

First edition
2006-08-01

Corrected version
2012-10-15

**Health informatics — HL7 version 3 —
Reference information model —
Release 1**

*Informatique de santé — HL7 version 3 — Modèle d'information de
référence — Version 1*



Reference number
ISO/HL7 21731:2006(E)

AVA Documents
Document Sharing Hub
© ISO/HL7 2006

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 and Health Level Seven International 2006

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the Internet or an Intranet, without prior written permission. Permission can be requested from either ISO or HL7 at the respective address below:

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

Health Level Seven International.
Standards Publishing Department
3300 Washtenaw Avenue, Suite 227 • Ann Arbor, MI 48104 • USA
Tel. +1 734 677 77 77
Fax +1 734 677 66 22
E-mail hq@HL7.org
Web www.HL7.org

Reproduction or use may be subject to royalty payments or licensing agreements.

The Material provided with this publication is protected by copyright law. The unauthorized reproduction or distribution of the copyright work is illegal and may be punishable by criminal law.

The license to the Material provided with this publication is limited to a personal use and includes the right to download for study purposes only. The modification or the implementation, in whole or in part, of the software is not permitted. Under no circumstances may it be sublicensed, copied, transferred, or placed on a network of any sort without the authorization of the copyright owner.

Membership to HL7 is required to obtain non-exclusive, worldwide, rights to download, copy and share the Material with employees and consultants of your company or organization for internal and study purposes, or to utilize the software for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute Compliant Products. Information about such membership can be obtained by contacting HL7 at the address above.

Violators may be prosecuted.
Published in Switzerland.

HL7, CDA, CCD, and Health Level Seven are registered trademarks of Health Level Seven International Reg. U.S. Pat & TM Office.

HEALTH LEVEL SEVEN REFERENCE INFORMATION MODEL, RELEASE 1

Modeling & Methodology
Co-Chair

Editor

George Beeler, Jr., PhD.
Beeler Consulting LLC

James Case, DVM, PhD.
University of California, Davis

Modeling & Methodology
Co-Chair

Jane Curry
Sierra Systems & CIHI - HL7 Canada

Modeling & Methodology
Co-Chair

Ann Hueber
Pathology Associates Medical Laboratories

Editor

Lloyd Mckenzie
IBM Canada & CIHI - HL7 Canada & Alberta Wellnet

Editor

Gunther Schadow, M.D., PhD.
Regenstrief Institute for Health Care

Modeling & Methodology
Co-Chair

Abdul-Malik Shakir
Shakir Consulting LLC

Health Level Seven and HL7 are registered trademarks of Health Level Seven, Inc.

*Health Level Seven, 3300 Washtenaw Avenue, Suite 227, Ann Arbor, MI 48104-4261. Phone: 734-677-7777
Fax: 734-677-6622, Email: HQ@HL7.org; Internet: <http://www.HL7.org>*

CONTENTS	PAGE
Foreword	vi
0 Introduction	vi
0.1 Uses of a Reference Information Model (RIM) in Health Informatics	vi
0.2 Further information	vii
1 Scope	1
2 Conformance	1
3 Normative references	1
4 Terms and definitions	1
5 Interpretation of the Specification	7
5.1 Specification contents	7
5.2 Understanding the RIM	8
5.3 Graphic Diagrams of the RIM	9
6 Subject areas	15
6.1 FoundationClasses	15
6.1.1 Acts	15
6.1.2 Entities	16
6.1.3 Roles	16
7 Classes	16
7.1 Classes in subject area Acts	16
7.2 Classes in subject area Entities	61
7.3 Classes in subject area Roles	77
8 Associations	85
8.1 (0..*)ActRelationship :: source :: (1..1)Act :: outboundRelationship	85
8.2 (0..*)ActRelationship :: target :: (1..1)Act :: inboundRelationship	85
8.3 (1..1)Entity :: languageCommunication :: (0..*)LanguageCommunication :: entity	85
8.4 (0..*)Participation :: act :: (1..1)Act :: participation	85
8.5 (0..*)Participation :: role :: (1..1)Role :: participation	85
8.6 (0..*)Role :: player :: (0..1)Entity :: playedRole	85
8.7 (0..*)Role :: scoper :: (0..1)Entity :: scopedRole	85
8.8 (0..*)RoleLink :: source :: (1..1)Role :: outboundLink	85
8.9 (0..*)RoleLink :: target :: (1..1)Role :: inboundLink	85
9 Normative Vocabulary Contents	86
9.1 RIM Structural Vocabulary	86
9.2 ActClass	87
9.3 ActMood	97
9.4 ActRelationshipCheckpoint	99
9.5 ActRelationshipJoin	99
9.6 ActRelationshipSplit	100
9.7 ActRelationshipType	100
9.8 ActStatus	107

9.9	ContextControl	108
9.10	EntityClass	110
9.11	EntityDeterminer	112
9.12	EntityStatus	113
9.13	ManagedParticipationStatus	113
9.14	ParticipationType	113
9.15	RelationshipConjunction	118
9.16	RoleClass	118
9.17	RoleLinkType	125
9.18	RoleStatus	125
Annex A	(informative) Background of RIM Development	126
A.1	History of the RIM	126
A.2	Implications of Balloting the RIM	126
Annex B	(informative) Overview of RIM Design Principles	128
B.1	Purpose	128
B.2	Overview	128
B.3	The Rationale Behind the RIM's Design	128
B.4	Linking Acts Together: The Semantics of ActRelationship	130
B.5	Definitions of the Six Core Rim Classes	131
B.6	Data Type and Vocabulary Specifications	132
B.7	HL7 Version 3 Methodology and the RIM	133
Annex C	(informative) Summary of Version 3 Data Types	134
C.1	Overview of Data Types	134

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 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.

A pilot project between ISO and Health Level Seven Inc. (HL7) has been formed to develop and maintain a group of ISO/HL7 standards in the field of medical devices as approved by Council resolution 7/2002. Under this pilot project, HL7 is responsible for the development and maintenance of these standards with participation and input from ISO member bodies.

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/HL7 21731 was prepared by HL7 and Technical Committee ISO/TC 215, *Health informatics*.

This corrected version incorporates important new copyright and licensing agreement information on the cover page of this document.

0 Introduction

This introduction is confined to discussion of the requirement for a Reference Information Model in standardization. Further information on the development of this model and the rationale for advancing it as a standard can be found in Annex A.

0.1 Uses of a Reference Information Model (RIM) in Health Informatics

0.1.1 Use of the RIM in ISO TC215

ISO TC215 – Health Informatics has previously advanced ISO 17113, a specification for a framework for developing health data interchange standards. This framework specification calls for messaging standards to be based on a single, comprehensive model of health information. The RIM presented in the current specification provides one such model. Further, the RIM may provide a reference document that can facilitate the harmonization of the health informatics standards and related specifications within ISO TC 215.

0.1.2 Use of the RIM by HL7

The HL7 RIM is a critical component of the V3 development process. It is the root of all information models and structures developed as part of the V3 development process.

The HL7 V3 standard development process is a model-driven methodology in which a network of inter-related models are developed that depict the static and behavioral aspects of the requirements and design of HL7 standards, as well as the underlying semantics and business rules that govern them.

0 INTRODUCTION

The RIM provides a static view of the information needs of HL7 V3 standards. It includes class and state-machine diagrams and is accompanied by use case models, interaction models, data type models, terminology models, and other types of models to provide a complete view of the requirements and design of HL7 standards. The classes, attributes, state-machines, and relationships in the RIM are used to derive domain-specific information models that are then transformed through a series of constraining refinement processes to eventually yield a static model of the information content of an HL7 standard.

The HL7 V3 standard development process defines the rules governing the derivation of domain information models from the RIM and the refinement of those models into HL7 standard specifications. The rules require that all information structures in derived models be traceable back to the RIM and that their semantic and related business rules not conflict with those specified in the RIM. The RIM therefore is the ultimate source for all information content in HL7 V3 standards.

The RIM is used by HL7 international affiliates to extend HL7 V3 standards to meet local needs. Through a process known as localization, V3 standard specifications are extended using the RIM as the source for new information content. This new information is derived from the RIM and refined in the same manner used to create the original specification.

0.1.3 Uses of the RIM Outside of HL7

The RIM is primarily for use by HL7 and its international affiliates. However, others outside of HL7 have also found the RIM useful. Although HL7 maintains a copyright on the expression of this standard, HL7 does not seek to license or otherwise control the use of information structures or programs that implement this specification. Early adopters of the V3 standards development process have used the RIM to develop HL7-like message specifications in their own environments. These early adopters include vendors, large integrated delivery networks, and government agencies within the United States and internationally. These same early adopters are extremely active in HL7 and provide practical input to the RIM and other aspects of V3 the development process.

Some HL7 member organizations have reported using the RIM as a source of input to their enterprise information architectures or as a starting place for systems analysis and design. The RIM may indeed be useful for such purposes; however, HL7 provides no assurance that the RIM is useful for anything other than as a reference model for HL7 standards development.

The RIM is only one model of healthcare information needs. The abstract style of the RIM and the ability to extend the RIM through vocabulary specifications make the RIM applicable to any conceivable healthcare system information interchange scenario. In fact, it is conceptually applicable to any information domain involving entities playing roles and participating in acts.

The universal applicability of the RIM makes it particularly useful for an organization like HL7 that has to consider the needs of a large and diverse membership. The style of the RIM makes it extremely stable, which is another important characteristic for HL7. The HL7 standards development process calls for the creation of domain specific models derived from the RIM and the incremental refinement of those models into design models that are specific to the problem area. These problem area specific design models narrow the abstractness of the RIM and include constraints on attribute values and class relationships that are use case specific. External organizations considering using the HL7 RIM are advised to adopt a similar process of deriving design models as a transformation of the RIM.

0.2 Further information

Questions or comments about the content of the standard may be addressed to HL7 at (www.hl7.org), to one of the HL7 International Affiliate organizations, or to the Secretariat of ISO TC215 – Health Informatics.

Health informatics — HL7 version 3 — Reference information model — Release 1

1 Scope

The Health Level Seven (HL7) Reference Information Model (RIM) is a static model of health and health care information as viewed within the scope of HL7 standards development activities. It is the combined consensus view of information from the perspective of the HL7 working group and the HL7 international affiliates. The RIM is the ultimate source from which all HL7 version 3.0 protocol specification standards draw their information-related content. In the context of ISO TC215 – Health Informatics, the RIM provides a reference model that may be used in developing further health informatics specifications.

2 Conformance

An information model such as the RIM specified in this standard may serve as the basis for other information models that are directly derived from it, and may provide a foundation to support the design of data bases and other information structures. Nevertheless, neither ISO TC 215 nor HL7 believe that it is reasonable to define tests of whether a particular implementation may conform to this standard. Therefore users of this standard shall not claim conformance to this standard. Further, ISO TC215 and HL7, as developers of this standard request that users inform them of particular requirements which caused the users to deviate from this standard or to extend it. This will allow subsequent releases of the standard to meet a broader range of requirements.

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 17113, *Health informatics — Exchange of information between healthcare information systems — Method for development of messages*

ISO/IEC 19501, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

ANSI/HL7 V3 DT, R1-2004, HL7 Version 3 Standard: Data Types — Abstract Specification, Release 1

4 Terms and definitions

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

4.1

ANSI

American National Standards Institute

4.2**association**

A reference from one [class](#) to another class or to itself, or a connection between two [objects](#) (instances of classes).

4.3**association role name**

A name for each end of an [association](#). The name is a short verb phrase depicting the [role](#) of the [class](#) at the opposite end of the association from the perspective of the class adjacent to the role.

4.4**attribute**

An abstraction of a particular aspect of a [class](#). Attributes become the data values that are passed in HL7 messages.

4.5**bag**

A form of [collection](#) whose members are unordered, and need not be unique.

4.6**cardinality**

Property of a data element (the number of times a data element may repeat within an individual occurrence of an object view).

4.7**class**

An abstraction of a thing or concept in a particular application [domain](#).

4.8**classifier attribute**

An [attribute](#) used in [generalization hierarchies](#) to indicate which of the [specializations](#) is the focus of the [class](#).

4.9**coded attribute**

An [attribute](#) in the [Reference Information Model](#) (RIM) with a base [data type](#) of CD, CE, CS, or CV.

4.10**coding strength**

An [extensibility qualifier](#) that specifies whether or not a code set can be expanded to meet local implementation needs.

4.11**coding system**

A scheme for representing concepts using (usually) short concept identifiers to denote the concepts that are members of the system; defines a set of unique concept codes. Examples of coding systems are ICD-9, LOINC and SNOMED.

4.12**collection**

An aggregation of similar [objects](#). The forms of collection used by HL7 are [set](#), [bag](#), and [list](#). Objects which may be found in collections include [data types](#).

4.13**connection**

In an [information model](#), a specified relationship between two [classes](#).

4.14**data type**

The structural format of the data carried in an [attribute](#). It may constrain the set of values an attribute may assume.

4.15**distal class**

From the perspective of a [class](#) in an [information model](#), it is the class at the opposite end of an [association](#) between the two.

4.16**domain**

1. A particular area of interest. For example, the domain for HL7 is healthcare.
2. The set of possible values of a [data type](#), [attribute](#), or data type component. See also [vocabulary domain](#).

4.17**event**

1. A stimulus that causes a noteworthy change in the [state](#) of an [object](#), or a signal that invokes the behavior of an object.
2. A [vocabulary domain](#) value for Mood.

4.18**extensibility qualifier**

A [vocabulary domain qualifier](#) used in a domain specification, which indicates whether or not the existing [vocabulary domain](#) can be extended with additional values. There are two possible values: CNE (coded, no extension) and CWE (coded with extension).

4.19**generalization**

An [association](#) between two [classes](#), referred to as [superclass](#) and [subclass](#), in which the subclass is derived from the superclass. The subclass inherits all properties from the superclass, including [attributes](#), relationships, and [states](#), but also adds new ones to extend the capabilities of the parent class. Essentially, a [specialization](#) from the point-of-view of the subclass.

4.20**generalization hierarchy**

All [superclasses](#) and [subclasses](#) with a common root superclass.

4.21**graphical expression**

A visual representation of a [model](#) that uses graphic symbols to represent the components of the model and the relationships that exist between those components.

4.22**Health Level Seven****HL7**

A standards developing organization based in the United States of America.

4.23**identifier attribute**

An [attribute](#) used to identify an [instance](#) of a [class](#).

4.24**information model**

A structured specification, expressed graphically and/or narratively, of the information requirements of a [domain](#). An information model describes the [classes](#) of information required and the properties of those classes, including [attributes](#), relationships, and [states](#). Examples in HL7 are the Domain Reference Information Model, [Reference Information Model](#), and Refined Message Information Model.

4.25**inheritance**

In a [generalization](#) relationship, the [subclass](#) inherits all properties from the [superclass](#), including [attributes](#), relationships, and [states](#), unless otherwise specified.

4.26**instance**

A case or an occurrence. For example, an instance of a [class](#) is an [object](#).

4.27**list**

A form of [collection](#) whose members are ordered, and need not be unique.

4.28**literary expression**

A representation of a [model](#) in text. The literary expression seeks to balance the need for a rigorous, unambiguous description of the model with the need for a rendition that can be easily read and interpreted by individuals who understand the general concepts underlying object-oriented models, but who may not be schooled in formal model definition languages.

4.29**mandatory**

If an [attribute](#) is designated as mandatory, all message elements which make use of this attribute must contain a non-null value or they must have a default that is not null.

4.30**mandatory association**

An [association](#) with a [multiplicity](#) minimum greater than zero on one end. A fully mandatory association is one with a multiplicity minimum greater than zero on both ends.

4.31**methodology**

Methods or rules followed in a particular discipline.

4.32**model**

A representation of a [domain](#) that uses abstraction to express the relevant concepts.

4.33**multiplicity**

In the [information model](#), multiplicity is a specification of the minimum and maximum number of [objects](#) from each [class](#) that can participate in an [association](#). Multiplicity is specified for each end of the association.

4.34**namespace**

A namespace is a part of the model in which names are defined and used, where each name has a unique meaning.

4.35**object**

An [instance](#) of a [class](#). A part of an information system containing a collection of related data (in the form of [attributes](#)) and procedures (methods) for operating on that data.

4.36**object identity**

The feature that the existence of an [object](#) is independent of any values associated with the object.

4.37**object-based**

Any method, language, or system that supports [object identity](#), classification, and encapsulation. An object-based system does not support [specialization](#). Ada is an example of an object-based implementation language.

4.38**property**

Any [attribute](#), [association](#), method, or [state](#) model defined for a [class](#) or [object](#).

4.39**Reference Information Model****RIM**

The HL7 [information model](#) from which all other information models (e.g., R-MIMs) and messages are derived.

4.40**role**

1. A function or position.
2. A [Reference Information Model class](#) that defines the competency of an Entity class. Each [role](#) is played by one Entity (the Entity that is in the role) and is usually scoped by another.
3. In UML, each end of an [association](#) is designated as a role to reflect the function that class plays in the association.

4.41**role name**

See [association role name](#).

4.42**set**

A form of [collection](#) which contains an unordered [list](#) of unique elements of a single type.

4.43**specialization**

An [association](#) between two [classes](#) (designated [superclass](#) and [subclass](#)), in which the subclass is derived from the superclass. The subclass inherits all properties from the superclass, including [attributes](#), relationships, and [states](#), but also adds new ones to extend the capabilities of the superclass.

4.44**state**

A named condition of a [class instance](#) ([object](#)) that can be tested by examination of the instance's [attributes](#) and [associations](#).

4.45**state attribute**

An [attribute](#) describing the current [state](#) of an [object](#).

4.46**state diagram**

A graphical representation of a [state transition model](#) showing [states](#) as vertices (nodes) and [state transitions](#) as directed arcs (arrows) between the nodes.

4.47**state machine**

A description of the life cycle for [instances](#) of a [class](#), defined by a [state transition model](#).

4.48**state transition**

A change in the [state](#) of an [object](#), as a result of a change in its [attributes](#) or [associations](#).

4.49**state transition model**

A graphical representation of the life cycle of a [class](#). The [model](#) depicts all of the relevant [states](#) of a class, and the valid transitions from state to state.

4.50**subclass**

A [class](#) that is the [specialization](#) of another class ([superclass](#)).

4.51**subject area**

A convenient aggregation of [model classes](#) used to partition large models into manageable subsets.

4.52**sub-state**

An identifiable [state](#) of a [class](#) that has a more specific definition than, and is entirely encompassed within the scope of, its [super-state](#).

4.53**superclass**

A [class](#) that is the [generalization](#) of one or more other classes ([subclasses](#)).

4.54**super-state**

A [state](#) of a [class](#) that encompasses two or more independent [sub-states](#).

4.55**Unified Modeling Language (UML)**

A language for the creation of [domain models](#). UML was created in order to unify several well-known object-oriented modeling [methodologies](#), including those of Booch, Rumbaugh, Jacobson, and others.

4.56**vocabulary**

The set of valid values for a [coded attribute](#) or field.

4.57**vocabulary domain**

The set of all concepts that can be taken as valid values in an [instance](#) of a [coded attribute](#) or field; a constraint applicable to code values.

4.58**vocabulary domain qualifier**

Part of a vocabulary domain specification. The two existing qualifiers are [extensibility](#) and realm.

4.59**W3C**

The World Wide Web Consortium, an international industry consortium

5 Interpretation of the Specification

5.1 Specification contents

The RIM consists of classes assigned to one or more subject area packages. Attributes, Relationships, and State Machines are associated with classes.

Each class within the RIM represents information about a concept that must be documented and communicated within the health care environment. The names that are assigned to these classes are drawn from normal language, but the use of these names is necessarily constrained to the "namespace" of the RIM. The meaning of these classes is entirely embodied in the definition of the class, and the definitions of the properties (attributes and associations) assigned to that class. Thus, for example, the meaning of the "Role" class can only be understood by studying the definition provided and the properties assigned. Definitions from another context or dictionary definitions for the name are not relevant within the context of the RIM namespace.

The RIM is expressed using the Unified Modeling Language (UML) with HL7 specific tags as extensions to the UML model element metadata. All standard UML model element metadata values are normative but only the following HL7 extensions are also normative:

- Class.stateAttribute
- Class.classCode

- Attribute.mandatoryInclusion
- Attribute.cardinality
- Attribute.vocabDomain
- Attribute.vocabStrength

5.2 Understanding the RIM

The RIM uses a very abstract modeling style. The back-bone classes of the RIM and their structural attributes are its core. An understanding of these classes and attributes is essential to understanding the RIM. This section describes how the abstractions are represented in UML and controlled through the application of controlling vocabulary that is part of this specification. An "executive overview" or high-level tutorial that provides examples of how these abstractions can be used to represent more detailed health information is contained in Annex B.

5.2.1 RIM as an abstract model

The RIM is comprised of six "back-bone" classes:

- **Act** which represents the actions that are executed and must be documented as health care is managed and provided;
- **Participation** which expresses the context for an act in terms such as who performed it, for whom it was done, where it was done, etc.;
- **Entity** which represents the physical things and beings that are of interest to, and take part in health care;
- **Role** which establishes the roles that entities play as they participate in health care acts;
- **ActRelationship** which represents the binding of one act to another, such as the relationship between an order for an observation and the observation event as it occurs; and
- **RoleLink** which represents relationships between individual roles.

Three of these classes -- Act, Entity and Role -- are further represented by a set of specialized classes, or sub-types. In the HL7 representation, a sub-type is only added to the RIM if it requires one or more attributes or associations that are not inherited from its parents. Classes that represent distinct concepts, but which need no further attributes or associations are represented solely as a unique code in the controlling vocabulary. Therefore, these three classes include the following coded attributes, which serve to further define the concept being modeled:

- **classCode** (in Act, Entity and Role) represents the exact class or concept intended, whether or not that class is represented as a class in the RIM hierarchy;
- **moodCode** (in Act) and **determinerCode** (in Entity) an attribute that distinguishes whether the class represents an instance or a kind of Act or Entity. If the class is a specialization of Act then moodCode further delineates the instance as an occurrence or an intent;
- **code** (in Act, Entity and Role) provides for further classification within a particular classCode value, such as a particular type of observation within the Observation class.

The other three RIM back-bone classes -- Participation, ActRelationship and RoleLink -- are not represented by generalization-specialization hierarchies. Nevertheless, these classes represent a variety of concepts, such as different forms of participation or different kinds of relationships between acts. These distinctions are represented by a **typeCode** attribute that is asserted for each of these classes.

5.2.2 Representation of RIM class structure

As noted previously, the RIM is modeled using a subset of the semantics embodied in UML. The RIM is a set of UML classes, each containing one or more attributes, which are assigned a data type based on an independent specification of data types. The classes are linked either by a set of association relationships, identified by unique role names, or by generalization relationships.

Each of these elements includes a textual definition. The appearance of attributes and associations is controlled by cardinality and related constraints applied to the attributes and to the roles that link the associations to the classes.

5.2.3 Representation of controlling vocabulary

Several of the attributes in the RIM are coded with the CS data type, which means that the value set to be used for those attributes must be drawn from the set of codes that have been defined by HL7. The classCode and typeCode attributes mentioned above are examples of attributes with a CS data type.

All of the code sets for these attributes are included as part of this ballot and are subject to the same voting principles as the classes in the RIM. Each code set is represented as a vocabulary domain -- the set of all concepts that can be taken as valid values in an instance of a coded field or attribute. It is important to note that a vocabulary domain consists of a set of concepts, not a set of words or codes.

5.2.4 Related specifications

As noted above, each attribute in the RIM is assigned a data type. The formal specification for these data types differs depending upon whether this model is being used within the context of HL7 or the context of ISO TC215. For use in HL7, the normative specification for the datatypes is the "HL7 Data Types Abstract Specification ((need current name)). Within the ISO TC215 framework, work-items to specify a relevant data type specification are being advanced as ISO 22??xx??. In either case, an informative summary of the relevant data type properties is included as Annex C.

5.3 Graphic Diagrams of the RIM

The classes in the Normative content of the RIM are represented in the following diagrams:

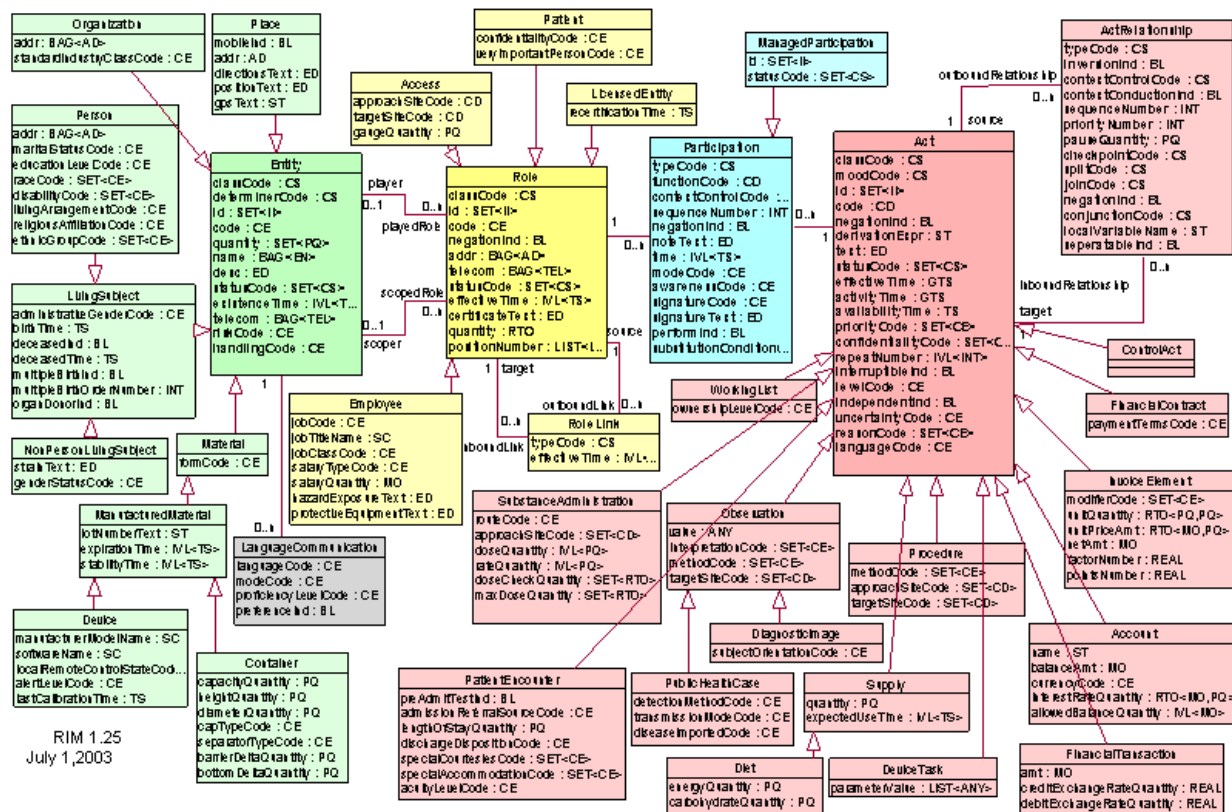


Figure 1: UML diagram of the classes in all subject areas.

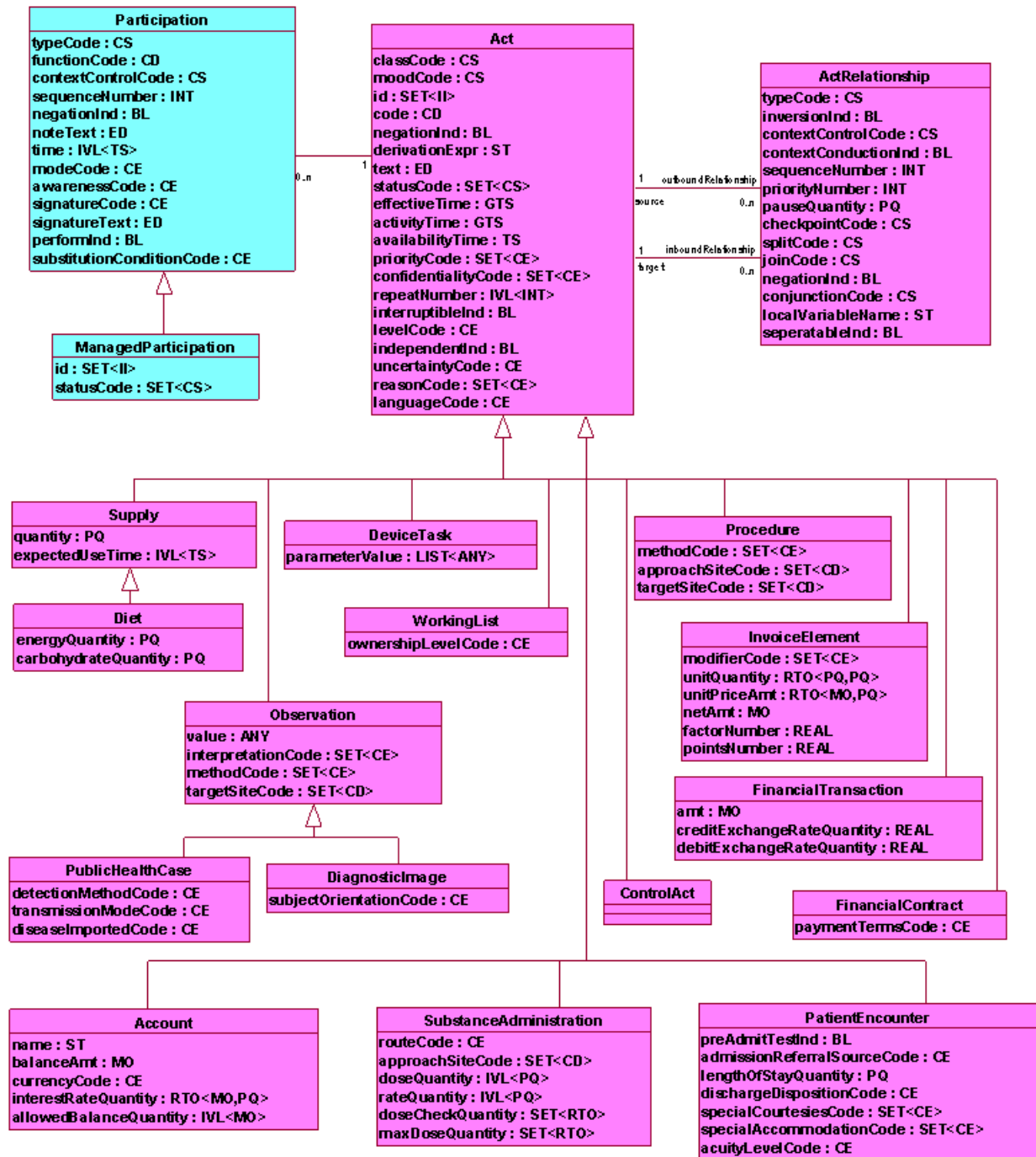


Figure 2: UML diagram of the classes in the Acts subject area.

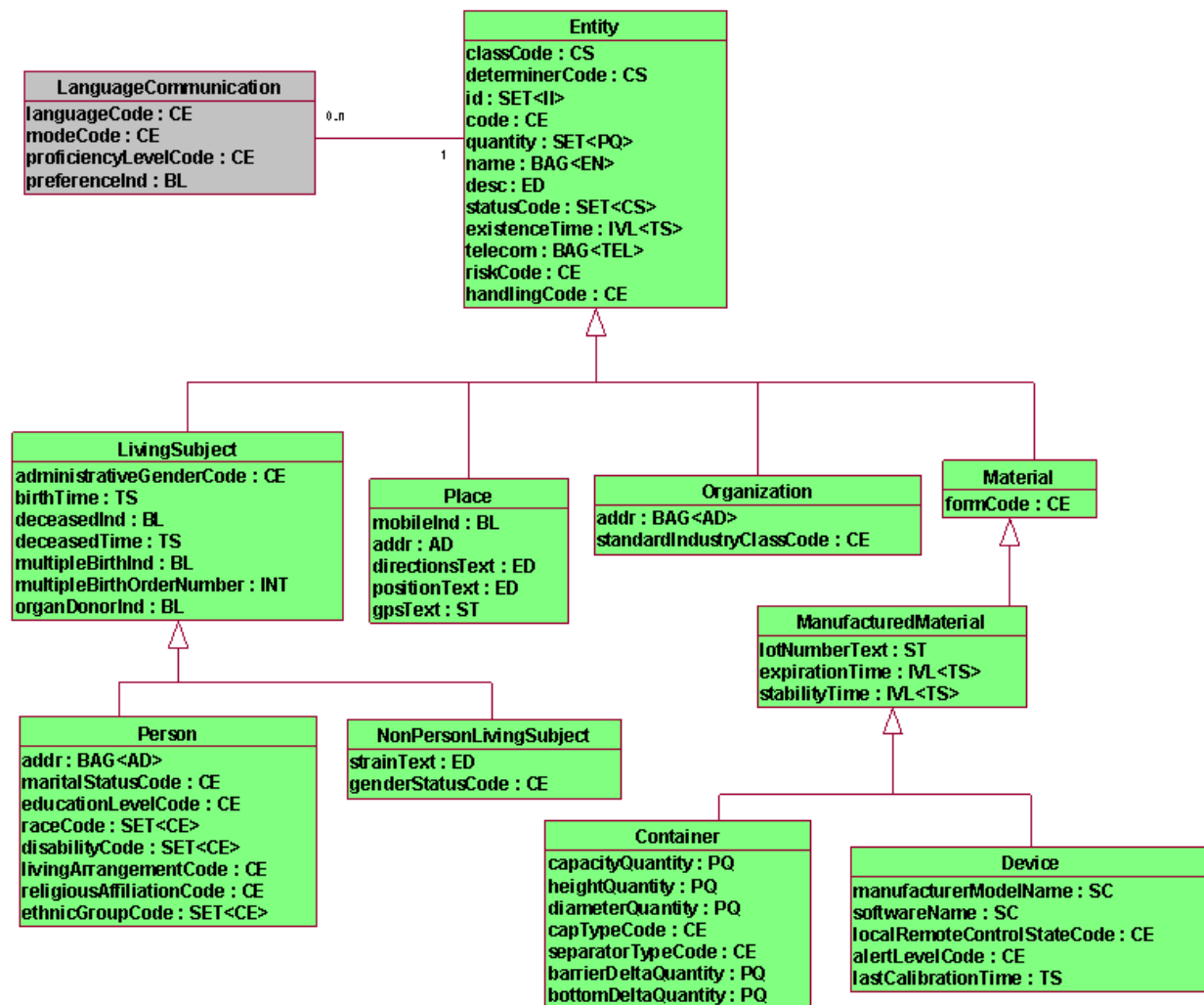


Figure 3: UML diagram of the classes in the Entities subject area.

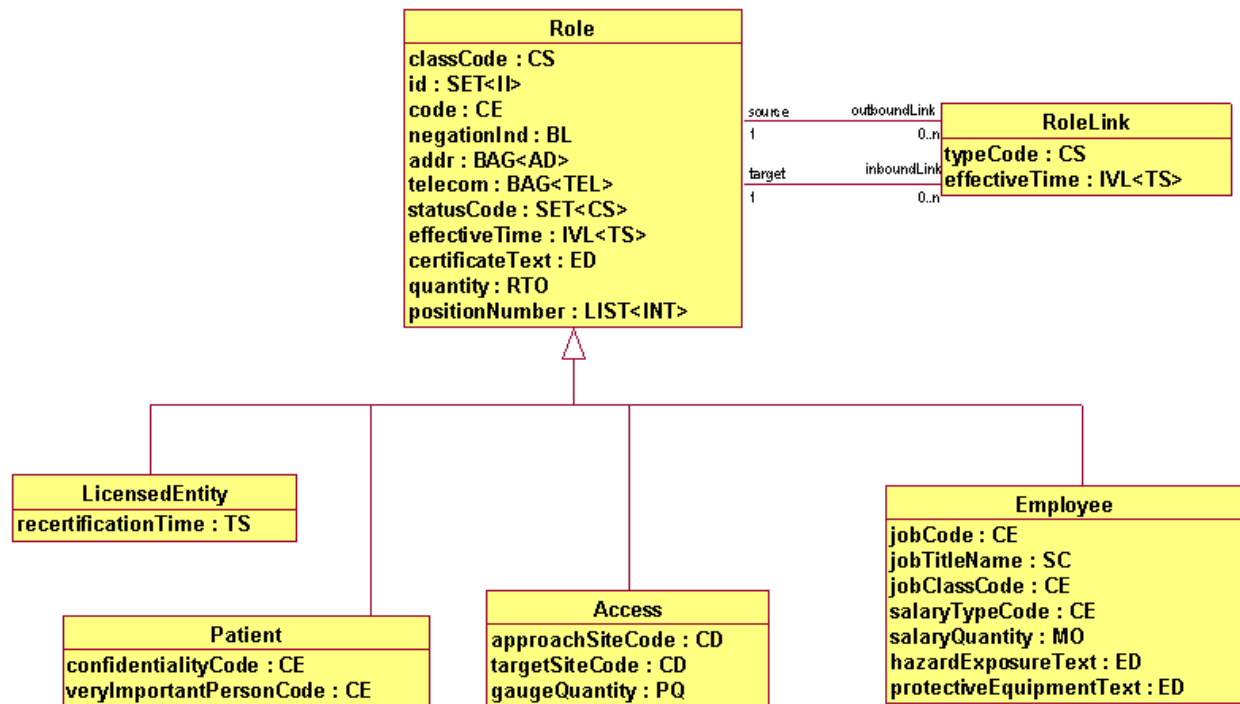


Figure 4: UML diagram of the classes in the Roles subject area.

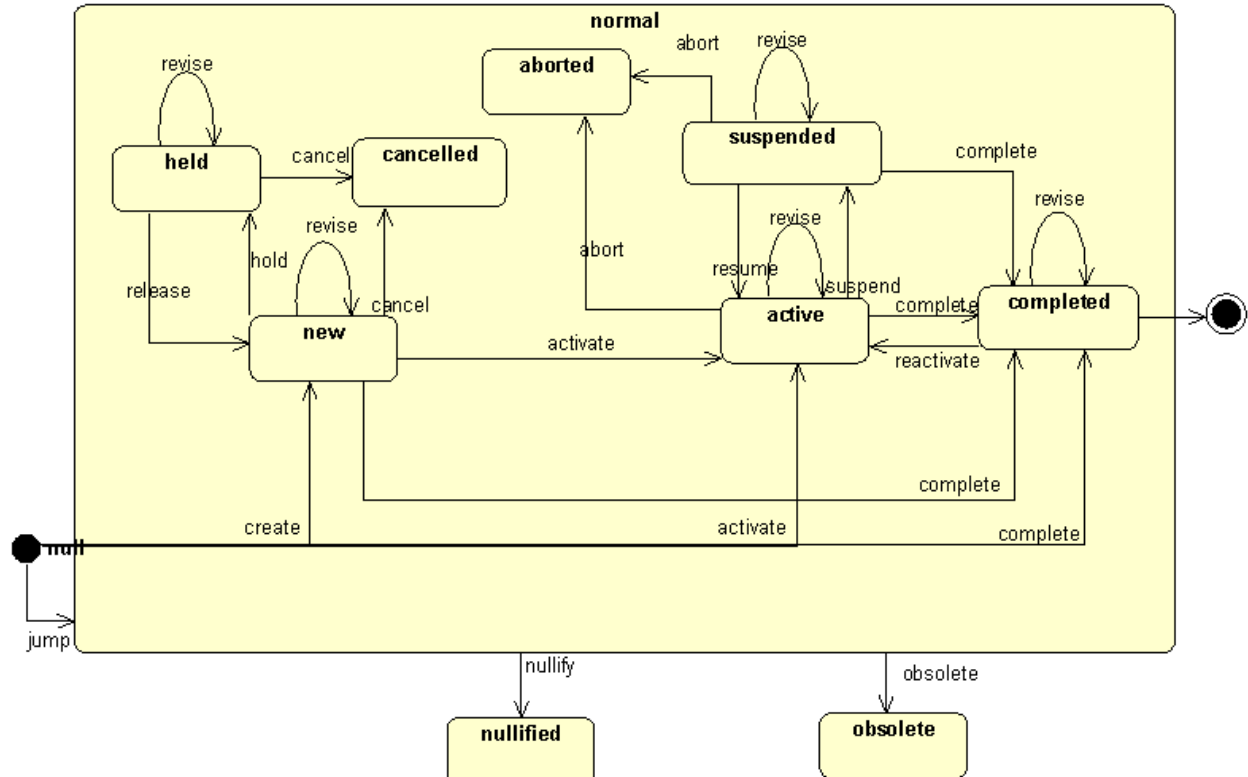


Figure 5: State-machine diagram for Act class.

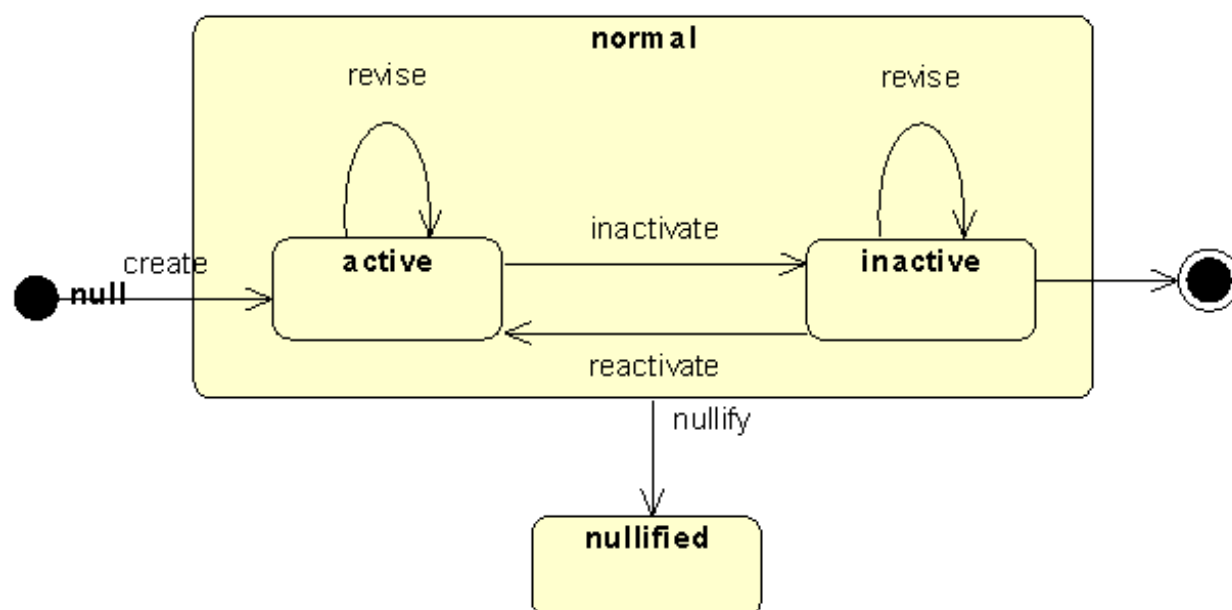


Figure 6: State-machine diagram for Entity class.

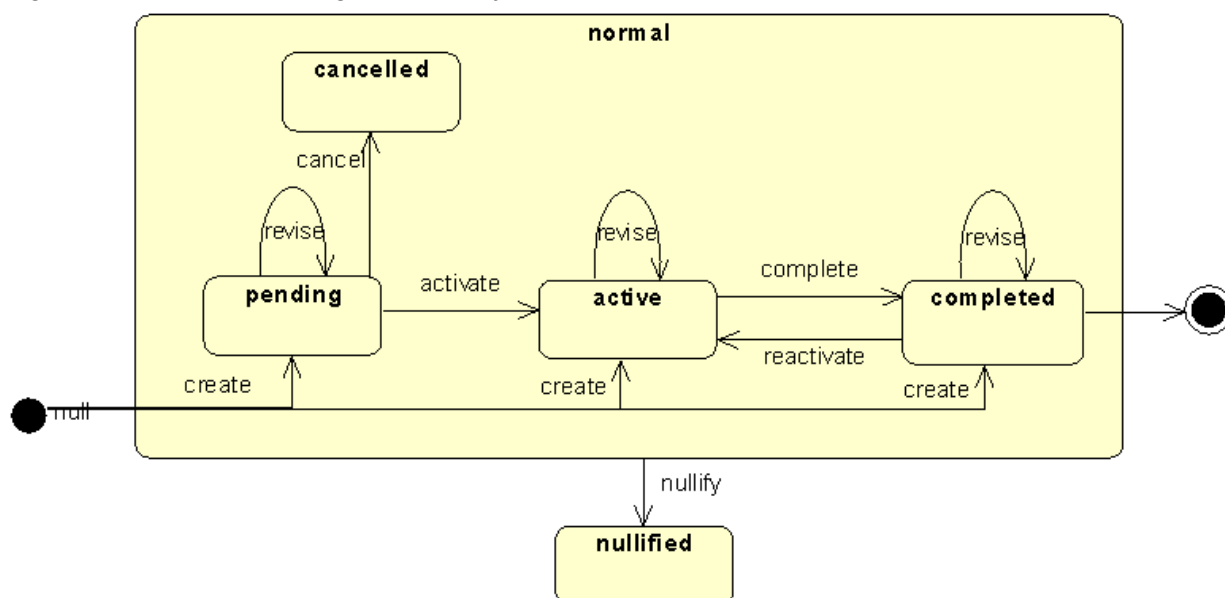


Figure 7: State-machine diagram for ManagedParticipation class.

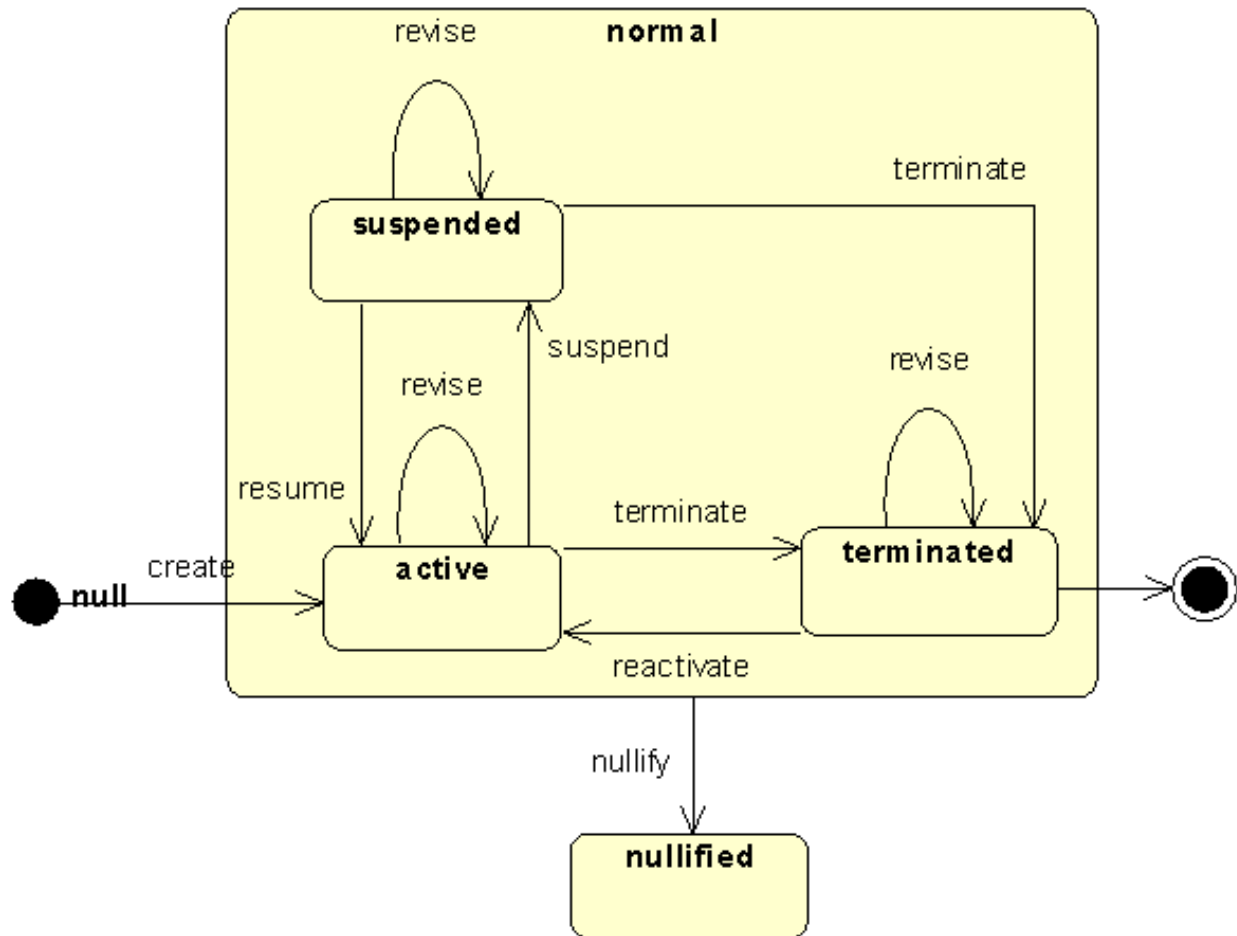


Figure 8: State-machine diagram for Role class.

6 SUBJECT AREAS IN: HEALTH LEVEL SEVEN REFERENCE INFORMATION MODEL, RELEASE 1

FoundationClasses

6.1 FoundationClasses (in [base model](#))

This collection of classes and their associations represent the "normative" content of the HL7 RIM. The content of this subject area has been balloted within HL7 as a normative document.

Link to a [class diagram](#) of this subject area.

- FoundationClasses contains subject areas:

[Acts](#)

[Entities](#)

[Roles](#)

6.1.1 Acts (in [FoundationClasses](#))

A collection of classes including the Act class and its specializations. These relate to the actions and events that constitute health care services.

Link to a [class diagram](#) of this subject area.

- Acts contains classes:

Account	FinancialContract	PatientEncounter
Act	FinancialTransaction	Procedure
ActRelationship	InvoiceElement	PublicHealthCase
DeviceTask	ManagedParticipation	SubstanceAdministration
DiagnosticImage	Observation	Supply
Diet	Participation	WorkingList

6.1.2 Entities (in [FoundationClasses](#))

A collection of classes related to the Entity class, its specializations and related qualifying classes. The classes represent health care stakeholders and other things of interest to health care.

Link to a [class diagram](#) of this subject area.

- Entities contains classes:

Container	LivingSubject	Organization
Device	ManufacturedMaterial	Person
Entity	Material	Place
LanguageCommunication	NonPersonLivingSubject	

6.1.3 Roles (in [FoundationClasses](#))

A collection of classes related to the Role class and its specializations. These classes focus on the roles participants may play in health care.

Link to a [class diagram](#) of this subject area.

- Roles contains classes:

Access	LicensedEntity	Role
Employee	Patient	RoleLink

7 Classes in: [Health Level Seven Reference Information Model, Release 1](#)

Each of the classes is listed below. The sort order is based on the following three criteria:

1. Normative Content first
2. Primary subject area name, alphabetically
3. Class name, alphabetically

7.1 Classes in subject area Acts

7.1.1 Class: Act (in [Acts](#))

- Class code: [ACT](#)

- Attributes of Act:

classCode :: CS	statusCode :: SET<CS>	interruptibleInd :: BL
moodCode :: CS	effectiveTime :: GTS	levelCode :: CE
id :: SET<II>	activityTime :: GTS	independentInd :: BL
code :: CD	availabilityTime :: TS	uncertaintyCode :: CE

negationInd :: BL	priorityCode :: SET<CE>	reasonCode :: SET<CE>
derivationExpr :: ST	confidentialityCode :: SET<CE>	languageCode :: CE
text :: ED	repeatNumber :: IVL<INT>	

- Associations of Act:

outboundRelationship :: (0..*) **ActRelationship** :: **source** :: (1..1)
inboundRelationship :: (0..*) **ActRelationship** :: **target** :: (1..1)
participation :: (0..*) **Participation** :: **act** :: (1..1)

- Act generalizes:

Account	InvoiceElement	SubstanceAdministration
DeviceTask	Observation	Supply
FinancialContract	PatientEncounter	WorkingList
FinancialTransaction	Procedure	

Description of Act:

A record of something that is being done, has been done, can be done, or is intended or requested to be done.

Examples: The kinds of acts that are common in health care are (1) a clinical observation, (2) an assessment of health condition (such as problems and diagnoses), (3) healthcare goals, (4) treatment services (such as medication, surgery, physical and psychological therapy), (5) assisting, monitoring or attending, (6) training and education services to patients and their next of kin, (7) and notary services (such as advanced directives or living will), (8) editing and maintaining documents, and many others.

Discussion and Rationale: Acts are the pivot of the RIM; all domain information and processes are represented primarily in Acts. Any profession or business, including healthcare, is primarily constituted of intentional actions, performed and recorded by responsible actors. An Act-instance is a record of such an intentional action. Intentional actions are distinguished from something that happens by forces of nature (natural events). Such natural events are not Acts by themselves, but may be recorded as observed (Observation).

Acts connect to Entities in their Roles through Participations and connect to other Acts through ActRelationships. Participations are the authors, performers and other responsible parties as well as subjects and beneficiaries (which includes tools and material used in the performance of the act, which are also subjects). The moodCode distinguishes between Acts that are meant as factual records, vs. records of intended or ordered services, and the other modalities in which act can appear.

One of the Participations that all acts have (at least implicitly) is a primary author, who is responsible of the Act and who "owns" the act. Responsibility for the act means responsibility for what is being stated in the Act and as what it is stated. Ownership of the act is assumed in the sense of who may operationally modify the same act. Ownership and responsibility of the Act is not the same as ownership or responsibility of what the Act-object refers to in the real world. The same real world activity can be described by two people, each being the author of their Act, describing the same real world activity. Yet one can be a witness while the other can be a principal performer. The performer has responsibilities for the physical actions; the witness only has responsibility for making a true statement to the best of his or her ability. The two Act-instances may even disagree, but because each is properly attributed to its author, such disagreements can exist side by side and left to arbitration by a recipient of these Act-instances.

In this sense, an Act-instance represents a "statement" according to Rector and Nowlan (1991) [Foundations for an electronic medical record. Methods Inf Med. 30.] Rector and Nowlan have emphasized the importance of understanding the medical record not as a collection of facts, but "a faithful record of what clinicians have heard, seen, thought, and done." Rector and Nowlan go on saying that "the other requirements for a medical record, e.g., that it be attributable and permanent, follow naturally from this view." Indeed the Act class is this attributable statement, and the rules of updating acts (discussed in the state-transition model, see Act.statusCode) versus generating new Act-instances are designed according to this principle of permanent attributable statements.

Rector and Nolan focus on the electronic medical record as a collection of statements, while attributed statements, these are still mostly factual statements. However, the Act class goes beyond this limitation to attributed factual statements, representing what is known as "speech-acts" in linguistics and philosophy. The notion of speech-act includes that there is pragmatic meaning in language utterances, aside from just factual statements; and that these utterances interact with the real world to change the state of affairs, even directly cause physical activities to happen. For example, an order is a speech act that (provided it is issued adequately) will cause the ordered action to be physically performed. The speech act theory has culminated in the seminal work by Austin (1962) [How to do things with words. Oxford University Press].

An activity in the real world may progress from defined, through planned and ordered to executed, which is represented as the mood of the Act. Even though one might think of a single activity as progressing from planned to executed, this progression is reflected by multiple Act-instances, each having one and only one mood that will not change along the Act-instance's life cycle. This is because the attribution and content of speech acts along this progression of an activity may be different, and it is often critical that a permanent and faithful record be maintained of this progression. The specification of orders or promises or plans must not be overwritten by the specification of what was actually done, so as to allow comparing actions with their earlier specifications. Act-instances that describe this progression of the same real world activity are linked through the ActRelationships (of the relationship category "sequel").

Act as statements or speech-acts are the only representation of real world facts or processes in the HL7 RIM. The truth about the real world is constructed through a combination (and arbitration) of such attributed statements only, and there is no class in the RIM whose objects represent "objective state of affairs" or "real processes" independent from attributed statements. As such, there is no distinction between an activity and its documentation. Every Act includes both to varying degrees. For example, a factual statement made about recent (but past) activities, authored (and signed) by the performer of such activities, is commonly known as a procedure report or original documentations (e.g., surgical procedure report, clinic note etc.). Conversely, a status update on an activity that is presently in progress, authored by the performer (or a close observer) is considered to capture that activity (and is later superceded by a full procedure report). However, both status update and procedure report are acts of the same kind, only distinguished by mood and state (see statusCode) and completeness of the information.

Attributes of Act:

7.1.1.1 Act.classCode :: CS (1..1) Mandatory

Vocabulary domain: ActClass (CNE)

Attribute description:

A code specifying the major type of Act that this Act-instance represents.

Constraints: The classCode domain is a tightly controlled vocabulary, not an external or user-defined vocabulary.

Every Act-instance must have a classCode. If the act class is not further specified, the most general Act.classCode (ACT) is used.

The Act.classCode must be a generalization of the specific Act concept (e.g., as expressed in Act.code), in other words, the Act concepts conveyed in an Act must be specializations of the Act.classCode. Especially, the classCode is not a "modifier" or the Act.code that can alter the meaning of a class code. (See Act.code for contrast.)

7.1.1.2 Act.moodCode :: CS (1..1) Mandatory

Vocabulary domain: [ActMood](#) (CNE)

Attribute description:

A code distinguishing whether an Act is conceived of as a factual statement or in some other manner as a command, possibility, goal, etc.

Constraints: An Act-instance must have one and only one moodCode value.

The moodCode of a single Act-instance never changes. Mood is not state.

To describe the progression of a business activity from defined to planned to executed, etc. one must instantiate different Act-instances in the different moods and link them using ActRelationship of general type "sequel". (See ActRelationship.typeCode.)

Discussion: The Act.moodCode includes the following notions: (1) event, i.e., factual description of an actions that occurred; (2) definition of possible actions and action plans (the master file layer); (3) intent, i.e., an action plan instantiated for a patient as a care plan or order; (4) goal, i.e., an desired outcome attached to patient problems and plans; and (5) criterion, i.e., a predicate used to evaluate a logical expression.

The Act.moodCode modifies the meaning of the Act class in a controlled way, just as in natural language, grammatical form of a verb modify the meaning of a sentence in defined ways. For example, if the mood is factual (event), then the entire act object represents a known fact. If the mood expresses a plan (intent), the entire act object represents the expectation of what should be done. The mood does not change the meaning of individual act properties in peculiar ways.

Since the mood code is a determining factor for the meaning of an entire Act object, the mood must always be known. This means, whenever an act object is instantiated, the mood attribute must be assigned to a valid code, and the mood assignment cannot change throughout the lifetime of an act object.

As the meaning of an act object is factored in the mood code, the mood code affects the interpretation of the entire Act object and with it every property (attributes and associations). Note that the mood code affects the interpretation of the act object, and the meaning of the act object in turn determines the meaning of the attributes. However, the mood code does not arbitrarily change the meaning of individual attributes.

Inert vs. descriptive properties of Acts: Acts have two kinds of act properties, inert and descriptive properties. Inert properties are not affected by the mood, descriptive properties follow the mood of the object. For example, there is an identifier attribute Act.id, which gives a unique identification to an act object. Being a unique identifier for the object is in no way dependent on the mood of the act object. Therefore, the "interpretation" of the Act.id attribute is inert with respect to the act object's mood.

By contrast, most of the Act class' attributes are descriptive for what the Act statement expresses. Descriptive properties of the Act class give answer to the questions who, whom, where, with what, how and when the action is done. The questions who, whom, with what, and where are answered by Participations, while how

and when are answered by descriptive attributes and ActRelationships. The interpretation of a descriptive attribute is aligned to the interpretation of the entire act object, and controlled by the mood.

Examples: To illustrate the effect of mood code, consider a "blood glucose" observation:

The DEFINITION mood specifies the Act of "obtaining blood glucose". Participations describe in general the characteristics of the people who must be involved in the act, and the required objects, e.g., specimen, facility, equipment, etc. involved. The Observation.value specifies the absolute domain (range) of the observation (e.g., 15-500 mg/dl).

In INTENT mood the author of the intent expresses the intent that he or someone else "should obtain blood glucose". The participations are the people actually or supposedly involved in the intended act, especially the author of the intent or any individual assignments for group intents, and the objects actually or supposedly involved in the act (e.g., specimen sent, equipment requirements, etc.). The Observation.value is usually not specified, since the intent is to measure blood glucose, not to measure blood glucose in a specific range. (But compare with GOAL below).

In REQUEST mood, a kind of intent, the author requests to "please obtain blood glucose". The Participations are the people actually and supposedly involved in the act, especially the placer and the designated filler, and the objects actually or supposedly involved in the act (e.g., specimen sent, equipment requirements, etc.). The Observation.value is usually not specified, since the order is not to measure blood glucose in a specific range.

In EVENT mood, the author states that "blood glucose was obtained". Participations are the people actually involved in the act, and the objects actually involved (e.g., specimen, facilities, equipment). The Observation.value is the value actually obtained (e.g., 80 mg/dL, or <15 mg/dL).

In event-CRITERION mood, an author considers a certain class of "obtaining blood glucose" possibly with a certain value (range) as outcome. The Participations constrain the criterion, for instance, to a particular patient. The Observation.value is the range in which the criterion would hold (e.g. > 180 mg/dL or 200-300 mg/dL).

In GOAL mood (a kind of criterion), the author states that "our goal is to be able to obtain blood glucose with the given value (range)". The Participations are similar to intents, especially the author of the goal and the patient for whom the goal is made. The Observation.value is the range which defined when the goal is met (e.g. 80-120 mg/dl).

Rationale: The notion of "mood" is borrowed from natural language grammar, the mood of a verb (lat. modus verbi).

The notion of mood also resembles the various extensions of the logic of facts in modal logic and logic with modalities, where the moodCode specifies the modality (fact, possibility, intention, goal, etc.) under which the Act-statement is judged as appropriate or defective.

7.1.1.3 Act.id :: SET<II> (0..*)

Attribute description:

A unique identifier for the Act.

7.1.1.4 Act.code :: CD (0..1)

Vocabulary domain: ActCode (CWE)

Attribute description:

A code specifying the particular kind of Act that the Act-instance represents within its class.

Constraints: The kind of Act (e.g. physical examination, serum potassium, inpatient encounter, charge financial transaction, etc.) is specified with a code from one of several, typically external, coding systems. The coding system will depend on the class of Act, such as LOINC for observations, etc.

Conceptually, the Act.code must be a specialization of the Act.classCode. This is why the structure of ActClass domain should be reflected in the superstructure of the ActCode domain and then individual codes or externally referenced vocabularies subordinated under these domains that reflect the ActClass structure.

Act.classCode and Act.code are not modifiers of each other but the Act.code concept should really imply the Act.classCode concept. For a negative example, it is not appropriate to use an Act.code "potassium" together with an Act.classCode for "laboratory observation" to somehow mean "potassium laboratory observation" and then use the same Act.code for "potassium" together with Act.classCode for "medication" to mean "substitution of potassium". This mutually modifying use of Act.code and Act.classCode is not permitted.

Discussion: Act.code is not a required attribute of Act. Rather than naming the kind of Act using an Act.code, one can specify the Act using only the class code and other attributes and properties of the Act. In general and more commonly, the kind of Act is readily specified by an ActRelationship specifying that this Act instantiates another Act in definition mood. Or, even without reference to an act definition, the act may be readily described by other attributes, ActRelationships and Participations. For example, the kind of SubstanceAdministration may be readily described by referring to the specific drug, as the Participation of an Entity representing that drug.

7.1.1.5 Act.negationInd :: BL (0..1)**Attribute description:**

An indicator specifying that the Act statement is a negation of the Act as described by the descriptive attributes.

Examples: Used with an Observation event, it allows one to say "patient has NO chest pain". With an Observation criterion it negates the criterion analogously, e.g., "if patient has NO chest pain for 3 days ...", or "if systolic blood pressure is **not** within 90-100 mm Hg ..."

Discussion: The negationInd works as a negative existence quantifier. This is best explained on Acts in criterion mood, and then translates into all other moods. In criterion mood without negation, one usually only specifies a few critical attributes and relationships (features) of an Act, i.e., only those that are needed to test the criterion. The more features one specifies, the more constrained (specific) is the criterion. For example, to test for "systolic blood pressure of 90-100 mm Hg", one would use only the descriptive attributes Act.code (for systolic blood pressure) and Observation.value (for 90-100 mm Hg). If one would also specify an effectiveTime, i.e., for "yesterday", the criterion would be more constrained. If the negationInd is true for the above criterion, then the meaning of the test is whether a systolic blood pressure of 90-100 mm Hg yesterday does **not exist** (independent of whether any blood pressure was measured).

The negationInd negates the Act as described by the descriptive properties (including Act.code, Act.effectiveTime, Observation.value, Act.doseQty, etc.) and any of its components. The inert properties such as Act.id, Act.moodCode, Act.confidentialityCode, and particularly the Author-Participation are **not** negated. These inert properties always have the same meaning: i.e., the author remains to be the author of the negative observation. Also, most ActRelationships (except for components) are not included in the negation.

For example, a highly confidential order written by Dr. Jones, to explicitly **not** give "succinyl choline" for the "reason" (ActRelationship) of a history of malignant hyperthermia (Observation) negates the descriptive properties "give succinyl choline" (Act.code), but it is still positively an order and written by Dr. Jones and for patient John Smith, and the reason for this order is the patient's history of malignant hyperthermia.

However, additional detail in descriptive attributes will be part of the negation which then limits the effectiveness of the negated statement. For example, had the order not to give a substance included a doseQuantity, it would mean that the substance should not be given at that particular dose (but any other dose might still be O.K.).

An act statement with negationInd is still a statement about the specific fact described by the Act. For instance, a negated "finding of wheezing on July 1" means that the author positively denies that there was wheezing on July 1, and that he takes the same responsibility for such statement and the same requirement to have evidence for such statement than if he had not used negation. Conversely, negation indicator does **not** just negate that the fact was affirmed or that the statement had been made. This holds for all moods in the same way, e.g., a negated order is an order **not** to do the described act, not just the lapidary statement that there is no such order.

7.1.1.6 Act.derivationExpr :: ST (0..1)

Attribute description:

A character string containing a formal language expression that specifies how the Act's attributes are derived from input parameters associated with derivation relationships.

Discussion: Derived observations can be defined through association with other observations using ActRelationships of type "derivation". For example, to define a derived observation for Mean Corpuscular Hemoglobin (MCH) one will associate the MCH observation with a Hemoglobin (HGB) observation and a Red Blood cell Count (RBC) observation. And the derivation expression encodes the formula: $MCH = HGB / RBC$.

The derivation expression is represented as a character string.

[Note: The syntax of that expression is yet to be fully specified. There would be a single standard expression language rather than an optional choice between many expression languages. The syntax would be based on a de-facto standard for many object-oriented languages, such as C++, Java, OCL etc. A concrete specification of this expression language is being worked on now and drafts can be expected within the year 2003.]

7.1.1.7 Act.text :: ED (0..1)

Attribute description:

A textual or multimedia description of the Act.

Examples: For act definitions, the Act.text can contain textbook-like information about that act. For act orders, the description will contain particular instructions pertaining only to that order.

Constraints: No restriction on length or content is imposed on the Act.text attribute.

The content of the description is not considered part of the functional information communicated between computer systems.

Free text descriptions are used to help an individual interpret the content and context of the act, but all information relevant for automated functions must be communicated using the proper attributes and associated objects.

For Acts that involve human readers and performers, however, computer systems must show the Act.text field to a human user, who has responsibility for the activity; or at least must indicate the existence of the Act.text information and allow the user to see that information.

7.1.1.8 Act.statusCode :: SET<CS> (0..*)

Vocabulary domain: ActStatus (CNE)

Attribute description:

A code specifying the state of the Act.

7.1.1.9 Act.effectiveTime :: GTS (0..1)

Attribute description:

A time expression specifying the focal or operative time of the Act, the primary time for which the Act holds, the time of interest from the perspective of the Act's intention.

Examples: For clinical Observations, the effectiveTime is the time at which the observation holds (is effective) for the patient.

For contracts, the effectiveTime is the time for which the contract is in effect.

For consents, the effectiveTime is the time for which the consent is valid.

For substance administrations, the effective time is the time over which the substance is to be administered, including the frequency of administration (e.g. TID for 10 days)

For a surgical procedure (operation), the effectiveTime is the time relevant for the patient, i.e., between incision and last suture.

For transportation acts, the effective time is the time the transported payload is en route.

For patient encounters, this is the "administrative" time, i.e., the encounter start and end date required to be chosen by business rules, as opposed to the actual time the healthcare encounter related work is performed.

Discussion: The effectiveTime is also known as the "primary" time (Arden Syntax) or the "biologically relevant time" (HL7 v2.x). This attribute is distinguished from activityTime.

For observations, the time of the observation activity may be much later than the time of the observed feature. For instance, in a Blood Gas Analysis (BGA), a result will always come up several minutes after the specimen was taken, meanwhile the patient's physiological state may have changed significantly.

For essentially physical activities (surgical procedures, transportations, etc.), the effective time is the time of interest for the Act's intention, i.e., since the intention of a transportation is to deliver a payload from location A to B, the effectiveTime is the time this payload is underway from A to B. However, the Act usually also includes accidental work which is necessary to perform the intention of the Act, but is not relevant for the Act's purpose.

For example, the time a driver needs to go to the pick-up location A and then return from drop-off location B to some home base, is included in the physical activity, but does not matter from the perspective of the payload's transportation. Another example is: a person's work hours (effectiveTime) may be from 8 AM to 5 PM, no matter whether that person needs 10 minutes for the commute or 2 hours. The commute is necessary to be at work, but it is not essential for the work hours.

7.1.1.10 Act.activityTime :: GTS (0..1)

Attribute description:

A time expression specifying the time in which the activity referred to by the Act statement happened, or -- depending on the mood -- is ordered to happen, scheduled to happen, when it can possibly happen, etc.

Discussion: See also, effectiveTime. The activityTime is the total time of activity including preparation and clean-up actions that may not be considered relevant to the main action. Thus the activityTime may be longer than the effectiveTime of the same act or it may be entirely different. For example, in retrospective observations, the observation activityTime is much later than the effectiveTime for which the observation is made.

For most use cases, designers should first consider effectiveTime as the primary relevant time for an act.

7.1.1.11 Act.availabilityTime :: TS (0..1)

Attribute description:

The point in time at which information about Act-instance (regardless of mood) first became available to a system reproducing this Act.

Examples: An Act might record that a patient had a right-ventricular myocardial infarction effective three hours ago (see Act.effectiveTime), but we may only know about this unusual condition a few minutes ago (Act.availabilityTime). Thus, any interventions from three hours ago until a few minutes ago may have assumed the more common left-ventricular infarction, which can explain why these interventions (e.g., nitrate administration) may not have been appropriate in light of the more recent knowledge.

Discussion: The availabilityTime is a subjective secondary piece of information added (or changed) by a system that reproduces this Act, and is not attributed to the author of the act statement (it would not be included in the material the author would attest to with a signature). The system reproducing the Act is often not the same as the system originating the Act, but a system that received this Act from somewhere else, and, upon receipt of the Act, values the availabilityTime to convey the time since the users of that particular system could have known about this Act-instance.

When communicating availabilityTime to another system, the availabilityTime of an Act A is attributed to the author of another Act B, that refers to or includes A. For example, if a medical record extract is compiled for reporting adverse events, availabilityTimes are attributed to the author who compiles that report.

7.1.1.12 Act.priorityCode :: SET<CE> (0..*)

Vocabulary domain: ActPriority (CWE)

Attribute description:

A code or set of codes (e.g., for routine, emergency), specifying the urgency under which the Act happened, can happen, is happening, is intended to happen, or is requested/demanded to happen.

Discussion: This attribute is used in orders to indicate the ordered priority, and in event documentation it indicates the actual priority used to perform the act. In definition mood it indicates the available priorities.

7.1.1.13 `Act.confidentialityCode` :: `SET<CE>` (0..*)

Vocabulary domain: Confidentiality (CWE)

Attribute description:

A code that controls the disclosure of information about this Act, regardless of mood.

Discussion: It is important to note that the necessary confidentiality of the medical record cannot be achieved solely through confidentiality codes to mask individual record items from certain types of users. There are two important problems with per-item confidentiality: one is inference and the other is the danger of holding back information that may be critical in a certain care situation. Inference means that filtered sensitive information can still be assumed given the other information that was not filtered. The simplest form of inference is that even the existence of a test order for an HIV Western Blot test or a T4/T8 lymphocyte count is a strong indication for an existing HIV infection, even if the results are not known. Very often, diagnoses can be inferred from medication, such as Zidovudin for treatment of HIV infections. The problem of hiding individual items becomes especially difficult with current medications, since the continuing administration of the medication must be assured.

To mitigate some of the inference-risk, aggregations of data should assume the confidentiality level of the most confidential action in the aggregation.

7.1.1.14 `Act.repeatNumber` :: `IVL<INT>` (0..1)**Attribute description:**

An interval of integer numbers stating the minimal and maximal number of repetitions of the Act.

Examples: An oral surgeon's advice to a patient after tooth extraction might be: "replace the gauze every hour for 1 to 3 times until bleeding has stopped completely." This translates to repeatNumber with low boundary 1 and high boundary 3.

Discussion: This attribute is a member of the workflow control suite of attributes.

The number of repeats is additionally constrained by time. The act will repeat at least the minimal number of times and at most, the maximal number of times. Repetitions will also terminate when the time exceeds the maximal Act.effectiveTime, whichever comes first.

7.1.1.15 `Act.interruptibleInd` :: `BL` (0..1)**Attribute description:**

An indicator specifying whether Act is interruptible by asynchronous events.

Discussion: This attribute is part of the suite of workflow control attributes. Act events that are currently active can be interrupted in various ways. Interrupting events include: (1) when an explicit abort request is received against the Act (2) when the time allotted to this Act expires (timeout); (3) a "through condition" ceases to hold true for this Act (see ActRelationship.checkpointCode); (4) the Act is a component with the joinCode "kill" and all other components in that same group have terminated (see Act.joinCode); and (5) when a containing Act is being interrupted.

If an Act receives an interrupt and the Act itself is interruptible, but it has currently active component-Acts that are non-interruptible, the Act will be interrupted when all of its currently active non-interruptible component-acts have terminated.

7.1.1.16 Act.levelCode :: CE (0..1)

Vocabulary domain: ActContextLevel (CWE)

Attribute description:

Code specifying the level within a hierarchical Act composition structure and the kind of contextual information attached to composite Acts ("containers") and propagated to component Acts within those containers. The levelCode signifies the position within such a containment hierarchy and the applicable constraints.

Discussion: Readers should be aware that **this attribute may be declared "obsolescent"** in the next normative release of the HL7 RIM. An alternate representation of this concept using a specified hierarchy of Act classCode values is being considered. If the change is adopted, HL7's RIM maintenance procedures state that the levelCode would be declared "obsolescent" in the next RIM release, and then become "obsolete" in the release following that. Users are advised to check with the latest HL7 internal definitions of the RIM, before using this attribute.

The levelCode concepts have been defined to meet specific health record transfer requirements. While these concepts are known to be applicable to some other types of transactions, they are not intended to be a complete closed list. Options exist for other sets of orthogonal levels where required to meet a business purpose (e.g. a multiple patient communication may be subdivided by a super-ordinate level of subject areas).

Examples: The "extract level" and the "folder level" must contain data about a single individual, whereas the "multiple subject level" may contain data about multiple individuals. While "extract" can originate from multiple sources, a "folder" should originate from a single source. The "composition" level usually has a single author.

Constraints: The constraints applicable to a particular level may include differing requirements for participations (e.g. patient, source organization, author or other signatory), relationships to or inclusion of other Acts, documents or use of templates. The constraints pertaining to a level may also specify the permissible levels that may be contained as components of that level. Several nested levels with the same levelCode may be permitted, prohibited (or limited). Instances of the next subordinate level are usually permitted within any level but some levels may be omitted from a model and it may be permissible to skip several layers.

7.1.1.17 Act.independentInd :: BL (0..1)**Attribute description:**

An indicator specifying whether the Act can be manipulated independently of other Acts or whether manipulation of the Act can only be through a super-ordinate composite Act that has this Act as a component. By default the independentInd should be true.

Examples: An Act definition is sometimes marked with independentInd=false if the business rules would not allow this act to be ordered without ordering the containing act group.

An order may have a component that cannot be aborted independently of the other components.

7.1.1.18 Act.uncertaintyCode :: CE (0..1)

Vocabulary domain: ActUncertainty (CNE)

Attribute description:

A code indicating whether the Act statement as a whole, with its subordinate components has been asserted to be uncertain in any way.

Examples: Patient might have had a cholecystectomy procedure in the past (but isn't sure).

Constraints: Uncertainty asserted using this attribute applies to the combined meaning of the Act statement established by all descriptive attributes (e.g., Act.code, Act.effectiveTime, Observation.value, SubstanceAdministration.doseQuantity, etc.), and the meanings of any components.

Discussion: This is not intended for use to replace or compete with uncertainty associated with Observation.value alone or other individual attributes of the class. Such pointed indications of uncertainty should be specified by applying the PPD, UVP or UVN data type extensions to the specific attribute. Particularly if the uncertainty is uncertainty of a quantitative measurement value, this must still be represented by a PPD<PQ> in the value and **not** using the uncertaintyCode. Also, when differential diagnoses are enumerated or weighed for probability, the UVP<CD> or UVN<CD> must be used, not the uncertaintyCode. The use of the uncertaintyCode is appropriate only if the entirety of the Act and its dependent Acts is questioned.

Note that very vague uncertainty may be thought related to negationInd, however, the two concepts are really independent. One may be very uncertain about an event, but that does not mean that one is certain about the negation of the event.

7.1.1.19 Act.reasonCode :: SET<CE> (0..*)

Vocabulary domain: ActReason (CWE)

Attribute description:

A code specifying the motivation, cause, or rationale of an Act, when such rationale is not reasonably represented as an ActRelationship of type "has reason" linking to another Act.

Examples: Example reasons that might qualify for being coded in this field might be: "routine requirement", "infectious disease reporting requirement", "on patient request", "required by law".

Discussion

Most reasons for acts can be clearly expressed by linking the new Act to another prior Act using an ActRelationship of type "has reason". This simply states that the prior Act is a reason for the new Act (see ActRelationship). The prior act can then be a specific existing act or a textual explanation. This works for most cases, and the more specific the reason data is, the more should this reason ActRelationship be used instead of the reasonCode.

The reasonCode remains as a place for common reasons that are not related to a prior Act or any other condition expressed in Acts. Indicators that something was required by law or was on the request of a patient etc. may qualify. However, if that piece of legislation, regulation, or the contract or the patient request can be represented as an Act (and they usually can), the reasonCode should not be used.

7.1.1.20 Act.languageCode :: CE (0..1)

Vocabulary domain: HumanLanguage (CWE)

Attribute description:

The primary language in which this Act statement is specified, particularly the language of the Act.text.

7.1.1.21 State machine for *Act*

Link to [state diagram](#) for this class.

States of *Act*:

- **aborted** (sub-state of *normal*): Active service object is exceptionally terminated.
- **active** (sub-state of *normal*): Service object is active.
- **cancelled** (sub-state of *normal*): Service object has been abandoned before activation.
- **completed** (sub-state of *normal*): Service object is completed.
- **held** (sub-state of *normal*): Service object is on hold so it can not be activated before it is released.
- **new** (sub-state of *normal*): Service object is in preparation awaiting to be activated.
- **normal**: Encompasses the expected states of a service object, but excludes "nullified" and "obsolete" which represent unusual terminal states for the life-cycle.
- **nullified**: Service object should not have been instantiated, and is therefore nullified.
- **obsolete**: Service object has been superseded by a new service object.
- **suspended** (sub-state of *normal*): Active service object is temporarily suspended.

State transitions of *Act*:

- **abort** (from *active* to *aborted*)
- **revise** (from *active* to *active*)
- **complete** (from *active* to *completed*)
- **suspend** (from *active* to *suspended*)
- **reactivate** (from *completed* to *active*)
- **revise** (from *completed* to *completed*)
- **cancel** (from *held* to *cancelled*)
- **revise** (from *held* to *held*)
- **release** (from *held* to *new*)
- **activate** (from *new* to *active*)
- **cancel** (from *new* to *cancelled*)
- **complete** (from *new* to *completed*)
- **hold** (from *new* to *held*)
- **revise** (from *new* to *new*)
- **nullify** (from *normal* to *nullified*)
- **obsolete** (from *normal* to *obsolete*)
- **activate** (from *null* to *active*)
- **complete** (from *null* to *completed*)
- **create** (from *null* to *new*)
- **jump** (from *null* to *normal*)
- **abort** (from *suspended* to *aborted*)
- **resume** (from *suspended* to *active*)
- **complete** (from *suspended* to *completed*)

- **revise** (from *suspended* to *suspended*)

7.1.2 Class: ActRelationship (in *Acts*)

- Attributes of ActRelationship:

<code>typeCode :: CS</code>	<code>priorityNumber :: INT</code>	<code>negationInd :: BL</code>
<code>inversionInd :: BL</code>	<code>pauseQuantity :: PQ</code>	<code>conjunctionCode :: CS</code>
<code>contextControlCode :: CS</code>	<code>checkpointCode :: CS</code>	<code>localVariableName :: ST</code>
<code>contextConductionInd :: BL</code>	<code>splitCode :: CS</code>	<code>seperatableInd :: BL</code>
<code>sequenceNumber :: INT</code>	<code>joinCode :: CS</code>	

- Associations of ActRelationship:

target :: (1..1) **Act** :: **inboundRelationship** :: (0..*)
source :: (1..1) **Act** :: **outboundRelationship** :: (0..*)

Description of ActRelationship:

A directed association between a source Act and a target Act. ActRelationship on the same source Act are called the "outbound" act relationships of that Act. ActRelationships on the same target Act are called the "inbound" relationships of that Act. The meaning and purpose of an ActRelationship is specified in the ActRelationship.typeCode attribute.

Examples: 1) An electrolyte observation panel may have sodium, potassium, pH, and bicarbonate observations as components. The composite electrolyte panel would then have 4 outbound ActRelationships of type "has component".

2) The electrolyte panel event has been performed in fulfillment of an observation order. The electrolyte panel event has an outbound ActRelationship of type "fulfills" with the order as target.

3) A Procedure "cholecystectomy" may be performed for the reason of an Observation of "cholelithiasis". The procedure has an outbound ActRelationship of type "has reason" to the cholelithiasis observation.

Discussion: Consider every ActRelationship instance an arrow with a point (headed to the target) and a butt (coming from the source). The functions (sometimes called "roles") that source and target Acts play in that association are defined for each ActRelationship type differently. For instance in a composition relationship, the source is the composite and the target is the component. In a reason-relationship the source is any Act and the target is the reason or indication for the source-Act.

The relationships associated with an Act are considered properties of the source act object. This means that the author of an Act-instance is also considered the author of all of the act relationships that have this Act as their source. There are no exceptions to this rule.

See ActRelationship.typeCode for more overview of the different kinds of ActRelationships.

The ActRelationship class is used to construct action plans and to represent clinical reasoning or judgments about action relationships. Prior actions can be linked as the reasons for more recent actions. Supporting evidence can be linked with current clinical hypotheses. Problem lists and other networks of related judgments about clinical events are represented by the ActRelationship link.

One of the most commonly used ActRelationship types is "has component" to describe the composition and de-composition of Acts. The relationship type allows specifying detail of Acts to varying degrees.

The composition relationship can group actions into "batteries," e.g., LYLES, CHEM12, or CBC, where multiple routine laboratory tests are ordered as a group. Some groupings, such as CHEM12, appear more arbitrary; others, such as blood pressure, seem to naturally consist of systolic and diastolic pressure.

The composition relationships can be arranged in a sequence to form temporal and conditional (non-temporal) action plans (e.g., care plan, critical path, clinical trials protocol, drug treatment protocols). There is a group of attributes in both Act and ActRelationship that we refer to as the "workflow Control suite of attributes", and which allow the detailed specification of executable action plans. These attributes are:

ActRelationship.sequenceNumber arranges the components of an Act as a sequence or as concurrent collections of components, expressing logical branches as well as parallel tasks (tasks carried out at the same time). The ActRelationship attributes splitCode and joinCode control how branches are selected or executable in parallel.

Act.activityTime and ActRelationship.pauseQty allow one to explicitly time an action plan. Act.repeatNumber allows specifying act to repeat (loop).

The ActRelationship type has-precondition allows plan steps to be conditional on the status or outcome of previous actions. The ActRelationship.checkpointCode specifies when pre-conditions of acts are tested during the flow of control.

The composition ActRelationship allows these constructs to be organized in multiple layers of nesting to fully support workflow management. This nesting and the workflow control attributes are designed in analogy to a block-structured programming language with support for concurrency (fork, join, interrupts), and without "goto" statements. It is important to note that ALL plans are established through sequencing components (steps) in a composite act (block) as can be depicted in "Nassi-Schneiderman" diagrams (also known as "Chap Charts" or "Structograms"), not by chain-linking acts as in a flowchart diagram.

With the composition relationship, the detail of Acts can be revealed to different levels for different purposes, without the structure of the Act hierarchy needing to be rearranged. This allows supporting multiple viewpoints on the same business processes. For instance, a billing-viewpoint of a laboratory test battery may be as a single billable act. A clinician's view of the same laboratory test battery is as a set of individual observations, where the ordering among the observations is irrelevant. The laboratory's view of this act will be more detailed, including action plan steps that are never reported to the clinician (e.g., centrifugation, decantation, aliquoting, running certain machines etc.). The laboratory's viewpoint warrants a thorough specification of action plans (that can be automated). During this specification, more and more nested sub-activities will be defined. Still the Act is the same, with varying degrees of detail uncovered in the de-composition relationship.

We described the nature of varying detail saying that Acts are "fractal", ever more decomposable, just as the movements of a robotic arm can be decomposed in many fine control steps.

Attributes of ActRelationship:

7.1.2.1 ActRelationship.typeCode :: CS (1..1) Mandatory

Vocabulary domain: [ActRelationshipType](#) (CNE)

Attribute description:

A code specifying the meaning and purpose of every ActRelationship instance. Each of its values implies specific constraints to what kinds of Act objects can be related and in which way.

Discussion: The types of act relationships fall under one of 5 categories:

- 1.) (De)-composition, with composite (source) and component (target).
- 2.) Sequel which includes follow-up, fulfillment, instantiation, replacement, transformation, etc. that all have in common that source and target are Acts of essentially the same kind but with variances in mood and other attributes, and where the target exists before the source and the source refers to the target that it links back to.
- 3.) Pre-condition, trigger, reason, contraindication, with the conditioned Act at the source and the condition or reason at the target.
- 4.) Post-condition, outcome, goal and risk, with the Act at the source having the outcome or goal at the target.
- 5.) A host of functional relationships including support, cause, derivation, etc. generalized under the notion of "pertinence".

7.1.2.2 [ActRelationship.inversionInd](#) :: **BL** (0..1)

Attribute description:

An indicator specifying that the [ActRelationship.typeCode](#) should be interpreted as if the roles of the source and target Acts were reversed. The inversion indicator is used when the meaning of [ActRelationship.typeCode](#) must be reversed.

7.1.2.3 [ActRelationship.contextControlCode](#) :: **CS** (0..1)

Vocabulary domain: [ContextControl](#) (CNE)

Attribute description:

A code that specifies how this [ActRelationship](#) contributes to the context of the current Act, and whether it may be propagated to descendent Acts whose association allows such propagation (see [ActRelationship.contextConductionInd](#)).

Rationale: In the interest of reducing duplication, humans tend to rely on context when interpreting information. For example, when reading a report taken from a folder containing a patient's medical record, the reader will infer that the report deals with the patient, even if there is no direct reference to the patient on the form. However, other pieces of information such as the author of the folder (the hospital that maintains it) may sometimes apply to the contents of the folder (e.g. a report generated by a doctor at the hospital) and other times not (e.g. a copy of a report from another institution). Humans are quite good at making the necessary inferences about what context should be propagated from an item to something within that item. However, incorrect inferences can occur (perhaps the report in the patient's record deals with a relative). Furthermore, computers have substantially more difficulty making such inferences, even though they can be essential for decision-support systems.

Discussion: This attribute allows the clear specification of whether an association adds to the context associated with a particular item (e.g. adding an additional author) or whether it replaces (overrides) part of the context associated with a particular item (e.g. identifying a sole author, independent of the containing item). It also indicates whether the association applies to only this act (non-propagating), or whether it may apply to derived acts as well (propagating).

This attribute is closely linked with [ActRelationship.contextConductionInd](#) which determines whether associations that have been marked as propagating will actually be conducted to a child Act. For example,

an author participation might be marked as propagating, but still not conducted to a hyperlink to an external document.

If no value or default is specified for this attribute (i.e. it is null), no inference can be made about context. Systems must make their own assumptions on the basis of what data is being represented. (For this reason, HL7 committees are encouraged to specify a default or fixed value for this attribute as part of their designs to ensure consistency of interpretation.)

Examples: An observation event has a patient participation marked "additive, propagating" (AP) and has component observation events linked through act relationships that are marked propagating. This means that the patient participation behaves as a patient participation of those component observation events in addition to the parent observation event.

A composite order is created containing a pharmacy order as well as requests for several lab tests. The composite order has participations for patient and author, and an act relationship to a diagnosis, all marked as "additive, propagating". The "component" association between the composite order and the pharmacy order is marked as conductive (contextConductionInd is TRUE). The pharmacy order has an author participation marked as "additive, non-propagating" (AN), and a reason relationship to a diagnosis, marked as "overriding, propagating" (OP). The order further has a relationship to a dispense event, marked as conductive, and an association to a drug protocol marked as non-conductive (contextConductionInd is FALSE). The meaning would be as follows:

The pharmacy order is interpreted as having the patient from the composite order, and having two authors (the one from the composite order, and the one on the pharmacy order itself). The diagnosis for the pharmacy order would **only** be the diagnosis specified on the pharmacy order, not the one specified on the composite order. The dispense event would carry the patient from the composite order and the diagnosis from the pharmacy order, but no author. The drug protocol would not be associated with a patient, diagnosis or author.

7.1.2.4 [ActRelationship.contextConductionInd](#) :: **BL** (0..1)

Attribute description:

If true, associations in the parent act are conducted across the ActRelationship to the child act.

Discussion: Only associations that have been added to the context of an Act and are marked "propagating" will be conducted. (See contextControlCode on ActRelationship and Participation)

The identification of an Act as a parent or child (and therefore the direction context will be conducted) is determined by how the association is traversed when it is serialized. The first act to be encountered is considered to be the parent. Context conducts across the ActRelationship to the second (child) Act.

Refer to ActRelationship.contextControlCode for rationale and examples.

7.1.2.5 [ActRelationship.sequenceNumber](#) :: **INT** (0..1)

Attribute description:

An integer specifying the relative ordering of this relationship among other like-types relationships having the same source Act.

Discussion: This attribute is part of the workflow control suite of attributes. An action plan is a composite Act with component Acts. In a sequential plan, each component has a sequenceNumber that specifies the

ordering of the plan steps. Multiple components with the same sequenceNumber make a branch. Branches can be exclusive (case-switch) or can indicate parallel processes indicated by the splitCode.

7.1.2.6 [ActRelationship.priorityNumber](#) :: INT (0..1)

Attribute description:

An integer specifying the relative preference for considering this relationship before other like-typed relationships having the same source Act. Relationships with lower priorityNumber values are considered before and above those with higher values.

Examples: For multiple criteria specifies which criteria are considered before others. For components with the same sequence number, specifies which ones are considered before others. Among alternatives or options that are being chosen by humans, the priorityNumber specifies preference.

Discussion: The ordering may be a total ordering in which all priority numbers are unique or a partial ordering, which assigns the same priority to more than one relationship.

7.1.2.7 [ActRelationship.pauseQuantity](#) :: PQ (0..1)

Attribute description:

A quantity of time that should elapse between when an Act is ready for execution and the actual beginning of the execution.

Discussion: This attribute is part of the workflow control suite of attributes. An action plan is a composite Act with component Acts. In a sequential plan, each component has a sequenceNumber that specifies the ordering of the plan steps. Before each step is executed and has preconditions these conditions are tested and if the test is positive, the Act has clearance for execution. At that time the pauseQuantity timer is started and the Act is executed after the pauseQuantity has elapsed.

7.1.2.8 [ActRelationship.checkpointCode](#) :: CS (0..1)

Vocabulary domain: [ActRelationshipCheckpoint](#) (CNE)

Attribute description:

A code specifying when in the course of an Act a precondition for the Act is evaluated (e.g., before the Act starts for the first time, before every repetition, after each repetition but not before the first, or throughout the entire time of the Act).

Discussion: This attribute is part of the workflow control suite of attributes. An action plan is a composite Act with component Acts. In a sequential plan, each component has a sequenceNumber that specifies the ordering of the plan steps. Before each step is executed, those with preconditions have their conditions tested; where the test is positive, the Act has clearance for execution. The repeatNumber may indicate that an Act may be repeatedly executed. The checkpointCode is specifies when the precondition is checked and is analogous to the various conditional statements and loop constructs in programming languages "while-do" vs. "do-while" or "repeat-until" vs. "loop-exit".

For all checkpointCodes, except "end", preconditions are being checked at the time when the preceding step of the plan has terminated and this step would be next in the sequence established by the sequenceNumber attribute.

When the checkpointCode for a criterion of a repeatable Act is "end", the criterion is tested only at the end of each repetition of that Act. When the condition holds true, the next repetition is ready for execution.

When the checkpointCode is "entry" the criterion is checked at the beginning of each repetition (if any) whereas "beginning" means the criterion is checked only once before the repetition "loop" starts.

The checkpointCode "through" is special in that it requires the condition to hold throughout the execution of the Act, even throughout a single execution. As soon as the condition turns false, the Act should receive an interrupt event (see Act.interruptibleInd) and will eventually terminate.

The checkpointCode "exit" is only used on a special plan step that represents a loop exit step. This allows an action plan to exit due to a condition tested inside the execution of this plan. Such exit criteria are sequenced with the other plan components using the ActRelationship.sequenceNumber.

7.1.2.9 ActRelationship.splitCode :: CS (0..1)

Vocabulary domain: [ActRelationshipSplit](#) (CNE)

Attribute description:

A code specifying how branches in an action plan are selected among other branches.

Discussion: This attribute is part of the workflow control suite of attributes. An action plan is a composite Act with component Acts. In a sequential plan, each component has a sequenceNumber that specifies the ordering of the plan steps. Branches exist when multiple components have the same sequenceNumber. The splitCode specifies whether a branch is executed exclusively (case-switch) or inclusively, i.e., in parallel with other branches.

In addition to exclusive and inclusive split the splitCode specifies how the pre-condition (also known as "guard conditions" on the branch) are evaluated. A guard condition may be evaluated once when the branching step is entered and if the conditions do not hold at that time, the branch is abandoned. Conversely execution of a branch may wait until the guard condition turns true.

In exclusive wait branches, the first branch whose guard conditions turn true will be executed and all other branches abandoned. In inclusive wait branches some branches may already be executed while other branches still wait for their guard conditions to turn true.

7.1.2.10 ActRelationship.joinCode :: CS (0..1)

Vocabulary domain: [ActRelationshipJoin](#) (CNE)

Attribute description:

A code specifying how concurrent Acts are resynchronized in a parallel branch construct.

Discussion: This attribute is part of the workflow control suite of attributes. An action plan is a composite Act with component Acts. In a sequential plan, each component has a sequenceNumber that specifies the ordering of the plan steps. Branches exist when multiple components have the same sequenceNumber. Branches are parallel if the splitCode specifies that more than one branch can be executed at the same time. The joinCode then specifies if and how the branches are resynchronized.

The principal re-synchronization actions are (1) the control flow waits for a branch to terminate (wait-branch), (2) the branch that is not yet terminated is aborted (kill-branch), (3) the branch is not re-synchronized at all and continues in parallel (detached branch).

A kill-branch is only executed if there is at least one active wait (or exclusive wait) branch. If there is no other wait branch active, a kill-branch is not started at all (rather than being aborted shortly after it is started).

Since a detached branch is unrelated to all other branches, active detached branches do not protect a kill-branch from being aborted.

7.1.2.11 [ActRelationship.negationInd](#) :: **BL** (0..1)

Attribute description:

An indicator that asserts that the meaning of the link is negated.

Examples: If the relationship without negation specifies that Act A has Act B as a component, then the negation indicator specifies that Act A does **not** have Act B as a component. If B is a reason for A, then negation means that B is **not** a reason for A. If B is a pre-condition for A, then negation means that B is **not** a precondition for A.

Discussion: As the examples show, the use of this attribute is quite limited, notably contrast this with the [Act.negationInd](#) that actually requires that the described Act not exist, not be done, etc. whereas the [ActRelationship.negationInd](#) merely negates this relationship between source and target act, but does not change the meaning of each Act. This is mostly used for clarifying statements.

Note also the difference between negation and the contrary. A contraindication is the contrary of an indication (reason) but not the negation of the reason. The fact that lower back pain is not a reason to prescribe antibiotics doesn't mean that antibiotics are contraindicated with lower back pain.

7.1.2.12 [ActRelationship.conjunctionCode](#) :: **CS** (0..1)

Vocabulary domain: [RelationshipConjunction](#) (CNE)

Attribute description:

A code specifying the logical conjunction of the criteria among all the condition-links of Acts (e.g., and, or, exclusive-or).

Constraints: All AND criteria must be true. If OR and AND criteria occur together, one criterion out of the OR-group must be true and all AND criteria must be true also. If XOR criteria occur together with OR and AND criteria, exactly one of the XOR criteria must be true, and at least one of the OR criteria and all AND criteria must be true. In other words, the sets of AND, OR, and XOR criteria are in turn combined by a logical AND operator (all AND criteria and at least one OR criterion and exactly one XOR criterion). To overcome this ordering, Act criteria can be nested in any way necessary.

7.1.2.13 [ActRelationship.localVariableName](#) :: **ST** (0..1)

Attribute description:

A character string name for the input parameter from which the source Act of this [ActRelationship](#) derives some of its attributes. The local variable name is bound in the scope of the [Act.derivationExpr](#) with its value being an Act selected based on the input parameter specification.

7.1.2.14 [ActRelationship.seperatableInd](#) :: **BL** (0..1)

Attribute description:

This attribute indicates whether or not the source Act is intended to be interpreted independently of the target Act. The indicator cannot prevent an individual or application from separating the Acts, but indicates the author's desire and willingness to attest to the content of the source Act if separated from the target Act. Note that the default for this attribute will typically be "TRUE". Also note that this attribute is orthogonal and unrelated to the RIM's context/inheritance mechanism. If the context of an Act is propagated to nested Acts, it is assumed that those nested Acts are not intended to be interpreted without the propagated context.

7.1.3 Class: Participation (*in Acts*)

- Attributes of Participation:

typeCode :: CS	noteText :: ED	signatureText :: ED
functionCode :: CD	time :: IVL<TS>	performInd :: BL
contextControlCode :: CS	modeCode :: CE	substitutionConditionCode :: CE
sequenceNumber :: INT	awarenessCode :: CE	
negationInd :: BL	signatureCode :: CE	

- Associations of Participation:

act :: (1..1) **Act** :: **participation** :: (0..*)
role :: (1..1) **Role** :: **participation** :: (0..*)

- Participation generalizes:

ManagedParticipation

Description of Participation:

An association between an Act and a Role with an Entity playing that Role. Each Entity (in a Role) involved in an Act in a certain way is linked to the act by one Participation-instance. The kind of involvement in the Act is specified by the Participation.typeCode.

Examples: 1) Performers of acts (surgeons, observers, practitioners).

2) Subjects of acts, patient, devices

3) Locations

4) Author, co-signer, witness, informant

5) Addressee, information recipient

Rationale: Participations represent performance while Roles represent competence. Participations specify the actual performance of an Entity in a certain Act, and thus a Participation is scoped by its specific Act. Conversely, Roles specify the competence of an Entity (i.e., how it may principally participate in what kinds of acts) irrespective of any individual Act.

For example, the professional credentials of a person (Role) may be quite different from what a person actually does (Participation). A common example is interns and residents performing anesthesia or surgeries under (more or less) supervision of attending specialists.

An Act can have multiple participations of the same type, which indicates collaborative activities or group involvement. The notion of multiple performing Participations partially overlaps with sub-activities (Act components). Whenever multiple actors are involved in an act, each actor performs a different task (with the extremely rare exception of such symmetrical activities as two people pulling a rope from either end). Thus, the presence of multiple actors could be equally well represented as an act consisting of sub-acts where each act would have only one performing actor

For example, a record of a surgical service may include the actors of type: (a) consentor, (b) primary surgeon, and (c) anesthetist. These three actors really perform different tasks, which can be represented as three related acts: (a) the consent, (b) the surgery proper, and (c) the anesthesia act in parallel to the surgery. If we used the sub-acts, the consentor, surgeon and anesthetist could simply be of actor type "performer."