HL7 Version 3 GELLO
Implementation Guide:
Clinical Decision Support;
Model Definition Language for
GELLO,
Release 1

April 2017

HL7 Normative Standard

Sponsored by:
Clinical Decision Support Work Group
IMPORTANT NOTES:
HL7 licenses its standards and select IP free of charge. If you did not acquire a free license from HL7 for this document, you are not authorized to access or make any use of it. To obtain a free license, please visit http://www.HL7.org/implement/standards/index.cfm.

If you are the individual that obtained the license for this HL7 Standard, specification or other freely licensed work (in each and every instance "Specified Material"), the following describes the permitted uses of the Material.

A. HL7 INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS, who register and agree to the terms of HL7’s license, are authorized, without additional charge, to read, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part without paying license fees to HL7.

INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS wishing to incorporate additional items of Special Material in whole or part, into products and services, or to enjoy additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS as noted below, must become ORGANIZATIONAL MEMBERS of HL7.

B. HL7 ORGANIZATION MEMBERS, who register and agree to the terms of HL7’s License, are authorized, without additional charge, on a perpetual (except as provided for in the full license terms governing the Material), non-exclusive and worldwide basis, the right to (a) download, copy (for internal purposes only) and share this Material with your employees and consultants for study purposes, and (b) utilize the Material for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, Compliant Products, in all cases subject to the conditions set forth in this Agreement and any relevant patent and other intellectual property rights of third parties (which may include members of HL7). No other license, sublicense, or other rights of any kind are granted under this Agreement.

C. NON-MEMBERS, who register and agree to the terms of HL7’s IP policy for Specified Material, are authorized, without additional charge, to read and use the Specified Material for evaluating whether to implement, or in implementing, the Specified Material, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part.

NON-MEMBERS wishing to incorporate additional items of Specified Material in whole or part, into products and services, or to enjoy the additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS, as noted above, must become ORGANIZATIONAL MEMBERS of HL7.

Please see http://www.HL7.org/legal/ippolicy.cfm for the full license terms governing the Material.

Ownership. Licensee agrees and acknowledges that HL7 owns all right, title, and interest, in and to the Materials. Licensee shall take no action contrary to, or inconsistent with, the foregoing.

Licensee agrees and acknowledges that HL7 may not own all right, title, and interest, in and to the Materials and that the Materials may contain and/or reference intellectual property owned by third parties (“Third Party IP”). Acceptance of these License Terms does not grant Licensee any rights with respect to Third Party IP. Licensee alone is responsible for identifying and obtaining any necessary licenses or authorizations to utilize Third Party IP in connection with the Materials or otherwise. Any actions, claims or suits brought by a third party resulting from a breach of any Third Party IP right by the Licensee remains the Licensee’s liability.

Following is a non-exhaustive list of third-party terminologies that may require a separate license:

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Owner/Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Procedures Terminology</td>
<td>American Medical Association</td>
</tr>
<tr>
<td>SNOMED CT</td>
<td>International Healthcare Terminology Standards Developing Organization (IHTSDO) <a href="http://www.ihtsdo.org/snomed-ct/get-snomed-ct">http://www.ihtsdo.org/snomed-ct/get-snomed-ct</a> or <a href="mailto:info@ihtsdo.org">info@ihtsdo.org</a></td>
</tr>
<tr>
<td>Logical Observation Identifiers</td>
<td>Regenstrief Institute</td>
</tr>
<tr>
<td>Names &amp; Codes (LOINC)</td>
<td></td>
</tr>
<tr>
<td>International Classification of</td>
<td>World Health Organization (WHO)</td>
</tr>
<tr>
<td>Diseases (ICD) codes</td>
<td></td>
</tr>
<tr>
<td>Taxonomy code set</td>
<td>AMA licensing contact: 312-464-5022 (AMA IP services)</td>
</tr>
</tbody>
</table>
Model Definition Language for Clinical Decision Support – GELLO Implementation Guide

Document Editor
Andrew McIntyre MBBS, FRACP; Medical-Objects
Kensaku Kawamoto, MD, PhD; University of Utah
Peter R. Tattam; Tattam Software Enterprises Pty Ltd
David E. Shields; University of Utah
Peter Scott, MBBS; Medical-Objects

Project Sponsor
HL7 Clinical Decision Support

HL7 Project #184
Normative Standard
May 2016
# Table of Contents

Table of Contents .................................................................................................................. 4

1 Executive Summary ............................................................................................................. 5

2 GELLO Implementation Guide for Model Definition Language ..................................... 6
   2.1 Grammar for GELLO class definitions (Normative) ..................................................... 6
   2.2 GELLO base types class definition (Informative) ......................................................... 8
   2.3 ISO21090 data types definition (Informative) .............................................................. 11
   2.4 vMR definition example (Informative) .......................................................................... 26
   2.5 Detailed Clinical Model example (Informative) ............................................................ 34
1 Executive Summary

A Model Definition Language for Clinical Decision Support (CDS) is a grammar for representing clinical information inputs and outputs that can be used by CDS engines and local clinical information systems, through mechanisms such as CDS services, or execution of standardised clinical decision support logic by a GELLO execution engine. This model definition language is needed to enable the design and development of scalable CDS resources that can be used across multiple healthcare institutions and health information systems. Existing clinical data, from any source, is virtualised to present a façade compliant with the required class model. Clinical logic can then be executed against the data represented in the model, such as the HL7 Virtual Medical Record (vMR) model. GELLO is one standard language that can be used with the model, but other languages can also use the resulting model.

The objective of this Implementation Guide is to define a BNF for defining a model which allows tight integration of a clinical model with a GELLO compiler or any other rule execution engine or authoring environment, ensuring good interoperability of CDS logic between institutions. This guide describes a grammar for defining the classes and data types. It does not attempt to define the final structure of the model, but defines a standard mechanism for defining classes and data types. The final models would be encoded in this format allowing compilers access to a standard, rigorously defined model. This format will be identical to the semantics of a UML XMI representation, but is compiler friendly and more human readable. While GELLO has been defined by HL7 CDS workgroup for encoding standards based decision support logic, the models as defined by this format could be used in any execution or authoring environment to tightly define data types, class structures and methods. The classes and data types defined by this format are built on the base GELLO types which are based on the OCL types used in UML. The format is designed to be language neutral, and should be useable by any object based execution environment that supports the base UML data types.

This is the normative ballot for this material. It is envisaged that all elements of the final target data model, when this is finalised, will be represented in both UML (.xmi files), class diagrams, and this format. As this format is both human and machine readable it is intended to make this format a more readily understandable form of the data model standard.

It is possible, by the use of this format, to define a class model which represents a concrete instance of a DCM (Detailed Clinical Model). A package containing the required classes to represent data conformant to a DCM could then be used to extend the available classes in a execution environment.

Note: Earlier versions of this specification was called the “Virtual Medical Record (vMR) for Clinical Decision Support – GELLO Implementation Guide.” The title of the specification, as well as the specification content, have been updated to clarify the independence of this specification from the vMR data model.
2 GELLO Implementation Guide for Model Definition Language

2.1 Grammar for GELLO class definitions (Normative)

This is the GELLO class BNF which uses the Extended BNF syntax as used in the GELLO standard. This BNF is the normative part of this standard.

```
GelloPackage ::= PackageStatement

PackageStatement ::= <PACKAGE> NameOrLiteral ImportsStatement? PackageElement* <ENDPACKAGE>

ImportsStatement ::= <IMPORTS> ImportName ("," ImportName)*

ImportName ::= NameOrLiteral

PackageElement ::= PackageStatement | ClassDefinition

ClassDefinition ::= <CLASS> ClassDefName (<EXTENDS> ClassRef)? ClassMember* |
                   <CLASS> ClassDefName <EQUAL> ClassRef |
                   <CLASS> ClassName EnumerationType

ClassDefName ::= NameOrLiteral |
                NameOrLiteral "(" ClassDefParams ")"

ClassDefParams ::= ClassRefOrParam ("," ClassRefOrParam)*

ClassRefOrParam ::= ClassRef |
                  Name ":" ClassRef

ClassRef ::= ClassName |
            ClassName "(" ClassRef ")" |
            <TUPLE>

ClassMember ::= NameOrLiteral ( "(" ClassFormalParams? ")" )? ":" 

ClassRef

ClassFormalParams ::= ClassFormalParam ("," ClassFormalParam)*

ClassFormalParam ::= Name ":" ClassRef

EnumerationType ::= <ENUM> "(" (EnumLiteral ("," EnumLiteral )*) ")"

EnumLiteral ::= NameOrLiteral

NameOrLiteral ::= Name | <STRING_LITERAL>

ClassName ::= NameWithPath | <STRING_LITERAL>

NameWithPath ::= Name (":" Name)*

Name ::= <ID> (":" <ID>)*
```
<STRING_LITERAL: ("\n\r\n\"\r\n\"\") >

<ID: ["A"-"Z","a"-"Z",0-9] (["a"-"z","A"-"Z",0-9] | "_"(["a"-"z","A"-"Z",0-9])+)* >

<ENUM: "Enum">

<TUPLE: "Tuple">

<EQUAL: "="> 

<ENDPACKAGE: "endPackage" | "EndPackage" | "endpackage">

<class| "Class">

<EXTENDS: "extends">

<IMPORTS: "Imports" | "imports">

<PACKAGE: "Package" | "package">
2.2 GELLO base types class definition (Informative)

These are the base GELLO types which are enhanced OCL types. They are used by the ISO 21090 data types and are included as an example of the use of the grammar for informative purposes.

Package System

Class Any
"="(o: Any): Boolean
"<"(o: Any): Boolean
oclIsDefined(): Boolean
oclAsType(t: T): T
oclIsTypeOf(t: T): Boolean
oclIsKindOf(typename: String): Boolean
oclIsInState(statename: String): Boolean
allInstances(): Set(T)

Class Real extends Any
"-"(): Real
"+"(rhs: Real): Real
"-"(rhs: Real): Real
"*"(rhs: Real): Real
"/"(rhs: Real): Real
abs(): Real
floor(): Integer
round(): Integer
"max"(rhs: Real): Real
"min"(rhs: Real): Real
"="(rhs: Real): Boolean
"<"(rhs: Real): Boolean
"<="(rhs: Real): Boolean
">"(rhs: Real): Boolean
">="(rhs: Real): Boolean
toChar():String
toCharFormat(decimalplaces:Integer): String

Class Integer extends Real
"-"(): Integer
"+"(rhs: Integer): Integer
"-"(rhs: Integer): Integer
"*"(rhs: Integer): Integer
"div"(rhs: Integer): Integer
"mod"(rhs: Integer): Integer
"max"(rhs: Integer): Integer
"min"(rhs: Integer): Integer
abs(): Integer
toChar():String

Class String extends Any
"="(rhs: String): Boolean
"<"(rhs: String): Boolean
size(): Integer
concat(rhs: String): String
substring(lower:Integer, upper: Integer): String
toInteger(): Integer
toReal(): Real
toUpper(): String
toLower():String
lpad(size:Integer, padChar:String):String
rpad(size:Integer, padChar:String):String
ltrim(trimChars:String):String
rtrim(trimChars:String):String
replace(replaceThis:String, withThis:String):String

Class Boolean extends Any
"="(rhs: Boolean): Boolean
"<>“(rhs: Boolean): Boolean
"or”(rhs: Boolean): Boolean
"xor”(rhs: Boolean): Boolean
"and”(rhs: Boolean): Boolean
"not”(): Boolean
implies(rhs: Boolean): Boolean

Class Collection extends Any

count(o: T): Integer
excludes(o: T): Boolean
excludesAll(c: Collection): Boolean
includes(o: T): Boolean
includesAll(c: Collection): Boolean
isEmpty(): Boolean
notEmpty(): Boolean
size(): Integer
sum(): T
"max”(): T
"min”(): T

"="(s: Collection): Boolean
"<>“(s: Collection): Boolean

asSet(): Set(T)
asBag(): Bag(T)
asSequence(): Sequence(T)

exists(e: Boolean): Boolean
forAll(e: Any): Boolean
collect(e: Any): Bag(T)
sortedBy(e: Any): Sequence(T)
Average():Real
Stdev():Real
Variance():Real
Median():Real
Mode():Real
Join(): Collection(Tuple)
Like(): Collection(String)
NotLike(): Collection(String)
Between(string1: String, string2: String): Collection(String)
Distinct(): Set(T)

Class "Set" extends Collection
"="(S: Set(T)): Set(T)
union(c: Bag(T)): Bag(T)
union(s: Set(T)): Set(T)
intersection(b: Bag(T)): Bag(T)
intersection(s: Set(T)): Set(T)
including(o: T): Set(T)
excluding(o: T): Set(T)
symmetricDifference(s: Set(T)): Set(T)
flatten(): Set(T)

select(e: Boolean): Set(T)
reject(e: Boolean): Set(T)

Class "Bag" extends Collection
union(b: Bag(T)): Bag(T)
union(s: Set(T)): Bag(T)
intersection(b: Bag(T)): Bag(T)
intersection(s: Set(T)): Bag(T)
including(o: T): Bag(T)
excluding(o: T): Bag(T)
flatten(): Bag(T)

select(e: Boolean): Bag(T)
reject(e: Boolean): Bag(T)

Class "Sequence" extends Collection
union(s: Sequence(T)): Sequence(T)
including(o: T): Sequence(T)
excluding(o: T): Sequence(T)

append(o: T): Sequence(T)
at(i: Integer): T
first(): T
flatten(): Sequence(T)
indexOf(o: T): Integer
insertAt(i: Integer, o: T): Sequence(T)
last(): T
prepend(o: T): Sequence(T)
subSequence(lower: Integer, upper: Integer): Sequence(T)

select(e: Boolean): Sequence(T)
reject(e: Boolean): Sequence(T)
lastN(number:Integer): Sequence(T)
firstN(number:Integer): Sequence(T)
Reverse(): Sequence(T)

EndPackage
These data types are a concrete implementation of the HL7 V3 Release 2 data types which are likely to be used for the final VMR model. They were originally imported from the ISO21090 XML schema file (source: http://gforge.hl7.org/svn/hl7v3/trunk/dt/iso/iso-21090-datatypes.xsd) but also define operations. This is included as an example and the definition as represented here is indicative of the use of the grammar and is informative. The HL7 V3 Release 2 data type specification is abstract and cannot be used directly. The ISO21090 XML schema files define content, but not operations. Data types defined in this manner could be used by an execution engine or authoring environment. The data types in the final data model may be simplified further.

**Package iso_21090_datatypes -- Imported from iso-21090-datatypes.xsd**

```java
class HXIT
  controlInformationExtension: String
  controlInformationRoot: Uid
  validTimeHigh: String
  validTimeLow: String


class UpdateMode Enum("A", "D", "R", "AR", "N", "U", "K")

class ANY extends HXIT
  flavorId: String
  nullFlavor: NullFlavor
  updateMode: UpdateMode
  -- operations
  equals(other : ANY): Boolean

class XP
  code: String
  codeSystem: String
  codeSystemVersion: String
  language: Code
  nullFlavor: NullFlavor
  value: String


class ADXP extends XP
  type: AddressPartType


class AD extends ANY
  isNotOrdered: Boolean
  part: Sequence(ADXP)
  use: Set(PostalAddressUse)
  useablePeriod: QSET_TS

class BL extends ANY
  value: Boolean
  -- operations
  "and"(other : BL) : BL
```
"or"(other : BL) : BL
"xor"(other : BL) : BL
implies(other : BL) : BL

"not"() : BL

class CodingRationale Enum("O", "P", "R", "OR", "PR")

class CD extends ANY
  code: String
  codeSystem: Uid
  codeSystemName: String
  codeSystemVersion: String
  codingRationale: CodingRationale
  displayName: ST
  id: String
  originalText: ED
  source: XReference
  translation: Sequence(CD)
  valueSet: Uid
  valueSetVersion: String
  --operations
  implies(other : CS) : BL
  implies(other : CD) : BL

class CO extends QTY
  code: CD
  value: Decimal
  --operations
  "+"(other : CO) : CO
  "-"(other : CO) : CO
  "min"(other : CO) : CO
  "max"(other : CO) : CO

class CS extends ANY
  code: String
  --operations
  codeSystem : Uid
  codeSystemName: String
  codeSystemVersion : String
  implies(other : CS) : BL
  implies(other : CD) : BL

class Decimal extends Real

class Compression Enum("DF", "GZ", "ZL", "Z", "BZ", "Z7")

class Code extends String

class IntegrityCheckAlgorithm Enum("SHA1", "SHA256")

class ED extends ANY
  charset: Code
  compression: Compression
  data: String
  description: ST
  integrityCheck: String
  integrityCheckAlgorithm: IntegrityCheckAlgorithm
  language: Code
  mediaType: String
  reference: TEL
  thumbnail: ED
  translation: Sequence(ED)
  value: String
  xml: Any

class EIVL_TS extends QSET_TS
  event: TimingEvent
  offset: IVL_PQ

class EntityNamePartQualifier Enum("LS", "AC", "NB", "PR", "HON", "BR", "AD", "SP", "MID", "CL", "IN", "PFX", "SFX")

class EntityNamePartType Enum("FAM", "GIV", "TITLE", "DEL")

class ENXP extends XP
  qualifier: Set(EntityNamePartQualifier)
  type: EntityNamePartType

class EN extends ANY
  part: Sequence(ENXP)
  use: set_EntityNameUse

--operations
  canonical() : EN

class GLIST_PQ extends ANY
  denominator: Integer
  head: PQ
  increment: QTY
  period: Integer

class GLIST_REAL extends ANY
  denominator: Integer
  head: REAL
  increment: QTY
  period: Integer

class GLIST_TS extends ANY
  denominator: Integer
  head: TS
  increment: QTY
  period: Integer

class IdentifierReliability Enum("ISS", "VRF", "UNV")

class IdentifierScope Enum("BUSN", "OBJ", "VER", "VW")

class II extends ANY
  displayable: Boolean
  extension: String
  identifierName: String
  reliability: IdentifierReliability
  root: Uid
  scope: IdentifierScope

--operations
  "+"(other : INT) : INT
  "-"(other : INT) : INT
  "*"(other : INT) : INT
  "/"(other : INT) : REAL
  "/"(other : REAL) : REAL
  abs() : INT
  "div"(other : INT) : INT
  "mod"(other : INT) : INT

"min"(other : INT) : INT
"max"(other : INT) : INT

class IVL_CO extends QSET_CO
  any: CO
  high: CO
  highClosed: Boolean
  low: CO
  lowClosed: Boolean
  width: QTY

class IVL_INT extends QSET_INT
  any: INT
  high: INT
  highClosed: Boolean
  low: INT
  lowClosed: Boolean
  width: QTY

class IVL_MO extends QSET_MO
  any: MO
  high: MO
  highClosed: Boolean
  low: MO
  lowClosed: Boolean
  width: QTY

class IVL_PQ extends QSET_PQ
  any: PQ
  high: PQ
  highClosed: Boolean
  low: PQ
  lowClosed: Boolean
  width: QTY

class IVL_QTY extends QSET_QTY
  any: QTY
  high: QTY
  highClosed: Boolean
  low: QTY
  lowClosed: Boolean
  width: QTY

class IVL_REAL extends QSET_REAL
  any: REAL
  high: REAL
  highClosed: Boolean
  low: REAL
  lowClosed: Boolean
  width: QTY

class IVL_TS extends QSET_TS
  any: TS
  high: TS
  highClosed: Boolean
  low: TS
  lowClosed: Boolean
  width: QTY

class MO extends QTY
  currency: Code
  value: Decimal
  --operations
  "+"(other : MO) : MO
  "-"(other : MO) : MO
  "*"(other : REAL) : MO
551 "/(other : REAL) : MO
552 "min"(other : MO) : MO
553 "max"(other : MO) : MO
554
class CalendarCycle Enum("CY", "MY", "CM", "CW", "WM", "WY", "DM", "CD",
                             "DY", "DW", "HD", "CH", "NH", "CN", "SN", "CS")
555
class PIVL_TS extends QSET_TS
556 alignment: CalendarCycle
557 count: INT
558 frequency: RTO
559 isFlexible: Boolean
560 period: PQ
561 phase: IVL_TS
562
class PQ extends QTY
563 codingRationale: CodingRationale
564 translation: Sequence(PQR)
565 unit: Code
566 value: Decimal
567
568 --operations
569
570 "+"(other : PQ) : PQ
571 "-"(other : PQ) : PQ
572 "+"(other : PQ) : PQ
573 "+"(other : Real) : PQ
574 "+"(other : Real) : PQ
575 "+"(other : Integer) : PQ
576 "+"(other : PQ) : PQ
577 "+"(other : Real) : PQ
578 "+"(other : Real) : PQ
579 "+"(other : Integer) : PQ
580 "+"(other : PQ) : PQ
581 "+"(other : Real) : PQ
582 "+"(other : Real) : PQ
583 "+"(other : Integer) : PQ
584 inverted() : PQ
585 abs() : PQ
586 "+"(other : PQ) : PQ
587 "+"(other : Real) : PQ
588 "+"(other : Real) : PQ
589 toInterval() : IVL_PQ
590 convert(unit: Code):PQ
591
592 class PQR extends CD
593 value: Decimal
594
595
class QSC_TS extends QSET_TS
596 code: CD
597
class QSD_PQ extends QSET_PQ
598 minuend: QSET_PQ
599 subtrahend: QSET_PQ
600
class QSD_TS extends QSET_TS
601 minuend: QSET_TS
602 subtrahend: QSET_TS
603
class QSET_CO extends ANY
604 originalText: ED
605
class QSET_INT extends ANY
606 originalText: ED
607
class QSET_MO extends ANY
608 originalText: ED
609
class QSET_PQ extends ANY
class QSET_QTY extends ANY

class QSET_REAL extends ANY

class QSET_TS extends ANY

class QSI_PQ extends QSET_PQ
term: Sequence(QSET_PQ)

class QSI_TS extends QSET_TS
term: Sequence(QSET_TS)

class QSP_PQ extends QSET_PQ
high: QSET_PQ
low: QSET_PQ

class QSP_TS extends QSET_TS
high: QSET_TS
low: QSET_TS

class QSS_PQ extends QSET_PQ
term: Sequence(PQ)

class QSS_TS extends QSET_TS
term: Sequence(TS)

class QSU_PQ extends QSET_PQ
term: Sequence(QSET_PQ)

class QSU_TS extends QSET_TS
term: Sequence(QSET_TS)

class UncertaintyType Enum("U", "N", "LN", "G", "E", "X2", "T", "F", "B")

class QTY extends ANY
expression: ED

class REAL extends QTY
value: Decimal

operations
"+"(other : REAL) : REAL
"-"(other : REAL) : REAL
"*"(other : REAL) : REAL
"/"(other : REAL) : REAL
"min"(other : REAL) : REAL
"max"(other : REAL) : REAL
"-"() : REAL
abs() : REAL
ceiling() : INT
floor() : INT
round() : INT
inverted() : REAL
power(other : REAL) : REAL
toInterval() : IVL_REAL

class RTO extends QTY
denominator: QTY
numerator: QTY

class SC extends ST
code: CD

class SD_TEXT extends ANY
ID: String
language: Code
mediaType: String
styleCode: set_Code

class SD_TITLE extends ANY
ID: String
language: Code
mediaType: String
styleCode: set_Code

class SLIST_INT extends ANY
digit: Sequence(INT)
origin: INT
scale: QTY

class SLIST_PQ extends ANY
digit: Sequence(INT)
origin: PQ
scale: QTY

class SLIST_REAL extends ANY
digit: Sequence(INT)
origin: REAL
scale: QTY

class SLIST_TS extends ANY
digit: Sequence(INT)
origin: TS
scale: QTY

class ST extends ANY
language: Code
translation: Sequence(ST)
value: String
operations
mediaType() : String
size() : Integer
concat(other : ST) : ST
substring(lower : INT, upper : INT) : ST
toInteger() : INT
toReal() : REAL
toTimestamp() : TS

class TelecommunicationCapability Enum("voice", "fax", "data", "tty", "sms")

class TelecommunicationAddressUse Enum("H", "HP", "HV", "WP", "DIR", "PUB", "BAD", "TMP", "AS", "EC", "MC", "PG")

class TEL extends ANY
capabilities: Set(TelecommunicationCapability)
use: Set(TelecommunicationAddressUse)
useablePeriod: QSET_TS
value: String

--operations
canonical() : TEL

class TS extends QTY
value: String
--operations
"+"(other : PQ) : TS
"-"(other : PQ) : TS
"-"(other : TS) : PQ
"min"(other : TS) : TS
"max"(other : TS) : TS
toInterval() : IVL_TS
precision() : Integer

class UVP_AD extends ANY
probability: Decimal
value: AD

class UVP_BL extends ANY
probability: Decimal
value: BL

class UVP_CD extends ANY
probability: Decimal
value: CD

class UVP_CO extends ANY
probability: Decimal
value: CO

class UVP_CS extends ANY
probability: Decimal
value: CS

class UVP_ED extends ANY
probability: Decimal
value: ED

class UVP_EN extends ANY
probability: Decimal
value: EN

class UVP_I2 extends ANY
probability: Decimal
value: II

class UVP_INT extends ANY
probability: Decimal
value: INT

class UVP_MO extends ANY
probability: Decimal
value: MO

class UVP_PQ extends ANY
probability: Decimal
value: PQ

class UVP_REAL extends ANY
probability: Decimal
value: REAL
class UVP_RTO extends ANY
  probability: Decimal
  value: RTO

class UVP_SC extends ANY
  probability: Decimal
  value: SC

class UVP_ST extends ANY
  probability: Decimal
  value: ST

class UVP_TEL extends ANY
  probability: Decimal
  value: TEL

class UVP_TS extends ANY
  probability: Decimal
  value: TS

class Uid extends String

class Uri extends String

class XReference
  xref: String

class XmlID extends String

class XmlIDREF extends String

class set_Code
  value: Set(Code)

class set_CodingRationale
  value: Set(CodingRationale)

class set_EntityNamePartQualifier
  value: Set(EntityNamePartQualifier)

class set_EntityNameUse
  value: Set(EntityNameUse)

class set_IDREF
  value: Set(XmlIDREF)

class set_TelecommunicationAddressUse
  value: Set(TelecommunicationAddressUse)

class set_TelecommunicationCapability
  value: Set(TelecommunicationCapability)

class COLL_AD extends ANY

class COLL_BL extends ANY

class COLL_CD extends ANY

class COLL_CO extends ANY

class COLL_CS extends ANY

class COLL_ED extends ANY

class COLL_EN extends ANY
class COLL_II extends ANY
class COLL_INT extends ANY
class COLL_MO extends ANY
class COLL_PQ extends ANY
class COLL_REAL extends ANY
class COLL_RTO extends ANY
class COLL_SC extends ANY
class COLL_ST extends ANY
class COLL_TEL extends ANY
class COLL_TS extends ANY

class BAG_AD extends COLL_AD
  value: Bag(AD)

class BAG_BL extends COLL_BL
  value: Bag(BL)

class BAG_CD extends COLL_CD
  value: Bag(CD)

class BAG_CO extends COLL_CO
  value: Bag(CO)

class BAG_CS extends COLL_CS
  value: Bag(CS)

class BAG_ED extends COLL_ED
  value: Bag(ED)

class BAG_EN extends COLL_EN
  value: Bag(EN)

class BAG_II extends COLL_II
  value: Bag(II)

class BAG_INT extends COLL_INT
  value: Bag(INT)

class BAG_MO extends COLL_MO
  value: Bag(MO)

class BAG_PQ extends COLL_PQ
  value: Bag(PQ)

class BAG_REAL extends COLL_REAL
  value: Bag(REAL)

class BAG_RTO extends COLL_RTO
  value: Bag(RTO)

class BAG_SC extends COLL_SC
  value: Bag(SC)

class BAG_ST extends COLL_ST
  value: Bag(ST)

class BAG_TEL extends COLL_TEL
value: Bag(TEL)

class BAG_TS extends COLL_TS
value: Bag(TS)

class DSET_AD extends COLL_AD
value: Sequence(AD)

class DSET_BL extends COLL_BL
value: Sequence(BL)

class DSET_CD extends COLL_CD
value: Sequence(CD)

class DSET_CO extends COLL_CO
value: Sequence(CO)

class DSET_CS extends COLL_CS
value: Sequence(CS)

class DSET_ED extends COLL_ED
value: Sequence(ED)

class DSET_EN extends COLL_EN
value: Sequence(EN)

class DSET_II extends COLL_II
value: Sequence(II)

class DSET_INT extends COLL_INT
value: Sequence(INT)

class DSET_MO extends COLL_MO
value: Sequence(MO)

class DSET_PQ extends COLL_PQ
value: Sequence(PQ)

class DSET_REAL extends COLL_REAL
value: Sequence(REAL)

class DSET_RTO extends COLL_RTO
value: Sequence(RTO)

class DSET_SC extends COLL_SC
value: Sequence(SC)

class DSET_ST extends COLL_ST
value: Sequence(ST)

class DSET_TEL extends COLL_TEL
value: Sequence(TEL)

class DSET_TS extends COLL_TS
value: Sequence(TS)

class HIST_AD extends LIST_AD

class HIST_BL extends LIST_BL

class HIST_CD extends LIST_CD

class HIST_CO extends LIST_CO

class HIST_CS extends LIST_CS
class HIST_ED extends LIST_ED
class HIST_EN extends LIST_EN
class HIST_II extends LIST_II
class HIST_INT extends LIST_INT
class HIST_MO extends LIST_MO
class HIST_PQ extends LIST_PQ
class HIST_REAL extends LIST_REAL
class HIST_RTO extends LIST_RTO
class HIST_SC extends LIST_SC
class HIST_ST extends LIST_ST
class HIST_TEL extends LIST_TEL
class HIST_TS extends LIST_TS
class LIST_AD extends COLL_AD
  value: Sequence(AD)
class LIST_BL extends COLL_BL
  value: Sequence(BL)
class LIST_CD extends COLL_CD
  value: Sequence(CD)
class LIST_CO extends COLL_CO
  value: Sequence(CO)
class LIST_CS extends COLL_CS
  value: Sequence(CS)
class LIST_ED extends COLL_ED
  value: Sequence(ED)
class LIST_EN extends COLL_EN
  value: Sequence(EN)
class LIST_II extends COLL_II
  value: Sequence(II)
class LIST_INT extends COLL_INT
  value: Sequence(INT)
class LIST_MO extends COLL_MO
  value: Sequence(MO)
class LIST_PQ extends COLL_PQ
  value: Sequence(PQ)
class LIST_REAL extends COLL_REAL
  value: Sequence(REAL)
class LIST_RTO extends COLL_RTO
  value: Sequence(RTO)
class LIST_SC extends COLL_SC
  value: Sequence(SC)
1079 class LIST_ST extends COLL_ST
1080     value: Sequence(ST)
1081
1082 class LIST_TEL extends COLL_TEL
1083     value: Sequence(TEL)
1084
1085 class LIST_TS extends COLL_TS
1086     value: Sequence(TS)
1087
1088 class NPPD_AD extends ANY
1089     value: Sequence(UVP_AD)
1090
1091 class NPPD_BL extends ANY
1092     value: Sequence(UVP_BL)
1093
1094 class NPPD_CD extends ANY
1095     value: Sequence(UVP_CD)
1096
1097 class NPPD_CO extends ANY
1098     value: Sequence(UVP_CO)
1099
1100 class NPPD_CS extends ANY
1101     value: Sequence(UVP_CS)
1102
1103 class NPPD_ED extends ANY
1104     value: Sequence(UVP_ED)
1105
1106 class NPPD_EN extends ANY
1107     value: Sequence(UVP_EN)
1108
1109 class NPPD_II extends ANY
1110     value: Sequence(UVP_II)
1111
1112 class NPPD_INT extends ANY
1113     value: Sequence(UVP_INT)
1114
1115 class NPPD_MO extends ANY
1116     value: Sequence(UVP_MO)
1117
1118 class NPPD_PQ extends ANY
1119     value: Sequence(UVP_PQ)
1120
1121 class NPPD_REAL extends ANY
1122     value: Sequence(UVP_REAL)
1123
1124 class NPPD_RTO extends ANY
1125     value: Sequence(UVP_RTO)
1126
1127 class NPPD_SC extends ANY
1128     value: Sequence(UVP_SC)
1129
1130 class NPPD_ST extends ANY
1131     value: Sequence(UVP_ST)
1132
1133 class NPPD_TEL extends ANY
1134     value: Sequence(UVP_TEL)
1135
1136 class NPPD_TS extends ANY
1137     value: Sequence(UVP_TS)
1138
1139 class StrucDoc_Align Enum("left", "center", "right", "justify", "char")
1140
1141 class StrucDoc_Br
1142
1143 class StrucDoc_CMTitle
1144     br: StrucDoc_Br
abbr: String
axis: String
colspan: Integer
headers: set_IDREF
rowspan: Integer
scope: StrucDoc_CellScope
class StrucDoc_TRow extends StrucDoc_TableItem
class StrucDoc_TRowGroup extends StrucDoc_TableItem
t: Sequence(StrucDoc_TRow)
class StrucDoc_Table extends StrucDoc_Captioned
border: StrucDoc_Length
cellpadding: StrucDoc_Length
cellspacing: StrucDoc_Length
col: Sequence(StrucDoc_Col)
colgroup: Sequence(StrucDoc_ColGroup)
frame: StrucDoc_Frame
rules: StrucDoc_Rules
summary: String
tbody: Sequence(StrucDoc_TRowGroup)
tfoot: StrucDoc_TRowGroup
thead: StrucDoc_TRowGroup
width: StrucDoc_Length
class StrucDoc_TableItem
align: StrucDoc_Align
char: String
charoff: StrucDoc_Length
valign: StrucDoc_VAlign
class StrucDoc_TitleFootnote
class StrucDoc_VAlign Enum("top", "middle", "bottom", "baseline")
EndPackage
2.4 vMR definition example (Informative)

A sample of a mock vMR is represented below to indicate how a vMR would be defined using the grammar and imports the ISO21090 package. This example is based on an import of a Green CDA schema and is not representative of the final vMR but is an informative example of the use of the grammar. In this example the Context classes (classes extending ContextClass), and Factory are specific to a GELLO execution environment.

Package GreenVMR
--example only and not normative

imports iso_21090_datatypes, HL7V3Rim

class addr
  city: String
  nullFlavor: nullFlavorType
  postalCode: String
  state: String
  streetAddressLine: String

class admissionTiming extends pivlTs

class advanceDirective
doctorCustodian: organizationInformation
effectiveDate: IVL_TS
status: CE
type: CD

class allergy
  adverseEventDate: IVL_TS
  adverseEventType: CD
  product: Sequence(CD)
  reaction: Sequence(CD)
  severity: Sequence(CD)

class author
  address: addr
  authorId: II
  authorName: pnm
  authorTime: TS
  reference: Sequence(reference)
telecom: tele

class birthPlace extends addr
  name: String

class causeOfDeath
  ageAtDeath: Integer
timeOfDeath: TS

class condition
  asAct: Act
  ageAtOnset: Integer
  causeOfDeath: causeOfDeath
diagnosisPriority: Integer
problemCode: CD
problemDate: IVL_TS
problemName: String
problemStatus: CE
problemType: CD
treatingProvider: Sequence(treatingProvider)
class confidentialityType Enum("N", "R", "V")

class contactType
  contactAddress: Sequence(addr)
  contactMothersMaidenName: onm
  contactName: Sequence(pnm)
  contactTelecom: Sequence(tele)
  relationship: relationshipType
  type: supportType


class custodian
  custodianAddress: addr
  custodianID: Sequence(II)
  custodianName: onm
  custodianTelecom: tele

class doseRestriction
  denominator: PQ
  numerator: PQ

class encounter
  asAct: Act
  admissionType: CD
  dischargeDisposition: String
  encounterDateTime: IVL_TS
  encounterID: Sequence(II)
  encounterProvider: Sequence(organizationInformation)
  encounterType: encounterType
  facilityLocation: facilityLocation
  reasonForVisit: reasonForVisit

class encounterType extends CD

class facilityLocation
  locationDuration: IVL_TS

class FamilyHistory extends Any
  ClinicalGenomicChoice: Sequence(ClinicalGenomicChoice)
  Informant: informationSource
  PatientID: ST
  PedigreeAnalysisResults: Sequence(result)
  Relatives: Sequence(Relative)

class ClinicalGenomicChoice extends Any
  ClinicalObservation: ClinicalObservation
  GeneticLoci: Sequence(result)
  Relative: Relative

class ClinicalObservation extends Any
  causeOfDeath: BL
  code: CD
  DataEstimatedAge: IVL_REAL
  negationInd: BL

class fillNumber
  nullFlavor: nullFlavorType
  value: String

class fills
  value: String

class fillStatus
value: statusType

class fulfillmentHistory
dispenseDate: IVL_TS
dispensingPharmacyLocation: addr
fillNumber: fillNumber
fillStatus: fillStatus
prescriptionNumber: II
provider: organizationInformation
quantityDispensed: PQ

class guarantorInformation
partyAddress: Sequence(addr)
partyName: pnm
partyPhone: Sequence(tele)
responsibilityEffectiveDate: IVL_TS

class header
certainty: CD
custodian: custodian
documentID: II
documentTimestamp: TS
healthcareProviders: healthcareProviders
informationSource: informationSource
language: CD
languagesSpoken: Sequence(languageSpoken)
personalInformation: personalInformation
supports: Sequence(support)
title: String
version: version

class healthcareProvider
dateRange: IVL_TS
patientID: II
providerEntity: providerInformation
role: CD

class healthcareProviders
careProvisionDateRange: IVL_TS
healthcareProvider: Sequence(healthcareProvider)

class immunization
administeredDate: TS
medicationInformation: Sequence(medicationInformation)
medicationSeriesNumber: PQ
performer: organizationInformation
reaction: Sequence(CD)
refusalInd: String
refusalReason: String

class indication extends CD
freeTextRef: String
status: statusType

class informationSource
author: author
informationSourceName: informationSourceName

class informationSourceName
organizationName: onm
personName: pnm
relatedPersonName: pnm

class ContextClass extends Any
Factory: Factory
dataRange: IVL_TS
advanceDirectives: Sequence(advanceDirective)
allergies: Sequence(allergy)
conditions: Sequence(condition)
encounters: Sequence(encounter)
immunizations: Sequence(immunization)
informationSource: informationSource
insuranceProviders: Sequence(insuranceProvider)
medications: Sequence(medication)
planOfCare: planOfCare
pregnancies: Sequence(pregnancy)
procedures: Sequence(procedure)
provider: organizationInformation
qualifier: qualifier
results: results
vitalSigns: vitalSignsOrganizer

class SinglePatient extends ContextClass
  FamilyHistory: FamilyHistory
  Patient: personInformation

class Population extends ContextClass
  FamilyHistory: Sequence(FamilyHistory)
  Patients: Sequence(SinglePatient)

class Template extends SinglePatient
  TemplateID: String
  RootAct: Act
  FocusAct: Act
  LocalVariable(VarName: String): Any

class eMeasure extends Template

class insuranceProvider
  financialResponsibilityPartyType: CD
  guarantorInformation: Sequence(guarantorInformation)
  healthInsuranceType: CD
  healthPlanName: String
  memberInformation: memberInformation
  payer: organizationInformation
  resultID: II
  subscriberInformation: subscriberInformation

class languageSpoken
  languageCode: CS
  modeCode: CE
  preferenceInd: preferenceInd

class medication
  admissionTiming: Sequence(admissionTiming)
  deliveryMethod: Sequence(CD)
  dose: PQ
  doseIndicator: String
  doseRestriction: doseRestriction
  fulfillmentHistory: Sequence(fulfillmentHistory)
  fulfillmentInstructions: String
  indicateMedicationStopped: IVL_TS
  indication: Sequence(indication)
  medicationInformationStopped: medicationInformation
  orderInformation: Sequence(orderInformation)
  patientInstructions: String
  productForm: CD
  reaction: CD
  route: CD
  site: Sequence(CD)
statusOfMedication: statusOfMedication

typeOfMedication: CD

vehicle: Sequence(vehicle)

class medicationInformation

codedBrandName: CD
codedProductName: CD
drugManufacturer: organizationInformation

class memberInformation

healthPlanCoverageDates: IVL_TS

memberID: II

patientAddress: Sequence(addr)

patientDateOfBirth: TS

patientName: pnm

patientPhone: Sequence(tele)

patientRelationshipToSubscriber: CD

class nullFlavorType Enum("NP", "NI", "MSK", "NA", "OTH", "NINF", "PINF", 
"UNK", "NASK", "TRC", "SKU", "NAV")

class onm

nullFlavor: nullFlavorType

class orderInformation

fills: fills

orderDateTime: IVL_TS

orderExpirationDateTime: TS

orderNumber: II

quantityOrdered: PQ

class organizationInformation

organizationAddress: Sequence(addr)

organizationID: Sequence(II)

organizationName: onm

organizationTelecom: Sequence(tele)

class patientInformation

personAddress: Sequence(addr)

personID: II

personInformation: Sequence(personInformation)

personPhone: Sequence(tele)

class personalInformation

patientInformation: patientInformation

class personInformation

birthPlace: birthPlace

ethnicity: CD

gender: CD

maritalStatus: Sequence(CD)

personDateOfBirth: TS

personName: Sequence(pnm)

race: Sequence(CD)

religiousAffiliation: CD

class pivlTs

alignment: String

institutionSpecified: Boolean

nullFlavor: nullFlavorType

operator: String

period: PQ

phase: IVL_TS

value: String

class plannedEvent
planFreeText: String
planId: II
plannedTime: IVL_TS
planType: CD

class planOfCare
plannedAct: plannedEvent
plannedObservation: plannedEvent
plannedProcedure: plannedEvent

class pnmm
family: String
given: String
nullFlavor: nullFlavorType
prefix: String
suffix: String

class preferenceInd
nullFlavor: nullFlavorType
value: String

class pregnancy extends CD

class procedure
bodySite: CD
procedureDateTime: IVL_TS
procedureFreeTextType: String
procedureID: Sequence(II)
procedureProvider: Sequence(organizationInformation)
procedureType: CD


class providerInformation
providerAddress: Sequence(addr)
providerID: II
providerName: pnmm
providerOrganizationName: onmm
providerTelecom: Sequence(tele)
providerType: CD


class qualifier
name: CD
value: ANY


class reasonForVisit
reason: CD


class reference
referenceDocumentID: II
referenceDocumentURL: String


class relationshipType Enum("CHILD", "SON", "HUSB")


class Relative extends Any
ClinicalGenomicChoice: Sequence(ClinicalGenomicChoice)
deceasedEstimatedAge: REAL
livingEstimatedAge: REAL
naturalFatherID: ST
naturalMotherID: ST
patientID: ST
person: pnmm
Relationship: CD


class result
resultDateTime: IVL_TS
resultID: Sequence(II)
resultInterpretation: CD
resultReferenceRange: Sequence(resultReferenceRange)
resultStatus: CS
resultType: CD
resultValue: ANY

class resultReferenceRange

class results
  resultsOrganizer: resultsOrganizer

class resultsOrganizer
  code: CD
effectiveTime: IVL_TS
  result: Sequence(result)

class simpleCode
  code: String
codeSystem: String
codeSystemName: String
displayName: String
nullFlavor: nullFlavorType

class statusOfMedication extends CD
  status: statusType

class statusType Enum("normal", "active", "cancelled", "pending", "suspended", "terminated", "nullified", "completed")

class subscriberInformation
  subscriberAddress: addr
  subscriberDateOfBirth: TS
  subscriberID: II
  subscriberName: pnm
  subscriberPhone: Sequence(tele)

class support
  contact: Sequence(contactType)
date: IVL_TS
guardian: contactType

class supportType Enum("GUARD", "AGNT")

class tele
  nullFlavor: nullFlavorType
  use: String
  useablePeriod: Sequence(useablePeriod)
  value: String

class translation
  code: String
codeSystem: String
codeSystemName: String
codeSystemVersion: String
displayName: String
nullFlavor: nullFlavorType
originalText: ED
  translation: String
  value: String

class treatingProvider
  treatingProviderID: II

class useablePeriod
  nullFlavor: nullFlavorType
operator: String
value: String
class valueType Enum("CE", "BL", "BN", "BIN", "CR", "UID", "ID", "URL", "AD", "BL", "EN", "QTY", "PQR", "RTO")
class vehicle extends CD
  name: String
class version
  number: String
  setID: II
class vitalSignsOrganizer
  code: CD
  effectiveTime: IVL_TS
  vitalSign: Sequence(result)
EndPackage
2.5 Detailed Clinical Model example (Informative)

A Mock Urine laboratory report is used to illustrate how the grammar could be used to define a detailed clinical model for use in clinical decision support. The content is informative and is included to provide example usage of the grammar for defining a class structure that is based on a detailed clinical model. A DCM could also be incorporated by using the generic information structure included in the base vMR, but this templates constraining the structure to be semantically equivalent to the DCM.

```
Package PKG_CEN_UrineCulture_V1
  imports VMR
  class DipStick
    Blood: Observation
    Glucose: Observation
    Protein: Observation
    PH: Observation
    SpecificGravity: Observation
  class Microscopy
    WhiteCells: Observation
    RedCells: Observation
    EpithelialCells: Observation
    OtherCells: Observation
    Other: Observation
  class Pathogen
    Organism: Observation
    Number: Observation
    Ampicillin: Observation
    Augmentin: Observation
    Trimethoprim: Observation
    Co_Trimoxazole: Observation
    Nitrofurantoin: Observation
    Norfloxacin: Observation
    Cephalexin: Observation
    Gentamicin: Observation
  class Isolate
    Growth: Observation
    Pathogen: Pathogen
  class Culture
    Isolate: Sequence(Isolate)
  class NonReportableItems
    WorkingNotes: Observation
  class CEN_UrineCulture_V1
    SpecimenType: Observation
    OtherSpecimenType: Observation
    DipStick: DipStick
    Microscopy: Microscopy
    Culture: Culture
    Comments: Observation
    NonReportableItems: NonReportableItems

EndPackage
```