the responsibility of one patient from one provider to another within the same healthcare organization. This usually also implies changing the patient location inside the organization.
Description	The service shall allow the registering of all available/relevant information related to the transfer of a patient in a healthcare organization from one organizational unit to another. The classes that will be involved in such transaction are subject of care described in 7.2.3.2 of ISO 12967-2:2009, agent described in 7.6.3.2 of ISO 12967-2:2009 and contact described in 7.2.3.5 of ISO 12967-2:2009.

6.4.3.2.6 Method merge

Method	Merge
Scope	Shall be used to merge contact data from source to target by moving all dependencies to the target and by annulling the source Contact.
Description	The service shall require as input the identifiers of the two contacts to merge. The classes that will be involved in such transaction are contact described in 7.2.3.5 of ISO 121967-2:2009, and possibly all classes associated to the contact as specified in that same subclause.
Example	It is a routine procedure to assign a temporary contact identifier when it is necessary to provide treatment before having performed a fully-fledged admission or in the case that the connection to the dedicated patient management system handling the master contact index is unavailable. When the admission is finally completed or the connection is re-established, the contacts shall be merged.

6.4.3.2.7 Method getContact

Method	getContact
Scope	Shall be used to retrieve full detailed information on one contact.
Description	The service shall require as input the identifier of the contact, the information of which shall be retrieved. The class that will be involved in such transaction is contact described in 7.2.3.5 of ISO 12967-2:2009.

6.4.3.2.8 Method getSOC

Method	getSOC
Scope	Shall be used to retrieve full detailed information on one subject of care.
Description	The service shall require as input the identifier of the subject of care, the information on whom shall be retrieved. The service shall retrieve all available/relevant information related to the person. This information covers not only the person and subject of care class but also several other related classes, such as the patient's general practitioner (i.e. an agent or healthcare provider), the patient's relatives (i.e. handled in the person class), etc. The information managed in the person and subject of care classes and handled through this method is described in 7.2.3.2 of ISO 12967-2:2009. The information regarding the agent is described in 7.6.3.2 of ISO 12967-2:2009.

6.4.4 Interfaces supporting the "Clinical information workflow"

6.4.4.1 General

The middleware shall provide at least one computational object providing interfaces to operations supporting the following fundamental business activities related to the Clinical Information Workflow, described in 9.5 of ISO 12967-1:2009, in which also some examples use cases are found.

6.4.4.2 List of methods

6.4.4.2.1 General

At least the following methods shall be provided by the middleware

6.4.4.2.2 Method annul

Method	**Annul**
Scope	Shall be used to annul (declare invalid, cancel) some clinical information.
Description	The service shall require as input the identifier of the clinical information that will be annulled. The information managed in the clinical information class handled through this method is described in 7.4.3.1 of ISO 12967-2:2009.
Example	In case of errors in the registration of clinical information, one possible way of using this method in a use case to rectify the issue is to create a new instance of the clinical information with the correct information, annul the previous one, and associate the new instance to the old through the association class reason for association of C.I.

6.4.4.2.3 Method registerCI

Method	**registerCI**
Scope	Shall be used to register (commit in the HISA middleware) the clinical information.
Description	The service shall require as input the information of the clinical information that will be handled. The information managed in the clinical information class handled through this method is described in 7.4.3.1 of ISO 12967-2:2009.
Example	This method would allow the insertion of the daily measurement of the patient's body temperature.

6.4.4.2.4 Method validateCI

Method	**validateCI**
Scope	Shall be used to certify the validity of the clinical information by an authorized agent.
Description	The service shall require as input the information of the clinical information that will be validated and of the agent actually validating it. The information managed in the clinical information class handled through this method is described in 7.4.3.1 of ISO 12967-2:2009.
Example	The presence of clinical information coming, for example, from laboratory instrumentation has to be validated and certified by a competent clinician, before it is considered valid and disclosed in clinical reporting.

6.4.4.2.5 generateAggregation

Method	**generateAggregation**
Scope	Shall be used to create an aggregated structure of the clinical information, structuring elementary clinical information into meaningful aggregations, such as reports, extracts, etc.
Description	The service shall require, as input, the information of the clinical information that will be handled. The information managed in the clinical information class handled through this method is described in 7.4.3.1 of ISO 12967-2:2009.
Example	The existing elementary clinical information in the system may be aggregated into more complex aggregates according to structures that are commonly used for communication information, such archetypes.

6.4.5 Interfaces supporting the "Activity management workflow"

6.4.5.1 General

The middleware shall provide at least one computational object providing interfaces to operations supporting the following fundamental business activities related to the activity management workflow, described in 9.6 of ISO 12967-1:2009, in which also some example use cases are to be found. Subclause 9.6.1 of ISO 12967-1:2009 also specifies the states in which an activity will find itself within its life cycle.

6.4.5.2 List of methods

6.4.5.2.1 General

At least the following methods shall be provided by the middleware.

6.4.5.2.2 Method request

Method	**Request**
Scope	Shall be used to request the execution of one or more activities.
Description	The service shall require as input the information of the activities that are requested. The service registers new actual activities with the resources, clinical information and any other associated property, which are planned for their execution. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes (e.g. resources, clinical Information, etc.).

6.4.5.2.3 Method accept

Method	**Accept**
Scope	Shall be used to accept the responsibility to actually execute the requested activities by the activity provider.
Description	The service shall require as input the information of the activities that have been requested. The activity is specified in 7.3.3.2 of ISO 12967-2:2009.

6.4.5.2.4 Method annul

Method	**Annul**
Scope	Shall be used to annul (declare invalid, cancel) the activity for certain reasons.
Description	The service shall require as input the identifier of the activity that is to be annulled. The activity is specified in 7.3.3.2 of ISO 12967-2:2009.

6.4.5.2.5 Method plan

Method	**Plan**
Scope	Shall be used to explicitly plan (i.e. assign to date, time and service point of execution) one set of activities.
Description	The service shall require as input the identifiers of the activities that are planned, together with the information of the agents that will act to perform the activities, and eventual resources that are deemed necessary for carrying out the activity. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes. The agent class is specified in 7.6.3.2 of ISO 12967-2:2009.

6.4.5.2.6 Method registerExecution

Method	**registerExecution**
Scope	Shall be used to register (commit in the HISA middleware) the execution details of the activity.
Description	The service shall require as input the identifiers of the activities that have been executed, plus all information regarding the execution, such as the actual resources used and eventual clinical information that has been generated as a result of the activity. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes.

6.4.5.2.7 Method report

Method	Report
Scope	Shall be used to create the report of the execution of the activity.
Description	The service shall require as input the identifier of the activity that has been executed, plus all information regarding the execution, for example the information regarding the actual resources used and eventual clinical information that has been generated as a result of the activity. Output shall be then aggregated into a report. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes involved in the transaction.

6.4.5.2.8 Method setReady

Method	setReady
Scope	Shall be used to set the status of the activity as “ready for execution”, meaning that all preliminary activities have been completed and the execution of the activity may actually start.
Description	The service shall require as input the identifier of the activity that has to be executed, plus any further information regarding the execution, for example the information regarding the actual resources used and eventual clinical information that has been generated as a result of the activity. The status of the activity shall be set accordingly. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes involved in the transaction.
Example	Examinations at a laboratory are planned in advance, possibly a matter of days. Once the resources planned to be used for the examination are prepared for the time slot and the subject of care attends, registering his/her presence to the acceptance desk, the activity is set to the appropriate status through this method.

6.4.5.2.9 Method startExecution

Method	startExecution
Scope	Shall be used to set the status of the activity as “in progress”.
Description	The service shall require as input the identifier of the activity that is being executed, plus any further information regarding the execution. The status shall be set accordingly. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes involved in the transaction.

6.4.5.2.10 Method terminate

Method	Terminate
Scope	Shall be used to terminate a group of activities.
Description	The service shall require as input the identifiers of the activities that have been executed, plus any further information regarding the execution, for example the information regarding the actual resources used and eventual clinical information that has been generated as a result of the activity. The status shall be set accordingly. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes involved in the transaction.

6.4.5.2.11 Method suspend

Method	Suspend
Scope	Shall be used to suspend a group of activities.
Description	The service shall require as input the identifiers of the activities that have been suspended, plus any further information regarding the suspension. The status shall be set accordingly. The activity is specified in 7.3.3.2 of ISO 12967-2:2009, which also references the clauses specifying the associated classes involved in the transaction.

A further overall example use case highlighting the activity life cycle is represented by the following situation, which illustrates the usage of some of the methods specified above.

- Outpatient clinic requests an exam for a certain out-patient to the radiology department. The functionality is achieved using the request method.
- Request is accepted at the radiology department through the accept method and the activity is scheduled by using the plan method.
- Before being admitted for the actual execution of the exam, the patient has to pay the cost of the service to a third administrative office, which is different both from the outpatient clinic and from the radiology. Once this is achieved the activity status is set to “ready for execution”, using the setReady method.
- After the execution of the exam, the procedure needs to be completed, for example the administrative office shall be informed of the detail of the activities actually executed, to invoice the patient additional costs, should additional exams (not foreseen at the beginning) have been executed. This shall be achieved through the method registerExecution.
- Process shall then be closed by terminating the activity using the method terminate.

6.4.6 Behavioural specifications, common to the complex services

It is up to the client of the operation to pass valid data as input for the class-specific properties and for those system and version attributes for which the default values should be overridden.

The system shall generate appropriate values for the common system and version attributes of the HISA classes when they are involved in the transaction, for those parameters that are not passed by the client of the operation. In particular the following.

- Instance identifier(s) shall be unique within the open distributed processing environment of the system within the enterprise. The client of the operation may specify a unique identifier of each HISA class involved in the operation. In this case the service shall verify the uniqueness of the identifier.
- System shall take care of managing state changes of the involved HISA object(s) by adding dynamically and automatically the values that record the changes occurred in the specific attributes of the class(es), thus keeping track of the life cycle of the instance(s) during time.
- Operations shall return to the client at least the set of common attributes related to the HISA object(s) involved in the operation. Such information shall include a value indicating successful execution of the operation or an error indicating the reason for failure.
- Operations shall implement all integrity checks with respect to the information passed by the client of operation, aborting it if the operation leads to an inconsistent informational state within the common information heritage.
- For operations that return multiple elements:
    - input parameter shall allow the user to specify the maximum number of elements that are expected to be in the output of the method call;
    - input parameter shall allow an offset parameter to be specified to fetch remaining elements with respect to the previous maximum number of elements parameter;
    - output parameter shall specify the actual number of elements fetched;
    - output parameter shall specify whether there are still elements to fetch.

- For operations that return the detail of text and extended data items:
  - input parameter shall allow the user to specify the maximum number of characters (or bytes) that are expected to be in the output of the method;
  - input parameter shall allow to specify an offset parameter to fetch remaining characters (bytes) with respect to the previous maximum number of characters (bytes) parameter;
  - output parameter shall specify the actual number of characters (bytes) fetched;
  - output parameter shall specify whether there are still characters (bytes) to fetch.

6.5 Common requirements of the interfaces

6.5.1 Interface documentation and organization

It shall be under the responsibility of the designer/supplier of a HISA-conformant middleware to describe and document the interfaces provided according to a consistent set of specifications, according to the requirements of this part of ISO 12967 and to the prescriptions of the ISO-ODP documents.

6.5.2 Naming criteria

Criteria according to which interfaces and methods are named and structured shall be common to the whole middleware, shall allow an intuitive identification of the semantics and of the involved objects by human beings, and shall be independent of the technological platform adopted for the implementation of the middleware.

In particular, in an object-oriented approach, the system shall be organized in packages, according to the division in clusters of objects identified in ISO 12967-1. A further package shall contain the objects related to the general purpose interface.

The packages shall comprise objects corresponding to the computational objects and exposing properties and operations. The properties and operations shall follow the naming criteria identified in the previous sections.

On the other hand, where the design and development is done in a non-object-oriented practice, the system shall be organized in areas corresponding to the clusters of objects identified in ISO 12967-1. A further area shall contain the operations related to the general purpose interface. In such cases the operations related to each section shall be named in a manner that allows identifying the area by prefixing the operation with:

- two characters identifying the cluster of objects or the general purpose interface, according to the following rationale:

1)	Subject of care objects	**sc**
2)	Activity management objects	**am**
3)	Clinical information objects	**ci**
4)	Users and authorization objects	**ua**
5)	Resources objects	**ro**
6)	Classification objects	**co**
7)	Messaging objects	**mo**
8)	General purpose interface	**gp**

- acronym of the HISA Class involved in the operation, where it is a basic operation.

6.5.3 Data types

The data types are tightly coupled to the technological platform on which middleware interfaces will be developed and released. Where the data types do not correspond to the ones identified in the HISA multi-part standard, the vendor shall provide mapping tables highlighting the correspondences with the HISA data types.

6.5.4 Structure and organization of the interfaces

All interfaces are organized according to the following common criteria:

datatype servicename (in datatype i_parameter1, …, in datatype i_parameterN out datatype pn, out datatype o_yyy)

where:

- first datatype is the data type of the value returned by the method; this value will always represent the eventual error or success code related to the execution of the service, exceptions shall be based on such a return value.
- servicename is the name of the method of the interface;
- in keyword specifies that the parameter is input to the method;
- out keyword specifies that the parameter is output to the method;
- between the in/out keywords the datatype of the parameter specifies the type of information exchanged through the methods.

Annex A
(informative)

Examples of services

The following examples represent the service specification using the structure just defined in a non-object-oriented practice.

EXAMPLE 1 Add a person instance

integer scPersonAdd (in [System attributes set], in [Version attributes set], in [Class-specific attribute set], out [Set of system attributes])

Where:

[Set of system attributes]: [ObjectIdentifier instanceID, String displayName, Identifier userCode, InternalTimestamp timestamp, DateTime creationTime, ObjectIdentifier creationAgent, ObjectIdentifier creationUnit, DateTime updateTime, ObjectIdentifier updateAgent, ObjectIdentifier updateUnit, String authorization, Boolean isFrozen]

NOTE 1 The exact and full specification of the system attribute set is found in 6.3.5 of ISO 12967-2:2009.

[Set of version attributes]: [Ordinal sequence, DateTime startValidityDate, DateTime endValidityDate]

NOTE 2 The exact and full specification of the version attribute set is found in 6.3.6 of ISO 12967-2:2009.

[Class-specific attribute set]: [String id, String fname, String lname, DateTime birthTime, DateTime deceasedTime, String Gender, String address, URI telcom]

NOTE 3 The exact and full specification of the person class-specific attribute set is found in 7.2.3.3 of ISO 12967-2:2009.

NOTE 4 The data type returned by the service in the example has been defined as an integer in accordance with common practice in service implementation.

NOTE 5 The service name in the example has been chosen to ease the reader. The actual naming of the services shall follow the rules of the technological environments in which the services shall be developed and released.

EXAMPLE 2 Get the full detail of a person instance

Integer scPersonDetail (in [Identifiers in System attributes set], in [Identifiers in Class-specific attribute set], out [Set of system attributes], out [Set of version attributes], out [Class-specific attribute set])

where:

[Set of system attributes]: [ObjectIdentifier instanceID, String displayName, Identifier userCode, InternalTimestamp timestamp, DateTime creationTime, ObjectIdentifier creationAgent, ObjectIdentifier creationUnit, DateTime updateTime, ObjectIdentifier updateAgent, ObjectIdentifier updateUnit, String authorization, Boolean isFrozen]

NOTE 6 The exact and full specification of the system attribute set is found in 6.3.5 of ISO 12967-2:2009.

[Set of version attributes]: [Ordinal sequence, DateTime startValidityDate, DateTime endValidityDate]

NOTE 7: The exact and full specification of the version attribute set is found in 6.3.6 of ISO 12967-2:2009.

[Class-specific attribute set]: [String id, String fname, String lname, DateTime birthTime, DateTime deceasedTime, String Gender, String address, URI telcom]

NOTE 8: The exact and full specification of the person class-specific attribute set is found in 7.2.3.3 of ISO 12967-2:2009.

[Identifiers in System attributes set] represent the identifiers in the property set.

[Identifiers in Class-specific attribute set] represent the identifiers in the property set.

NOTE 9 The data type returned by the service in the example has been defined as an integer in accordance to common practice in service implementation.

NOTE 10 The service name in the example has been chosen to ease the reader. The actual naming of the services shall follow the rules of the technological environments in which the services are developed and released.

Bibliography

[1] REYNOLDS, M. and WEJERFELD, I., *Short Strategic Study — Health Information Infrastructure — Final report*, CEN/TC 251/N00-074. Sept. 2000. Available from http://www.hisa-standard.org

[2] RM-ODP: *The Reference Model for Open Distributed Processing*. Available from http://www.rm-odp.net/

[3] AMBLER, S.W., White paper: *The Design of a Robust Persistence Layer For Relational Databases*. Available from http://www.ambysoft.com/essays/persistenceLayer.html

[4] EN 13940-1, *Health informatics — System of concepts to support continuity of care — Part 1: Basic concepts*

[5] EN 14822-2, *Health informatics — General purpose information components — Part 2: Non-clinical*

[6] EN 14822-3, *Health informatics — General purpose information components — Part 3: Clinical*

[7] CEN/TS 14796, *Health Informatics — Data types*

[8] ISO 13606-1, *Health informatics — Electronic health record communication — Part 1: Reference model*

[9] ISO 13606-4, *Health informatics — Electronic health record communication — Part 4: Security*

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

[11] ISO/IEC 10746-2, *Information technology — Open Distributed Processing — Reference Model: Foundations*

[12] ISO/IEC 10746-3, *Information technology — Open Distributed Processing — Reference Model: Architecture*

[13] ISO/IEC 10746-4, *Information technology — Open Distributed Processing — Reference model: Architectural semantics — Part 4*

[14] ISO/IEC 15414, *Information technology — Open distributed processing — Reference model: Enterprise Language*

[15] ISO/IEC FCD 19793, *Information technology — Open Distributed Processing — Use of UML for ODP system specifications*

[16] ISO 9000, *Quality management systems — Fundamentals and vocabulary*

ICS 35.240.80

Price based on 27 pages