

The Unified² Service Action Model

A Proposal to the HL7 Technical Committees Orders/Observations and Patient Care

Gunther Schadow

Regenstrief Institute for Health Care, Indiana University
School of Medicine, Indianapolis, IN

Dan Russler

McKesson HBOC, Atlanta, GA

Charlie Mead

Carecentric Solutions, Duluth, GA

Jim Case

University of California, Davis

Clem McDonald

Regenstrief Institute for Health Care, Indiana University
School of Medicine, Indianapolis, IN

Contents

PART A	1
1 Introduction	1
The Proposed Model	5
2.1.1 Data Types	6
2.2 The Service Action Centered View	8
2.2.1 Attributes of class Service.....	9
2.2.1.1 <i>Service.id</i> : SET{II}.....	9
2.2.1.2 <i>Service.mood_cd</i> : SET{CV}.....	9
2.2.1.3 <i>Service.type_cd</i> : CD.....	10
2.2.1.4 <i>Service.descr</i> : ED.....	10
2.2.1.5 <i>Service.status_cd</i> : CV.....	11
2.2.1.6 <i>Service.total_time</i> : GTS.....	12
2.2.1.7 <i>Service.critical_time</i> : GTS.....	12
2.2.1.8 <i>Service.method_cd</i> : CD.....	12
2.2.1.9 <i>Service.body_site_cd</i> : CD.....	12
2.2.1.10 <i>Service.interpretation_cd</i> : SET{CV}.....	13
2.2.1.11 <i>Service.confidentiality_cd</i> : SET{CV}.....	14
2.2.1.12 <i>Service.max_repeat_nmb</i> : INT default: 1.....	15
2.2.1.13 <i>Service.interruptible_ind</i> : BL default: true.....	16
2.2.1.14 <i>Service.substitution_cd</i> : CV default: N.....	16
2.2.1.15 <i>Service.priority_cd</i> : SET{CV} default: {R}.....	16
2.2.1.16 <i>Service.orderable_ind</i> : BL default: true.....	17
2.3 Actors and targets	17
2.3.1 Actor	17
2.3.1.1 <i>Actor.type_cd</i> : SET{CV}.....	18
2.3.1.2 <i>Actor.tmr</i> : IVL{TS}.....	20
2.3.1.3 <i>Actor.note_txt</i> : ED.....	20
2.3.1.4 <i>Actor.signature_cd</i> : CV.....	20
2.3.2 Target.....	21
2.3.2.1 <i>Target.type_cd</i> : SET{CV}.....	21
2.3.2.2 <i>Target.tmr</i> : SET{CV}.....	22
2.3.2.3 <i>Target.awareness_cd</i> : CV.....	23
2.4 Action Structures	23
2.4.1 Attributes of class service relationship	25
2.4.1.1 <i>Service_relationship.type_cd</i> : CV.....	25
2.4.1.2 <i>Service_relationship.inversion_ind</i> : BL default: false.....	28
2.4.1.3 <i>Service_relationship.sequence_nmb</i> : INT default: 1.....	29
2.4.1.4 <i>Service_relationship.priority_nmb</i> : INT default: 1.....	29
2.4.1.5 <i>Service_relationship.pause_qty</i> : PQ ~ 1 s default: 0 s.....	29
2.4.1.6 <i>Service_relationship.checkpoint_cd</i> : CV default: B.....	29
2.4.1.7 <i>Service_relationship.split_cd</i> : CV default: II.....	29
2.4.1.8 <i>Service_relationship.join_cd</i> : CV default: W.....	30
2.4.1.9 <i>Service_relationship.negation_ind</i> : BL default: false.....	30
2.4.1.10 <i>Service_relationship.conjunction_cd</i> : BL default: AND.....	30
2.5 Timed and conditioned care plans	31
2.5.1 Plans.....	31
2.5.1.1 Background.....	31
2.5.1.1.1 Sequential control constructs	32
2.5.1.1.2 Concurrent control constructs	33
2.5.1.2 HL7 Activity Plans	33
2.5.2 Conditions.....	35

2.5.3	Timing.....	36
2.5.3.1	<i>A duration is a physical quantity with a time unit</i>	37
2.5.3.2	<i>Points on the absolute real time axis</i>	37
2.5.3.3	<i>Intervals</i>	37
2.5.3.4	<i>Recurring events, periodic time</i>	38
2.5.3.5	<i>Frequency f or period duration T are equivalent</i>	39
2.5.3.6	<i>Periodic points vs. intervals and the phase ϕ</i>	39
2.5.3.7	<i>Periodic points and intervals as sets of time points</i>	39
2.5.3.8	<i>Stochastic and exact timing</i>	40
2.5.3.9	<i>Alignment to calendars</i>	40
2.5.3.10	<i>Literal expression and the General Time Specification (GTS)</i>	41
2.5.3.11	<i>Time related attributes</i>	43
2.6	Special kinds of Services.....	44
2.6.1	Observation.....	45
2.6.1.1	<i>Observation.value : ANY</i>	45
2.6.1.2	<i>Observation.derivation_expr : ST</i>	46
2.6.1.3	<i>Observation.property_cd : CV</i>	46
2.6.1.4	<i>Observation.type_cd : CD inherited from Service</i>	46
2.6.1.4.1	Diagnoses and Allergies	47
2.6.1.4.2	Demographic observations, age, gender, and race, and their use in reference ranges.	48
2.6.1.5	<i>Observation.descr : ED inherited from Service</i>	50
2.6.1.6	<i>Observation.critical_time : GTS inherited from Service</i>	50
2.6.2	Procedure	50
2.6.2.1	<i>Procedure.entry_site_cd : CD</i>	50
2.6.2.2	<i>Procedure.body_site_cd : CD inherited from Service</i>	50
2.6.3	Medication	50
2.6.3.1	<i>Medication.form_cd : CD</i>	51
2.6.3.2	<i>Medication.route_cd : CD</i>	51
2.6.3.3	<i>Medication.dose_qty : PQ</i>	52
2.6.3.4	<i>Medication.strength_qty : PQ default: 1</i>	52
2.6.3.5	<i>Medication.rate_qty : PQ ~ 1 s</i>	52
2.6.3.6	<i>Medication.dose_check_qty : PQ</i>	53
2.6.3.7	<i>Medication.method_cd : CD inherited from Service</i>	53
2.6.4	Condition node.....	54
2.6.4.1	<i>Condition_node.type_cd : CD inherited, default: no information</i>	54
2.6.4.2	<i>Condition_node.mood_cd : CD inherited, default: EVN</i>	55
2.6.5	Consent	55
2.6.5.1	<i>Consent.type_cd : CD inherited from Service</i>	55
2.6.5.2	<i>Consent.descr : ED inherited from Service</i>	56
2.6.5.3	<i>Consent.time : GTS inherited from Service</i>	56
2.6.5.4	<i>Consent.critical_time : GTS inherited from Service</i>	57
2.6.6	Transportation.....	57
2.6.6.1	<i>Transportation.type_cd : CD inherited from Service</i>	57
2.6.6.2	<i>Transportation.critical_time : GTS inherited from Service</i>	58
2.6.7	Supply	58
2.6.7.1	<i>Supply.qty : PQ</i>	58
2.6.7.2	<i>Supply.method_cd : CD</i>	58
2.6.8	Diet service	59
2.6.8.1	<i>Diet.energy_qty : PQ ~ 1 kcal/d</i>	59
2.6.8.2	<i>Diet.carbohydrate_qty : PQ ~ 1 g/d</i>	59
2.6.8.3	<i>Diet.type_cd : CD inherited from Service</i>	59
2.6.8.4	<i>Diet.method_cd : CD inherited from Service</i>	60
2.7	Substance: the Material class.....	60
2.7.1	Attributes of class Material	60
2.7.1.1	<i>Material.id : SET(II)</i>	60
2.7.1.2	<i>Material.type_cd : CD</i>	61
2.7.1.3	<i>Material.form_cd : CV</i>	62

2.7.1.4	<i>Material.descr</i> : ED.....	62
2.7.1.5	<i>Material.status_cd</i> : SET{CV}.....	62
2.7.1.6	<i>Material.extent_tmr</i> : IVL{TS}.....	62
2.7.1.7	<i>Material.lot_nmb</i> : ST.....	63
2.7.1.8	<i>Material.handling_cd</i> : CD.....	63
2.7.1.9	<i>Material.danger_cd</i> : CD.....	63
2.7.1.10	<i>Material.qty</i> : SET{PQ} default: {1}.....	64
2.7.2	Material relationships to other Material.....	65
2.7.2.1	<i>Material_relationship.type_cd</i> : CV.....	65
2.7.2.2	<i>Material_relationship.inversion_ind</i> : BL default: false.....	66
2.7.2.3	<i>Material_relationship.tmr</i> : IVL{TS}.....	66
2.7.2.4	<i>Material_relationship.position_nmb</i> : LIST{NM}.....	67
2.7.2.5	<i>Material_relationship.qty</i> : PQ.....	67
2.7.3	Responsibilities of Stakeholders for Material.....	68
2.7.3.1	<i>Responsibility.type_cd</i> : CV.....	68
2.7.3.2	<i>Responsibility.tmr</i> : IVL{TS}.....	69
2.7.3.3	<i>Responsibility.material_id</i> : SET{II}.....	69
2.8	The Roles of Material.....	69
2.8.1	Specimen.....	69
2.8.1.1	<i>Specimen.body_site_cd</i> : CD.....	70
2.8.1.2	<i>Material.type_cd</i> : CD.....	70
2.8.1.3	<i>Material.extent_tmr</i> : IVL{TS}.....	71
2.8.1.4	<i>Material.qty</i> : SET{PQ} default: {1}.....	71
2.8.2	Container.....	72
2.8.2.1	<i>Container.capacity_qty</i> : PQ.....	72
2.8.2.2	<i>Container.height_qty</i> : PQ ~ 1 cm.....	72
2.8.2.3	<i>Container.diameter_qty</i> : PQ ~ 1 cm.....	72
2.8.2.4	<i>Container.barrier_delta_qty</i> : PQ ~ 1 cm.....	72
2.8.2.5	<i>Container.bottom_delta_qty</i> : PQ ~ 1 cm.....	72
2.8.2.6	<i>Container.separator_type_cd</i> : CD.....	72
2.8.2.7	<i>Container.cap_type_cd</i> : CD.....	72
2.8.3	Therapeutic agent.....	73
2.8.4	Devices.....	73
2.8.5	Access tubes and drains.....	73
2.8.5.1	<i>Access.gauge_qty</i> : PQ.....	73
2.8.5.2	<i>Access.entry_site_cd</i> : CD.....	74
2.8.5.3	<i>Access.body_site_cd</i> : CD inherited from Service.....	74
2.8.6	Location as a Role of Material.....	75
2.8.7	Food.....	75
2.8.7.1	<i>Food.preference_cd</i> : CD default: NVEG.....	75
2.9	Stakeholder Owned Service Lists.....	76
2.9.1	Service List.....	76
2.9.1.1	<i>Service_list.id</i> : SET{II}.....	76
2.9.1.2	<i>Service_list.type_cd</i> : CV.....	76
2.9.1.3	<i>Service_list.name</i> : ST.....	77
2.9.1.4	<i>Service_list.descr</i> : ED.....	77
2.9.2	List Item.....	77
2.9.2.1	<i>List_item.sequence_nmb</i> : REAL default: 1.....	77
2.9.2.2	<i>List_item.priority_nmb</i> : REAL.....	77
2.9.2.3	<i>List_item.note_txt</i> : ED.....	77

List of Figures

Figure 1: This is the complete class diagram of the USAMP-II model, covering the clinical and ancillary part of the entire HL7 RIM. This includes the traditional RIM areas: orders, service event, master service, scheduling, and patient care. The three service related class hierarchies (formally called master service, service order, and service event have been merged into one Service hierarchy. The attribute “mood_cd” distinguished between different nuances of a service (e.g., whether it’s the service as ordered, as scheduled, as performed, as reported in history taking, as a goal, etc.) The second important novelty is the unification of Material that includes all the substantial “things” (except people) that services deal with. In spite the dramatic decrease in attributes, all current application layer requirements of HL7 version 2.3.1 are covered.....	5
Figure 2: A surgeon intends to perform a cholecystectomy because there are gallstones found by a radiologist. The medical record must attribute the “reason” link properly to the surgeon so as to avoid ambiguity as if the radiologist has suggested the cholecystectomy. The rule of attribution is that all service relationships are attributed to the responsible actor of the source service.....	24
Figure 3: Simple model of the essential sequential programming constructs: A statement may be primitive (not specialized.) A conditional consists of a number of branches, each branch having a condition and a statement. A block is a sequence of statements that is executed once, while a loop is just a block executed more than once. Blocks (loops) can be exited with the exit statement, that is usually embedded in a conditional.....	31
Figure 4: Example of sequential plan construction for laparoscopic cholecystectomy. Edged boxes are Service instances (all in definition “master” mood.) Rounded boxes are Service_relationship instances of type: plan-component. The sequence_nmb attribute orders the relationships into a sequence. Each service can in turn be decomposed into plan-components.....	32
Figure 5: Example of plan construction with a concurrent region. Edged boxes are Service instances. Rounded boxes are Service_relationship instances of type: plan-component showing only the sequence_nmb attribute. Since the components A3.1, A3.2 and A3.3 all have the same sequence number (3) they may be executed in parallel.....	34
Figure 6: A simple condition representing a PRN medication order: Acetaminophen 1000 mg tab p.o. PRN for pain. Criteria always refer to the most recent service if the time attribute is not specified.....	35
Figure 7: Example of a complex criterion as a precondition for Service X, representing the Boolean expression ((A OR B) AND C) XOR (D OR E).....	36
Figure 8: Periodic events can be described in terms of a rotation or wave. The period T , frequency f , and angular velocity ω are equivalent measures of how often the event occurs in a unit of time. The phase ϕ is a measure for the offset between the wave and a reference wave of the same frequency. The phase may be measured in a unit of plane angle or as a difference of time Δt . Thus, all parameters describing the periodic event can be measured as time duration.....	38
Figure 9: Example of constructing a complex schedule as the intersection of various periodic intervals of time. The schedule is: every other day from Monday to Friday 8:00 to 10:00 AM for six times. Regardless of how the set of time was constructed, it has one outer bound interval and can be enumerated as a sequence of intervals, each representing one occurrence of the service.....	40
Figure 10: A service as ordered by a doctor is only part of what needs to actually happen. Typically a real service consists of preparation and cleanup work that may take just as much time as the biologically relevant part of the service. While the medical reasoning will focus on the biologically relevant aspect of the service, scheduling must consider the entire service.....	44
Figure 11: Reference ranges are frequency distributions of observations among populations. The reference ranges (mood: REF) are associated with the master observation (upper left, mood: DEF) through <i>reference</i> links. The reference value is usually an interval of physical quantity (low–high) or, with nominal observations, a set of value codes. If a reference range has no	

criterion, it is the typical “normal” range, based on the not further specified healthy population. If criteria are associated with the reference, the criteria can be any observation (mood: EVN+CRT), but sex and age are the most common reference range criteria. An actual observation (upper right, mood: EVN) may be linked with the applicable reference range in order to specify which range has been applied to determine the interpretation (abnormal) flag “H” on the service report. 49

List of Tables

Table 1: Overview of HL7 version 3 data types and mapping to HL7 v2.3	7
Table 2: Service action moods – the boldfaced moods are the conventional moods of HL7 v2.	9
Table 3: Meaning of the description attribute depending on the mood of the service.	11
Table 4: Body site concepts (from HL7 v2.3 table 0163).....	13
Table 5: Interpretation codes.	13
Table 6: Confidentiality policies	14
Table 7: Service substitution code.....	16
Table 8: Service priority code.....	16
Table 9: Actor types	18
Table 10: Actor signature code.....	21
Table 11: Target types	21
Table 12: Awareness code.....	23
Table 13: Service relationship types.....	25
Table 14: Condition checkpoint code.....	29
Table 15: Branch split code.....	30
Table 16: Branch join code.....	30
Table 17: Criteria conjunction code	30
Table 18: Period Identifiers in the Gregorian (western) Calendar	41
Table 19: Examples for literal expressions for time sets.	42
Table 20: Choice of observation value data types	45
Table 21: Observation type codes for diagnoses, allergies, problems and complaints.....	47
Table 22: Observation type codes for demographic observations used to define populations	48
Table 23: Route of administration	51
Table 24: Methods of administering medication (from HL7 v2.3 table 0165).....	53
Table 25: Service relationship types especially relevant for conditions	54
Table 26: Types of consents and other legal documents	56
Table 27: Actors and targets in an example transport service	57
Table 28: Types of transportation services	57
Table 29: Supply methods for pharmacy dispensing services (HL7 v2.3 table 0321).....	58
Table 30: Medically relevant diet types, not including patient preferences.....	59
Table 31: Concept repertoire for forms of material.....	62
Table 32: Material handling code	63
Table 33: Material danger code	63
Table 34: Kinds of quantities for amounts of material	64
Table 35: Material relationship types	65
Table 36: Material responsibility type code	68
Table 37: Material type codes for specimen.....	70
Table 38: Devices commonly used to administer medication (from HL7 v2.3 table 0164).....	73
Table 39: Preferences, or “styles” of food.....	76
Table 40: Types of stakeholder owned service lists.	77

The Unified² Service Action Model

PART A

Specification of the Model

1 Introduction and Overview

This white paper is a proposal to the Patient Care, Orders/Observations, and Scheduling/Referral HL7 Technical Committees (TCs) for a substantive change in the overall structure of the Reference Information Model (RIM). In addition, even though the aforementioned TCs are the primary audience for whom this proposal is intended, various other committees will be interested in some – if not all – of the concepts and details discussed in this proposal. The authors are particularly interested in critical input from the Lab Automation Special Interest Group (SIG), Image Management SIG, Patient Administration/Financial Management, and of course, Modeling and Methodology.

This proposal, which reflects the composite thinking of a group of active members of the "primary target" committees, outlines the main structures for, as well as some of salient details involved in, a fundamental restructuring of the "clinical" part of the HL7 RIM. The proposed changes, if adopted, would result in the replacement of 34 classes and 325 attributes with a newly-defined set of 23 classes and 78 attributes. The motivation of this effort has been to simplify and rationalize the RIM, while simultaneously evolving it so that it could more easily accommodate the various new messaging requirements being generated by both a changing United States Healthcare Delivery system, and/or HL7's desire to integrate and harmonize its efforts with those of other international healthcare standards bodies. Additionally, the work was done with a consistent awareness of not losing any of the knowledge represented in previous versions of HL7, particularly the most recent (and therefore complete) v2.3. Thus, the proposed model was developed after a careful study of all relevant parts of HL7 version 2.3, with special attention to the messaging functionality described in v2.3.1 Chapters 4, 7 (except clinical trials,) 8, 10, and 12, and parts of Chapters 3 and 6 (AL1, DG1, PR1). (Of some interest is the fact that the revised model fits on one letter-size page. Although the physical presentation of the model was certainly not a primary driver during model development, the authors strongly believe that the clarity, conciseness, and compactness of presentation of the HL7 RIM will ultimately aid in the important and ongoing task of securing broad-based understanding and consequent acceptance of the RIM.)

This document has two major parts. The first part describes the proposed model in detail, and provides precise and complete definitions for all classes, attributes and associations. Also included in this section are reasonably complete "concept repertoires" (i.e. code tables) for each coded attribute. The notable exception to code-table definition occurs for attributes whose values would expected to be

drawn from general healthcare terminologies. Rather than include exhaustive code tables for these attributes, we have simply included references to examples of appropriate external coding systems (e.g. SNOMED, ICD, NANDA etc.). It should be noted, however, that the number of attributes where such "generally available" terminologies are referenced has been significantly reduced and normalized to one of three types reference:

1. Service/Activity/Action names, modifiers and methods;
2. Names of Material "things;" and
3. Anatomic Structures and Systems.

The second part of this proposal provides evidence for the authors' claim that the revised model addresses all functionality of HL7 v2.3.1 in those areas potentially affected by the proposed changes. The validity of this claim is substantiated by means of a detailed mapping of all the segments and fields of the affected areas between HL7 2.3 and the proposed model. The authors believe that the mapping not only delineates domain completeness and compatibility with the v2.3 message set, but also demonstrates improvements in both the clarity and expressiveness of certain message elements when compared to the same or similar elements as defined in HL7 v2.3.1. Of particular note is the removal of most free text and/or character string fields defined in v2.3.1, and the replacement of those fields with well-defined, interoperable, table-resident codes.

Based on the information presented in this document's two parts, the authors believe that the proposed model will pave the way to new messaging opportunities, including quality management, outcome assessment, decision support, cost control, and authenticated, accurate, electronic medical record communication, while simultaneously providing a clear link back to the existing HL7 version 2.x standards. (It should be noted in the context of the mapping portion of this document the proposed model also preserves substantive amounts of the message design work that has been done during the development of HL7 version 3.)

The model is framed around three central constructs (see Figure 1):

1. Unification and abstraction of the "clinically-relevant activities/actions" and "things" that fall within the scope of HL7's charge into two base classes: "Clinically-relevant activities/actions" are represented by the class "Service," while "things" are represented by the class "Material;"
2. Formalization of the fact that any activity/action (or thing) represented as an instance of the Service (or Material) class can itself be either further decomposed into a set of more finely-granulated component activities/actions (or things), or, alternatively itself be included in the composition of a more coarsely-granulated composite activity/action (or thing). The relationships between various activities/actions (or things) involved in various compositions/decompositions is modeled as a reflexive relationship between the Service (or Material) class and the class Service_relationship (or Material_relationship).
3. Three clearly defined -- as well as flexible and expandable -- associations between the Service and Material classes and the people and/ organizations who are, respectively, Actors or Targets of Services, and/or Responsible Parties of Material.

The authors believe that the most notable contribution of the model to the evolution of the existing HL7 RIM is the clear identification of two collections of classes: the Service class/subclass hierarchy, and the Material class and its collection of associated classes. Through these two collections, the model clarifies and unifies a previously widely-distributed, fragmented, and inconsistently abstracted set of related classes. The unification occurs by means of:

1. A single attribute -- mood_cd -- of the Service class; and
2. The set of classes associated with instances of the Material class via well-defined semantic "roles."

The authors acknowledge that the modeling framework presented here is somewhat different from that

used in much of traditional information modeling, where two distinct layers are typically identified: a "knowledge" layer describing things that *may* exist, *might* be observed, or *may be* done; and an "information" layer describing things that *actually* exist, *have been* observed, or *have been* done. A close examination of models constructed using this two-layered approach reveals that, other than being differentiable based on a fundamental difference in "mood" (i.e. the knowledge layer represents things that have a mood of "potential" which the information layer represents things that have a mood of "actual"), the content and structure of the two layers is often nearly identical. Such is the case in the current version of the RIM. In contrast, the present model simultaneously collapses the two-layered approach into a single layer in which both knowledge and information instances can accurately be represented using an appropriately-chosen value of the "mood_cd" attribute (e.g. "actual" vs "intent"). It then further expands the expressiveness of the model gained by use of the concept of mood through use of a set of semantically rich "moods" which allow several types of both "actual" and/or "intent" instances to be defined (see Table 2).

The notion of "mood" -- which is discussed in detail in the document -- is of pivotal importance to the model, and is the fundamental cornerstone whereby a departure from the traditional two-layer modeling approach is enabled. In particular, much of the contextual semantics assumed by an instance of class Service are fundamentally distinguishable based on the value of the mood_cd attribute (e.g. "possible," "actual," "intended," "ordered," "expected," "suggested," "aimed," "feared," etc.).

The concept of "mood" is consistent with the framework of human language, a framework to which the discipline of electronic data interchange has often looked for metaphor and/or inspiration. Specifically, human language has taught us the importance of the fundamental distinction of subject and verb, the power of the verb in defining contextual meaning, and the importance of combinatorics (i.e. compositional grammar) in non-ambiguous but expressive communication. In particular, much of the power and expressiveness of human language comes through factoring common modifiers (e.g. verb tenses and moods) and then combining the factored output in context-specific ways. This USAM--II model attempts to bring a similar type of combinatoric expressiveness and power to the domain of HL7 "language."

However, the authors also realize that the combinatorically-derived expressive power of human language potentially poses a major problem to automated computer processing of shared electronic data, a problem towards which much effort has been directed over the past 50 years: the problem of *meaning*. In particular, it has become painfully clear that meaning that is often the case that meaning that is obvious to humans is largely unintelligible to computers, particularly in the context of computer-to-computer messaging. Therefore, a substantial amount of the effort spent in the design of this model has been spent on finding ways to simplify and normalize information so as to unambiguously define -- from the computer's perspective -- the meaning well-known to a human sender or receiver, but often remarkably opaque to a computer. If the model is successful in this effort, the authors are confident that the increased flexibility and expressiveness facilitated by it will lead not to increased chaos in HL7 messaging, but rather to improved interoperability.

The name of this model is the "Unified² Service Action Model Proposal (USAMP-II)" because it is inspired by, and is the logical extension of, the first Unified Service Action Model Proposal (USAM) that was introduced in the RIM in Spring, 1998. Without the initial efforts of USAM's creators and authors -- Tim Snyder, Charlie Mead and Dan Russler -- USAM--II could not possibly exist today. In addition to USAM's primary architects, Clem McDonald, Gunther Schadow, Linda Quade, Debra Weiss, and Anita Benson made contributions of vital importance to the initial USAM effort. USAM--II owes acknowledgment to Mark Shafarman, Joyce Spindler, and Wayne Tracy. This work has been made possible by the prudence and support of Clem McDonald and the Regenstrief Institute for Health Care.

2 The Proposed Model

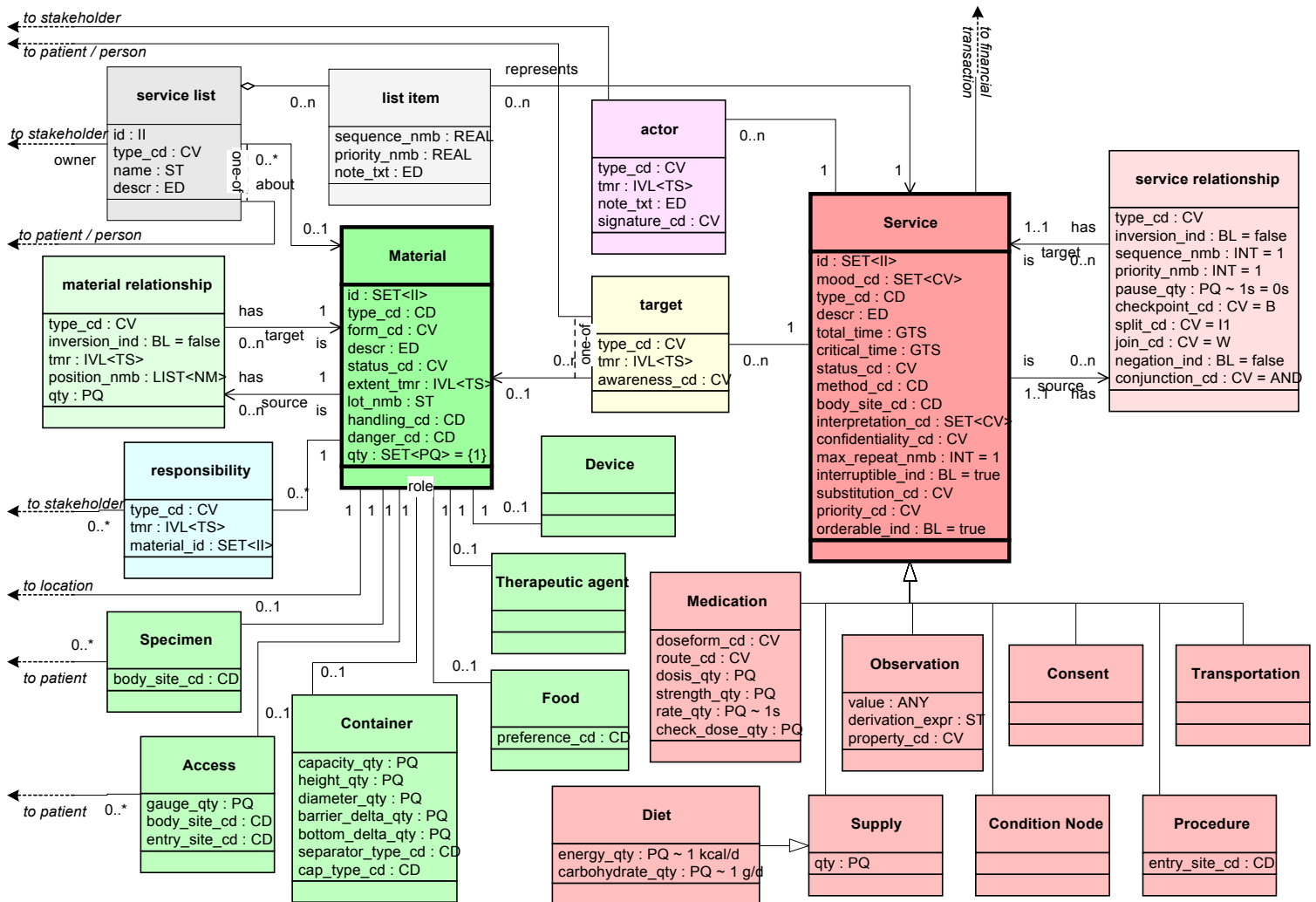


Figure 1: This is the complete class diagram of the USAMP-II model, covering the clinical and ancillary part of the entire HL7 RIM. This includes the traditional RIM areas: orders, service event, master service, scheduling, and patient care. The three service related class hierarchies (formally called master service, service order, and service event) have been merged into one Service hierarchy. The attribute “mood_cd” distinguished between different nuances of a service (e.g., whether it’s the service as ordered, as scheduled, as performed, as reported in history taking, as a goal, etc.) The second important novelty is the unification of Material that includes all the substantial “things” (except people) that services deal with. In spite of the dramatic decrease in attributes, all current application layer requirements of HL7 version 2.3.1 are covered.

This section will provide a detailed introduction in the USAMP-II model as shown in the class diagram of Figure 1. Most of the description will focus on what the model inherently is and how it is used rather than in comparing to previous HL7 models of version 2 or various revisions of the HL7 version 3 RIM. A very detailed mapping guide between HL7 version 2.3.x and this model will follow in Section 2.5.3.1 below.

The Unified Service Action Model (USAM) divides the world into two major categories: action and substance. Substance is all the tangible people and things in the world, while actions assess or change the state of substances.

Quite naturally, substance is divided into Stakeholders and Material. Stakeholders are subjects having legal rights and obligations. Stakeholder includes both individual Person and Organization. Material is everything else that has physical existence in space and time. Material is a large class of all kinds of things, including devices (both durable and disposable equipment), chemicals, food, specimen, and containers, as well as locations and living subjects (non-human species only at this point.)

One could argue that a third important category of entities in health care be “information;” indeed, isn’t the medical record a collection of health related information about a patient? A critical part of the USAMP approach, however, is not to consider “information” entities independent from actions. Of course, health care computing in general and this model in particular is all about information, but computing, communication and information modeling only exists in order to support actions influencing substance for the benefit of people. It is important therefore to always hold on to the focus and purpose of information, that is substance and action.

Main Entry: **ob-ser-va-tion**

Function: *noun*

Etymology: Middle French, from Latin *observatio*, *observatio*, from *observare*

Date: 1535

1 : an act or instance of observing a custom, rule, or law : OBSERVANCE

2 a : an act of recognizing and noting a fact or occurrence often involving measurement with instruments (*weather observations*) **b** : a record or description so obtained

3 : a judgment on or inference from what one has observed; *broadly* : REMARK, STATEMENT

Exhibit 1: Webster’s definition of “observation”

The strong bound between information and action is most obvious with the Observation action. An observation, according to Webster’s, is an “act of recognizing and noting a fact [...] often involving measurement with instruments” and at the same time an observation is also “a record or description so obtained” [i.e. obtained through recognizing and noting]. Thus an observation is both, the action or measurement “procedure” and the resulting information that was obtained. The Unified Service Action Model understands the result to be entirely dependent on the observation action, and thus models the result as a component (attribute) of the Observation action rather than an independent entity.

All other classes in the model that are not substance or actions are *associative classes* and only exist in order to support relationships among and between substance and actions. The recursive relationship classes for Material and Service support relationships among (not between) substance and actions respectively. Relationships *between* substance classes and actions are established through the actor and target classes. The class Service-list is another kind of relationship between substance (Stakeholders) and actions. Service-list accounts for the fact that different stakeholders may assign different priorities to the same actions.

2.1.1 Data Types

In order to understand this model some knowledge about the new HL7 data types is needed. The version 3 data types have been designed in parallel to this model and both designs have fertilized each other. Where specific new features of the new data types are required, we will explain these in line.

The following table tries to give an short overview of the defined data types that are commonly assigned to attributes.

This table is neither complete nor detailed enough to provide anything more than a coarse overview. The complete and detailed definition of HL7 data types is found in the *HL7 version 3 Data Type Specification* (currently under development, see <http://aurora.rg.iupui.edu/v3dt>.) The RIM also contains a special non-normative subject area where data types are represented as information model classes.

Table 1: Overview of HL7 version 3 data types and mapping to HL7 v2.3

Name	Symbol	Description	V2.3
Boolean	BL	The Boolean type stands for the values of two-valued logic. A Boolean value can be either true or false.	ID
Character String	ST	Used when the appearance of text does not bear meaning, which is true for formalized text and all kinds of names. Do not use this data type for attributes intended to contain free text. Scarcely any attribute will be declared directly as a Character String.	ST
Encapsulated Data	ED	Can convey any data that is primarily meant to be shown to human beings for interpretation. Whenever attributes should contain text entered and shown to users, the Encapsulated Data type should be used, not the plain character string type. Encapsulated Data can be any kind of text, whether unformatted or formatted written language or other multi-media data. Instead of the data itself, an ED may contain only a reference (URL.)	TX, FT, ED, RP
Instance Identifier	II	Used to uniquely identify some individual entity, a piece of data or a real world entity. Examples are medical record number, placer and filler order id, service catalog item number, etc.	ID, IS, CE, HD, EI
Telecommunication Address	TEL	A telephone number or e-mail address specified as a URL. In addition this type contains a time specification when that address is to be used, plus a code describing the kind of situations and requirements that would suggest that address to be used (e.g., work, home, pager, answering machine, etc.)	TN, XTN
Code Value	CV	Exactly one symbol in a code system. The meaning of the symbol is defined exclusively and completely by the code system that the symbol is from. Used primarily for technical concepts, concepts which is crucial to HL7 operations, and concepts which are defined or adopted under the discretion of HL7.	ID, CE
Concept Descriptor	CD	A descriptor for a real world ("natural") concept, such as a finding, a diagnosis, or of any semantic field, that is not under the sole discretion of HL7. A given concept may be expressed in multiple terms where each term is a translation of some other term, or is a (re-)encoding of the original human readable text, that can also be sent in this data type. This data type is suitable for multi-axial code systems.	CE
Integer Number	INT	Integer numbers are precise numbers that are results of counting and enumerating. Integer numbers are discrete, the set of integers is infinite but countable. No arbitrary limit is imposed on the range of integer numbers. Two special integer values are defined for positive and negative infinity.	NM
Real Number	REAL	Fractional numbers as approximations to real numbers. Fractional numbers occur whenever quantities of the real world are measured or estimated, or where quantities are the result of calculations based on other floating point numbers. This type preserves the precision in terms of significant digits.	NM
Physical Quantity	PQ	A dimensioned quantity expressing the result of a measurement. Consists of a floating point value and a physical unit. Physical Quantities should be preferred instead of two attributes expressing a number and a unit separately. Physical quantities are often constrained to a certain dimension. This can be by specifying some unit representing the dimension (e.g. m, kg, s, kcal/d, etc.)	CQ
Monetary Amount	MO	The amount of money in some currency. Consists of a value and a denomination (e.g., U.S.\$, Pound sterling, Euro, Indian Rupee.)	MO
Point in Time	TS	A scalar defining a point on axis of natural time.	TS

Ratio	RTO	A ratio quantity is a quantity that comes about through division of any numerator quantity with any denominator quantity (except zero.) Ratios occur in laboratory medicine as "titers", i.e., the maximal dissolutions at which an analyte can still be detected. Other Ratios are price expressions, such as dollar per gram.	SN
Postal and Residential Address	AD	The main use of such declared data is to be printed on mailing labels (postal address,) or to allow a person to physically visit a location (residential address.) The difference between postal and residential address is whether or not there is just a post box.	AD, XAD
Person Name	PN	Used for one full name of a natural person. Names usually consist of several name parts that can be classified as given, family, nickname etc. This data type is intended to be used only in the Person_name class. Instead of directly using this data type for an attribute of another class, one should consider drawing an association to the Person_name class.	PN, XPN
Organization Name	ON	Used to name an organization. Similar but simpler than the name of a natural person.	XON

Generic (Parameterized) Data Types

Set Collection	SET<t>	A collection of values of any type T without a specifying an order among the elements.	
List Collection	LIST<t>	An ordered set of values of any type T.	
Bag Collection	BAG<t>	An unordered set of values of any type T where each value can occur more than once (rare.)	
Interval	IVL<t>	Ranges (intervals) of values of type T. An interval is a set of consecutive values of any ordered data type, such as, integer, floating point, point in time, physical quantity, monetary amount, and ratio.) Intervals should be preferred instead of two attributes expressing a start and an end separately.	SN, XNM

Note that some data types that existed in HL7 version 2 no longer exist in version 3. Many of the old composite types, such as CN, contain multiple concepts, and are now represented more explicitly in the information model as either attributes or classes. Other types, such as ID, IS, and CE, received a more rigorous definition so that an automatic 1:1 mapping is often not possible. Other types, such as PN have been promoted to an information model class.

2.2 The Service Action Centered View

Healthcare is a series of intentional actions (or “services”) that are performed to benefit patients. Actions occur within a context of who, whom, where, when, how, and why. Actions in human language are verbs that unite all the nominal phrases, the actor (nominative), the targets (accusative), and beneficiaries (dative). Where the nominal entities contribute most of the information content of a sentence, the one essential key to the meaning of the sentence is the verb.

For example, “Dr. Smith examines Mrs. Doe,” represents the action *to examine*, with Dr. Smith as actor and Mrs. Doe as target. “MicroLab tests a specimen of Mrs. Doe” is another action *to test*, with “MicroLab” as actor, and specimen as direct object.

Any representation of an action should identify the kind of action (*what happens*), the actors *who* accomplish to the action, the objects or targets *whom* the action influences. Adverbs of location (*where*), time (*when*), manner (*how*), and other information about circumstances, such as reasons (*why*) or motives (*what for*) are additional pieces of information that may be required or optional in given situations.

2.2.1 Attributes of class Service

2.2.1.1 Service.id : SET{II}

This is an instance identifier of a particular Service object. For example, whenever a service is carried out, there is a new service object instantiated with an identifier that uniquely distinguishes this service object from every other service object.

2.2.1.2 Service.mood_cd : SET{CV}

Webster's dictionary defines mood as a "distinction of form [...] of a verb to express whether the action or state it denotes is conceived as **fact** or in some other manner (**as command, possibility, or wish**)" This definition of mood can be directly applied to the USAMP-II model, where the service action (corresponding to a *verb* in natural language) may be conceived as an event that happened (fact), an ordered service (command), a possible service (master), and a goal (wish) of health care. The mood code is critical to the design of this model. Without the mood_cd, the model above would be at least three times as big, in order to distinguish service events, from orders, schedules, goals, and master service items.

One of the "infinitive" moods is used for describing potential services that can have actual services associated with them. Common use of the infinitive mood is for dictionary entries (so called "master service") and all "knowledge" links (e.g., possible reason, cause, manifestation, etc.) Other special infinitives are goal and trigger mood. A goal describes a wish for a certain outcome (typically an observation) to be achieved in the future. An observation in goal mood is not actually made, thus is an infinitive. Goals are evaluated later. Triggers are service descriptions that can match actual services (like a query.) When a trigger matches, it enables, suggests, or demands the associated action to be performed. Triggers are most often used to fully describe PRN medication orders, but can serve to build reminder systems too.

Main Entry: ²**mood**

Function: *noun*

Etymology: alteration of ¹mode

Date: 1569

1 : the form of a syllogism as determined by the quantity and quality of its constituent propositions

2 : distinction of form or a particular set of inflectional forms of a verb to express whether the action or state it denotes is conceived as fact or in some other manner (as command, possibility, or wish)

3 : MODE 1b

Exhibit 2: Webster's definition of "mood".

Table 2: Service action moods – the boldfaced moods are the conventional moods of HL7 v2.¹

Concept	Implies	Code	Definition
event		EVN	A service that actually happens, may be an ongoing service or a documentation of a past service
order		ORD	An order for a service (has at least two actors: placer and filler)
intent		INT	An intention or plan to perform a service (like an order but only one actor)

¹ **How to read the vocabulary tables in this document?** We mostly define vocabulary in a table with four columns. **1. Concept** lists a short name or phrase for each concept. **2. Implies** refers to another concept (row) in the table which is understood as a generalization of this row's concept (e.g., if a cat *is an* animal, the presence of a cat *implies* the presence of an animal.) The hierarchy established by the implies column is also visualized through indenting the concept names. **3. Code** is a short one to 5-letter abbreviation (all caps) useful for computer communication. **4. Definition** seeks to define the concept.

schedule	intent	SCH	A planned or scheduled service instance
give notice	schedule	RXG	A reminder notice to administer a medication
goal		GOL	The goal that the service predicate may apply in the future (usually of observations)
control setting	goal	CTR	A preset value of a feedback auto-control device, e.g., the temperature set on thermostat, minute volume set on an SIMV ventilator, etc.
risk		RSK	The risk that some service predicate might apply (usually observations)
option		OPT	An option, alternative or preference for a set of parameters of the service from the point of view of the actor specified. May be the (ordering) physician's preferences or the patient's preferences. Services in option mood are not complete, they depend on a main service for which they are defined. Usually options come in a group among which a preference ranking can be specified.
requirement		REQ	A suggested consequence, a service that should be performed if the associated criteria (see below) are true. A requirement is often like a suggested order (e.g., if patient is on K ⁺ sparing diuretics suggest potassium controls.)
definition		DEF	A definition of a service ("master")
reference		REF	Service reference values, such as "normal" values
In addition the following flag can be set to turn the statement of event, intent, or goal into a predicate.			
criterion		CRT	A criterion or condition that must apply for an associated service to be considered
trigger	criterion	TRG	A condition or criterion that, if it applies triggers a consequential action. May be evaluated asynchronously (to specify alerts or reminder)

2.2.1.3 Service.type_cd : CD

A code for the *kind* of action (e.g., physical examination, serum potassium, etc.), used to be called "universal service identifier". The Service.type_cd specifies the service conceptually by using a code from a code system. We often refer to the Service.type_cd as the "name" of the Service. In any case, the type_cd or "name" is a handle on the concept of the action, not on the individual action instance.

Different code systems cover different kinds of services, which is why there is not one single code system to be used for the Service.type_cd. Furthermore, the data type Concept Descriptor (CD) allows the action to be named by multiple code systems at the same time, whereby each term from a coding system is assumed to be a synonym. For example, a Thrombectomy service may be named "34001" using the CPT-4 code, "P1-30322" in SNOMED, or "38.00" in ICD-10-PCS.

2.2.1.4 Service.descr : ED

The description of a service is a piece of free text or multimedia data that describes the service in all necessary detail. There is no restriction on length or content imposed on the description attribute. However, the content of the description is not considered part of the functional information communicated between systems. Descriptions are meant to be shown *to specially interested* human individuals. All information relevant for automated functions *must* be communicated using the proper attributes and associated objects.

Note that the description attribute is not a service "name." All names of the service can be communicated in the Service.type_cd attribute as codes together with readable print-names.

As with any attribute of class Service, the meaning of the description attribute is subject to the Service.mood_cd. For service definitions, the description can contain textbook like information about

that service. For service orders, the description will contain particular instructions pertaining only to that order. Filler order systems must show the description field to a performing provider.

For Service records of actual services (Service.mood_cd = actual,) the description is an important part of the documentation. The description will contain textual reports on the service. This is true for any service, in particular for surgical procedures, where the description attribute is the place to put the entire surgery report. If the surgical procedure is reported as multiple interrelated Service instances, each instance may contain the part of the report pertinent to that step of the procedure. **However, there is no need to break a service report apart into sub-services *only* to break the textual report apart into multiple sections.** The Encapsulated Data type is capable of handling formatted textual reports in HTML, PDF, or word processor formats. In addition, the HL7 PRA working group defines standards to use XML as a markup language for report documents.²

Note that textual reports should always be sent in the Service.descr; this includes reports of observation services. The Observation.value field is reserved for information that is processed automatically and that is accessible to automated processes. Human authored free text reports are not easily accessible to automated processing, which is why they should be communicated in the Service description attribute. Of course, free text documents can be analyzed by natural language parsers and similar tools. We encourage that any output of such natural language parsers be communicated in the Observation.value attribute in the form of structured machine accessible data. Since narrative text and observation value are in different attributes, they can be communicated together, without interfering with each other.

Table 3: Meaning of the description attribute depending on the mood of the service.

Mood		Meaning of the Description
Code	Concept	
EVN	event	Textual report of what has been done in the service, what noteworthy events happened, and what results have been achieved.
ORD	order	Instructions of what the filler is supposed to be doing, necessary rationale and caveats.
INT	intent	Notes about what is intended to be done, necessary rationale and caveats.
DEF	definition	Textbook-like description of the service.

for other moods the description attribute has analogously variant meaning.

2.2.1.5 Service.status_cd : CV

The state of the action (e.g., newly ordered, in process, completed.) The state is communicated in coded form. The codes are strictly defined by the state-transition model of a service class. No alternative coding system can be used for the status_cd attribute (CNE, coded no exceptions.)

No particular State-Transition model has been included in this specification as of yet. Various state transition models have been proposed in the orders committee in the past (although these have never been reconciled.) This proposed model understands that, in principle, it must (and can) accommodate any state-transition logic that has been defined using the previous model. In particular, the extensive

² The role of the PRA work in this context is not quite clear: is PRA defining formats for chiefly narrative information or does PRA seek to structure information to be readily computer-processible? Clearly, the PRA started out to provide a format for clinical narrative and not to force information into a predefined semantic structure, according to the motto "First Do No Harm" (L. Alschuler.) What is not clear yet I where the PRA project will end up. At this point, however, PRA can support only narrative text forms, though in a shareable structure, plus some indexes to help in accessing and interpreting the text. Therefore, we consider PRA documents a kind of text not a computer-accessible value.

merging of classes and the mood code attribute do not impact the scope and definition of currently proposed state-transition models.

2.2.1.6 Service.total_time : GTS

This is the time when the action happened, is ordered or scheduled to happen, or when it can possibly happen (depending on the mood of the Service object.) The timing of actions is a very important concept that will be explained in greater detail in Section 2.5.3 below.

2.2.1.7 Service.critical_time : GTS

This is the “biologically relevant” time of an action. The concept is best understood with observations, where the time of the observation action 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. Even more so in history taking, when the doctor records an episode of Hepatitis A under which the patient suffered last year for several weeks. For surgical procedures the time between first cut and last suture is taken as the critical time of the procedure. For transport and supply services the critical time is the time en route or time of delivery respectively. Critical time and total time of a service may often be related in a certain way, which will be discussed below (cf. Figure 10.)

2.2.1.8 Service.method_cd : CD

For any service there may be several different methods to achieve by and large the same result, but may be important to a more thorough interpretation (e.g., blood pressure method: arterial puncture vs. Riva-Rocci, sitting vs. supine position, etc.)

Method concepts can be “pre-coordinated” in the Service definition, so that there is never an option to select different methods. Pre-coordinating methods into the service code (type_cd) avoids having to standardize on method codes. There are so many possible methods which all depend heavily on certain kinds of services, so that defining a vocabulary domain of all methods is close to impossible. The pre-coordinated approach avoids relying on the impossible to be done.

However, a code system might be designed such that it specifies a set of available methods for each defined service concept. Thus, a user ordering a service could select one of several variances of the service by means of the method code. Available method variances may also be defined in a master service catalog for each defined service. In service definition records (Service.mood_cd = DEF) the method_cd attribute is a set of all available method codes that a user may select while ordering, or expect while receiving results.

Although the authors of this proposal believe that the pre-coordinated approach to methods goes a long way and should be followed as far as possible, the same information structure can handle both the pre-coordinated and the post-coordinated approach.

2.2.1.9 Service.body_site_cd : CD

Most health care services have a focus on a particular anatomic structure of the patient (the “target” of service.) This information is found in body_site_cd. The coding system to be used for anatomic site is not specified in detail. Anatomic sites, body parts, and functional body systems are huge and highly complex domains that require a very sophisticated terminology system. Candidates are Galen,

SNOMED, or Read codes. Alternatively, a simple local coding system can be used to identify exactly the common body sites used.

Some body sites can also be “pre-coordinated” in the Service definition, so that there is never an option to select different body sites. The same information structure can handle both the pre-coordinated and the post-coordinated approach.

For administrative body sites (i.e. where medications are administered) HL7 used to define a table (0163) that must be used as defined in Table 4 below.

Table 4: Body site concepts (from HL7 v2.3 table 0163)

Concept	Code	Concept	Code	Concept	Code
Bilateral Ears	BE	Left Mid Forearm	LMFA	Right External Jugular	REJ
Bilateral Eyes	OU	Left Naris	LN	Right Eye	OD
Bilateral Nares	BN	Left Posterior Chest	LPC	Right Foot	RF
Buttock	BU	Left Subclavian	LSC	Right Gluteus Medius	RG
Chest Tube	CT	Left Thigh	LT	Right Hand	RH
Left Arm	LA	Left Upper Arm	LUA	Right Internal Jugular	RIJ
Left Anterior Chest	LAC	Left Upper Abd Quadrant	LUAQ	Rt Lower Abd Quadrant	RLAQ
Left Antecubital Fossa	LACF	Left Upper Forearm	LUFA	Right Lower Forearm	RLFA
Left Deltoid	LD	Left Ventragluteal	LVG	Right Mid Forearm	RMFA
Left Ear	LE	Left Vastus Lateralis	LVL	Right Naris	RN
Left External Jugular	LEJ	Nebulized	NB	Right Posterior Chest	RPC
Left Eye	OS	Perianal	PA	Right Subclavian	RSC
Left Foot	LF	Perineal	PERIN	Right Thigh	RT
Left Gluteus Medius	LG	Right Arm	RA	Right Upper Arm	RUA
Left Hand	LH	Right Anterior Chest	RAC	Right Upper Abd Quadrant	RUAQ
Left Internal Jugular	LIJ	Right Antecubital Fossa	RACF	Right Upper Forearm	RUFA
Left Lower Abd Quadrant	LLAQ	Right Deltoid	RD	Right Vastus Lateralis	RVL
Left Lower Forearm	LLFA	Right Ear	RE	Right Ventragluteal	RVG

2.2.1.10 Service.interpretation_cd : SET(CV)

This attribute allows for a very rough interpretation of the course or outcome of a service action. This is sometimes called “abnormal flags”, however, the judgement of normalcy is just one of the common rough interpretations, and is often not relevant. For example, for the observation of a pathologic condition, it doesn’t make sense to state the normalcy, since pathologic conditions are never considered “normal.” The following concepts are defined for the interpretation code:

Table 5: Interpretation codes.

Code	Implies	Definition
Normality, Abnormality, Alert. At most one allowed.		
N		Normal (for all service types)
A		Abnormal (nominal observations, all service types)
L	A	Below low normal (quantitative observations)
H	A	Above high normal (quantitative observations)

AA	A	Abnormal alert (nominal observations and all service types)
LL	L, AA	Below lower alert threshold (quantitative observations)
HH	H, AA	Above upper alert threshold (quantitative observations)
Severity of conditions (diagnoses or allergies.) At most one allowed.		
MI		Mild
MIMO		Mild to moderate
MO		Moderate
MOSV		Moderate to severe
SV		Severe
MISV		Mild to severe (this form is used indeed even though it spans the entire range)
Change of quantity and/or severity. At most one of B or W and one of U or D allowed.		
B		Better (of severity or nominal observations)
W		Worse (of severity or nominal observations)
U		Significant change up (quantitative observations, does not imply B or W)
D		Significant change down (quantitative observations, does not imply B or W)
Microbiology: interpretations of minimal inhibitory concentration (MIC) values. At most one allowed.		
R		Resistant
I		Intermediate
MS		Moderately susceptible
S		Susceptible
VS		Very susceptible
Technical exceptions. At most one allowed. Does not imply normality or severity.		
<		Below absolute low-off instrument scale (does not imply LL or L)
>		Above absolute high-off instrument scale (does not imply HH or H)

2.2.1.11 Service.confidentiality_cd : SET(CV)

This is a code that limits the disclosure of information about this service. The codes refer to confidentiality policies as listed in the following table:

Table 6: Confidentiality policies

Concept	Code	Definition
By accessing subject / role and relationship based rights (one and only one)		
normal	N	Normal confidentiality rules (according to good health care practice) apply, that is, only authorized individuals with a medical or business need may access this item.
clinician	D	only clinicians may see this item, billing and administration persons can not access this item without special permission.
restricted individual	R	Restricted access, i.e. only to providers having a current care relationship to the patient.
low	L	Access only to individual persons who are mentioned explicitly as actors of this service and whose actor type warrants that access (cf. to actor type code.)
		No patient record item can be of low confidentiality. However, some service objects are not patient related and therefore have low confidentiality.

business	B	Since the service class can represent knowledge structures that may be considered a trade or business secret, there is sometimes (though rarely) the need to flag those items as of business level confidentiality. However, no patient related information may ever be of this confidentiality level.
----------	---	---

Modifiers of role based access rights (multiple allowed)		
sensitive	S	Information for which the patient seeks heightened confidentiality. Flag can be set or cleared on patient's request. Sensitive information is not to be shared with family members. Information reported by the patient about family members is sensitive by default.
taboo	T	Information not to be disclosed or discussed with patient except through physician assigned to patient in this case. This is usually a temporary constraint only, example use is a new fatal diagnosis or finding, such as malignancy or HIV.
celebrity	C	Celebrities are people of public interest including employees, whose information require special protection.
By information type, only for service catalog entries (multiples allowed) Not to be used with actual patient data!		
HIV		HIV and AIDS related item
PSY		Psychiatry related item
ETH		Alcohol/drug-abuse related item
SDV		Sexual assault / domestic violence related item

Confidentiality policies may vary from institution to institution and not all systems are capable of abiding by all details of the confidentiality policies suggested in the above table. However, these are the items that are being used in some institutions and which implementers may want to consider supporting.

It is important to note that good confidentiality of the medical record can *not* be achieved through confidentiality codes only to filter out individual record items to 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 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.

A similar confidentiality code attribute is therefore required in the Patient class to cover the entire patient record. But this is outside the scope of the present proposal.

The flags HIV, PSY, ETH, and SDV may only be used on service items that are not patient related. Typically, they are used in the service definition object ("master" service) to indicate a generic disclosure policy of any actual service item of that type.

Aggregations of data may assume the privacy level of the most private action in the aggregation.

2.2.1.12 Service.max_repeat_nmb : INT **default: 1**

This is the maximum number of repetitions of a service. Typical values are 1, some other finite number, and the infinity (a specific null value for numbers.) See the discussion on service plans below on how specifically this is used.

2.2.1.13 Service.interruptible_ind : BL **default: true**

Indicates whether a service is interruptible by asynchronous events (such as “through”-conditions to turn false, or time running up.) See discussion on activity plans below.

2.2.1.14 Service.substitution_cd : CV **default: N**

Indicates whether an ordered or intended service may be or has been substituted for a different service. The fact that the actual service differs from the planned or ordered service, and the details of the variance can be seen by comparing the service as planned or ordered from the service as performed. Both service records should be sent in a message where this difference is important. The Service.substitution_cd attribute is mainly used in an order, to specify whether an ordered service may be substituted and in what way it may be substituted.

Table 7: Service substitution code

Concept	Code	Definition
no	N or 1	No substitution happened (no substitution allowed.)
generic	G	A generic has been (may be) substituted for a brand product.
therapeutic	T	A therapeutic substitution was done (is permitted.)
no selection	0	No product selection indicated, i.e. this is a generic (non-brand) order / service.
patient	2	Substitution on patient request.
actor	3	Substitution selected by actor (e.g. pharmacist.)
out of stock	4	Substitution because requested product or generic is not in stock.
brand as generic	5	Substitution of a brand as a generic.
brand law	6	No substitution, brand product mandated by law.
unavailable	7	Substitution because product not available in the marketplace.

2.2.1.15 Service.priority_cd : SET{CV} **default: {R}**

This attribute encodes the urgency under which the service is to be scheduled and performed, or was performed. This attribute is used in orders to indicate the ordered priority. It is also used in the service event documentation to indicate the actual priority used to perform the service, which is used to determine the charge. In master service definitions it indicates the available priorities.

Table 8: Service priority code.

Concept	Implies	Code	Definition
stat		S	With highest priority (e.g., emergency.)
ASAP		A	As soon as possible, next highest priority after stat.
routine		R	Routine service, do at usual work hours.
preop		P	<i>the use of this code is unclear, it is left here only to not break backwards compatibility without need. Deadline for a definition is September 17, 2000, after that date, this code will be removed.</i>
callback for scheduling		CS	Filler should contact the placer (or target) to schedule the service. (Was “C” in HL7 version 2.3’s TQ-priority component.)
callback placer for scheduling	callback for scheduling	CSP	Filler should contact the placer (or target) to schedule the service. (Was “C” in HL7 version 2.3’s TQ-priority component.)
contact recipient for	callback for scheduling	CSR	Filler should contact the service recipient (target) to schedule the service. (Was “C” in HL7 version 2.3’s TQ-priority component.)

scheduling		
timing critical	T	It is critical to come as close as possible to the requested time (e.g., for a trough antimicrobial level.)
as needed	PRN	an “as needed” order should be accompanied by a description of what constitutes a need. This description is represented by an observation service predicate as a precondition.
callback results	CR	Filler should contact the placer as soon as results are available, even for preliminary results. (Was “C” in HL7 verion 2.3’s reporting priority.)
rush reporting	RR	A report should be prepared and sent as quickly as possible.

2.2.1.16 Service.orderable_ind : BL

default: true

This attribute indicates whether this service can be requested independently from other services. Some services can only occur as subordinate to a super-service; others are abstractions of services or service groups that should not be ordered in one piece. Since in principle everything that can be done can potentially be requested, this attribute is *true* by default. It is usually only used for master file definitions.

2.3 Actors and targets

All people, things and locations involved in a Service (or for scheduling purposes “all resources of an activity”) are associated with the Service as either actors or targets. Actors are mostly professional provider personnel, but also the patient (for self-administered services,) or a next of kin. Targets are the recipients of care (e.g., patient), but also consumable material, durable equipment, locations, and anything that is needed or of notable presence in the service.

2.3.1 Actor

Actors can participate in an action in different ways. For example, primary surgeon, assistant surgeon, sterile nurse, and nurse assistant are all actors in a surgical procedure, who are more or less immediately involved in the action. However, payers, supervisors, provider organizations (e.g., “MicroLab”) and their delegates may be actors too, even though they might not be individual persons who have their “hands on” the action. The patient himself is a performing actor in self-care procedures (e.g. fingerstick blood glucose, insulin injection, etc.)

The Stakeholders, people and organizations that can be actors and targets of a service action are not shown in the model in Figure 1. Those classes are left untouched in their present definition by the most recent RIM. The intention is to allow Actor bindings only to entities that are capable of and accountable for their independent decisions. Capability of independent decision and accountable usually applies only to persons under the law, including both organizations and natural (human) persons. This “legal person” as a subject of legal rights and obligations is a very obvious interpretation of the RIM Stakeholder construct (it is a well-known legal notion.)

The notion of multiple actors with specific functions touches and partially overlaps on two “sides” with related concepts of the RIM, and understanding the distinctions is important to use the RIM constructs correctly. On the one “side” actor functions look similar to Stakeholder roles (e.g., healthcare practitioner, guarantor, contact-person,) and capability and certification (e.g., certified surgeon vs. resident, certified nurse midwife vs. other midwife practitioner, registered nurse vs. other nurse practitioner.) The professional credentials of a person may be quite different from what a person actually does. The most common example is interns and residents performing anesthesia or surgeries under (more or less) supervision of attending specialists. The opposite example is people who are both

medical doctors and registered nurses and who perform the function of a nurse. Thus, roles and certification refer to the static capabilities of a person (person-related) while actors and Actor.type_cd refer to the particular function an actor played in the service (activity-related.)

On the other “side” the actor concept interferes with sub-activities. Whenever multiple actors are involved in a service, 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 modeled as a service consisting of sub-services (through the Service_relationship class,) where each service 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 services: (a) the consent, (b) the surgery proper, and (c) the anesthesia service in parallel to the surgery. If we used the sub-services, the consentor, surgeon and anesthetist could simply be of actor type “performer.” Thus the more sub-services we use, the fewer different actor types need to be distinguished, and the fewer sub-services we use, the more distinct actor types we need.

Note that the perception of a task as “atomic” or “composite” (of sub-tasks) depends on local business rules and may differ from department to department. In principle, every task can be thought of as being a composite of sub-tasks. We thus say that actions are “fractal.” The paradigmatic example of the fractal nature of activities is a “robotic arm” doing some simple action as reaching for a tool in front of it. The seemingly simple activity of the robotic arm decomposes into complex control and coordination procedures and movements, action of separate motors and switches, etc. (We sometimes use the key-phrase “robotic arm discussion” to recall the fractal nature of actions, since this example has been brought up over and over again, independently by different people.)

As a rule of thumb, sub-tasks should be considered instead of multiple actors when each sub-task requires special scheduling, or billing, or if overall responsibilities for the sub-tasks are different. In most cases, however, human resources are scheduled by teams (instead of individuals,) billing tends to lump many sub-tasks together into one position, and overall responsibility often rests with one attending physician, chief nurse, or head of department. This model allows both the multi-actor and the multi-service approach to represent the business reality, with a slight bias towards “lumping” minor sub-activities into the overall service.

2.3.1.1 Actor.type_cd : SET(CV)

Identifies a particular function or a set of functions that a person performs in the Service. Defined actor types are:

Table 9: Actor types

Concept	Implies	Code	Definition
performer		PRF	A person who actually and principally carries out the service. Need not be the principal responsible actor, e.g. a surgery resident operating under supervision of attending surgeon, and may be the patient in self-care, e.g. fingerstick blood sugar.
supervisor		SPV	A person who is legally responsible for the service (event, order, etc.) carried out by a performer as a delegate. A supervisor is not necessarily present in an action, but is accountable for the action through the power to delegate, and the duty to review actions with the performing actor after the fact (e.g. head of a biochemical laboratory.)
assistant		ASS	A person assisting in a service through his substantial presence and involvement (excludes mere advisorship.)
advisor		ADV	A person providing advice as to whether or how to perform a service without being

			substantially present in the service (and without being supervisor.)
technician	TEC		A person who carries out technical tasks, e.g., preparation and analysis of specimen.
witness	WIT		A person witnessing the action happening without necessarily doing anything.
placer	PLC		The placer of an ordered service, which may be an organization or a person, such as a physician or nurse.
filler	FIL		The (designated) filler of an ordered service. The filler of the ordered service is supposed to become the performer of the actual service. The filler is often a department (organization) such as lab, pharmacy, or nursing, but may also be an individual person such as a specific consulting physician or a patient.
originator	ORG		A first-hand source of reported information.
informant	INF		A second-hand source of reported information (e.g., in history taking.)
data entry	ENT		A person entering the data into the originating system.
clarifier	CLQ		A contact (often not individual) to whom immediate questions for clarification should be directed (e.g., a care facility to be called by phone number.)
consenter	CNS		The person giving consent to the service (usually the patient himself or a legal guardian.) A consenting person is an actor in the sense of asking or delegating an action to happen upon himself.
consent obtainer	CSO		The healthcare practitioner obtaining an informed consent from the patient (or legal guardian.) This includes explaining the procedure, rationale, risks, and alternatives, answering questions, and making sure the patient has understood the issue. (If consents are modeled as an associated service, the "consent obtainer" would be the "performer" of the consent service.)
attester	ATT		A person attesting to the service as represented.
verifier	VER		A person who verifies the correctness and appropriateness of the service (plan, order, event, etc.) and hence takes on accountability.
tracker	TRC		A person who receives copies of exchange about this service (e.g., a primary care provider receiving copies of results as ordered by specialist.)
referrer	REF		A person having referred the subject of the service to the performer (referring physician.) Typically, a referring physician will receive a report.
reviewer	REV		A person who is to review the details of a service (order or documentation) after the fact.
patient reviewer	PRV		A person reviewing the outcome of a service with a patient (or legal guardian.)
attending	ATT		The attending physician for a patient (encounter) is a well-known role in some health care systems (notably in the U.S.) While attending is an "encounter practitioner" type, the notion of encounter is often ambiguous between institutions. Hence, attending is modeled as a service-related function of an actor too.
escort	ESC		For patient transports a person who escorts the patient.

Typical actor types in surgical procedures

primary surgeon	performer	SPR	In a typical surgery setting the primary performing surgeon.
first assistant	assistant	SA1	In a typical surgery setting the assistant facing the primary surgeon. The first assistant performs parts of the operation and assists in others (e.g., incision, approach, electrocoagulating, ligatures, sutures.)
second assistant	assistant	SA2	In a typical surgery setting the assistant who primarily holds the hooks.
sterile nurse	assistant	SNI	In a typical surgery setting the nurse in charge of the instrumentation.
third assistant	assistant	SA3	In a typical surgery setting there is rarely a third assistant (e.g., in some Hip operations the third assistant postures the affected leg.)
nurse assistant	witness	SNA	In a typical surgery setting the non-sterile nurse handles material supply from the stock, forwards specimen to pathology, and helps with other non-sterile tasks (e.g., phone calls, etc.)

Typical actor types in anesthesia			
anesthetist	performer	ANS	In a typical anesthesia setting an anesthesiologist or anesthesia resident in charge of the anesthesia and life support, but only a witness to the surgical procedure itself. To clarify responsibilities anesthesia should always be represented as a separate service related to the surgery.
anesthesia nurse	assistant	ANN	In a typical anesthesia setting the nurse principally assisting the anesthesiologist during the critical periods.
Other health care functions			
midwife		MWF	A person (usually female) helping a woman deliver a baby. Responsibilities vary by locale, ranging from a mere assistant to a full responsibility for (normal) births and pre- and post-natal care for both mother and baby.

Note that the Actor.type_cd designates the actual function performed in the service. This is quite different from a role associated with a person or a profession- or specialty-code, although, some of the Actor.type_cd values may suggest that they are the same. While a person's role, a profession code, or a specialty code may signify a general capability and authority of that person, an Actor.type_cd rather represents a part or sub-task of the associated Service activity.

Most notably the role "performing surgeon" is not necessarily played by a certified surgeon, but in many cases by a resident (in which case an attending surgeon is designated as the "responsible" actor.) The same is true for the "anesthetist" which doesn't have to be an "anesthesiologist" and will in most cases be a resident or sometimes even a nurse.

An actor can do multiple of such functions identified by the Actor.type_cd at the same time. This can be represented in two ways: either the same Actor instance can be used with multiple Actor.type_cd values (hence, it's a set of code values). Alternatively, one can use multiple Actor-instances with just one Actor.type_cd value relating to the same Stakeholder. A mixture of both approaches is also possible. The rationale when to use just one or the other approach can be, e.g., an actor playing three roles two of which are closely related whereas the third role may have a different time range. A more concrete ruling on a standardized use may follow in the future.

2.3.1.2 Actor.tmr : IVL(TS)

This attribute can specify the time range during which the associated person was an actor of the specified Actor.type_cd in the associated service. This may be needed when the actor's involvement spans only part of the service, and if this fact is worth mentioning. The Actor.tmr does not need to be used in cases where this detail is irrelevant.

2.3.1.3 Actor.note_txt : ED

An actor can make a comment about this service item in the note attribute.

2.3.1.4 Actor.signature_cd : CV

Some Actors must provide a signature on the service instance. For example, a procedure report requires a signature of the performing and responsible surgeon. Or a consent requires the signature of the consentor. This attribute allows to specify whether or not such a signature is on file, it does not itself provide evidence for the signature.

Table 10: Actor signature code

Concept	Code	Definition
required	X	A signature for the service is required of this actor.
signed	S	A signature for the service is on file from this actor.

In today's environment, with the advent of digital signatures, this treatment appears to be insufficient. We will continue to work on integrating this to a framework of digital signatures. However, there are severe technical obstacles to overcome: digital signatures do not exist over abstract information. Only concrete bit-representations of information can be signed. Since HL7 version 3 tries to separate abstract information from bit-encodings, it is not clear how such a digital signature could exist.

We are aware of the X.509 approach of Distinguished Encoding Rules (DER), but there is currently no definition for encoding HL7 data structures in DER, nor does it seem like the industry prefers DER as the principle message encoding rules. Furthermore, there needs to be a framework to integrate traditional paper-based signatures as well. Hence, we acknowledge that the Actor class may be the principle point of implementing electronically authenticated medical records, but we defer the elaboration of this approach to later.

2.3.2 Target

Targets can be all physical entities, including humans, other living subjects and inanimate material. The Target class maintains a choice to link to either a Person or a Material as its substance. Again, the existing RIM classes around Stakeholder are not touched by this proposal at this time. The approach to material however is a novelty of this proposal and will be discussed further below.

2.3.2.1 Target.type_cd : SET{CV}

Just as with actors, different participation types can be identified for targets. By "target" of an action we basically mean objects of a verb. Objects appear in different cases: direct objects, indirect objects or adverbial objects according to their roles in a sentence. Target participation type codes distinguish those different roles. For instance, patient, guarantor, custodian, or family members are examples of target participation types. These are in the role of direct target on whom (accusative) or the indirect beneficiary (on behalf of whom) the service action is performed. In addition any material, specimen, device, or location used or produced by a service is a target of the service.

Table 11: Target types

Concept	Implies	Code	Definition
direct target		DIR	Target that is substantially present in the service and which is directly affected by the service action (includes consumed material, devices, etc.)
subject	direct target	SBJ	The principle target that the service acts on (e.g. the patient in physical examination, a specimen in a lab observation. Consumables, and devices used as tools for a service are not subjects. However, a device may be a subject of a maintenance service.)
beneficiary		BEN	Target on behalf of whom the service happens, but that is not necessarily present in the service. Can occur together with <i>direct target</i> to indicate that a target is both.
recipient	subject	REC	Target that receives an intervention. Usually a patient, but may be a family member (teaching) or a device or room (cleaning, disinfection, housekeeping.) Note: not all direct targets are recipients (e.g., consumables are direct targets but not recipients.)
donor	subject	DON	In some organ transplantation services and rarely in transfusion services a donor will be a target participant in the service. However, in most cases transplantation is decomposed in three services: explantation, transport, and implantation. The

			identity of the donor (recipient) is often irrelevant for the explanation (implantation) service.
patient		PAT	All <i>clinical</i> observation and intervention services have a patient target that is both, beneficiary and subject. In non-clinical (e.g., laboratory) observations, or in teaching or management services, the subject may be different (e.g., specimen, next of kin, etc.) and the patient is just the beneficiary. An organ donor in a transplantation service is a direct target without being a beneficiary. Thus, the patient target type requires specification of direct target, beneficiary or both.
mother	patient	MTH	In an obstetric service, the mother.
baby	patient	BBY	In an obstetric service, the baby.
specimen	subject	SPC	The subject of non-clinical (e.g. laboratory) observation services is a specimen.
proxy	subject	NOK	Someone who is the subject of the service on behalf of the patient. For example, a family member who is the subject of a teaching service in the patient's matters.
product	direct target	PRD	A material target that is brought forth (produced) in the service (e.g., specimen in a specimen collection, access or drainage in a placement service, medication package in a dispense service.) It doesn't matter whether the material produced had existence prior to the service, or whether it is created in the service (e.g., in supply services the product is taken from a stock.)
consumable	direct target	CSM	Target that is taken up, is diminished, and disappears in the service.
therapeutic agent		TPA	Something incorporated in the subject of a therapy service to achieve a physiologic effect (e.g., heal, relieve, provoke a condition, etc.) on the subject. In an administration service the therapeutic agent is a consumable, in a preparation or dispense service, it is a product. Thus, consumable or product must be specified in accordance with the kind of service.
device	direct target	DEV	Something used in delivering the service without being substantially affected by the service (i.e. durable or inert with respect to that particular service.) Examples are: monitoring equipment, tools, but also access/drainage lines, prostheses, pace maker, etc.
non-reuseable device	device	NRD	A device that changes ownership due to the service, e.g., a pacemaker, a prosthesis, an insulin injection equipment (pen), etc. Such material may need to be restocked after he service.
reusable device	device	RDV	A device that does not change ownership due to the service, i.e., a surgical instrument or tool or an endoscope. The distinction between reuseable and non-reuseable must be made in order to know whether material must be re-stocked.
payload	subject	PYL	For transportation services, the transported passenger or goods.
location		LOC	The facility where the service is done. May be a static building (or room therein) or a moving location (e.g., ambulance, helicopter, aircraft, train, truck, ship, etc.)
origin	location	ORG	The location of origin for transportation services. May be a static building (or room therein) or a movable facility (e.g., ship.)
destination	location	DST	The destination for transportation services. May be a static building (or room therein) or a movable facility (e.g., ship.)
via	location	VIA	For transportation services, an intermediate location that specifies a path between origin an destination.
remote	location	RML	Some services take place at multiple concurrent locations (e.g., telemedicine, telephone consultation.) The location where the principal performing actor is located is taken as the primary location (LOC) while the other location(s) are considered "remote."

2.3.2.2 Target.tmr : SET(CV)

This is the time range in which the associated person or thing was a target of the specified Target.type_cd in the associated service.

2.3.2.3 Target.awareness_cd : CV

For person targets indicates whether the associated patient or family member is aware of the service, and especially of the observation made. For example, a patient (or his next family members) may not be aware of a malignancy diagnosis, the patient and family may be aware at different times, and some patients may go through a phase of denial. For other than person targets this attribute is not applicable and shall not be valued.

Table 12: Awareness code

Concept	Code	Definition
full awareness	F	Target person is fully aware of the issue.
uninformed	U	Target person has not yet been informed of the issue.
denying	D	Target person has been informed about the issue but currently denies it.
partial	P	Target person is partially aware of the issue.
marginal	M	Target person is marginally aware of the issue.
incapable	I	Target person is not capable of comprehending the issue.

2.4 Action Structures

Consider a surgical procedure, e.g. a laparoscopic cholecystectomy, as a typical action in health care. This action obviously consists of many smaller actions that must occur in the right order and relation to each other. Preoperative preparation is a precondition. Anesthesia is conducted in parallel to the entire surgical component. The operation itself includes a sequence of steps, such as incisions, preparation of the gall bladder, ligation of the vessels, excision and extraction of the gall bladder, sutures and bandages. Close analysis reveals that even the simplest of actions can be split into smaller actions. We thus say that actions are “fractal.” The paradigmatic example of the fractal nature of activities is a “robotic arm” doing some simple action as reaching for a tool in front of it. The seemingly simple activity of the robotic arm decomposes into complex control and coordination procedures and movements, action of separate motors and switches, etc.

Because actions are fractal for everyone to keep track of all the sub-actions is neither possible nor desirable. Since healthcare is a collaborative process involving many different perspectives, the level of detail needed may not be the same for everyone. Notably there is, in principle, no way to come to a standardized consensus on the “right” level of detail to be assumed in all of HL7’s scope. For example, the working group on laboratory automation is interested in more procedural detail of laboratory tests and maintenance activities. Other working groups focusing on clinical medicine may not want to deal with all that technical detail, but may consider other detail (e.g., medical reasoning) as relevant. Yet, HL7 is to suit both equally valid perspectives, and HL7 should do its best to allow both perspectives to cooperate smoothly.

An information model must describe the most fine-grained level of detail needed by any customer of the data. For instance, the surgeon reports on every major milestone of his operation for communication with the next surgeon and the legal system, but the payer usually only wants to know about the cholecystectomy at the very top level. Since the detail level needed may vary, the model must incorporate a method of mapping between “atomic” actions and collections of sub-actions.

Analysis of action relationships also revealed the need to associate individual actions to collections of past actions, e.g. this test was performed because of the results of two earlier tests.

In the initial USAM proposal therefore introduced a general recursive association, the class Service_relationship. The Service relationship class is a recursive associative class with two

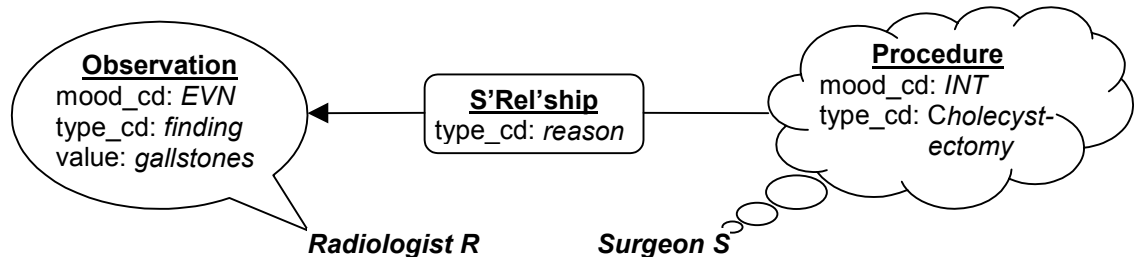


Figure 2: A surgeon intends to perform a cholecystectomy because there are gallstones found by a radiologist. The medical record must attribute the “reason” link properly to the surgeon so as to avoid ambiguity as if the radiologist has suggested the cholecystectomy. The rule of attribution is **that all service relationships are attributed to the responsible actor of the source service.**

associations to the Service class, one named “source” the other named “target”. Consider every Service_relationship instance an arrow with a point (headed to the target) and a butt (coming from the source.) For each relationship type the roles of source and target Service are different as specified in Table 13 below.

In principle the assignment of roles to each side of the relationship “arrow” is completely arbitrary. However since a service is the core element of a medical record, proper attribution of the medical record items is an important issue. The relationships associated with a Service are considered properties of the *source* service object. That means, that the originator of the information reported in a service object is not only responsible for the attribute values of that object, but also for all its outgoing relationships.

For example (Figure 2), consider a record item of an ultrasonography, authored by radiologist R, mentioning the presence of 4 gall stones about 1 cm in diameter, but no signs for acute inflammation of the gall bladder. The surgeon S reads this report and sees in it an indication for a laparoscopic cholecystectomy. S will be the author of the cholecystectomy service. So both, the cholecystectomy service and the indication will be authored by S. The indication link pointing to the ultrasonography service must not in any way be misunderstood as if the Radiologist was suggesting the cholecystectomy.

THE SERVICE RELATIONSHIPS ARE ALWAYS ATTRIBUTED TO THE RESPONSIBLE AUTHOR OF THE SERVICE AT THE SOURCE OF THE RELATIONSHIP (THE SOURCE-SERVICE)!

With this recursive service relationship one can group actions into “batteries,” e.g. LYTES, 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.

Actions may also be grouped longitudinal, in a sequence of sub-actions. Examples of such longitudinal grouping patterns include the phases of a clinical trial or the steps of the cholecystectomy outlined above. Actions may be explicitly timed, and may be conditioned on the status or outcome of previous actions. Concurrent collections of actions allow expressing logical branches as well as parallel tasks (tasks carried out at the same time.) These constructs can be organized in multiple layers of nesting, to support workflow management.

The relationship class is not only used to construct action plans but also to represent clinical reasoning or judgements about action relationships. Prior actions can be linked as the reasons for more recent actions. Supporting evidence can be linked with current clinical hypotheses. A flexible way of

managing problem lists consistent with the requirements addressed by Rector³ also uses the action relationship as its key component.

2.4.1 Attributes of class service relationship

2.4.1.1 Service_relationship.type_cd : CV

Determines the meaning of a relationship between two Services. This attribute is probably the most important attribute in this entire model besides the Service.mood_cd. It is a “structural” attribute inasmuch as each of its values implies specific constraints to what kinds of Service objects can be related and in which way.

The following table lists the concepts of the Service_relationship.type_cd attribute. For individual relationship type codes, the roles of source and target are specified in the table.

Table 13: Service relationship types⁴

Concept	Code	Implies	Definition	meaning of the service	
				source	target
has parts	PART		Collection of sub-services of any kind.	collection	element
has plan component	PLCM	PART	A collection of sub-services as steps or subtasks performed for the source service. Services may be performed sequentially or concurrently. See Section 2.5 below for detail.	plan	step
has list items	LELM	PART	Ordered collection of services of any kind. The sequence_nmb attribute must be filled.	list	element
has ingredient	INGR	PART	Allows expressing composition of medications from ingredients. Both, source (composite) and target (ingredients) are medications, where the ingredients need not necessarily be commonly ordered medications. Note: usually the class Material and Material_relationship should be used to express mixtures. However, for medical knowledge a composite substance (e.g. Cotrimoxazole or Tylenol III) can be thought of as a combined treatment service. The Medication class should be used mainly for generics while the detail of pharmaceutical products should be given through the Material class.	composite	ingredient
has species	SPEC		The generalization relationship (often ambiguously called “is-a”) can be used to express categorical knowledge about services. (e.g., <i>amilorid</i> , <i>triamterene</i> , and <i>spironolactone</i> medications are <i>potassium sparing diuretics</i> .)	genus	species

³ Rector AL, Nowlan WA, Kay S. Foundations for an electronic medical record. Meth Inform Med, 1991;30:179–86.

⁴ See footnote 1 on page 9 for a description of the vocabulary table convention. Here we use two additional convention in the **Implies** column: (1) an “exponent” –1 at the code in the “implies” column means that the currently defined concept implies the inverse meaning of the code in the “implies” column. Please refer to the description of the inversion_ind attribute about what inversion means and how it is used. (2) a minus sign in front of the code in the “implies” column indicates that the currently defined concept implies the logical negation of the concept referred to in the “implies” column. (e.g., contraindication is the negation of reason (“–RSON”).

has pre-condition	PRCN		A requirement to be met before a service is performed. The target can be any service in <i>criterion</i> mood. For multiple pre-condition a conjunction attribute (AND, OR, XOR) is applicable.	action	pre-condition
has trigger	TRIG	PRCN	A pre-condition that, if true would permit, suggest, or demand the source service (action) to be executed. The target is in <i>trigger</i> mood. A delay between the trigger and the triggered action can be specified.	action	trigger
has reason	RSON	PRCN	The reason or rationale for a service. A reason link is weaker than a trigger, it only suggests that some service may be or might have been a reason for some action, but not that this reason requires/required the action to be taken. Also, as opposed to the trigger, there is no strong timely relation between the reason and the action.	action	reason
suggests	SUGG	RSON ⁻¹	This is an inversion of the reason link, used to express recommendations or suggestions. For example, if the radiologist in the example of Figure 2 would have recommend the cholecystectomy, one would use the suggestion link.	reason	suggested action
has contra-indication	CIND	-RSON	A contraindication is just a negation of a reason, i.e. it gives a condition under which the action is not to be done. Both, source and target can be any kind of service, target service is in <i>criterion</i> mood. How the strength of a contraindication is expressed (e.g., relative, absolute) is left as an open issue. The priority_nmb attribute could be used.	action	contra-indication
has outcome	OUTC		An observation that should follow or does actually follow as a result or consequence of a condition or action (sometimes called "post-condition".) Target must be an observation as a <i>goal</i> , <i>risk</i> or any <i>criterion</i> . For complex outcomes a conjunction attribute (AND, OR, XOR) can be used. An outcome link is often inverted to describe an outcome assessment.	condition or action	outcome
has goal	GOAL	OUTC	A goal that one defines given a patient's health condition. Subsequently planned actions aim to meet that goal. Source is an observation or condition node, target must be an observation in <i>goal</i> mood.	observation, condition	goal
has objective	OBJV	OUTC	A desired outcome that a service action aims to meet. Source is a service, target must be an observation in <i>goal</i> mood.	service	goal
has risk	RISK	OUTC	A noteworthy undesired outcome of a patient's condition that is either likely enough to become an issue or is less likely but dangerous enough to be addressed.	observation, condition	risk
has support	SPRT		Used to indicate that an existing service is suggesting evidence for a new observation. The assumption of support is attributed to the same actor who asserts the observation. Source must be an observation, target may be any service (e.g., to indicate a status post.)	observation	supporting evidence
has explanation	EXPL	SPRT ⁻¹	This is the inversion of support. Used to indicate that a given observation is explained by another observation or condition.	observation	explaining observation or condition
is cause for	CAUS	SPRT	An assertion that a new observation was assumed to be the cause for another existing observation. The assumption is attributed to the same actor who asserts the observation. This is stronger and more	cause	effect

			specific than the support link. For example, a growth of <i>Staphylococcus aureus</i> may be considered the cause of an abscess. The source (cause) is typically an observation, but may be any service, while the target must be an observation.		
is manifestation of	MFST	CAUS ⁻¹	An assertion that a new observation may be the manifestation of another existing observation or action. This assumption is attributed to the same actor who asserts the manifestation. This is stronger and more specific than an inverted support link. For example, an agitated appearance can be asserted to be the manifestation (effect) of a known hyperthyroxia. This expresses that one might not have realized a symptom if it would not be a common manifestation of a known condition. The target (cause) may be any service, while the source (manifestation) must be an observation.	manifestation	cause
is derived from	DRIV	SPRT	A derivation link serves to explicitly associate a derived observation with its input parameters. Both, source and target must be observations, typically numerical observation. E.g., an anion-gap observation can be associated as being derived from given sodium-, (potassium-,) chloride-, and bicarbonate-observations.	output parameter	input parameter
has reference values	REFV	SPRT ⁻¹	Reference ranges are essentially descriptors of a class of result values assumed to be "normal", "abnormal", or "critical." Those can vary by sex, age, or any other criterion. Source and target are observations, the target is in <i>reference</i> mood. The interpretation range is both a support link and a trigger, in case of alarms being triggered by critical results.	observation	range
assigns name	NAME	SPRT ⁻¹	Used to assign a "name" to a condition thread. Source is a condition node, target can be any service.	condition thread	name
is revision of	RVSN		A service description that is a modification of another service description. This includes revisions of protocols and orders.	revision	prior version
amends	AMND	RVSN	A service that amends a previously stated service. This is used, e.g., for corrections of reported observations.	amendment	prior version
updates (condition)	UPDT	RVSN	A condition thread relationship specifically links condition nodes together to form a condition thread. The source is the new condition node and the target links to the most recent node of the existing condition thread.	new head of thread	old head of thread
instantiates (master)	INST	RVSN	Used to capture the link between a potential service ("master" or plan) and an actual service, where the actual service instantiates the potential service. The instantiation may override the master's defaults.	instance	master
fulfills (order)	FLFS	RVSN	A service that was done in fulfillment of an ordered service description. A fulfilled service may differ from an ordered (or planned) service description.	fulfillment	order
dispensing for (order)	DISP	FLFS	Links a medication service (order) with a supply service, representing the dispensing of the drug (as order or event.)	supply order of event	medication order
substitutes (brand product)	SBST	RVSN	A special link between medications indicating that the source is a generic for the target.	generic	brand
matches	MTCH	RVSN	A trigger-match links an actual service (e.g., an	matching	trigger

(trigger)			observation or procedure that took place) with a service in <i>trigger</i> mood. For example if the trigger is "observation of pain" and pain is actually observed, and if that pain-observation caused the trigger to fire, that pain-observation can be linked with the trigger.	service	
evaluates (goal)	GEVL	RVSN	A goal-evaluation links an observation (intent or actual) to a goal to indicate that the observation evaluates the goal. Given the goal and the observation, a "goal distance" (e.g., goal – observation) can be "calculated" and need not be sent explicitly.	evaluation	goal
has option	OPTN		Multiple alternative options for an order, routing, or scheduling options and preferences may be specified for a planned (or ordered) service. The source (plan) is in either of the moods definition, order, intent or schedule. The Service.priority_nmb attribute is used to weigh options as preferred over other options.	plan	option

This table is not necessarily complete. The purpose of this table is to be as specific as possible but also to classify similar relationship types into categories. Some may miss an unspecific relationship type "pertains-to." This "pertinence" was not included in the above table, because, it is just a more polite way to say "relationship, not otherwise specified" or "other." The problem is that "other" terms have no specified meaning, but are merely the complement of all the currently existing relationship types. As new relationship types will be defined in the future, "other" will change its meaning rather drastically. When other relationship types are discovered, they should be communicated to the Order/Results or Patient/Care Technical Committees and should be standardized. In addition the HL7 Code Value (CV) data type allows one to express that a given concept is not-codeable with the applicable code system.

2.4.1.2 Service_relationship.inversion_ind : BL default: false

The inversion indicator is used when the meaning of the relationship type must be reversed. For example, we define a relationship type *reason* to express the reason for an action as in

- a) "A cholecystectomy was performed because of symptomatic cholelithiasis without signs for cholecystitis." (cholecystectomy has-reason cholelithiasis)

This statement of rationale is attributed to the responsible performer of the cholecystectomy. Now consider the following statement:

- b) "The finding of symptomatic gall stones (cholelithiasis) with no signs of acute cholecystitis suggests a cholecystectomy."

While sentence a) declares a reason for an action, sentence b) suggests an action. Reason and suggestion links are in a reciprocal, i.e., if X has-reason Y, then Y suggests X. The second statement would may have been made by the originator of the cholelithiasis finding.

In the "network" of interrelated services, we need to make sure that we do not lose proper attribution of statements to originators ("who said what?") Since attribution is so important, we adopt a very simple rule for it: **a service relationship is always attributed to the originator of the source service.** No exceptions to this rule are permitted whatsoever. If attribution needs to be different one can invert the relationship type by setting the inversion_ind attribute to *true*.

If the inversion indicator is *true*, source and target service swap their roles, that is, the reason and the suggested action swap their roles, so that cholecystectomy can be the source and colelithiasis can be the target. Note that the attribution rule is always unchanged, i.e., **the service relationship is always attributed to the responsible author of the source service**, no matter what the inversion_ind value is.

2.4.1.3 Service_relationship.sequence_nmb : INT default: 1

This integer number allows one to specify an ordering amongst the outgoing relationships of a service. This is used to represent sequences of actions in execution plans.

The ordering may be total or partial. A total ordering exists if every relationship in a relationship bundle has a distinct sequence number. (A relationship “bundle” is a sub-set of the relationships originating in the same service instance and usually having the same relationship type). If, however, some relationships in the bundle share the same sequence number, we have a partial ordering. In such a case the services with the same sequence number are concurrent.

2.4.1.4 Service_relationship.priority_nmb : INT default: 1

This integer number allows to specify another kind of ordering amongst the outgoing relationships of a service. This is used to represent the priority ordering of conditional branches in service execution plans, or priority ranking in pre-condition, outcome or support links, and preferences among options.

The ordering may be total or partial. A total ordering exists if every relationship in a relationship bundle (a relationship bundle is a sub-set of the relationships originating in the same service instance and usually having the same relationship type) has a distinct priority number. If, however, some relationships in the bundle share the same priority number, we have a partial ordering. Those links with the same priority will have undefined ordering of consideration.

2.4.1.5 Service_relationship.pause_qty : PQ ~ 1 s default: 0 s

The time that should elapse after this activity has got clearance to execute and the actual begin of execution. Any entering pre-conditions are tested before the slot is entered, so the pause specifies a minimal waiting time before the service is executed after its pre-conditions become true.

2.4.1.6 Service_relationship.checkpoint_cd : CV default: B

Indicates when associated pre-conditions are to be tested. The following values are defined:

Table 14: Condition checkpoint code

Concept	Code	Definition
entry	S	Condition is tested once before the service is executed (IF <i>condition</i> THEN <i>service</i>).
beginning	B	Condition is tested every time before execution of the service (WHILE <i>condition</i> DO <i>service</i> .)
end	E	Condition is tested at the end of a repeated service execution. The service is repeated only if the condition is true (DO <i>service</i> WHILE <i>condition</i> .)
through	T	Condition must be true throughout the execution and the service is interrupted (asynchronously) as soon as the condition turns false (asynchronous WHILE loop.) The service must be interruptible.
exit	X	Condition is a loop checkpoint, i.e. it is a step of an activity plan and, if negative causes the containing loop to exit.

2.4.1.7 Service_relationship.split_cd : CV default: I1

When an activity plan has a branch (indicated through multiple steps with the same item number) the split code specifies how branches are selected for execution. The values are defined as follows:

Table 15: Branch split code

Concept	Code	Definition
exclusive try once	E1	The pre-condition associated with the branch is evaluated once and if true the branch may be entered. All other exclusive branches compete with each other and only one will be selected. This implements a COND, IF and CASE conditionals, or "XOR-split." The order in which the branches are considered may be specified in the <code>Service_relationship.priority_nmb</code> .
exclusive wait	EW	A branch is selected <i>as soon as</i> the pre-condition associated with the branch evaluates to true. If the condition is false, the branch may be entered later, when the condition turns true. All other exclusive branches compete with each other and only one will be selected. Each waiting branch executes in parallel with the default join code <i>wait</i> (see below.) The order in which the branches are considered may be specified in the <code>Service_relationship.priority_nmb</code> .
inclusive try once	I1	A branch is executed if its associated preconditions permit. If associated preconditions do not permit, the branch is dropped. Inclusive branches are not suppressed and do not suppress other branches.
inclusive wait	IW	A branch is executed <i>as soon as</i> its associated conditions permit. If the condition is false, the branch may be entered later, when the condition turns true. Inclusive branches are not suppressed and do not suppress other branches. Each waiting branch executes in parallel with the default join code <i>wait</i> (see below.)

2.4.1.8 `Service_relationship.join_cd` : CV default: W

In a parallel branch construct the join code indicates how the concurrent activities are resynchronized.

Table 16: Branch join code

Concept	Code	Definition
wait	W	Wait for this branch to terminate.
kill	K	When all other concurrent branches are terminated, interrupt and discontinue this branch.
exclusive wait	X	Wait for any one of the branches in the set of exclusive wait branches to terminate, then discontinue all the other exclusive wait branches.
detached	D	Detach this branch from the other branches so it will not be resynchronized with the other branches.

A *kill* branch will only be 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 discontinued shortly after it is started.) A *detached* branch will be unrelated to all other branches, thus a *kill* branch will be discontinued no matter whether there are *detached* branches still running.

2.4.1.9 `Service_relationship.negation_ind` : BL default: *false*

For conditions and criteria links indicates whether the meaning is negative (condition must not be true.) Normally all conditions are interpreted as affirmative, i.e., the condition must be true. The `negation_ind` is part of the condition so that the Boolean outcome of the condition XOR-ed with the `negation_ind` of the condition link must be true. We thus say the "condition is true" even if the test was negative if the `negation_ind` is *true*.

2.4.1.10 `Service_relationship.conjunction_cd` : BL default: AND

In a bundle of precondition or outcome relationships, this code indicates the logical conjunctions of the criteria.

Table 17: Criteria conjunction code

Concept	Code	Definition
and	AND	This condition must be true.

or	OR	At least one of the condition among all OR conditions must be true.
exclusive or	XOR	One and only one of the XOR conditions must be true.

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. 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.)

2.5 Timed and conditioned care plans

Providing a concise way of representing care plans, schedules, protocols, guidelines, and workflow processes is one of the main concerns of this proposal. There is a huge amount of prior work on this area on which this proposal is based. In this section we present the three principle building blocks of timed and conditioned care plans, which is, (1) plans, (2) conditions and (3) timing.

2.5.1 Plans

2.5.1.1 Background

Defining a plan of activities is very much like programming a computer. To standardize the HL7 information structure for activity management (e.g., ordering, scheduling) it is of utmost importance to absolutely crisply define the meaning and functioning of all the data elements and data structures that are supposed to manage activities.

The advantage of likening business activity management with computer programming is that what is understandable to a computer usually makes sense to humans as well (but not vice versa.) In the following we speak about a service whose “conditions are checked,” “that is invoked or discontinued,” as if there was a microprocessor doing it all. In so speaking we do not imply that every system needs to have automated activity management functionality or a service execution “robot,” programmed and directed by HL7 messages. Instead of performing services automatically, many systems will choose to manage activities by active or passive reminder notices to personnel. However, systems should be able to map whatever their functionality is to HL7 interfaces precisely and interoperably. This can only be achieved if the HL7 specification of process wok-flow has some rigor.

In principle a computer program consists of *statements*. Every statement represents a step in an activity plan. Most computer programming languages provide a sequential computing model, where statements are executed one after

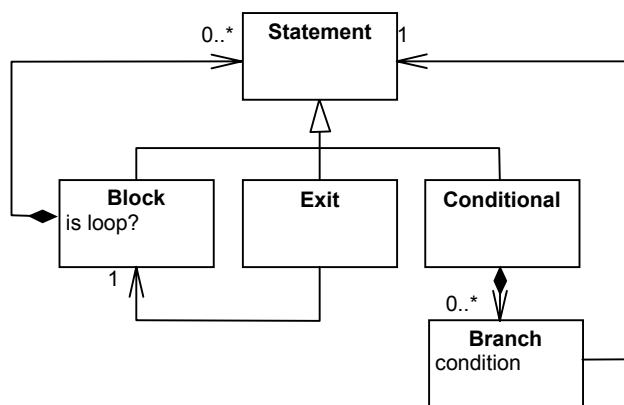


Figure 3: Simple model of the essential sequential programming constructs: A statement may be primitive (not specialized.) A conditional consists of a number of branches, each branch having a condition and a statement. A block is a sequence of statements that is executed once, while a loop is just a block executed more than once. Blocks (loops) can be exited with the exit statement, that is usually embedded in a conditional.

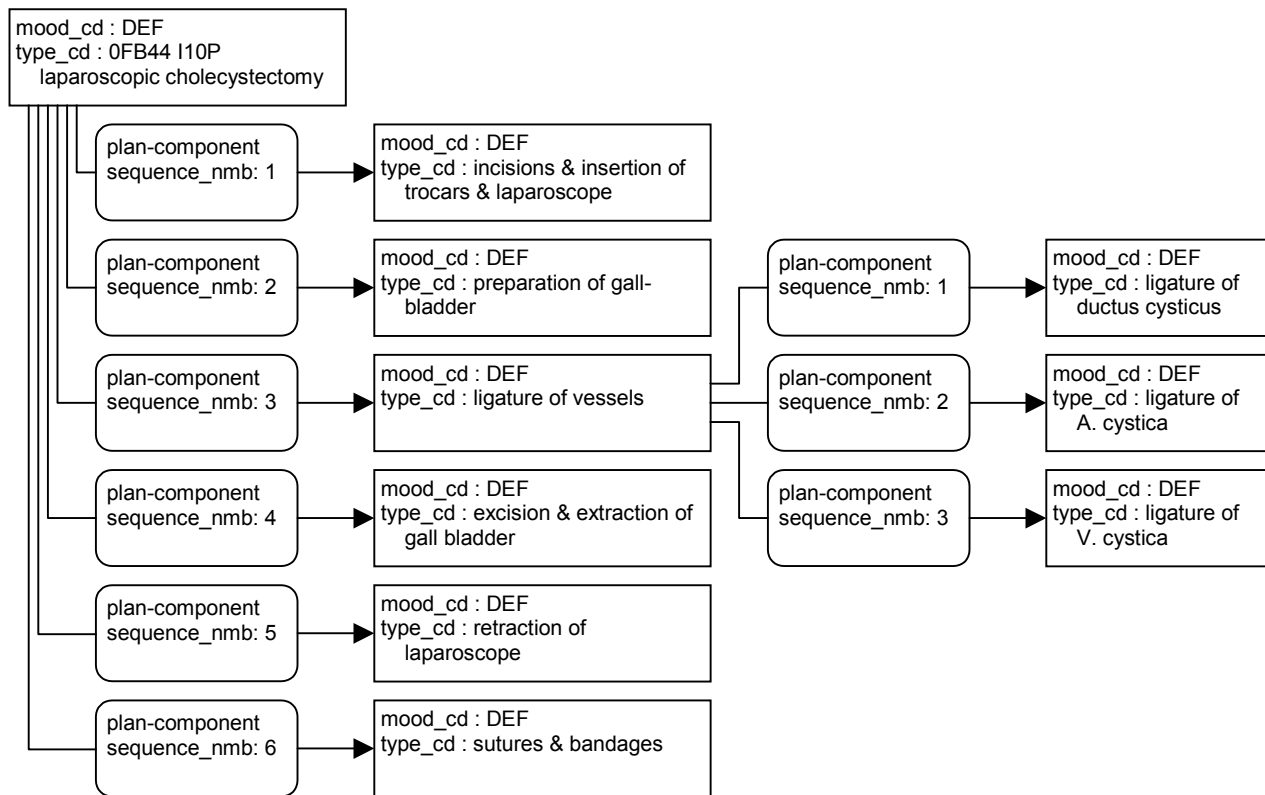


Figure 4: Example of sequential plan construction for laparoscopic cholecystectomy. Edged boxes are Service instances (all in definition “master” mood.) Rounded boxes are Service_relationship instances of type: plan-component. The sequence_nmb attribute orders the relationships into a sequence. Each service can in turn be decomposed into plan-components.

the other. For real world processes a purely sequential execution model is clearly inadequate, since real world activities are carried out cooperatively and concurrently by multiple actors who perform multiple sub-tasks all at the same time. However, there is considerable work in computer science on parallel processes, although that is not usually reflected in the more popular programming languages. Just like many of the proposed concurrency-aware programming languages do, we will reuse the fundamental concepts from a sequential execution model as far as possible and then add the concurrent constructs to it.

2.5.1.1.1 Sequential control constructs

In the sequential computing model, the primitive statements fall into two categories (1) data and I/O manipulation commands, and (2) control flow commands. The most primitive control flow command is the GOTO statement and the IF-THEN conditional branching. In the early 30 years of computer science the value of structured program design has been discovered and one finds agreement that even though the primitive control flow statements are all that is technically needed, a structured program model is favorable. The consensus model of structured programming that such diverse languages as LISP and the ALGOL family seem to agree to consists of the following components:

- (1) primitive statement (e.g., procedure call, value assignment.)
- (2) block (a sequence of statements.)
- (3) Conditional
 - (a) (COND (*condition*₁ *statement*₁) (*condition*₂ *statement*₂) ... (*condition*_{*n*} *statement*_{*n*}))
 - (b) IF *condition* THEN *statement*₁ ELSE *statement*₂

- (c) CASE *expression* OF *value*₁ : *statement*₁; *value*₂ : *statement*₂; ...; *value*_n : *statement*_n; END
- (4) Loop
 - (a) LOOP *statement* with nested conditional EXIT
 - (b) WHILE *condition* DO *statement*
 - (c) REPEAT *statement* UNTIL *condition*
 - (d) FOR *iterator* DO *statement*

The most general conditional statement is the COND term known from LISP: a list of condition-action pairs is worked down until a condition evaluates to true, only then the associated action is executed and its result becomes the result of the entire COND term. The final condition in the list should be a default (the *true* literal) so that the final action will be the default action, otherwise the COND term “fails.” IF-THEN-ELSE is simply a restricted COND term with only one condition and one default. The CASE statement is just a COND term with the condition being an equality test for the case expression with the case branch value.

The most general (and actually the most useful) loop statement is the LOOP-EXIT statement added late into the ALGOL family (with MODULA 2.) Pushing the condition test to the begin or end of the loop body as is done in WHILE and REPEAT loops reads nice in English but is seldom actually useful.

Therefore we will mark the primitive statement, the block, the COND term, and the LOOP-EXIT statement as a requirement for the HL7 sequential activity plan. A simple model of those minimal control constructs is shown in Figure 3.

2.5.1.1.2 Concurrent control constructs

The primitive of parallel activities are the notions of a process, a fork that spawns off a child process from a parent process, and inter-process communication through signals and semaphores. However, just as with sequential programming, a more structured approach is favorable. A typical approach to “parallelizing” sequential programming languages is to conceive parallel extensions of the sequential constructs. Thus one can correlate a statement with a process, a block with a multi-branch fork. The COND term can be parallelized as a multi-branch fork with guard-conditions on each branch. Parallel loop constructs are conceivable but their use seems rather limited.

2.5.1.2 HL7 Activity Plans

In HL7 version 3 an activity plan can be constructed as a brand new instance used once for a particular patient or can be defined in the “master file” and reused multiple times. Complex orders are an example for when one would construct brand new service plans. For protocols and guidelines, service plans are defined once as a “master” service. In any way, every activity in our plan construction model is represented as an instance of the service class.

Every “primitive” activity must be defined as a service “master”, there are no primitive statements defined by HL7 besides what is defined as “master” services. These plan construction features allow one to construct plans either as reuseable master plans or as ad-hoc plans constructed for just one special case. However, the primitives that the so constructed plan uses, must be predefined as master services.

A simple block statement can be formed through a service that has a set of target services associated with it through service relationships of type *plan-component*. In a simple sequential block, every relationship instance has a different *Service_relationship.sequence_nmb* value so that the order of the statements in the block is given from lowest to highest. For example, the laparoscopic cholecystectomy example would (partially) decompose as shown in Figure 4.

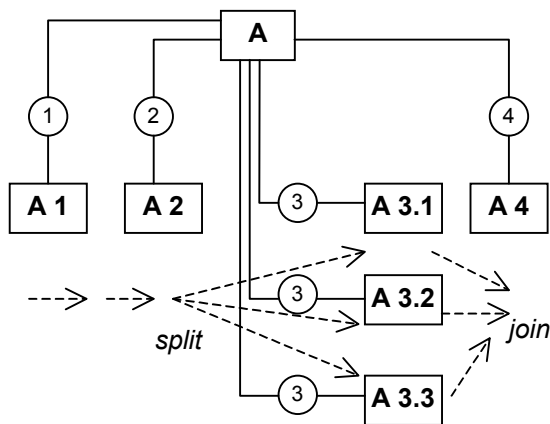


Figure 5: Example of plan construction with a concurrent region. Edged boxes are Service instances. Rounded boxes are Service_relationship instances of type: plan-component showing only the sequence_nmb attribute. Since the components A3.1, A3.2 and A3.3 all have the same sequence number (3) they may be executed in parallel.

An activity loop can be constructed by setting the Service.max_repetition to a value greater than 1. For any finite max_repetition > 1 the loop will cycle at most that number of times and then exit. If max_repetition is set to the positive infinity (a special kind of a null-value) the loop can be cycled indefinitely. The loop is still terminated when the time constraints or conditional constraints are no longer valid.

If the multiple plan-components have the same sequence number, this indicates a concurrent region or a branching of the flow of control. Concurrent region and (conditional) branching is actually expressed in the same way with the attributes split_cd, and join_cd to control the kind of branching or concurrent region. Figure 5 shows an example with three concurrent plan components. In this example there are four sequential steps: A1, A2, A4 and the *concurrent region* consisting of A3.1, A3.2 and A3.3. The control flow is symbolized with dashed arrows in Figure 5, with a

split (or fork) before the concurrent region and a join after the concurrent region.

When only one of the branches A3.1, A3.2, or A3.3 is selected we have a conventional conditional branch known from every programming language. In general a branch can only be selected for execution if all its preconditions are true. If all the branches have their split_cd attribute set to “exclusive” branching (E1 or EW) only one branch will be entered. If more than one branch is ready to be entered, the one with the lowest priority_nmb value is selected. If priority_nmb is ambiguous, the selection of the branch is undefined (subject to preferences of the performing actors or simply random.) The priority is ambiguous if more than one branch that have clearance to run have the same priority_nmb value and if no other branch with clearance has a higher priority (see about conditions below.)

If split_cd is “inclusive” (I1 or IW) and more than one branch have clearance to run, there will be a true concurrent region, i.e. all branches with clear preconditions will be executed in parallel. The synchronization of the concurrent region is controlled by the join_cd. If the join_cd for one branch is wait (W) the concurrent region ends not before that branch terminates. If the super-service is asynchronously stopped (by a time-out or by through-criteria to turn false,) the wait branches will be interrupted. (An interrupt may be blocked, however, if the actually running (sub-)service has the Service.interruptible_ind set to false.)

If the join_cd for a branch is kill (K) that branch is interrupted and terminated prematurely as soon as all other branches have terminated whose join_cd is neither kill nor detached. The interruption of the kill branch is subject to the Service.interruptible_ind. This means, if the branch to interrupt is currently executing a sub-activity that is not interruptible, the service will be interrupted only after that non-interruptible sub-activity is completed. For example, if the slot contains one wait-branch and one kill-branch, the kill-branch executes until the wait-branch terminates. A concurrent region can not consist of only kill branches, since none of it could ever be executed – there must be at least one wait or exclusive-wait branch.

Exclusive join codes behave like the reverse of an exclusive branch. For example, if all sub-activities A3.1, A3.2, and A3.3 run with exclusive wait (X) and, say, A3.3 terminates first, A3.1 and A3.2 will be

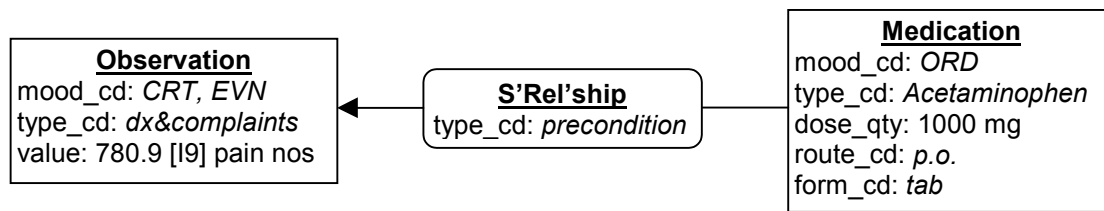


Figure 6: A simple condition representing a PRN medication order: Acetaminophen 1000 mg tab p.o. PRN for pain. Criteria always refer to the most recent service if the time attribute is not specified.

interrupted and discontinued. The *exclusive wait* join code lets you express a “whichever comes first” logic.

Finally the join code *detach* will spawn off the activity and run on its own. Detached branches will not be resynchronized to the parent service nor to any other branches running concurrently. Detached branches can not extend the runtime of kill-branches, and are not terminated by interrupts to the parent service. Detached branches are terminated only subject to their own conditions. For example, if A3.3 were detached, it could continue to execute even if A3.1 and A3.2 ends and A4 is begun. No synchronization with A3.1 and A3.2 or A4 will occur for the detached branch.

2.5.2 Conditions

Activities can not be planned ignorant of the varying circumstances of the real world. Variants of plans must be selectable depending on certain circumstances. Probably the most important functionality of the Unified² Service Action Model is its straight forward way to express conditions. Conditions can select sequential or concurrent branches in activity plans, and can control the repeated execution of activities (loops.) But conditions can also asynchronously invoke other activities (e.g., alert and reminder triggers) or can asynchronously interrupt current activities. Finally, activities as outcomes can be used to describe goals, whose evaluation can be initiated automatically.

Conditions in USAM are described uniformly as Service *predicates*. A service predicate is an instance of class Service in one of the predicate moods. In predicate mood, a service object is not indicating a service that has been or will actually be performed as such, but it specifies a pattern that actual services can be compared against. For example, when we want to prescribe PRN medication Acetaminophen 1000 mg p.o. once for “pain” we can formulate this as an Observation service in *criterion* mood as shown in Figure 6.

Thus any service that two systems share common definitions (defined in the “master” service catalog) can be used to formulate conditions. By default any field of a *criterion* service object is assigned the *no-information* (NULL) value, meaning that it would match any actual data. When testing a criterion a system will pick the most recent actual service instance of the specified Service.type_cd and will look at all fields that are assigned values and compare those with the actual service under consideration. All fields with assigned values must match the actual service.

Conditions are tested at different times in the execution of a service depending on the Service_relationship.checkpoint_cd defined in Table 14. By default (*beginning*, B) a precondition is tested every time before a service is executed. For repeating services it is tested before every occurrence and the repetition is ended the first time the condition evaluates to false. This implements a WHILE *condition* DO *service* loop.

Alternatively the condition may be tested only once before a repeating service is executed the first time (*start*, S). A start condition can thus not terminate a loop, but the loop can be terminated through other

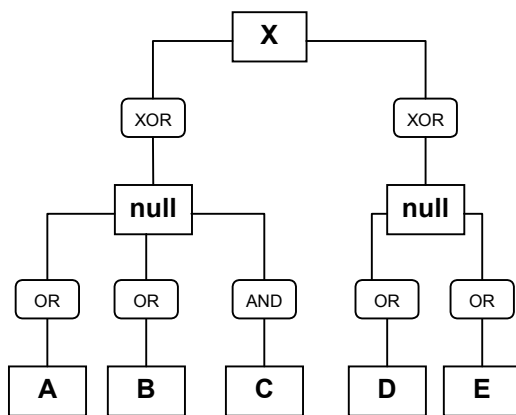


Figure 7: Example of a complex criterion as a precondition for Service X, representing the Boolean expression $((A \text{ OR } B) \text{ AND } C) \text{ XOR } (D \text{ OR } E)$.

conditions or through a timing constraint. This implements a simple IF *condition* THEN *statement* logic, where *statement* may be a loop terminated by other criteria.

A condition may be tested after the first execution of a repeating service and before the next cycle (*end*, E). This implements the positive version of the REPEAT *statement* UNTIL *condition* loop. The sense of the condition is always the same, i.e. the statement is executed or repeated when the condition is true and the loop is exited when the condition turns false (DO *statement* WHILE *condition*.)

With the `checkpoint_cd` set to *through* (T) that condition must true at all times throughout the execution of the service. If that condition turns false

the service is interrupted. Therefore it is an error for a service with a through condition to have an `interruptible_ind = false`. Sub-services may well block interrupts by setting their `interruptible_ind` to *false*, which will cause the service to end as soon as the currently running non-interruptible sub-service ends.

A condition can be negated by setting the `Service_relationship.negation_ind` to *false*. That way the service will be executed or repeated only if the condition is false. The negation indicator is part of the condition, that is, the Boolean outcome of the condition XOR the `negation_ind` of the condition link must be true. We thus say the “condition is true” even if the test was negative if the `negation_ind` is set to *true*.

Multiple criteria can be combined using Boolean logic operators in the attribute `Service_relationship.conjunction_cd`. The operators AND, OR and XOR are available (negation is also available with the `Service_relationship.negation_ind = true`.) The default conjunction is AND, i.e. by default all associated criteria must be true. When all criteria links have `conjunction_cd` OR, only one criteria needs to be true. With XOR, exactly one criterion must be true. Note that use of XOR in reasonable rules is rather rare. In one criteria bundle multiple conjunction operators can be mixed with the following semantics: all links are sorted into three groups by conjunction code. All AND criteria must be true, one or more of the OR criteria must be true, and exactly one of the XOR criteria must be true.

For more complex criteria that require nested Boolean expressions intermediate criteria can be constructed using Service objects without a `Service.type_cd` value. For example, the complex logic expression “ $((A \text{ OR } B) \text{ AND } C) \text{ XOR } (D \text{ OR } E)$ ” can be constructed as shown in Figure 7.

A criterion in *trigger* mood is tested asynchronously at any time the system may choose. Trigger conditions are used to trigger alerts or reminders. For example, the PRN order above could have been written using a trigger so that each time the patient reports pain (and that information is entered into the medical record) the associated PRN order is considered.

2.5.3 Timing

The timing of orders and schedules can be a very complex task, even the representation of all the parameters of scheduling is quite a challenge. HL7 v2.3.1 has left many questions open about the timing of orders, and the timing/quantity (TQ) data type has not been very widely appreciated. While

part of the complexity of the HL7 v2.x TQ data type is due to the complexity of the matter, we have put a lot of effort into revising the TQ and finding a simpler and more uniform alternative. Together with the version 3 data type working group we have investigated two representations of timing parameters.

One alternative was a minimal set of distinct attributes, to describe frequency, duration, pause intervals, etc. However, we found that the best way to specify such parameters in a way that is somewhat intuitive to humans would be a single “General Time Specification” (GTS) that is a literal syntax defining a complex set of periodic points and intervals of time.

A requirement for the definition of this GTS syntax was that it had to have a crisp semantic explanation that would be readily implementable in computer systems. Thus, we developed a semantic model of this General Time Specification, that, due to the nature of the problem, requires some mathematics to understand. In this section we will first explain the concepts that are fundamental to this General Time Specification, and only then describe the syntax. We are now confident that the approach works in general, but we are not yet finished with refining the specification, so, some changes may still occur in the near future..

2.5.3.1 A duration is a physical quantity with a time unit

HL7 version 3 distinguishes a number of different time related data types. The simplest of the time related data types is for *duration of elapsed time*, which is just a physical quantity (PQ, a real number with a unit) constrained to the dimension of time. That is, any unit of time is allowed (e.g., 1 s, 1 min, 1 h, 1 d, 1 wk, 1 mon, 1 a.)

2.5.3.2 Points on the absolute real time axis

Based on the duration, we define the *point in time* data type, i.e. any point on the real natural time axis. Because there is no natural origin of the absolute real time axis, one can reach any point on the time axis by agreeing on an epoch and communicating just the duration elapsed since that epoch. Astronomers, for example, communicate point in time as the number of days elapsed since the Julian date noon Monday, January 1 4713 B.C. Other than that, most people will not understand an epoch/duration form for points in time and use the Gregorian (or other) calendars instead. However, computers can best deal with the epoch/duration form of time points so system implementers will likely deal with the epoch/duration form internally, while presenting time points to humans in a calendar form.

The relationship between calendars and the even flow of real time is quite complex, which makes conversion between epoch/duration form and a calendar form of points in time rather difficult. Fortunately there are many programs available that help in this task, most notably such routines are implemented in most operating systems. In HL7 v3 we continue to allow points in time to be expressed as “Timestamps” (TS) in the traditional format “YYYYMMDDhhmmss.f f f [+ | -] ZZzz”, where digits can be left out from the right hand side..

The arithmetic difference between two points in time is a duration. The sum of a point in time and a duration is a point in time. No multiplication is defined for points in time. A duration can be multiplied by a real number (yielding a duration scaled by the factor) or any other physical quantity (yielding something other than a duration.)

2.5.3.3 Intervals

Intervals are continuous subsets of an ordered base data type. Intervals can be formed from numbers and physical quantities, including elapsed time. For example, the interval [0;1] of real numbers

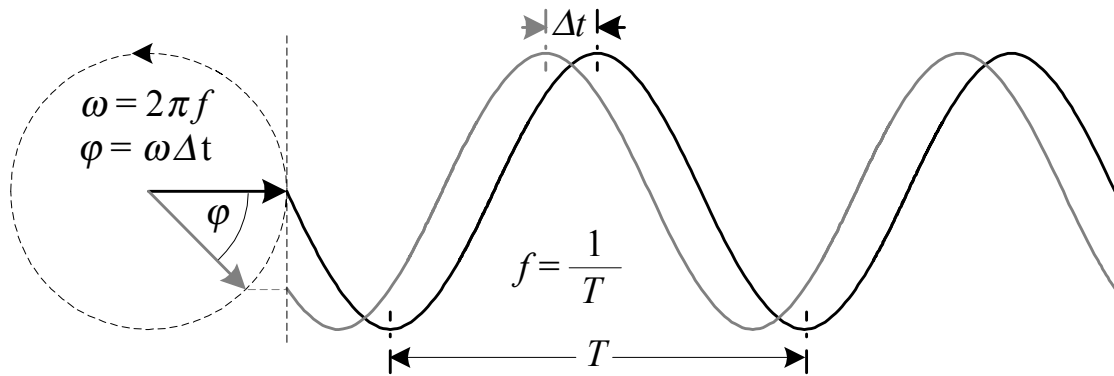


Figure 8: Periodic events can be described in terms of a rotation or wave. The period T , frequency f , and angular velocity ω are equivalent measures of how often the event occurs in a unit of time. The phase ϕ is a measure for the offset between the wave and a reference wave of the same frequency. The phase may be measured in a unit of plane angle or as a difference of time Δt . Thus, all parameters describing the periodic event can be measured as time duration.

contains all the real numbers between zero and one. In the same way one can form an interval of duration, such as [5 min; 10 min] containing all elapsed time from 5 to 10 minutes. The meaning of an interval is that all the values between the bounds are contained. For example, we can specify the normal range of turn-around times for an activity as “[5 min; 10 min]” (or simply “5-10 min”).

Intervals can also be formed from points in time. The most common case where this is done is to specify the time of a Service, that includes any point in time between a begin time and an end time. In the epoch/duration form of absolute time, such an interval of points in time is specified as an interval of elapsed time given a certain epoch. For example, if we set the epoch to be January 1 1996, the interval between September 1 1999 and September 3 1999 would be “1339-1341 d”. In calendar form we can specify that interval as “19990901-19990903”.

An interval is usually specified in terms of its low and high bounds. If both bounds are unknown, the width of the interval may still be known. For instance, we may know that a service requires 10 min but we may not know (yet) when the service is scheduled to start (or when it started in the past.) In that case one can specify only the width of an interval. In the example of the three days in September can be written as “[3 d]”. For the 10 minutes service we can write “[10 min]”.

2.5.3.4 Recurring events, periodic time

In scheduling and medication orders we need to deal with events that recur over time in some regular pattern. For example, amoxicillin 500 mg tablets 3 times a day for 10 days. U.S. doctors and pharmacist have developed abbreviations they use to communicate such timing information in some “coded” form. For example, BID means twice a day, TID three times a day, X10D means for 10 days, and Q6H means every six hours. These abbreviations have been used in HL7 version 2’s TQ data type.

For HL7 version 3 rather than using such symbols we apply a consistent quantitative schema to expressing such periodic time phenomena. The goal is to have a schema that has crisply defined semantics and that is actually computer processible. The basic building blocks for periodic time are: duration, intervals, set operations, and calendars.

2.5.3.5 Frequency f or period duration T are equivalent

Periodic events can be described with a **frequency** f or the **period** duration T . Period and frequency are related in a simple reciprocal way: $f = 1/T$. Apart from their reciprocity, period and frequency are the same.

We can describe the above amoxicillin example using the frequency $f = 3/d$ or period $T = 8$ h (note that there is no difference in choosing f or T .) Whenever there are two ways of expressing the same information, a standard should allow one and forbid the other so as to avoid unnecessary optionality. Since we can base elapsed time, absolute time and intervals of absolute time all on duration, we choose to base periodic time also on duration, the period duration.

2.5.3.6 Periodic points vs. intervals and the phase φ

Within the period of 8 hours we may need to specify some more precisely at what point the medication is to be given. This can be done by specifying an offset into the period. For example, if a service is to repeat once every day at 8 o'clock the period is $T = 1$ d. To adjust the timing to 8 o'clock every day we can set the offset (or **phase** φ) $\varphi = 8$ h. Thus, given that for $\varphi = 0$ every cycle starts a 12 o'clock midnight, moving the phase 8 hours later will set the recurrent time to every day 8 o'clock. In the same way one can specify times precisely up to the nanosecond if necessary, for example, every day at 08:12:18 and 675 milliseconds would be: $T = 1$ d, $\varphi = 8.2051875$ h. We call this construct **periodic point in time**.

One can use the period-phase notation to specify **periodic intervals of time**. For example, a radiation therapy is to be performed every other day for 10 minutes for five times in a row. We can specify this as: $T = 2$ d, $\varphi = [10 \text{ min}]$. When this service has been scheduled, say, for every other day at 9:30 to 9:40 AM, this is expressed as: $T = 2$ d, $\varphi = [9.5; 9.6667]$ min. A periodic *point* in time has a simple elapsed time as its phase φ , while a periodic *interval* has an interval of elapsed time as its phase.

2.5.3.7 Periodic points and intervals as sets of time points

A periodic point in time is an infinite set of discrete time points recurring at the same distance from all past to all eternity. Most real recurring events are limited to a certain maximum amount or maximum time in which the event repeats. Since both intervals of absolute time and periodic points in time are sets of absolute time, one can build the **intersection** between the interval and the recurring time to narrow the time set down to some finite number of recurrences. For example, by limiting the amoxicillin therapy to any interval of 10 days with 3 doses, we limited the amount to 30 doses. The set **union** operation is available to specify more irregular timings (e.g., Monday 8:00 AM and Thursday 11:30 AM.)

Every set of time S has an **outer bound interval**. The outer bound interval is the smallest interval of absolute time that includes the set of time S . The outer bound interval may be infinite in on both sides, but usually it is finite on the left and often it is finite on the right too. The outer bound interval is an important concept that a scheduling system needs to find out when a series of recurring services actually start and when it ends.

Every set of time S , with a finite outer bound interval, can be enumerated as a sequence of time intervals, regardless how it is constructed. For example, the radiation therapy, that we start on a Tuesday, can be enumerated as Tuesday 9:30-9:40, Thursday 9:30-9:40, Saturday 9:30-9:40, and the following week Monday 9:30-9:40, and Wednesday 9:30-9:40. A scheduling system will eventually have to turn every specification of periodic time as a complex set of times into such an enumerated sequence of occurrence intervals.

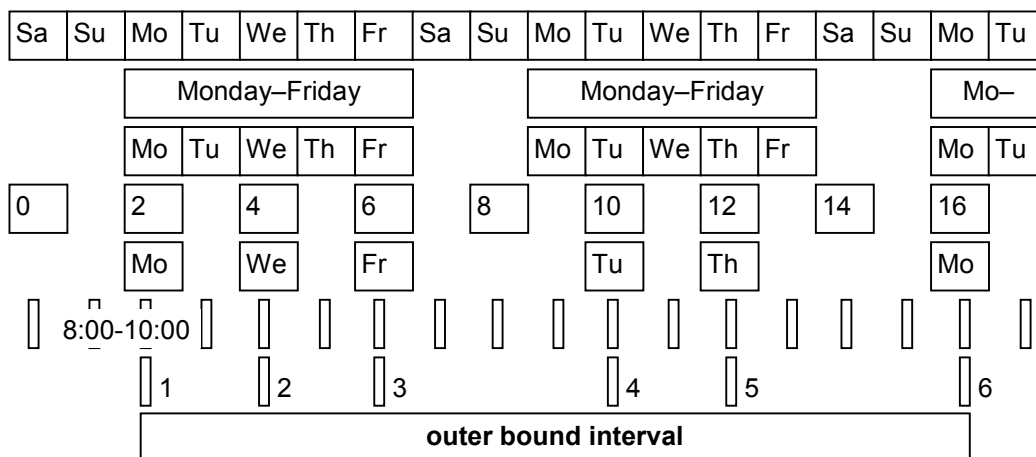


Figure 9: Example of constructing a complex schedule as the intersection of various periodic intervals of time. The schedule is: every other day from Monday to Friday 8:00 to 10:00 AM for six times. Regardless of how the set of time was constructed, it has one outer bound interval and can be enumerated as a sequence of intervals, each representing one occurrence of the service.

2.5.3.8 Stochastic and exact timing

If we measure the frequency of a recurring event, say, $f = 16 / \text{min}$ ($T = 3.75 \text{ s}$), this can mean two things: either the recurring event occurs evenly and exactly every 3.75 seconds (exact frequency) or it is distributed unevenly but at average there are 16 events per minute (stochastic frequency.) Note that choosing period T or frequency f in specifying the timing of recurrence does not decide whether the timing is exact or stochastic. Although traditionally we tend to think of “every 8 hours” as more exact as “three times a day” both statements are really the same, since period and frequency are related through $T = 1/f$.

By default all periodic points and intervals of time are stochastic, that is, the true times of occurrence is allowed to be randomly distributed around the times specified. Just how far the actual timing may differ from the specified timing is usually left up to the reasonable judgement of the performing actors. If the allowance needs to be constrained, one can do so using a probability distribution for the phase. For example, for ampicillin 500 mg i.v. every 8 hours starting at 6 AM, with an allowance of 30 minutes, one can specify: $T = 8 \text{ h}$, $\phi = 6 \pm 0.5 \text{ h}$. From the many probability distributions available, one will most often choose the *guess*, *uniform*, or *normal* distributions. See the HL7 version 3 data type manual for more on probability distributions.

2.5.3.9 Alignment to calendars

The time units 1 mon, and 1 a, are but averages over one or many years, because the length of an individual month may vary between 28 and 31 days, and an individual year may be 365 or 366 days. Sometimes one wants a schedule to be ignorant of the irregularities of a calendar (e.g., strictly every other day) and sometimes one wants to align to the calendar (e.g., every 15th of the month.) One can specify alignment to a calendar unit for both the period and the phase. Alignment of the period to a calendar unit means that at the beginning or end of the calendar unit there may be a short period. For example, with week of the month the first and last period of any month are usually shorter. Alignment to calendars is selected symbolically through a code referring to a calendar unit. See the data type specification for more detail.

2.5.3.10 Literal expression and the General Time Specification (GTS)

Periodic times constructed from periods and phases with optional alignment to certain calendar units is easy to understand and process for computers. However, for humans a more intuitive notation is in order. The HL7 v3 data type specification therefore defines the following literal expression for sets of time.

```

expr    := term [ "+" term ]
term    := factor [ [ " " ] factor ]
factor  := period [ list | [ range [ step ] | step ] ]
list    := number [ ", " list ]
range   := number "-" number
step    := "/" number [ "%" number ]
    
```

Exhibit 3: Definition of literal expressions for sets of real time in EBNF.

A period identifier is a short one or two letter code for a calendar cycle. Period identifier come in three forms: (1) continuous, (2) ordinal, and (3) implicit. A continuous period is measured from some initial date (e.g., the epoch or an order start date) and is not bound to the larger calendar cycles. For example, if something is to happen strictly every other day regardless whether months are 30 or 31 days long one would use a continuous period. Continuous periods are formed using the letter C before the period identifier (e.g., CD for continuous day.)

An ordinal period identifier is aligned to the larger calendar cycles. For example, if something is to happen on every odd day of the month (1, 3, 5, ..., 27, 29, (31)) an ordinal period is used. Ordinal periods are specified using two period identifiers, one for the period in which to count and another for the larger period which we want to align to, (e.g. DM ordinal day of the month, DW ordinal day of the week.) Ordinal periods are counted from either 0 or 1 depending on the customs of the calendar. For example, in the western calendar day of the month and month of the year is usually counted from 1, while hour of the day and minute of the hour is counted from 0.

Implicit periods are those periods identified by the one letter period code, because it is so common to use it in either the continuous or the ordinal sense. For example, the year is counted continuously because there is no larger cycle (except for decimal multiples decade and century, which are not real calendar cycles.) Weeks are usually counted in a continuous way (i.e. not aligned to the calendar year,) while most other calendar cycles are aligned to each other (month-day-hour-minute-second.)

This part of the specification needs review and validation through a reference implementation to make sure everything is covered, and, to refine the specification. At this point the specification is certainly not quite complete.

Table 18: Period Identifiers in the Gregorian (western) Calendar

Code		Meaning	Position	Length	Starts With
2-letter	1-letter				
CY	Y	year (anno domini)	1	4	0
YM	M	month of the year	2	2	1 (January)
CM		month (continuous)			0
CW	W	week (continuous)			0
WY		week of the year		2	1
DM	D	day of the month	3	2	1
CD		day (continuous)			0
DY		day of the year		3	1
DW	J	day of the week		1	1 (Monday)
HD	H	hour of the day	4	2	0

CH		hour (continuous)			0
NH	N	minute of the hour	5	2	0
CN		minute (continuous)			0
SN	S	second of the minute	6	2	0
CS		second (continuous)			0

The following table shows many example expressions.

Table 19: Examples for literal expressions for time sets.

Literal Expression	Meaning
M09	September
MY09	September (using explicit ordinal two letter code)
M0915	September 15
M091516	September 15 at 4 PM
M09151630	September 15 at 4:30 PM
M0915163034.12	September 15 at 4:30:34.12 PM
M09 D15 H16 N30 S34.12	September 15 at 4:30:34.12 PM (as the intersection of multiple sets)
M01,03,07	January, March, July
M/2	every even month of the year (February, April, ...)
M/2%1	every odd month of the year (January, March, ...)
M04-09	April 1 to September 30
M04-09/2	every second month from April to September (April, June, August)
J6	every Saturday
DW6	Saturday (using explicit two letter code)
J1,3,4	Monday, Tuesday, Thursday
J/2	every even day of the week (Tuesday, Thursday, Saturday)
J/2%1	every odd day of the week (Monday, Wednesday, Friday, Sunday)
J1-5	Monday to Friday
J1-5/2%1	Monday, Wednesday, Friday
W/2	every other week (continuous)
W/2 J2	every other Tuesday
WY/2	every other week of the year
Y1999 WY15	the 15 th calendar week in 1999
1999 WY15	the 15 th calendar week in 1999 (period code is optional for the highest calendar unit)
1969021919-20	February 19 th of 1969, 7 PM to 8 PM.
WM2	the second week of the month
DY128	the 128 th day of the year
WM2 J6	Saturday of the 2 nd week of the month
M05 WM2 J6	Saturday of the 2 nd week of May
M05 DM08-14 J6	Mother's day (second Saturday in May.)
J1-5 H0800-1600	Monday to Friday from 8 AM to 4 PM
J1-4 H0800-1600 + J5 H0800-1200	Monday to Thursday 8 AM to 4 PM and Friday 8 AM to 12 noon.
CD/2 [10 min]	every other day for 10 minutes.
[10 d] H/8	three times a day for 10 days (each time a 60 minutes interval).
H/8	every eighth hour (each time a 60 minutes interval)

H/8%0700	every eight hours starting at 7 AM (each time a 1 minute interval)
H/8@	every eight hours (a point in time)
H/8%0 (30 min)@	every eight hours with 30 min tolerance (guess distribution)
H/8%0 (30 min) [10 min]	every eight hours with 30 min tolerance for 10 minutes.

Symbolic time specification (to be resolved at the receiving system)

H/8 IST	three times a day at institution specified times
AM	every morning at institution specified times
PM	every evening at institution specified times
HS	at the hour of sleep
AC	before meal (ante cibus)
PC	after meal (post cibus)
IC	between meals (inter cibus)
ACM	before breakfast (ante cibus matutinus)
ACD	before lunch (ante cibus diurnus)
ACV	before dinner (ante cibus vespertinus)
PCM	after breakfast (post cibus matutinus)
PCD	after lunch (post cibus diurnus)
PCV	after dinner (post cibus vespertinus)
ICM	between breakfast and lunch
ICD	between lunch and dinner
ICV	between dinner and the hour of sleep

Common symbolic abbreviations

Abbreviation	Normal form
--------------	-------------

BID	H/12 IST	two times a day at institution specified time
TID	H/8 IST	three times a day at institution specified time
QID	H/6 IST	four times a day at institution specified time

2.5.3.11 Time related attributes

The Service class has two time attributes: **Service.total_time** and **Service.critical_time**. The two attributes exist because the time the work of a service was done may be different from the critical time which the service is about. For all laboratory observations on specimen, the critical time is the “biologically relevant” time, when the specimen was taken. A measurement may be performed hours or even days after the specimen was taken. The Service.critical_time may be an interval or even a complex set of time, but for such laboratory observations on specimen the interval does not mark begin and end of the biological condition. For example, begin and end of a certain potassium concentration in the patient’s serum are unknown, the only thing that is known is that the critical time is within that unknown time interval. Therefore the critical time will often be specified as only one point in time.

For clinical observation (made directly with the patient) or for direct care procedures on the patient, the Service.critical_time will usually lie within the Service.total_time but will still not be the same. Most procedures will require some preparation time before the procedure enters its critical phase, and will require some time to unwind after the critical phase is finished. So, the critical time in surgical procedures will reflect the time between first incision and last suture, for imaging the critical time will be the time the images were actually shot, etc. The total time on the other hand will contain the preparation time and time to clean up.

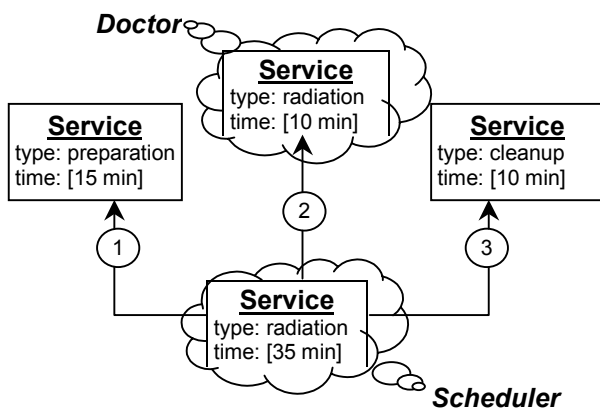


Figure 10: A service as ordered by a doctor is only part of what needs to actually happen. Typically a real service consists of preparation and cleanup work that may take just as much time as the biologically relevant part of the service. While the medical reasoning will focus on the biologically relevant aspect of the service, scheduling must consider the entire service.

A doctor who orders some therapy, such as an i.v. over 20 minutes or a radiation over 10 minutes will consider this timing to be the biologically relevant time. Almost every service will require a certain time for preparation and cleanup before and after the service. Therefore, if the service is to be scheduled, the margin around the critical phase of the service needs to be considered. Obviously there is some ambiguity in whether the service timing includes the preparation/cleanup margin or not. Figure 10 shows the general pattern of the situation.

As can be seen in Figure 10, the service ordered by the doctor and the service scheduled and performed are really two different entities. The Doctor's service is a part of the overall service to schedule. Today's systems will probably not want to clearly overcome the ambiguity between ordered medical service and performed actual service, which is why we have the biologically relevant time as an attribute.

Both biological time and service time are defined as a set of time as explained above. This allows points in time, intervals and periodic points or intervals to be specified. A doctor's order will typically contain a critical time specification. Only when the service is scheduled will it also contain a proper total time.

The **Service_relationship.pause_qty** attribute allows to specify a time duration which is to elapse after the preconditions are effective and before the action is performed. When there are no preconditions, the pause quantity will give a waiting time between the previous activity and this activity.

The **Service.max_repetitions** attribute is a generalization of a simple loop indicator. That is, whether or not the service should be executed multiple times depends chiefly on the **Service.max_repetition** attribute. By default **Service.max_repetitions** is 1, and thus, by default a service is to occur only once. When **Service.max_repetitions** is set to the positive infinity the service will repeat and the number of occurrences will be determined by the timing attribute and/or by associated conditional constraints. When **Service.max_repetitions** is a finite number greater than one, it will prohibit the number of repetitions to exceed that value, regardless whether the timing or conditional constraints would allow for more repetitions. That way, **Service.max_repetitions** has an effect on the outer bound interval of the **Service.time**, by limiting the number of occurrence intervals to that **Service.max_repetition** number.

2.6 Special kinds of Services

USAM divides actions into very coarse categories. The more common subclasses are displayed in the lower part of Figure 1. As usual, subclasses are identified mainly because different categories of actions have different basic properties, which are reflected in the attributes. Attributes of a sub-class should be both useful and unique to that sub-class. Each sub-class of action inherits the attributes described in the **Service**. The meaning of the inherited attributes may be interpreted slightly differently for each specialization of service.

In the following subsections each subclass of **Service** is described in detail. Even though a subclass may have no special attributes, it inherits all the attributes of the **Service** class. The meaning and use of the **Service** attributes vary slightly depending on the subclass. If we refer to that specialized meaning

and use of an attribute of a subclass that is inherited from Service, we will prefix the attribute name with the name of the subclass. For example, when we speak about the Service.type_cd attribute in the special context of the Observation class, we will refer to that attribute as Observation.type_cd. In the following subsections we will first describe the special attributes of a subclass and then explain the specialized meaning of inherited attributes, if there is a significant variance.

2.6.1 Observation

Observations are actions performed in order to determine an answer or *result value*. Observation result values are specific information about the observed object. The type and constraints of result values depend on the kind of action performed.

In USAM, the observation action and observation result are modeled as being the two sides of the same concept, just like the two faces of a coin are not separable from each other. Most other published healthcare models, including earlier HL7 RIM versions, separate the activity of observing and the observation result into different classes. These models label the kind of action in one class and the kind of observation result in the other. In most cases, however, the test name is a label for both activity and observation result. So, in merging action with the result, the two codes are now only one.

2.6.1.1 Observation.value : ANY

The result value of an observation action. As was true with HL7 v2, this value can be of any data type. However, there are clearly more or less reasonable choices of data types as indicated in the table below.

Table 20: Choice of observation value data types

Kind of observation	Data type	Notes
Quantitative measurements	PQ	Physical quantity (real number with unit.) This is the most usual choice. Note that numeric values must not be communicated as a simple character string (ST.)
Titer (e.g., 1:64) and other ratios (e.g. 1 out of 1000)	RTO	A ratio of two integer numbers (e.g., 1:128.) Sometimes by local conventions titers are reported as just the denominator (e.g., 32 instead of 1/32) Such conventions are confusing and should not be followed in HL7 messages.
Index (number without unit)	REAL	When a quantity does not have a proper unit, one can just send the number as a real number. Alternatively one can use a PQ with a dimensionless unit (e.g., 1 or %). An integer number should only be sent when the measurement is by definition an integer, which is an extremely rare case and then is most likely an ordinal (see below.)
Ranges (e.g., ≤ 3; 12–20)	IVL(PQ)	Interval of physical quantity. Note that sometimes such intervals are used to report the uncertainty of measurement value. For uncertainty there are dedicated data type extensions available.
Ordinals (e.g., stage “IIa”)	CV, INT	At this point, ordinals should be reported either as code values, (e.g., +, ++, +++; or I, IIa, IIb, III, IV) or as integers. In the future ordinals may be addressed by a separate data type.
Nominal results, “taxons” (e.g. organism type.)	CD	The Concept Descriptor (CD) is the most common data type to use for categorical results (e.g., diagnosis, complaint, color.) Such qualitative results are rarely simple Code Values (CV) if there is a tightly defined code system which everyone uses.
Image (still, movie)	ED	The encapsulated data type allows one to send an image (e.g., chest X-ray) or a movie (e.g., coronary angiography, cardiac echo.)
Waveform		Waveforms can be sent using the waveform template developed by the Automated Data SIG for version 2.3. A mapping onto version 3 is shown farther below. In addition one can use the Encapsulated Data (ED) type to send waveforms in other formats.

Formalized expressions	ST	The character string data type may exceptionally be used to convey formalized expressions that do not fit in any of the existing data types. However, use of the string data type is not allowed if the meaning can be represented by one of the existing data types. Note that many of the data types do have character string literal expressions too, so the field in the message can be formatted using character string literals and still have a distinct data type.
Bulk text reports	ED	A detailed procedure report should normally be sent in the attribute Service.descr. The Observation.value should be reserved for computer interpretable or automatically generated information. Note that the Encapsulated Data type (ED) can accommodate formatted text in such common formats as HTML, PDF, or Word Processor formats. The ED data type can also carry dictation that is not yet transcribed. We strongly discourage to send formatted text as result values. At the very least, the formatted text should be broken down into sections, one per sub-service object.

2.6.1.2 Observation.derivation_expr : ST

Derived observations can be defined through association with other observations using relationships of derivation type (Service_relationship.type_cd = *derivation*.) For example, to define a derived observation for Mean Corpuscular Hemoglobin (MCH) one will associate the MCH observation with an Hemoglobin (HGB) observation (Service_relationship.sequence_nmb = 1) and a Red Blood cell Count (RBC) observation (Service_relationship.sequence_nmb = 2) Since $MCH = HGB / RBC$, the value of the derivation expression would be “\$1 / \$2”.

The derivation expression is a character string with a simple syntax similar to that of the UNIX “expr” utility, or the expression subset of the PERL or TCL language. All observations that are cited in the formula must be associated with the derived observation through links of type *derivation* with a unique Service_relationship.sequence_nmb. Such observation values are referred to by that sequence number preceded by a dollar sign (\$).

Defined operators are addition (+), subtraction (−), multiplication (*) and division (/). Parentheses can be used to overcome the usual precedence (left to right, multiplication before addition.) In addition to the basic arithmetic operations the usual mathematical functions are defined.

2.6.1.3 Observation.property_cd : CV

This attribute describes the scientific property of the observation value. For quantitative observations, this is the kind of quantity. The code table to be used will represent all the concepts defined in the IUPAC *Compendium of Terminology and Nomenclature of Properties in Clinical Laboratory Sciences* (“Silver Book”.) This concept has a large intersection with the LOINC “Kind of Property” table, which is also featured by HL7 v2.3.1 as Table 0254.

2.6.1.4 Observation.type_cd : CD inherited from Service

For observations the type_cd attribute will preferably use a standard code system for observations. The Logical Observation Identifier Names and Codes (LOINC) system is currently the most widely used and most interoperable coding scheme available. Therefore it is strongly recommended to communicate a LOINC code as the primary code of the Observation.type_cd. The Concept Descriptor (CD) data type, however, allows for multiple codes to be sent as synonyms, so that local codes as well as codes from other coding systems can be sent all together. Typically, in the U.S. one will also use CPT-4 codes, in Europe EUCLIDES codes may be an option. SNOMED also provides codes for observation names.

2.6.1.4.1 Diagnoses and Allergies

Diagnoses and Allergies are kinds of observations whose `type_cd` values should be more tightly restricted, because applications must find those observations reliably, even if they lack support for multiple coding systems. This is to a certain extent also the case for complaints and problems. In earlier versions of HL7 and the RIM, diagnoses and allergies were modeled as separate classes that have been merged recently into the observation class.

Current systems however depend on diagnoses and allergies being clearly labeled as such. Therefore the following Table 21 contains codes that must be used whenever these kinds of diagnosis and allergy observations are communicated. This code must be used besides any other code system. The code symbols are deliberately constructed such that they bear meaning and so they reflect the hierarchy of concepts. Thus, simple ancillary or administrative systems can continue to extract the necessary information using their usual lexical methods, without having to consult complex terminological data bases.

Table 21: Observation type codes for diagnoses, allergies, problems and complaints.

Kind of observation	Code ⁵	Definition	LOINC
Diagnosis	DX	The most general not further specified kind of diagnosis.	
Clinical Diagnosis	DXC	A diagnosis issued by a clinician.	
Admission Diagnosis	DXA	A preliminary diagnosis issued by a clinician as a reason for admission into a hospital. May often be issued by a referring physician, reported to and entered by an admission clerk.	
Working Diagnosis	DXW	A diagnosis on which current clinical work (confirmation and treatment) is based.	
Emergency Room Diagnosis	DXWE	A working diagnosis issued in an emergency room setting. ER diagnoses are pragmatic, to determine the most effective therapeutic actions, not necessarily to make the most fine-grained diagnostic distinctions.	11301-9 11300-1
Pre-operative Diagnosis	DXWP	A concept used (only?) in cancer surgery to set a landmark on the stage of the tumor before intervention. This is the mark typically used for prognosis and treatment studies.	
Post-operative Diagnosis	DXWQ	Occurs together with the preoperative diagnosis. The intra-operative findings concerning a tumor may often revise the staging and prognosis to the worse. Post-operative diagnosis does <i>not</i> refer to the improved state achieved by the operative intervention. So, it's really an "intra-operative" diagnosis.	
Differential Diagnosis	DXD	A diagnosis that is considered as a possible alternative to the working diagnosis, but that is not primarily pursued in subsequent workup and treatment.	
Discharge Diagnosis	DXD	A discharge diagnosis is a rendition of the current working diagnosis written on the discharge letter. It reflects the clinician's best guess of the course of the inpatient encounter. Discharge diagnoses are sometimes slightly biased towards higher certainty, higher specificity and higher severity than is actually justified from the facts.	
Ancillary Diagnosis	DXX	An ancillary diagnosis is made from the standpoint of a specialist who will concentrate on his area of specialty and considering primarily the facts that are accessible to his methods.	

⁵ Note that the **Implies** column is not shown in this table, since the hierarchical implication of the general concept in the special concept is coded in the code column. We acknowledge that hierarchically meaningful code symbols is not "good vocabulary practice." Yet, for the very purpose of this table, we have determined the ease of automated interpretation more important than the generalized re-usability of the codes.

Tissue Diagnosis	DXXT	A diagnosis issued by a pathologist from histological analysis of tissue.	
Frozen Sections	DXXTF	A tissue examination from a section prepared quickly using deep frozen tissue. This examination is performed intra-operatively in cancer surgery. The surgeon interrupts the procedure and waits for the result to decide how to continue the operation.	
Final Paraffin Sections	DXXTTP	Frozen sections will only give a cursory result and is always followed by a thorough histologic examination of paraffin sections to find out the type and spread of the cancer cells more exactly.	
Radiology Diagnosis	DXXR	A diagnosis issued in the radiology department, based on imaging methods such as conventional X-ray, CAT scans, MRI scans, PET scans, etc. Ultrasonography typically is a radiology discipline in the U.S., while in European countries is done by Internists.	
Billing Diagnosis	DXB	A diagnosis made for billing purpose.	
Allergy	AL	An allergy is a diagnosis too. However, allergy information is most often stated based only on historical information gathered from the patient. Most allergies of record are not confirmed. The observation result will code one or more allergens.	
Allergy History	ALH		
Confirmed Allergy	ALC		19058-7
Symptom	SX	A subjective finding reported by the patient or a next of kin that the patient perceives as pertinent to his problem.	
Chief Complaint	SXC	The primary symptom for which the patient seeks help.	8661-1
Signs & Physical Findings	PX	Typically categorical observations made by the physician in the physical examination.	

2.6.1.4.2 Demographic observations, age, gender, and race, and their use in reference ranges.

There is a small set of medical observations which traditionally are communicated in administrative data elements (called demographics.) Typically this is gender and age, but also species, breed/race, and strain. This proposal does not aim to discourage reporting this data as part of patient “demographics.” However, we need a way to specify those observations as genuine medical observations, especially for defining patient collectives, and most notably for special population-based reference ranges.

Reference ranges are a short hand representation of a frequency distribution of a measured value over a population. Usually the “normal” healthy human is used as the reference population, but sometimes values are distributed differently in different populations, so that the population on which reference ranges are based must be identified. This used to be done in HL7 v2.3 in special fields or components using special codes and conventions. In HL7 v3 we will represent all population characteristics as observations.

Table 22: Observation type codes for demographic observations used to define populations

Kind of observation	Code	Definition	LOINC
Gender	SEX	The clinical gender. The concept repertoire includes the concepts male (M), female (F) but also many more as provided by some other clinical terminology of genders.	21840-4
Age since birth	AGEB	The elapsed time since birth. The age can not always be calculated from the date of birth, especially if the date of birth is unknown. Note that with low ages (neonates) the gestational age becomes more relevant than the age since birth.	21612-7
Gestational age	AGEG	The age since conception. Gestational age is not just applicable to an unborn fetus, but to neonates as well, for which the gestational age is a more accurate measure of age than birth	18185-9

		age. There are many methods of calculating gestational ages, which can be differentiated using other LOINC codes..
Species	SPEC	The species of a target of service. Many laboratories do blood chemistry for veterinary medicine as well as for human medicine, so the differentiation by species is quite common. The species of humans is <i>homo sapiens sapiens</i> .
Race/breed	RACE	<i>An actually or potentially interbreeding group within a species; a taxonomic category (as a subspecies) representing such a group; a division of mankind possessing traits that are transmissible by descent and sufficient to characterize it as a distinct human type.</i> [Webster Dictionary]. Race is a problematic concept since it is a very fuzzy categorization and history of man is full of racial discrimination. However, human descent is still a very important determinant for clinical conditions (e.g., 90 % of Thais have a β -Galactosidase deficiency, Africans have a significantly higher rate of Sickle Cell Anemia, etc.) For the terminology of race values, it is important to distinguish clinically relevant race from ethnicity or nationality, which is more of a cultural phenomenon (although behavioral aspects of cultures may sometimes be a determining factor for health conditions as well.)
Strain	STRAIN	<i>A group of presumed common ancestry with clear-cut physiological but usually not morphological distinctions; [...] a specified infraspecific group [...].</i> [Webster Dictionary]

For example, if we want to specify reference ranges for Aldosterone (a test to help monitor hypertension) we have to distinguish different age groups. Figure 11 shows how Aldosterone normal

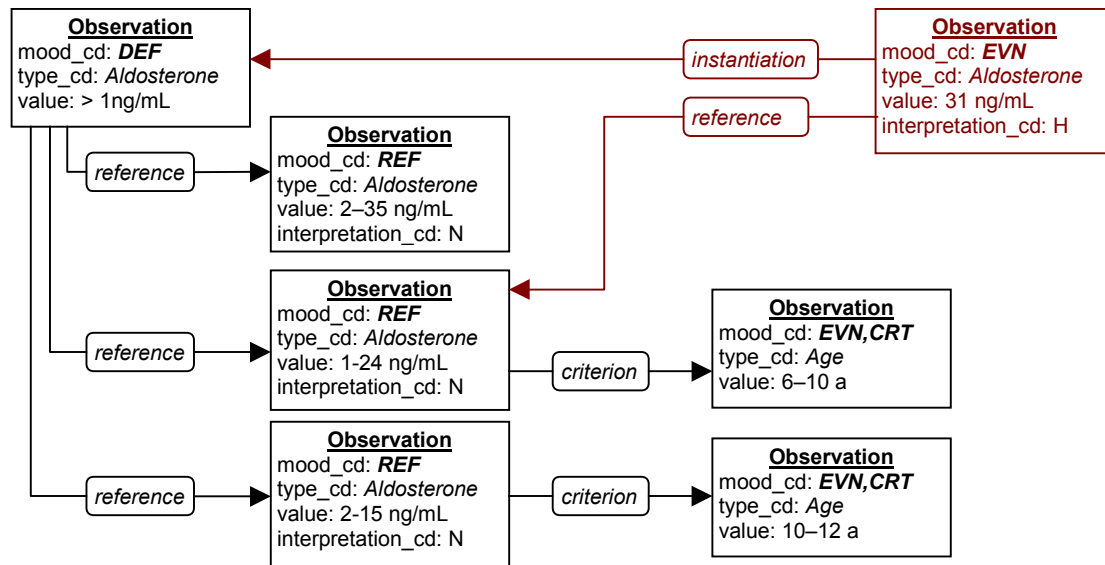


Figure 11: Reference ranges are frequency distributions of observations among populations. The reference ranges (mood: REF) are associated with the master observation (upper left, mood: DEF) through *reference* links. The reference value is usually an interval of physical quantity (low–high) or, with nominal observations, a set of value codes. If a reference range has no criterion, it is the typical “normal” range, based on the not further specified healthy population. If criteria are associated with the reference, the criteria can be any observation (mood: EVN+CRT), but sex and age are the most common reference range criteria. An actual observation (upper right, mood: EVN) may be linked with the applicable reference range in order to specify which range has been applied to determine the interpretation (abnormal) flag “H” on the service report.

values for age groups 1–10 years and 10–12 years are specified. It also shows how the applicable reference range is connected to a particular observation report.

2.6.1.5 Observation.descr : ED **inherited from Service**

In an observation report (mood_cd = actual) the attribute Observation.descr is used to store textual reports. The Observation.value field is reserved for information that is processed automatically and that is accessible to automated processes. Human authored free text reports are not easily accessible to automated processing, which is why they should be communicated in the description attribute. Of course, free text documents can be analyzed by natural language parsers and similar tools. We encourage that any output of such natural language parsers be communicated in the Observation.value attribute in the form of structured machine accessible data.

2.6.1.6 Observation.critical_time : GTS **inherited from Service**

An observation report (mode_cd = actual) contains the physiologically relevant time of an observation. In that case it is either a simple point in time or an interval of time that lies within the period for which the observation is believed to be representative. For observations on specimen this relevant time is the time of specimen collection, where it does not usually matter whether the exact start and end time of a specimen collection service are marked. The purpose of the Observation.critical_time is to indicate the time for which the assertion is valid for the patient

2.6.2 Procedure

The term procedures typically stands for surgical procedures. But the procedure class covers all direct care activities, whether performed by physicians, nurses, physiotherapy providers, etc.

2.6.2.1 Procedure.entry_site_cd : CD

All procedures other than dermatological has an anatomic site of access or entry and an anatomic site which the procedure is targeted at and that is reached through the entry site. For example an arteria pulmonalis catheter targets a pulmonary artery but the access site is typically the vena carotis interna or the vena subclavia, at the neck or the fossa subclavia respectively.

The coding system is the same as for Service.body_site.

2.6.2.2 Procedure.body_site_cd : CD **inherited from Service**

This is the anatomical target site of the procedure. For example, a pulmonary artery catheter will have the target site arteria pulmonalis with or without a known laterality.

The coding system is the same as for Service.body_site.

2.6.3 Medication

Medication is an indirect care-intervention using a material substance as a therapeutic agent. The effect of the therapeutic substance is typically established on a biochemical basis, however, that is not a requirement. For example, radiotherapy can largely be described in the same way, especially if it is a systemic therapy such as radio-iodine. Whether or not radiotherapy will be covered by a separate class is open.

Medication as a service indicates the administration of a generic class of medication to a patient. The administration of a particular preparation (in the U.S. typically represented by NDC code) requires the association of the material class with the Medication service. The material information is usually added to the order by the pharmacist when the prescription is filled as a revision or substitution to the original order.

Because medication deploys material substances, a number of attributes arguably pertain to the material rather than the procedure. At this point, we decided to allow that information to be represented in two ways, as attributes of the medication service or as attributes of the material. This problem is especially obvious with the kind of substance applied. For example, an Amoxicillin treatment can be described as `Medication.type_cd = Amoxicillin` or as `Medication.type_cd = administer` with associated Material target of type Amoxicillin. At this point naming the Service Action after the generic administered substance is the preferred strategy.

The goal is to allow simple medications to be described without having to use the Material class. Only if such actions as dispensing, or such information as the manufacturer are relevant, or if a recipe prescription is written, should one have to deploy the Material class.

2.6.3.1 Medication.form_cd : CD

The dose form of the therapeutic substance. Examples are *tablet*, *capsule*, *suppository*, etc.

2.6.3.2 Medication.route_cd : CD

The route of the medication. Medication route is similar to an anatomic body site through which the therapeutic agent is incorporated or otherwise applied to the body. It is an open issue whether a specialized `route_cd` could be replaced by a general anatomic site code. The typical routes are per os (PO), sublingual (SL), rectal (PR), per inhalationem (IH), ophtalmic (OP), nasal (NS), otic (OT), vaginal (VG), intra-dermal (ID), subcutaneous (SC), intra-venous (IV), and intra-cardial (IC).

However, as the table below suggests there are other routes and there are many variations as to how to access a specific route. For instance, an oral administration with the patient swallowing will usually have the same effect as if the same substance is given through a gastric tube. A more systematic approach to analyze the route into components such as site of primary entry (e.g. oral, nasal), site/system of substance uptake (e.g. gastrointestinal, bronchial, nasal mucosa), method (e.g., swallow, inhale), and device (e.g., gastric tube, tracheal tube) should be considered. At this point the version 2.x code table is used.

Table 23: Route of administration

Concept	Code	Concept	Code	Concept	Code
Apply Externally	AP	Intramuscular	IM	Oral	PO
Buccal	B	Intranasal	IN	Otic	OT
Dental	DT	Intraocular	IO	Perfusion	PF
Epidural	EP	Intraperitoneal	IP	Rectal	PR
Endotrachial Tube	ET	Intrasynovial	IS	Rebreather Mask	RM
Gastrostomy Tube	GTT	Intrathecal	IT	Soaked Dressing	SD
GU Irrigant	GU	Intrauterine	IU	Subcutaneous	SC
Immerse Body Part	IMR	Intravenous	IV	Sublingual	SL
Intra-arterial	IA	Mouth/Throat	MTH	Topical	TP
Intrabursal	IB	Mucous Membrane	MM	Tracheostomy	TRA

Intracardiac	IC	Nasal	NS	Transdermal	TD
Intracervical (uterus)	ICV	Nasogastric	NG	Translingual	TL
Intradermal	ID	Nasal Prongs	NP	Urethral	UR
Inhalation	IH	Nasotrachial Tube	NT	Vaginal	VG
Intrahepatic Artery	IHA	Ophthalmic	OP	Wound	WND

2.6.3.3 Medication.dose_qty : PQ

The dose is the amount of the therapeutic agent given at one administration event. This attribute can be used all by itself, or in combination with a strength. In theory, a physician's prescription could suffice with just the dose specification. For example, if Azythromycin is to be given at 80 mg once a day for three days, there is no need to specify a strength. The pharmacist can figure out the right preparation given what is available in stock or on the marketplace. When the pharmacist dispenses a particular preparation with a particular strength and packet size from a particular manufacturer, etc., this detail should be communicated using the Material class.

2.6.3.4 Medication.strength_qty : PQ default: 1

The strength of a medication is the amount of therapeutic agent per each unit of administration (entitic mass, amount of substance, etc.) If the dose form is continuously divisible (e.g., liquid, gas), the strength is a concentration (volumic mass, amount of substance, etc.)

We generally discourage using this attribute, because in theory, a physician's prescription could suffice with just the dose specification. For example, if Azythromycin is to be given at 80 mg once a day for three days, there is no need to specify a strength. The pharmacist can figure out the right preparation given what is available in stock or on the marketplace. When the pharmacist dispenses a particular preparation with a particular strength and packet size from a particular manufacturer, etc., this detail should be communicated using the Material class.

When the strength attribute is used, the actual administered amount is the product of dose_qty and strength_qty.

2.6.3.5 Medication.rate_qty : PQ ~ 1 s

With continuously divisible dose forms (e.g., liquids, gases) a dose rate can be specified. The Medication.rate_qty is specified as a physical quantity in time (a duration.) Hence, the rate_qty is really the denominator of the dose rate. For example, if an Ringer solution is to be given at 100 mL/h i.v., the dosis_qty would be 100 mL and the rate_qty would be 1 h. Note that there is no difference in the actual values of dosis_qty and rate_qty as long as the quotient of both has the same value. In this example, we could just as well specify dosis_qty as 50 mL and rate_qty as 30 min, or 200 mL and 2 h or any other combination where the quotient equals 100 mL/h.

Note that in principle one could again suffice with just the dosis_qty attribute specifying the rate right in that one attribute (e.g., dosis_qty = 100 mL/h.) However this practice is not allowed. Systems that implement the semantics of units according to the Unified Code for Units of Measure would have no problem noting the fact that a dose_qty is really a rate. Other system however will have difficulties to tell an at-once dose from a dose rate from just looking at the units. If a system wishes to deal only with a single quantity describing the dosage, it can always calculate such a quantity as

$$real_dosis_qty = dosis_qty \times strength_qty / rate_qty.$$

2.6.3.6 Medication.dose_check_qty : PQ

This attribute should not generally be used, it is only provided for a special purpose. In some countries, especially Japan, there is a regulatory requirement to note the total daily dose on the prescription and associated documentation. The purpose of this requirement obviously is to encourage and facilitate reviewing the total dose prescribed to avoid over- (or under-) dosage. Rather than to define a “total daily dose” attribute as HL7 v2.3 did, we define this general purpose dose_check_qty attribute that can be used in various ways as required by local business rules or regulations. For example, in Japan one would use this field as a total daily dose by calculating the “real” dose as noted above and then adjusting the denominator to 1 d. For example, with Erythromycin 250 mg 1 tablet 3 times a day one can calculate the total daily dose as

$$dosis_check_qty = dosis_qty (1) \times strength_qty (250 \text{ mg}) \times frequency (3 /d) = 750 \text{ mg/d.}$$

For the i.v. example above this term would be

$$dosis_check_qty = dosis_qty (100 \text{ ml}) \times strength_qty (1) / rate_qty (1 \text{ h}) = 100 \text{ mL/h}$$

which can be calculated on a daily basis as

$$dosis_check_qty = 100 \text{ mL/h} \times 24 \text{ h/d} = 2400 \text{ mL/d} = 2.4 \text{ L/d.}$$

So, in Japan, the denominator of the dosis_check_qty unit must always be 1 /d. In other countries the constraints on the dosis_check_qty may be different or, most likely, the attribute would not be used at all. In any case this dosis_check_qty attribute must not be used to carry any functional information.

2.6.3.7 Medication.method_cd : CD

inherited from Service

While there are little chances for a comprehensive method code to be available for all Services, HL7 v2.3 has defined a limited set of concepts that cover the more common methods of medication administration.

Table 24: Methods of administering medication
(from HL7 v2.3 table 0165)

Value	Description	Value	Description
CH	Chew	NB	Nebulized
DI	Dissolve	PT	Pain
DU	Dust	PF	Perfuse
IF	Infiltrate	SH	Shampoo
IS	Insert	SO	Soak
IR	Irrigate	WA	Wash
IVPB	IV Piggyback	WI	Wipe
IVP	IV Push		

Note, however, that in many cases the route implies a method quite strongly. For example, *per os* (PO) usually implies that the substance is swallowed; hence there has been no code for *swallow* as an administration method. The method is only necessary if it is different what would usually be expected. For example, if a tablet is to be taken PO, but should be chewed instead of swallowed right away, this can be indicated using the *chew* (CH) method code.

2.6.4 Condition node

The condition node service type is used to represent problems (conditions.) The primary purpose of the condition node is to arrange other services of the patient record into a longitudinal thread that represents the patient's condition. Condition nodes are lined up along the time axis through links of *type updates condition*. Thus, a Condition node instance is not a condition or problem in itself, the condition is the entire thread or network of chain-linked condition nodes.

Each condition node represents a revision of the problem in the form of added evidence, or changing of the "name" assigned to the problem. A "name" is assigned to a problem thread by a condition node that binds another observations (diagnosis) to the thread. Consequently, conditions may change their "names" over time to represent the progression of disease, and the changing of knowledge about the disease.

A condition thread may have more than one current name. Consequently, conditions may accumulate names over time as different practitioners develop opinions or descriptions of the condition. It will not be unusual that these names may be in conflict with one another, such as when two clinicians disagree about the nature of a condition. In addition, these names may also change over time to represent the progression of disease or the changing of knowledge about the disease.

There are a number of relevant service relationship types that are especially applicable to condition nodes, shown in Table 25. Please refer to Table 13 for a more complete description of relationship types.

Table 25: Service relationship types especially relevant for conditions

Concept	Code	Definition	Roles	
			source	target
updates condition	UPDT	Chains together condition nodes to a thread representing the condition.	new head of thread	old head of thread
assigns name	NAME	A support link where an observation (diagnosis) is taken as the currently representative name for the entire thread.	condition	observation (diagnosis) as name
has support	SPRT	Indicates that the condition is support for an associated conclusion or assumption.	condition	observation (diagnosis)
is cause for	CAUS	Indicates that one assumes a condition to be the cause of some observation.	condition as cause	observation as effect
is manifestation of	MFST	Indicates that one assumes some service or observation to be the cause of a condition.	condition as effect	any service as cause
has explanation	EXPL	Indicates that one explained an observation by a condition.	observation	explaining condition
has reason	RSON	Links any service to the condition thread that the service attempts to address.	action	condition node as reason
has outcome	OUTC	Links outcome assessment to a condition.	condition	outcome
has goal	GOAL	Used to declare a goal for a condition thread.	condition	goal
has risk	RISK	Used to note about a risk of the condition.	condition	risk

2.6.4.1 Condition_node.type_cd : CD

inherited, default: no information

Whereas in most other kinds of Services the type_cd is an important piece of information without which the entire Service record has almost no meaning, with condition nodes the type_cd is not very important. In fact, the type code of a condition node needs no mentioning at all. The purpose of a

condition node is chiefly to link other medical record items to a problem thread so as to provide proper attribution and context to that link.

2.6.4.2 Condition_node.mood_cd : CD inherited, default: EVN

Whereas most other kinds of Services can exist in multiple moods (e.g., intent, order, definition, actual), condition nodes usually occur in actual mode. Thus, the mood code of a condition node needs no mentioning at all, but is assumed to be actual (EVN) unless otherwise stated. There is currently no definition of the meaning and use of a Condition_node.mood_cd other than *actual*. Receiving systems should check for the mood code anyway and should raise an exception if they encounter anything else than the actual mood.

2.6.5 Consent

Obtaining informed consents is an important medico-legal activity. Consents need to be documented just as any other medical record information, with proper attribution, and all the context of who, whom, when, where, etc. The obtaining of a consent takes a considerable share of a physician's time and needs to be scheduled in a more or less formal way. The details of consents vary from procedure to procedure. Often an institution has a number of different consent forms for various kinds of procedures, that remind the physician about the topics to mention. Such forms also contain patient education material. In electronic medical record communication consents thus are information entities on their own and need to be managed similar to medical activities. Thus, consents are modeled as a special class of Services.

The "signatures" to the consent document are represented electronically through Actor instances to the consent object. Typically an informed consent has actors of type *performer* (the physician informing the patient, and *consenter*, the patient or legal guardian. Some consents may associate a *witness* or a notary public (e.g., living wills, advanced directives.) In consents where a physician is not required (e.g. living will,) the performer may be the patient himself or a notary public.

Some consents have an minimal required delay between the consent and the service, so as to allow the patient to rethink his decisions. This minimal delay can be expressed in the service definition by the service_relationship.pause_qty attribute that delays the service until the pause time has elapsed after the consent has been completed.

2.6.5.1 Consent.type_cd : CD inherited from Service

The Consent class is not only used for informed consents for invasive procedures, but include other forms of legal documentation of will and agreement too. Notable examples are advanced directive, living will, advanced beneficiary notice, etc. No terminology system exists that would include the various types of consents and other legal documents. Hence, the following code must be used where applicable. It is understood that there are very many types of consent forms in an institution and that those kinds are highly dependent on local and current legislation and business rules. Therefore, no attempt was made to list all possible consents in all detail.

The following table is only a rough categorization, to which institution wide master files would add many specializations. Those local specializations may use a local code as the Consent.type_cd in addition to this mandated code. However, note that such a local code is not required to manage different consents. If all that is needed is a unique identifier of the consent kind and consent form, the Service.id attribute is all that is needed.

Table 26: Types of consents and other legal documents

Concept	Code	Implies	Definition
procedure consent	PC		A formal informed consent that must generally be obtained from the patient or legal guardian before an invasive procedure can be carried out on the patient.
clinical trial consent	TC		An informed consent required for inclusion of a patient into a clinical trial.
advanced directive	AD		A document containing the patient's anticipated wish for how to clinically proceed in cases where the patient is critically ill and unable to make or express his own decisions.
living will	LW		A document containing the patients wish for how to proceed after the patient expires. This usually contains directives of how to distribute his belongings.
advanced beneficiary notice	AB		A document that expresses the patient's consent to bear costs of medical services that his insurance(s) may not provide coverage for. In the U.S. patient covered by Medicare are thus protected against unexpected charges they would incur for services not covered by Medicare.
necessity	AB1	AB	Service is subject to medical necessity procedures
agreed	AB2	AB	Patient has been informed of responsibility, and agrees to pay for service
ask payer	AB3	AB	Patient has been informed of responsibility, and asks that the payer be billed
against medical advice	AMA		A consent that the patient signs to understand that a permanent or transient leave from an inpatient encounter is against medical advice. Such forms are given for signature to patients to keep the provider free of liability actions in the rare case that such a leave would have negative consequences for the patient.
release of information	RI		In a release of information consent, a patient (or proxy) agrees to the disclosure of medical information to another party. The other party may be another provider to whom the patient will be referred. In another typical case the other party is a life-, health-, or disability-insurance.

2.6.5.2 Consent.descr : ED**inherited from Service**

The Consent.descr of an actual consent (mood_cd = EVN) contains all the detail of what has been consented to by the patient. For example, it could contain notes taken by a physician during the dialog with the patient (or legal guardian,) should contain questions asked and answers given to those questions, to the extent required by law and local business rules. In situations where this is legitimate and required, the Consent.descr could even contain an entire audio or even a video clip of the patient-physician dialog, or a transcript thereof. In many cases the Consent.descr could contain an electronic version of the filled consent form. This would include all the notes taken by the physician or the patient, as an electronic form, a scanned image, or just a reference to the consent form on paper.

For consents in definition mood, the consent.descr will most usually contain an electronic rendition of the consent form and necessary patient education material. Thus, when an invasive procedure is scheduled, the consent associated to that procedure as a precondition, could be automatically scheduled, and the consent form could be printed out from the data in the Consent.descr.

2.6.5.3 Consent.time : GTS**inherited from Service**

The time when the consent was created. This begins with the consent form being handed to the patient, includes the provider informing the patient, and ends with the time when the patient (or proxy) signs the consent. This time set may also contains just one singular point in time (time stamp) in which case it marks the date and time of the signature.

2.6.5.4 Consent.critical_time : GTS inherited from Service

This is the time range when the consent is valid. Many laws have consents expire after a certain period (e.g. after one year.) Furthermore, some kinds of consent are not valid immediately after signature, but require a period in which the signer may still change his mind and withdraw the consent. Hence, the Consent.critical time, for example, may be an interval spanning 24 hours to one year after the signature date.

2.6.6 Transportation

Transportation is an important support activity in the delivery of health services. Transportation is usually performed by other responsible parties than the health care providers who do the medical work on the transported payload. Therefore transportation is a service of its own right, with separate actors, separate scheduling, and separate billing.

Transportation is the moving of a payload from a location of origin to a destination location. Thus, any transport service has the three target instances of type *payload*, *origin*, and *destination*, besides the targets that are generally used for any service (i.e., *performer*, *device*, etc.)

For example, in the transport service of a patient (*John Doe*) in his bed (inventory number *1234567*), from the post-operative watch unit (*A5 west*) to the floor (*A7 north*), by the Nurse (*Jody Smith*), one would have the following actors and targets:

Table 27: Actors and targets in an example transport service

Participation	Type	Who/What
Actor	performer	Jody Smith, the nurse
Target	payload	John Doe, the patient
Target	origin	A5 west, the post-op. watch unit
Target	destination	A7 north, the floor
Target	device	bed, inventory number 1234567

Every Transport service must at least have three targets for origin, destination and payload.

2.6.6.1 Transportation.type_cd : CD inherited from Service

Since medical terminology sets will often not focus on transportation types, HL7 will maintain a code sets for those transportation types, that must be used by all HL7 compliant systems, besides local codes, that may also be used. The following table is a yet incomplete code for different kinds of transportation. The values are largely adopted form HL7 version 2.3.1.

Table 28: Types of transportation services

Concept	Code	Definition
patient transport	PAT	Any kind of patient transport
walking	WALK	Patient walks to diagnostic service
wheelchair	WHLC	Patient transported in wheelchair
cart	CART	Patient travels on cart or gurney
non-human transport		
	PORT	A portable device goes to location of use.
	VAN	An institutional van service provides transportation.

MAIL	Public postal service (for specimen)
COURIER	Using a third party express courier service

... to be continued ...

2.6.6.2 Transportation.critical_time : GTS inherited from Service

The time when the transportation actually occurred, i.e. when the payload was actually transported. This excludes the time a transporter is occupied without actually transporting the payload, e.g., time to drive to the pick-up location, and time to drive from the drop-off location back to the depot. This time set usually is one simple interval of point in time (start and end time-stamp.)

2.6.7 Supply

Supply orders and deliveries are very simple services that mainly focus on the delivered *product*. The product is associated with the supply service as a Material target of type *product* (PRD). Just as with Medication services there are in principle two ways to represent the type and identity of supplied material, i.e. as the Supply.type_cd or as the Material.type_cd of the target material (Target.type_cd = *product*.) With general supply orders the precise identification of the Material, its manufacturer, serial numbers, etc. is important, and supply services are only very marginal parts of the electronic patient record. Therefore, most of the detail information about the supply should be represented using the Material class.

Note that if delivery needs to be scheduled, tracked, and billed separately, one can associate Transportation services with the supply.

Pharmacy dispense services are represented as supply services, associated with a medication service. The medication class represents the administration of medication, while dispensing is supply.

2.6.7.1 Supply.qty : PQ

Specifies the quantity ordered or supplied (depending on the mood_cd.) This is a physical quantity (PQ) that must be from a constrained set of extensive “amount” kind of quantities. Refer to Section 2.7.1.10 for a definition of such “amount” quantities.

2.6.7.2 Supply.method_cd : CD

When a supply service represents a pharmacy dispense service, the method_cd may contain one of the following values for the dispense method. This is fully compatible with HL7 v2.3.

Table 29: Supply methods for pharmacy dispensing services (HL7 v2.3 table 0321)

Concept	Code	Definition
Traditional	TR	[we need a definition!!]
Unit Dose	UD	[we need a definition!!]
Floor Stock	F	The medication is dispensed from a stock on the care unit every day.
Automatic Dispensing	AD	[we need a definition!!]

2.6.8 Diet service

Diet services are very much like supply services, with some aspects resembling Medication services: the detail of the diet is given as a description of the Material associated as a target of type *product*. Medically relevant diet types may be communicated in the *Diet.type_cd*, however, the detail of the food supplied and the various combinations of dishes should be communicated as Material instances.

2.6.8.1 Diet.energy_qty : PQ ~ 1 kcal/d

The most important medically relevant parameter of a diet order is the supplied biologic energy (Calories) per day. This value may be specified in the *Diet.energy_qty* attribute as a physical quantity. This physical quantity should be convertible to 1 kcal/d (or 1 kJ/d.) Note, that there is a lot of confusion about what is a “calorie.” There is a “large Calorie” and a “small calorie.” On “nutrition facts” labels, the large “Calories” is used. More appropriately, however, one should use the small calorie, which is 1/1000 of a large Calorie. In the Unified Code for Units of Measure, the proper unit symbol for the large calorie is “[Ca1]” and for the small calorie it is “ca1”, or, more commonly used as a kilo-calorie “kca1”.

2.6.8.2 Diet.carbohydrate_qty : PQ ~ 1 g/d

For diabetes diet one typically restricts the amount of metabolized carbohydrates to a certain amount per day (e.g., 240 g/d). This restriction can be communicated in the *carbohydrate_qty*.

2.6.8.3 Diet.type_cd : CD inherited from Service

The following table is an incomplete set of medically relevant diet types that may be communicated in the *Diet.type_cd*. Note that details about the dishes and preparation are described in the Material class with the associated role class for Food.

Table 30: Medically relevant diet types, not including patient preferences

Concept	Code	Definition
normal diet	N	A normal diet, i.e. no special preparations or restrictions for medical reasons. This is notwithstanding any preferences the patient might have regarding special foods, such as vegetarian, kosher, etc.
(we call it “schonkost” in German)	SCH	A diet that avoids ingredients that might cause digestion problems, e.g., avoid excessive fat, avoid too much fiber (cabbage, peace, beans.)
(we call it “breikost” in German)	BR	A diet exclusively composed of oatmeal, semolina, or rice, so as to be extremely easy to eat and digest.
liquid	LQ	A strictly liquid diet, that can be fully absorbed in the intestine, and therefore may not contain fiber. Used before enteral surgeries.
tea only	T	This is not really a diet, since it contains little nutritional value, but is essentially just water. Used before coloscopy examinations.
fasting	FAST	No enteral intake of foot or liquids whatsoever, no smoking. Typically 6 to 8 hours before anesthesia.
diabetes mellitus diet	DM	A diet that uses carbohydrates sparingly. Typically with a restriction in daily energy content (e.g. 1600–2000 kcal.)
reduction diet	RD	A diet that seeks to reduce body fat, typically low energy content (800–1600 kcal.)
parenteral	PAR	Patient is supplied with parenteral nutrition, typically described in terms of i.v. medications.
low fat	LF	A diet low in fat, particularly to patients with hepatic diseases.

no fat	NF	A no fat diet for acute hepatic diseases.
low sodium	LS	A diet low in sodium for patients with congestive heart failure and/or renal failure.
low protein	LP	A low protein diet for patients with renal failure.
gluten free	GF	Gluten free diet for celiac disease.
phenylalanine free	PAF	Phenylketonuria diet.
low valine, leucine, isoleucine	VLI	Diet with low content of the amino-acids valin, leucin, and isoleucin, for "maplesirup disease."

2.6.8.4 Diet.method_cd : CD inherited from Service

Diet may need to be scrambled and may need to be applied through some gastric tube. This can be described using the method_cd attribute and as associated Materials representing access routes, such as naso-esophageo-gastric or -duodenal tube or a percutaneous endoscopic gastrotomy (PEG) tube.

2.7 Substance: the Material class

The Unified² Service Action Model divides the world into Substance and Actions with some glue classes (roles, participations, relationships) between them. The sections above have all dealt in depth with Actions. In this section we will turn to Substances. At this point, the HL7 RIM defines the classes Person and Organization with a common generalization called "Stakeholder". Although Stakeholder, Person and Organizations are classes that we referred to as part of the Universe of Substance, most Substance does not have legal rights and responsibilities.

We find that substance that is not considered a "Stakeholder" is much less organized in the current RIM than is the area of people and organizations. For example, we find such classes as Collected_specimen_sample, Durable_medical_equipment, Patient_service_location, and also Living_subject, scattered in the model, unrelated to each other, with each having their own set of relationships to other classes. This is a problem because all these classes are used and acted upon by Services. At this point, the Target_participation optionally links to any one of those many substance classes. In addition there is a number of other relationships from Service to, e.g., location, and between the scattered substance classes. This is troublesome, because there is more regularity and more system to these classes than the current model suggest.

The Unified Service Action Model therefore suggest the creation of a class Material, that assumes all the common attributes of substance. Webster's definitions for material are shown in Exhibit 4 and Exhibit 5 for reference. As can be seen the term "material" has a fairly broad meaning which we intentionally evoke for this Material class.

2.7.1 Attributes of class Material

2.7.1.1 Material.id : SET(II)

As a substantive class reflecting physical entities, material has instance identifiers. Note that an instance identifier is a pure identifier and not a classifier. That means, this identifier is not used to store information about what kind or type of material this is. Ideally each entity will have only one identifier assigned to it, however, since different systems will maintain different material data bases, there may be different instance identifiers assigned by different systems.

Note that for serial numbers assigned by specific manufacturers, catalog numbers of specific distributors, or for inventory numbers issued by owners, the attribute Responsibility.material_id :

Main Entry: ¹**ma·te·ri·al**

Pronunciation: m&-tir-E-&l

Function: *adjective*

Etymology: Middle English *materiel*, from Middle French & Late Latin; Middle French, from Late Latin *materialis*, from Latin *materia* matter – more at MATTER

Date: 14th century

1 a (1) : relating to, derived from, or consisting of matter; *especially* : PHYSICAL ⟨the *material* world⟩ (2) : BODILY ⟨*material* needs⟩ **b** (1) : of or relating to matter rather than form ⟨*material* cause⟩ (2) : of or relating to the subject matter of reasoning; *especially* : EMPIRICAL ⟨*material* knowledge⟩

2 : having real importance or great consequences ⟨facts *material* to the investigation⟩

3 a : being of a physical or worldly nature **b** : relating to or concerned with physical rather than spiritual or intellectual things ⟨*material* progress⟩

- **ma·te·ri·al·ly** /-E-&-IE/ *adverb*

- **ma·te·ri·al·ness** *noun*

synonyms MATERIAL, PHYSICAL, CORPOREAL, PHENOMENAL, SENSIBLE, OBJECTIVE mean of or belonging to actuality. MATERIAL implies formation out of tangible matter; used in contrast with spiritual or ideal it may connote the mundane, crass, or grasping ⟨*material* values⟩. PHYSICAL applies to what is perceived directly by the senses and may contrast with mental, spiritual, or imaginary ⟨the *physical* benefits of exercise⟩. CORPOREAL implies having the tangible qualities of a body such as shape, size, or resistance to force ⟨artists have portrayed angels as *corporeal* beings⟩. PHENOMENAL applies to what is known or perceived through the senses rather than by intuition or rational deduction ⟨scientists concerned with the *phenomenal* world⟩. SENSIBLE stresses the capability of readily or forcibly impressing the senses ⟨the earth's rotation is not *sensible* to us⟩. OBJECTIVE may stress material or independent existence apart from a subject perceiving it ⟨no *objective* evidence of damage⟩. **synonym** see in addition RELEVANT

Exhibit 4: Webster's definition of "material".

SET⟨II⟩ can also be used. This allows to more clearly express the fact that such a code is assigned by a specific party associated with that material. In any case, all values of Responsibility.material_id may occur in Material.id just as well.

2.7.1.2 Material.type_cd : CD

This code describes what kind of material this is. It is an arbitrarily precise classification. We do not expect any single terminology to provide all concepts that are types of material, since it is simply too broad a domain. Instead of limiting the Material.type_cd to a single domain, we allow various code systems to be used, and thus, the actual domain of Material.type_cd becomes the union of all the possible code systems for material.

For example, specimen types (e.g., whole blood, serum, urine) can be used in this attribute. For chemicals, IUPAC codes might be used here. For arbitrary products one can use the Universal Product Code (UPC) code or a particular manufacturer's serial number. For pharmacological substances yet another coding system may be applicable such as the U.S. National Drug Code (NDC.) The concept descriptor data type allows for multiple codes used as synonyms for each other, thus, one can

Main Entry: ²**material**

Function: *noun*

Date: 1556

1 a (1) : the elements, constituents, or substances of which something is composed or can be made (2) : matter that has qualities which give it individuality and by which it may be categorized ⟨*sticky material*⟩ ⟨*explosive materials*⟩ **b** (1) : something (as data) that may be worked into a more finished form ⟨*material* for a biography⟩ (2) : something used for or made the object of study ⟨*material* for the next semester⟩ (3) : a performer's repertoire ⟨a comedian's *material*⟩ **c** : MATTER **3b d** : CLOTH **e** : a person potentially suited to some pursuit ⟨*varsity material*⟩ ⟨*leadership material*⟩ **2 a** : apparatus necessary for doing or making something ⟨*writing materials*⟩ **b** : MATÉRIEL

Exhibit 5: Webster's definition of "material" cont'd.

specify an UPC code next to an NDC code and an IUPAC code.

2.7.1.3 Material.form_cd : CV

This is a classifier describing the form of the material. This includes the typical state of matter (solid, liquid, gas) and, for therapeutic substances, the dose form. The following concept repertoire is applicable:

Table 31: Concept repertoire for forms of material

Concept	Implies	Code	Definition
continuous		CNT	Continuously divisible form, typically amorphous. Continuously divisible material has typically no identity and comes in quantities measured as mass or volume, etc.
powder	continuous	PWD	A grained or powdered crystalline substance in solid state.
liquid	continuous	LQD	Substance is typically in liquid state.
gas	continuous	GAS	Substance is typically in gas state.
integral		INT	Integral solid form that can not be broken into pieces without destroying the form. Typically has a fixed shape. In the pharmacy this is sometimes called "eaches."
tablet	integral	TAB	A tablet (Note: tablets can be broken into pieces of $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, but not much less.)
capsule	integral	CAPS	A capsule (a container.)
suppository	integral	SUPP	Used to apply medication rectally.
vial	integral	VIAL	A container made from all-glass and closed through melting. Filled with crystalline powder or liquid.
bottle	integral	BTTL	A container closed by a cover.
... more ...			

Note that the above table is not complete. It needs to incorporate all medication dose forms.

2.7.1.4 Material.descr : ED

A free text description of the material. May contain multimedia, such as a drawing or image depicting the material.

2.7.1.5 Material.status_cd : SET(CV)

The status_cd tracks the state of the state-transition model of the material. This may be a rather trivial state-transition model, since the more concrete and detailed state-transition models may be assigned to the material role classes.

2.7.1.6 Material.extent_tmr : IVL(TS)

The time interval a certain material is in existence. The high boundary of this interval is the expiration date if it is defined for the material.

Expiration dates does not always have a "day" component; therefore, such a date may be transmitted as *YYYYMM*.

2.7.1.7 Material.lot_nmb : ST

The lot number is the number printed on the label attached to the container holding the substance and on the packaging which houses the container. A “lot” is a collection of products produced in one cycle. This means, for instance, if one bottle of a lot is spoiled, chances are high that the entire lot is spoiled. Conversely, product defects that occur in routine production are likely to be contained in one lot.

Note that a lot number is not meant to be a unique identifier, but is meaningful only when the product kind is identified.

2.7.1.8 Material.handling_cd : CD

A code to describe how the material needs to be handled to avoid damage.

Table 32: Material handling code

Concept	Implies	Code	Definition
room temperature		RMT	Keep at room temperature, about 20 °C
body temperature		BDT	Keep at body temperature, about 36 to 37 °C
cool		COO	Keep cool at about 5 to 8 °C
frozen		FRZ	Keep frozen below 0 °C
deep frozen		DFR	Keep deep frozen, below – 16 °C
nitrogen		NTR	Keep in liquid nitrogen
dry		DRY	Keep in a dry environment
dark		DRK	Protect against light
no shock		PSO	Protect against shock
upright		UPR	Keep upright, do not turn upside down
no shake		PSA	Do not shake
... more ...			

2.7.1.9 Material.danger_cd : CD

A code signaling whether there are certain dangers or hazards associated with this material.

Table 33: Material danger code

Concept	Implies	Code	Definition
tissue		TIS	The normal dangers associated with normal human or animal tissue. I.e. potential risk of unknown infections. Routine blood or excretions of humans and animals.
infectious		INF	Material known to be infectious with human pathogenic microorganisms. Those who handle this material must take precautions for their protection.
biohazard	infectious	BHZ	Material contains microorganisms that is an environmental hazard. Must be handled with special care.
radioactive		RAD	Material is a source for ionizing radiation and must be handled with special care to avoid injury of those who handle it and to avoid environmental hazards.
poison		POI	Material is poisonous to humans. Special care must be taken to avoid incorporation, even of small amounts.
acid		ACI	Material is acid and may cause severe injury to human skin and eyes. Avoid any unprotected contact.

inflammable		IFL	Material is highly inflammable and in certain mixtures (with air) may lead to explosions. Keep away from fire, sparks and excessive heat.
explosive	inflammable	EXP	Material is an explosive mixture. Keep away from fire, sparks, and heat.

2.7.1.10 Material.qty : SET<PQ>

default: {1}

For many materials, the individual thing has no relevance. Especially continuously divisible forms come only in “amounts” rather than as individuals. There is a specific class of physical quantities that can be used for amounts, count (number), amount of substance, mass, and volume. This class of physical quantities is called “extensive” quantities. A quantity is called extensive if it can be added up (if it is additive.) For example, if you have 1 gallon of water and you add another gallon of water, you have two gallons of water, since volume is an additive quantity. By contrast, if you have one gallon of Glucose 5% and add to it another gallon of Glucose 5% you still have Glucose 5%, thus, mass fraction is not an additive (extensive) kind of quantity.

Only extensive quantities are permitted as elements of the Material.qty set. Typically the kinds of quantities shown in Table 34 will occur. Extensive quantities are simpler to deal with than intensive quantities. Extensive quantities are never fractions or ratios, no denominator can cancel out the units of a numerator, and therefore, with extensive quantities we can conclude the kind of quantity from the unit of measure.

Table 34: Kinds of quantities for amounts of material

Kind of quantity	Typical Unit	Forms	Examples
Number	1	solid	Material that is large enough that it can be counted (“eaches”)
Mass	1 g	liquid, solid	Tissue, chemical substances, food.
Amount of substance	1 mol	all	Chemical substances, small particles.
Volume	1 L	liquid, gas	Chemical substances in liquid and gas state. Amorphous tissue.
Length	1 m	solid	Long material measured in length, e.g., tape, pipes, hose, etc.
Area	1 m ²	solid	Flat material measured in area, e.g., covers, foils, etc.
Energy	1 J, 1 kcal	solid, liquid	Chemical substances, especially food.
Catalytic amount	1 kat, 1 U, 1 i.U.	all	Enzymes and other chemical substances having catalytic activity.
Radioactivity	1 Bq, 1 Ci	all	Radioactive substances.
Reaction equivalent	1 Eq	all	Ionized chemical substances measured through titration. Deprecated, use proper amount of substance instead.

The Material.qty attribute permits to convey a collection of physical quantities. This collection feature must be used in the following way. When the set contains more than one quantity, the quantities must have different units. Furthermore, all quantities in the set must denote an equivalent amount. For example, for the material Glucose, we may specify an amount as the mass of 1 g. If we also want to specify the amount in amount of substance (moles) we must specify the equivalent of 1 g Glucose in mole, which is 5.556 mmol. For another example, if we specify the amount of a material Water as 1 L, and we want to provide a mass, the mass must be the mass of 1 L water, which is 1 kg.

By specifying the amount in multiple units representing multiple kinds of (extensive) quantities, we not only allow for flexibility. This brings about a simple yet powerful way to represent material constants, such as molar mass, molar volume, mass density, biologic energy content, etc. So, if we specify mass, amount of substance, volume and energy content of a substance, we can convert to any of those kinds of quantities given any other quantity.

2.7.2 Material relationships to other Material

Material relates to other material largely in some kind of whole-part or containment relationship. The special functioning of the material relationship depends on the role of material, i.e. whether the material is an discrete thing, a homogenous substance, a container, or a location. Material can be all of those forms which is explained in Section 2.8 below.

Analogous to the service relationship, the material relationship is a directed link between material entities. This means, the relationship is like an arrow with a butt and a point. The entity at the side of the butt is called the *source*, and the entity at the point is called the *target* of the relationship.

The following attributes can be ascribed to a material relationship.

2.7.2.1 Material_relationship.type_cd : CV

Material relationships can be of different types, i.e., may express different kinds of relationships. The relationship concepts are exhaustively defined in Table 35, that is, the concepts of that table must be used.

Every relationship type implies certain roles for the material at each side of the relationship. The notion of roles in a material relationship is very similar to material roles as defined in Section 2.8 below. Where in Table 35 the roles are so generic that they are not represented as a material role class in the model, that generic role name is printed in *italics*. Role names in upright font refer to the same concept as represented by the material role class of the same name. In general a material filling that role should be accompanied by the detail defined in the role class, but it is not an absolute requirement. For example, if a material is taken as a container but none of the container-specific attributes are applicable, the instance of the Container role class need not be present.

Table 35: Material relationship types

Concept	Code	Implies	Definition	meaning of the service	
				source	target
has part	PART		Relates a whole to its parts. A part may be an ingredient that is not separable from the whole, or a discrete part that may be identified separately and may, in principle, be disassembled from the part.	<i>whole</i>	<i>part</i>
has ingredient	INGR	PART	Relates a component to a mixture. E.g., Glucose and Water are ingredients of D5W, latex may be an ingredient in a tracheal tube.	<i>any material</i>	<i>any material</i>
has base	BASE	INGR	A base ingredient is what comprises the major part. E.g., Water in most i.v. solutions, or Vaseline in salves. Among all ingredients of a material, there should be only one base. A base substance that in turn be a mixture, e.g. base: 500 ml bottle D5W, additive: KCl 20 mmol.	<i>any material</i>	<i>any material</i>
has additive	ADTV	INGR	An ingredient that is added to a base, that amounts to a minor part of the overall mixture.	<i>any material</i>	<i>any material</i>
has active ingredient	ACTI	INGR	A therapeutically active ingredient in a mixture, where the mixture is typically a manufactured pharmaceutical.	therapeutic agent	therapeutic agent
has stabilizer	STBL	ADTV	A stabilizer is a substance added to a mixture in order to prevent the molecular disintegration	<i>any material</i>	<i>any material</i>

			of the main substance.		
has preservative	PRSV	ADTV	A substance added to a mixture to prevent microorganisms (fungi, bacteria) to spoil the mixture.	<i>any material</i>	<i>any material</i>
has flavor	FLVR	ADTV	A substance added to a mixture to make it taste a certain way. In food the use is obvious, in pharmaceuticals flavors can hide disgusting taste of the active ingredient (important in pediatric treatments.)	<i>any material</i>	<i>any material</i>
has color	COLR	ADTV	A substance influencing the optical aspect of material.	<i>any material</i>	<i>any material</i>
has content	CONT		Relates a material as the content to a container. Unlike ingredients, the content and a container remain separate (not mixed) and the content can be removed from the container. A content is not part of an empty container.	container	<i>any material</i>
has presence	PRSN		Relates any material to a location at which it is present in some way. This presence may be limited in time.	<i>any material</i>	location
has depot	DEPO	PRSN	Relates a material (e.g. a device) to a location at which it is normally found or stored when not used.	<i>any material</i>	location
has species	SPEC		Relates a generalized material concept to its specialization.	any material as genus	any material as species
has generic		SPEC ⁻¹	A special link between pharmaceuticals indicating that the target is a generic for the source.	therap. agent as brand	therap. agent as generic

2.7.2.2 Material_relationship.inversion_ind : BL default: false

The role type may be used in the opposite direction.

For example, instead of listing a material instance representing a mixture and subordinate to it mentioning the ingredients as target material instances, one can use one ingredient and subordinate to it mention the mixture in which it happens to exist. This is the common way of thinking of pharmaceuticals. In most pharmaceuticals, we have one main ingredient which we consider “therapeutically active” and which we mention, although we know that this substance always comes as an ingredient of a mixture containing diluents, stabilizers, preservatives, flavors and colors. This active ingredient can then be specified as the top material instance → *inverted ingredient* → mixture → *ingredient* → other ingredients.

Another notable example for inversion of the relationship type is for containers. The *content* relationship type allows one to first list a container (e.g. package) and then provide a list of content as subordinate (target) material. In other cases, one wants to mention the material first and by the way describe it being contained in a container. Therefore, when the content is the important thing and the container just goes with it (e.g., for most medications,) one will use the inverted content link.

2.7.2.3 Material_relationship.tmr : IVL<TS>

For some transient relationships between material one can specify a time in which the relationship is valid using the Material_relationship.tmr attribute. As with any interval of points in time, a start time, an end time, or a just a duration may be specified.

2.7.2.4 **Material_relationship.position_nmb : LIST<NM>**

Some containers have discrete positions in which content may be located. Depending on the geometry of the container, the position may be referenced as a scalar ordinal number, or as a vector of ordinal numbers (coordinates.) Coordinates always begin counting at 1.

Some containers may have customary ways of referring to the positions. Take a checkboard, for example, in which rows are specified A-H and columns specified 1-8. In these cases, the non-numeric coordinate must be converted into a numeric. In the absence of any specific regulation for a specific container type, the rule of thumb is that the coordinate that is changed earlier is positioned first. For the checkboard example, this means that the columns are changed or traversed first. When you start placing the figures in the start position, you chiefly align them in the columns, and only then you start moving them ahead in rows (and columns too.)

For an automated blood chemistry analyzer, with a square shaped tray, this means that the first coordinate is the one in which direction the tray moves at each step. Whereas the second coordinate is the one in which the tray moves only every 10 (or so) steps.

As a final example, the positions on a computer screen that works in usual left-to-right and top-to-bottom direction, the columns would be the first coordinate and the lines would be the second coordinate. (Note however, that this is just an example to clarify the rule. It does not mean that a character displayed on a screen would be an instance of the Material class. In fact, it's immaterial.)

2.7.2.5 **Material_relationship.qty : PQ**

This attribute specifies how much of the target material is contained in the source material of a relationship. For example, if a box contains 10 eggs, the box is the relationship source is the box and the relationship target is the egg, where the relationship quantity is 10. For mixtures with multiple ingredients, the relationship quantities specify the relative amounts of the ingredients in the mixture (proportion.)

The quantity must be a quantity that specifies an "amount" (refer to Table 34 in Section 2.7.1.10). The amounts specified as the proportion quantity for each ingredient are taken to be numerators over the same denominator. For example, D5W is a mixture consisting Water (H₂O) and 5% (= 50 g/L) Glucose (Glc.) The proportions can be either of the following pairs: H₂O:1 g + Glc:50 mg; H₂O:1 L + Glc:50 g; H₂O:500 mL + Glc:25 g; or any combination that amounts to the same concentration of Glucose in Water.

Note that the value of the proportion quantity does not matter as long as the proportion between the ingredients of a substance is kept invariant. If, for example, we specify D5W as having ingredients 500 mL of H₂O and 25 g of Glucose this does **not** mean that D5W could only be dispensed in multiples of 500 mL.

The benefit of specifying the proportion in terms of amounts is that it is simple and straightforward, and there is no ambiguity that we often face with intensive measures, such as concentrations, mass fractions vs. mass ratios, etc. For example, the unit percent (%) is ambiguous, since it could be a mass fraction or a volume fraction or any kind of ratio. All ratios are ambiguous since one needs to know what is the numerator substance and what the denominator substance. This ambiguity is all removed by specifying the proportion in terms of extensive measures (additive amounts.)

2.7.3 Responsibilities of Stakeholders for Material

Material can have many kinds of relationships with Stakeholders. We subsume all the relationships between material and stakeholders under the notion of Responsibility. The reason being that responsibility for the existence of material, any specific property of material, or performance of functional material (devices) is with some stakeholder. The underlying reason for stakeholder associations to material is that the material is somehow acted upon by the stakeholders. In that sense, one could subsume the Responsibility association under the Service action class. However, just as we chose to represent minor sub-activities around Services as Actors with various actor types, we allow the responsibilities that come from actions of stakeholders to be persistently “coined” on the material.

For example, manufacturing is certainly an activity (Service) with the manufacturer (Organization) as an Actor and the material as a Target of type product. However, in many cases we are not interested in the activity of manufacturing the material, when it took place and what its circumstances were, but what we are interested in is just: who made it? This interest in the manufacturer is chiefly one of responsibility and liability: if the material is different than expected, does not perform well, or does harm, one would probably consider holding the manufacturer liable. Responsibility and liability are concepts that form the very basis of a society based on the law, and emphasis on those terms should by no means imply an undue “legalization” of relationships.

Other relationship types between Material and Stakeholder are: owner, distributor, custodian/holder. All those relationships can be considered to be characterized by responsibilities. This even goes so far as **if** a human fetus would be considered Material, motherhood (and fatherhood!) would be a type of Responsibility between a Stakeholder (Person) and that fetus. This example shows that responsibility has two aspects: responsibility is not only being held liable by others for malfunctioning, disappointment, and harm caused by the material; responsibility also means an ethical responsibility towards the “material” and even to the extent of being held liable by society for neglect of one’s responsibility towards that “material.” This latter kind of responsibility is clearly present between fetus and parent, but also between animal and owner or custodian.

2.7.3.1 Responsibility.type_cd : CV

Specifies the kind of responsibility of the Stakeholder to the Material.

Table 36: Material responsibility type code

Concept	Implies	Code	Definition
manufacturer		MAN	Someone bringing a specific material instance into existence, or, if the material is not a specific instance, someone capable of doing so.
distributor		DST	Someone distributing material between a manufacturer and a buyer or retailer.
retailer	distributor	RET	Someone selling a material, also giving advice to prospective buyers.
transporter		TRP	Someone in transient possession of a material for the purpose of relocating it.
owner		OWN	Someone to whom law grants the right to call a material his own, which entitles him to make decisions about the disposition of that material.
holder		HLD	Someone who is currently in possession of the material, who holds, or uses it, usually based on some agreement with the owner.
trainer		TRN	Of a companion animal, someone who is training the animal on behalf of the animal’s owner.
parent		PRN	One of the two direct ancestors of a human fetus, in case a fetus is not considered a person.
father	parent	FTH	The male parent of a human fetus, in case a fetus is not considered a person.
mother	parent	MTH	The female parent of a human fetus, in case a fetus is not considered a person.

2.7.3.2 Responsibility.tmr : IVL<TS>

Allows to specify a limitation in time during which the responsibility holds.

2.7.3.3 Responsibility.material_id : SET<II>

The same piece of material may be given different identifiers by different responsible parties. For example, a manufacturer may assign a manufacturer id, a distributor may assign a catalog number, etc. All those identifiers can in principle occur under the Material.id attribute, i.e., as a property of the material itself. However, this attribute allows to make the scope of the id more clear, i.e. it helps to easily distinguish a specific manufacturer's id from a distributor's id much more directly and obvious as can be done using the assigning authority component of the instance identifier data type.

2.8 The Roles of Material

Material is used in different roles. We could have modeled the roles of material similar to the subtypes of Services (as specializations.) Role and specialization are similar in many ways. Notably there is a kind of "inheritance" of properties from a substance to a role, just as there is inheritance of properties from a general class to its specializations. A role can never exist without a substance that takes on that role, thus, all genuine properties of the substance are available regardless of which role it takes on. This is important because in the following we use the same convention to describe the interpretation of a Material attribute from the standpoint of a particular role.

The one important difference between a role class of a substantive class and a specialization of a general class is that specializations are exclusive whereas roles are inclusive. For example, the same substance (e.g., leaves of the eucalyptus plant) may be considered food and at the same time may be considered a therapeutic agent. Or, a bottle with an attached applicator is a container, and at the same time, is a device for administering the content of the container. Finally, an ambulance is a device for transportation, but at the same time, it may be a health care location (facility.)

Note that the role classes proposed for material are not very heavily used. If no strong properties are defined for these role classes in the near future, one can consider deleting them from the model. The same comment can be made on the subclasses of Service. At this point, the role classes are here to illustrate a principle. In the future, either we will use these role classes more strongly, or we may delete them from the model.

2.8.1 Specimen

According to Webster's dictionary, a specimen is "an individual, item, or part considered typical of a group, class, or whole" or "a portion or quantity of material for use in testing, examination, or study." In the practice of clinical medicine and especially in previous HL7 specifications, specimen was tightly related to the container which holds the specimen. However, there is an important difference between a container and a specimen. Through the material class with roles for both specimen and container one can manage containers separately from specimen. With the same class one can manage empty specimen containers (material management) the same way as the container filled with specimen.

In the prior models that merged specimen and container as the same thing, there are relatively few properties pertaining to the specimen itself as compared to the container. Only specimen type, source body site (or anatomical system,) quantity (typically volume,) and handling instructions pertain genuinely to the specimen. One can argue whether some of those parameters even pertain to the specimen collection activity rather than the specimen itself. For example, for a peripheral venous

blood sample it does not really matter whether it has been taken from the vena cephalica, the cubital venes, vena jugularis, vena femoralis, or any other peripheral vein.

2.8.1.1 Specimen.body_site_cd : CD

Body site has been retained as an attribute of the specimen, since it may be relevant in some cases, e.g., if multiple liver needle biopsies are taken from different lobes and locations of the liver. The value of the Specimen.body_site_cd should be identical to the value of the Service.body_site_cd of an associated specimen collection service. This attribute therefore is used only if such an associated specimen collection service is not communicated. When the rule is to always send a specimen along with a record of the specimen collection service, this attribute needs not and should not be valued.

2.8.1.2 Material.type_cd : CD

For material in the role of specimen, the Material.type_cd is a specimen type code. HL7 does not principally prescribe the coding system used for the material type code. However, the following concepts must be provided in the concept descriptor for specimen where applicable. The concept descriptor allows for arbitrarily many code translations, so one of the code translations must be a code from the table below. This table is taken from the LOINC 1.0M, and is largely the same as the specimen source table of HL7 v2.3.1 (with few exceptions.) We have removed one item from the LOINC table, which is “XXX – specimen type specified elsewhere,” since it is not applicable in the only field available for specimen type.

Table 37: Material type codes for specimen

Concept	Code	Concept	Code	Concept	Code
Abcess	ABS	Fistula	FIST	Seminal fluid	SMN
Amniotic fluid	AMN	Body fluid, unsp	FLU	Seminal plasma	SMPLS
Aspirate	ASP	Food sample	FOOD	Serum	SER
Basophils	BPH	Gas	GAS	Skin	SKN
Bile fluid	BIFL	Gastric fluid/contents	GAST	Skeletal muscle	SKM
Blood arterial	BLDA	Genital	GEN	Spermatozoa	SPRM
Blood bag	BBL	Genital cervix	GENC	Sputum	SPT
Blood capillary	BLDC	Genital fluid	GENF	Sputum - coughed	SPTC
Blood – cord	BLDCO	Genital lochia	GENL	Sputum - tracheal aspirate	SPTT
Blood product unit	BPU	Genital vaginal	GENV	Stone (use CALC)	STON
Blood venous	BLDV	Hair	HAR	Stool = Fecal	STL
Bone	BON	Inhaled Gas	IHG	Sweat	SWT
Breath (use EXG)	BRTH	Intubation tube	IT	Synovial fluid (Joint fluid)	SNV
Bronchial	BRO	Isolate	ISLT	Tears	TEAR
Burn	BRN	Lamella	LAM	Throat	THRT
Calculus (=Stone)	CALC	Leukocytes	WBC	Thrombocyte (platelet)	THRB
Cardiac muscle	CDM	Line	LN	Tissue, unspecified	TISS
Cannula	CNL	Line arterial	LNA	Tissue gall bladder	TISG
Catheter tip	CTP	Line venous	LNV	Tissue large intestine	TLGI
Cerebral spinal fluid	CSF	Liquid NOS	LIQ	Tissue lung	TLNG
Cervical mucus	CVM	Lymphocytes	LYM	Tissue placenta	TISPL

Cervix	CVX	Macrophages	MAC	Tissue small intestine Tissue ulcer	TSMI
Colostrum	COL	Marrow (bone)	MAR	Tissue ulcer	TISU
Conjunctiva	CNJT	Meconium	MEC	Tube, unspecified	TUB
Curette	CUR	Menstrual blood	MBLD	Ulcer	ULC
Cornea	CRN	Milk	MLK	Umbilical blood	UMB
Cyst	CYST	Breast milk	MILK	Unknown medicine	UMED
Dialysis fluid	DIAF	Nail	NAIL	Urethra	URTH
Dose med or substance	DOSE	Nose (nasal passage)	NOS	Urine	UR
Drain	DRN	Other	ORH	Urine clean catch	URC
Duodenal fluid	DUFL	Pancreatic fluid	PAFL	Urine catheter	URT
Ear	EAR	Patient	PAT	Urine sediment	URNS
Ear wax (cerumen)	EARW	Peritoneal fluid /ascites	PRT	Unknown substance	USUB
Electrode	ELT	Placenta	PLC	Vomitus	VOM
Endocardium	ENDC	Plasma	PLAS	Whole blood	BLD
Endometrium	ENDM	Plasma bag	PLB	Whole body	BDY
Eosinophils	EOS	Pleural fluid (thoracentesis fld)	PLR	Water	WAT
Erythrocytes	RBC	Polymorphonuclear neutrophils	PMN	Wick	WICK
Eye	EYE	Platelet poor plasma	PPP	Wound	WND
Exhaled gas (=breath)	EXG	Platelet rich plasma	PRP	Wound abscess	WNDA
Fibroblasts	FIB	Pus	PUS	Wound exudate	WNDE
Filter	FLT	Saliva	SAL	Wound drainage	WNDD

2.8.1.3 Material.extent_tmr : IVL<TS>

As with any other material, the extent specifies the time a material exists. With specimen, the low bound of the extent interval is especially important as the time the specimen was collected, or derived. Most chemistry lab specimen are disposed after use, and the time of disposal would be the high bound of the extent time range. In anatomic pathology many specimen are frozen and kept for a long time, in which case most specimen records will not have a value for the high boundary of the extent time range.

2.8.1.4 Material.qty : SET<PQ>

default: {1}

This is the amount of specimen. This attribute is mostly used for continuous forms, such as liquids, gases, or soil. In veterinary medicine, a number of animals may be taken as a specimen for a large population. Again the individual animal in such a set has no relevance, just as the individual grain of sand has no relevance in a soil sample, or the individual erythrocyte has no relevance in a blood sample.

Note that for an integral thing taken as a specimen, the Material.qty is 1. In these cases, the material quantity should not be used to communicate simple observations on that individual specimen. For example, if one Appendix has been sent to anatomical pathology, the length, mass, etc. would *not* be recorded in this field. Length, mass, and other measurements on an individual item must be represented as observations.

2.8.2 Container

The design of the Container role class is heavily influenced by the use of a specimen container. All attributes of this class are taken from the HL7 Lab Automation SIG / NCCLS proposal for an HL7 version 2.3.2 chapter.

A container can be related to a content material through a Material_ relationship of type *content*.

2.8.2.1 Container.capacity_qty : PQ

A capacity of a container is the maximum amount of content the container is designed to hold. See Section 2.7.1.10 about what an amount is. Note that the Material.qty for a container is used the same way as for any other material. The Material.qty does not describe the capacity of one container, but allows one to specify a quantity of containers if the individual container has no relevance.

The actual amount of content in a container can be specified by the Material.qty of the content material.

2.8.2.2 Container.height_qty : PQ ~ 1 cm

From NCCLS, a geometric property of the container. Issue: how do we know that we do not need to describe other arbitrary properties of containers? If we do, how do we do that?

2.8.2.3 Container.diameter_qty : PQ ~ 1 cm

From NCCLS, a geometric property of the container. Issue: how do we know that we do not need to describe other arbitrary properties of containers? If we do, how do we do that?

2.8.2.4 Container.barrier_delta_qty : PQ ~ 1 cm

From NCCLS, a geometric property of the container. Issue: how do we know that we do not need to describe other arbitrary properties of containers? If we do, how do we do that?

2.8.2.5 Container.bottom_delta_qty : PQ ~ 1 cm

From NCCLS, a geometric property of the container. Issue: how do we know that we do not need to describe other arbitrary properties of containers? If we do, how do we do that?

2.8.2.6 Container.separator_type_cd : CD

From NCCLS, the kind of separator material. Issue: code appears to be undefined. This attribute will be dropped if we do not get in a half-way complete concept repertoire by September 2000.

2.8.2.7 Container.cap_type_cd : CD

From NCCLS, the kind of cover cap ? Issue: code appears to be undefined. This attribute will be dropped if we do not get in a half-way complete concept repertoire by September 2000.

2.8.3 Therapeutic agent

A therapeutic agent is anything that is brought to interact with the human body in order to achieve therapeutic effects.

2.8.4 Devices

A device is anything used in an activity without being substantially changed through that activity. This includes durable (reuseable) medical equipment as well as disposable equipment.

Currently there are no attributes of device that would not also be applicable to any kind of material. This role class is shown anyway, in order to make the use of material for therapeutic agents obvious. If there are no properties defined for this class by September 2000 it will be deleted from the model.

In any way, there are a few device concepts defined by HL7 version 2.3 which are suggested for use in HL7 v2.3 as Material.type_cd values if the material is a device of one of the defined kinds and if it is not otherwise specified. See Table 38 below.

Table 38: Devices commonly used to administer medication (from HL7 v2.3 table 0164)

Value	Description	Value	Description
AP	Applicator	IVS	IV Soluset
BT	Buretrol	MI	Metered Inhaler
HL	Heparin Lock	NEB	Nebulizer
IPPB	IPPB	PCA	PCA Pump
IVP	IV Pump		

2.8.5 Access tubes and drains

Access tubes and drains are anything used to administer therapeutic agents (medication and vital elements) into the body, or to drain material (e.g., exsudat, pus, urine, air, blood) out of the body. Typically an access is a catheter, cannula or flexule proceeded into a compartment of the body.

Therefore, (target) body site and entry site are attributes of the access. Note that the Access role primarily exists in order to describe material actually deployed as an access, and not so much the fresh material as it comes from the manufacturer. For example, in supply ordering a box of catheters from a distributor, it is not necessary to use the access role class, since the material attributes will usually suffice to describe and identify the product for the order.

2.8.5.1 Access.gauge_qty : PQ

The gauge of an access is a measure for the inner diameter of the tube (the lumen.) Typically catheter gauge is measured in terms of units not seen elsewhere. Those units are defined in the Unified Code for Units of Measure, but are repeated here because they are so unusual.

Table 39: Common gauge measures for access devices

Name	Symbol	Synonym	Definition	Conversion
millimeter	mm		SI-unit, one thousandth of a meter	1 mm = 10 ⁻³ m
inch (international)	[in_i]		Customary unit as used today in the U.S. and Great Britain.	1 [in_i] = 25.4 mm
french	[Ch]	charrière	From Charrière, a French manufacturer of medical instruments. Defined as the gauge of a tube having a circumference of 1 mm.	1 [ch] = 1/π mm ≈ 0.3183 mm 1 mm ≈ 3.14 [Ch]
gauge	G		There is a variety of gauge units defined and used in the wire manufacturing industry. We are about to collect clear definitions on most of them (particularly the variant used with cannulas) but have not been quite successful. The general schema of wire gauges and their relationship to diameters is given through the formula on the right.	$d = d_0 \times 10^{-k g}$ $g = (\log_{10} d_0 - \log_{10} d) / k$; where k 1/10 or 1/20 depending on gauge variant. d the diameter g the gauge number d_0 e.g., 8.23 mm, 6.5 mm, depending on gauge variant.

2.8.5.2 Access.entry_site_cd : CD

The Access.entry_site_cd specifies the anatomic site where the access first enters the body. For example in a arteria pulmonalis catheter targets a pulmonary artery but the access entry site is typically the vena carotis interna at the neck, or the vena subclavia at the fossa subclavia.

The coding system is the same as for Service.body_site.

Entry site has been copied from the Procedure service class into the Access role class. The value of the Access.entry_site_cd should be identical to the value of the Procedure.entry_site_cd of an associated access placement service. This attribute is used if such an associated access placement service is not communicated. Since accesses are typically placed for a considerable period of time and since the access is used as a Target (resource) of many services, the entry site seems to have become an important attribute of the access itself. The entry site is one of the most distinctive descriptors that help in locating a specific access among many others.

2.8.5.3 Access.body_site_cd : CD **inherited from Service**

This is the anatomical target site of the access, i.e., the body compartment into which material is administered or from which it is drained. For example, a pulmonary artery catheter will have the target site arteria pulmonalis with or without a known laterality.

The coding system is the same as for Service.body_site.

Body site has been copied from the Service class into the Access role class. The value of the Access.body_site_cd should be identical to the value of the Service.body_site_cd of an associated access placement service. This attribute is used if such an associated access placement service is not communicated. Since accesses are typically placed for a considerable period of time and since the access is used as a Target (resource) of many services, the target body site seems to have become an important attribute of the access itself. The body site is an important information that determine what

kinds of substances may or may not administered (e.g., special care to avoid medication injections into an arterial access.)

2.8.6 Location as a Role of Material

Locations are treated as a role of Material. We anticipate this decision to be questioned, since it sounds quite unusual. In initial forms of this model, location was indeed modeled as a separate kind of substantive class, parallel to material. However, modeling location unrelated to material is challenged by a very simple problem: an ambulance. Obviously an ambulance is a material with a device role, used as a device for transportation services. However, an ambulance is also a location at which services are delivered.

This ambulance example shows that information analysis must separate the concept of position and region in a geometric coordinate system at which things are located and where activities happen, from a facility that is a resource for activities. For example an operating room is a service location that is clearly material in nature: the important parts of an operating room are its technical equipment that is tightly coupled to the walls, floor and ceiling of the room. Conversely, the position of the operating room, whether in terms of geographic coordinates or position in a building (floor, tract, room number, etc.), is quite irrelevant. So, while it is true that things are located at positions and services happen at positions, position itself is a fairly irrelevant aspect of what we call a service location. What is relevant is the location as a resource, just like an equipment.

When in occupational or environmental medicine locations become a target *subject* of service (i.e., the subject to be assessed or acted upon) the findings are similar: position is a fairly irrelevant aspect of a location, whereas the material features of the location are of much higher interest. It is the water, air, the soil, the floor, the wall, the thing at the position that is the target of a service, not the position.

Thus whereby position is an important property of locations, that mainly help in locating (finding) a location, the essence of location is in fact material. Thus, we can rightfully model location as a role of material.

This proposal does not pertain to the detail of the location class in the current RIM (called “master patient service location”) since we do not discuss the details of the location class, we do not even show that class in the model diagram. All we assert here about location is that it is a role of material, i.e., that there is an association `Material :: takes_on_role_of(0,1) :: * Location :: is_a_role_of(1)`.

2.8.7 Food

Food is a role of material. Food is anything that is ingested by humans to address hunger and provide nutrition to the body. Food is often combined into dishes, which are combined into full meals. Since the `Material_relationship` class can express this combination there are little additional properties needed in the food class. There is only one classifier attribute that seem to be relevant and special for food. We call that classifier “preference”, which is described below.

2.8.7.1 Food.preference_cd : CD

default: NVEG

The food preference describes the “style” and properties of the food that is selected mainly to meet the preference and customs of the recipient of the food. The term “preference” was selected to express that this property of food meets a preference of the consumer, not in order to limit this attribute to describe only the preferred style of food but not the actual style. The following concept repertoire is defined:

Table 40: Preferences, or “styles” of food

Concept	Implies	Code	Definition
non-vegetarian		NVEG	Supposedly every reasonable food is permitted.
no beef	NVEG	NBEF	Everything but beef (e.g., a hindu will absolutely not eat beef.)
no pork	NVEG	NPRK	Everything but pork (e.g., muslims and jews)
kosher	NPRK	KOSH	Prepared after the traditional jewish rules
no beef and no pork	non-veg.	NBOP	Everything except beef and pork (e.g., many hindus today will not eat beef, but will also stay away from pork), allowed meat is mutton, poultry, and fish.
no meat but fish	non-veg.	NMBF	Fish is the only allowed meat.
vegetarian		VEG	No meat at all. The only allowed animal product is egg and milk.
vegan	VEG	VEGN	vegetarian without eggs
... more ...			

2.9 Stakeholder Owned Service Lists

Service lists are ordered collections of services. A service list is owned by a stakeholder and that owner can sort the services on the list in any order. A service itself is not affected in any way by being or not being a member on any particular list. Such stakeholder owned lists represent the fact that all prioritization of facts is in principle subjective. Technical applications of service lists are: work-list or schedule of the day, prioritized list of patient problems, to-do items, etc.

An example for the subjectivity of ranking among problems is a patient with three problems: a hebephrenic schizophrenia, a hypertrophic obstructive cardiomyopathy (HOCM) with a history of ventricular tachycardia, and acute abdominal pain. This patient is seen by three different physicians, a psychiatrist, a cardiologist, and, most recently, a surgeon. Obviously each one of the physicians would determine a different prioritization of the patient problems according to the his specialty and role in the care of that patient.

These stakeholder owned lists are not considered part of the patient record but are private lists that only the owner maintains. The need for communicating those lists becomes obvious when the list owner is an organization, department or team, where updates of the lists are made from different work places and systems.

2.9.1 Service List

A service list is owned by one and only one Stakeholder, which may be an individual, or organization (e.g., care team.) A Stakeholder can have multiple service lists. Each service list has a subject, i.e., a material (e.g. today’s schedule for operating room 12a,) or a patient (patient’s problem list) or another person. If the service list has no subject but just an owner, the owner is considered the subject. Thus, any stakeholder can maintain personal to-do lists, diaries, logbooks, etc.

2.9.1.1 Service_list.id : SET<II>

Identifiers for the service lists, required to address the same service lists in multiple transactions.

2.9.1.2 Service_list.type_cd : CV

A kind of service list. Refer to the following Table for defined list types.

Table 41: Types of stakeholder owned service lists.

Concept	Implies	Code	Definition
schedule		SCH	A work-list, a schedule, or a personal to-do list of items intended to be done.
logbook		LOG	A diary of past services.
issues		ISS	A collections of any kinds of services as issues that need to be resolved.
problem list	issues	PRB	A patient's problem list as seen by a particular provider.
... more ...			

2.9.1.3 Service_list.name : ST

A short name that the owner of the list chooses to find this list among others. The name must be unique among all the lists that the stakeholder owns.

2.9.1.4 Service_list.descr : ED

A description of this list. This may be considerable amount of text that explains what the list is for and how it is used. This is especially relevant if the owner is an organization or work group.

2.9.2 List Item

Each list item represents one service on the list.

2.9.2.1 List_item.sequence_nmb : REAL default: 1

The items of the list can be sequenced using this attribute. It is a real number in order to allow dynamic insertion without having to renumber all the items every time an insertion or deletion is made.

2.9.2.2 List_item.priority_nmb : REAL

Items in the list can be ranked by priority. This is used to help deciding which item to address next when the items are not sequenced.

2.9.2.3 List_item.note_txt : ED

A note may be attached to each list. Since stakeholder owned lists are not part of the medical record, these notes are private notes of the list owner and are not subject to the rules of auditing and archiving that apply to medical record items.