



1

2

3

4

Virtual Medical Record (vMR) for Clinical Decision Support – GELLO Implementation Guide

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

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

Draft Standard for Trial Use

September 2011

24 **Table of Contents**

25 *Table of Contents* 2

26 *1 Executive Summary*..... 3

27 *2 GELLO implementation guide for VMR*..... 3

28

29

30 1 Executive Summary

31 A Virtual Medical Record (vMR) for Clinical Decision Support (CDS) is a data model
32 for representing clinical information *inputs* and *outputs* that can be used by CDS
33 engines and local clinical information systems, through mechanisms such as CDS
34 services, or execution of standardised clinical support logic by a GELLO execution
35 engine. A vMR for CDS is needed to enable the design and development of scalable
36 CDS resources that can be used across multiple healthcare institutions and health
37 information systems. Existing clinical data, from any source, is virtualised to
38 present a façade compliant with the vMR class model. Clinical logic can then be
39 executed against the data represented in the vMR model. GELLO is one standard
40 language that can be used with the vMR model, but other languages can use the
41 vMR model.

42

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

60

61 This is the first DSTU ballot for this material. Input is sought on the format rather
62 than the content, as a normative vMR model will be defined elsewhere using this
63 grammar. It is envisaged that all elements of the final vMR, when this is finalised,
64 will be represented in both UML (.xmi files), class diagrams, and this format. As this
65 format is both human and machine readable it is intended to make this format a
66 more readily understandable form of the vMR standard.

67

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

72 2 GELLO implementation Guide for VMR

73

74 2.1 Grammar for GELLO class definitions (Normative)

75

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

78

79 GelloPackage ::= PackageStatement

80

81 PackageStatement ::=

82 <PACKAGE> NameOrLiteral ImportsStatement? PackageElement* <ENDPACKAGE>

83

84 ImportsStatement ::=

85 <IMPORTS> ImportName ("," ImportName)*

86

87 ImportName ::= NameOrLiteral

88

89 PackageElement ::= PackageStatement | ClassDefinition

90

91 ClassDefinition ::=

92 <CLASS> ClassDefName (<EXTENDS> ClassRef)? ClassMember* |

93 <CLASS> ClassDefName <EQUAL> ClassRef |

94 <CLASS> ClassName EnumerationType

95

96 ClassDefName ::=

97 NameOrLiteral |

98 NameOrLiteral "(" ClassDefParams ")"

99

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

101

102 ClassRefOrParam ::= ClassRef |

103 Name ":" ClassRef

104

105 ClassRef ::=

106 ClassName |

107 ClassName "(" ClassRef ")" |

108 <TUPLE>

109

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

111 ClassRef

112

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

114

115 ClassFormalParam ::= Name ":" ClassRef

116

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

118

119 EnumLiteral ::= NameOrLiteral

120

121 NameOrLiteral ::= Name | <STRING_LITERAL>

122

123 ClassName ::= NameWithPath | <STRING_LITERAL>

124

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

126

127 Name ::= <ID> ("." <ID>)*

128

129 <STRING_LITERAL: ('\'\' (~['\', "\n", "\r"])* '\'| "\'\' (~['\', "\n",
130 "\r"])* "\'\') >
131
132 <ID: ["a"- "z", "A"- "Z"] (["a"- "z", "A"- "Z", "0"- "9"] | "_"(["a"- "z", "A"- "Z", "0"-
133 "9"])+)* >
134
135 <ENUM: "Enum" >
136
137 <TUPLE: "Tuple" >
138
139 <EQUAL: "=" >
140
141 <ENDPACKAGE: "endPackage" | "EndPackage" | "endpackage" >
142
143 <CLASS: "class" | "Class" >
144
145 <EXTENDS: "extends" >
146
147 <IMPORTS: "Imports" | "imports" >
148
149 <PACKAGE: "Package" | "package" >
150
151

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
  ">=" (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
```

```

214     ltrim(trimChars:String):String
215     rtrim(trimChars:String):String
216     replace(replaceThis:String, withThis:String):String
217
218 Class Boolean extends Any
219     "="(rhs: Boolean): Boolean
220     "<>"(rhs: Boolean): Boolean
221     "or"(rhs: Boolean): Boolean
222     "xor"(rhs: Boolean): Boolean
223     "and"(rhs: Boolean): Boolean
224     "not"(): Boolean
225     implies(rhs: Boolean): Boolean
226
227 Class Collection extends Any
228
229     count(o: T): Integer
230     excludes(o: T): Boolean
231     excludesAll(c: Collection): Boolean
232     includes(o: T): Boolean
233     includesAll(c: Collection): Boolean
234     isEmpty(): Boolean
235     notEmpty(): Boolean
236     size(): Integer
237     sum(): T
238     "max"(): T
239     "min"(): T
240
241     "="(s: Collection): Boolean
242     "<>"(s: Collection): Boolean
243
244     asSet(): Set(T)
245     asBag(): Bag(T)
246     asSequence(): Sequence(T)
247
248     exists(e: Boolean): Boolean
249     forAll(e: Any): Boolean
250     collect(e: Any): Bag(T)
251     sortBy(e: Any): Sequence(T)
252     Average():Real
253     Stdev():Real
254     Variance():Real
255     Median():Real
256     Mode():Real
257     Join(): Collection(Tuple)
258     Like(): Collection(String)
259     NotLike(): Collection(String)
260     Between(string1: String, String2: String): Collection(String)
261     Distinct(): Set(T)
262
263
264 Class "Set" extends Collection
265     "-"(s: Set(T)): Set(T)
266     union(c: Bag(T)): Bag(T)
267     union(s: Set(T)): Set(T)
268     intersection(b: Bag(T)): Bag(T)
269     intersection(s: Set(T)): Set(T)
270     including(o: T): Set(T)
271     excluding(o: T): Set(T)
272     symmetricDifference(s: Set(T)): Set(T)
273     flatten(): Set(T)
274
275     select(e: Boolean): Set(T)
276     reject(e: Boolean): Set(T)
277
278
279 Class "Bag" extends Collection

```

```

280     union(b: Bag(T)): Bag(T)
281     union(s: Set(T)): Bag(T)
282     intersection(b: Bag(T)): Bag(T)
283     intersection(s: Set(T)): Bag(T)
284     including(o: T): Bag(T)
285     excluding(o: T): Bag(T)
286     flatten(): Bag(T)
287
288     select(e: Boolean): Bag(T)
289     reject(e: Boolean): Bag(T)
290
291     Class "Sequence" extends Collection
292     union(s: Sequence(T)): Sequence(T)
293     including(o: T): Sequence(T)
294     excluding(o: T): Sequence(T)
295
296     append(o: T): Sequence(T)
297     at(i: Integer): T
298     first(): T
299     flatten(): Sequence(T)
300     indexOf(o: T): Integer
301     insertAt(i: Integer, o: T): Sequence(T)
302     last(): T
303     prepend(o: T): Sequence(T)
304     subSequence(lower: Integer, upper: Integer): Sequence(T)
305
306     select(e: Boolean): Sequence(T)
307     reject(e: Boolean): Sequence(T)
308     lastN(number: Integer): Sequence(T)
309     firstN(number: Integer): Sequence(T)
310     Reverse(): Sequence(T)
311
312     EndPackage
313

```


2.3 ISO21090 Data Types Definition (Informative)

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 vMR may be simplified further.

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

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

class NullFlavor Enum("NI", "INV", "OTH", "NINF", "PINF", "UNC", "DER",
"UNK", "ASKU", "NAV", "NASK", "QS", "TRC", "MSK", "NA")

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

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

class XP
  code: String
  codeSystem: String
  codeSystemVersion: String2
  language: Code
  nullFlavor: NullFlavor
  value: String

class AddressPartType Enum("AL", "ADL", "UNID", "UNIT", "DAL", "DINST",
"DINSTA", "DINSTQ", "DMOD", "DMODID", "SAL", "BNR", "BNN", "BNS", "STR",
"STB", "STTYP", "DIR", "INT", "CAR", "CEN", "CNT", "CPA", "CTY", "DEL", "POB",
"PRE", "STA", "ZIP", "DPID")

class ADXP extends XP
  type: AddressPartType

class PostalAddressUse Enum("H", "HP", "HV", "WP", "DIR", "PUB", "BAD",
"PHYS", "PST", "TMP", "ABC", "IDE", "SYL", "SRCH", "SNDX", "PHON")

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
```

```

375         "or"(other : BL) : BL
376         "xor"(other : BL) : BL
377         implies(other : BL) : BL
378         "not"() : BL
379
380     class CodingRationale Enum("O", "P", "R", "OR", "PR")
381
382     class CD extends ANY
383         code: String
384         codeSystem: Uid
385         codeSystemName: String
386         codeSystemVersion: String
387         codingRationale: CodingRationale
388         displayName: ST
389         id: String
390         originalText: ED
391         source: XReference
392         translation: Sequence(CD)
393         valueSet: Uid
394         valueSetVersion: String
395         --operations
396         implies(other : CS) : BL
397         implies(other : CD) : BL
398
399
400     class CO extends QTY
401         code: CD
402         value: Decimal
403         --operations
404         "+"(other : CO) : CO
405         "-"(other : CO) : CO
406         "min"(other : CO) : CO
407         "max"(other : CO) : CO
408
409     class CS extends ANY
410         code: String
411         --operations
412         codeSystem : Uid
413         codeSystemName : String
414         codeSystemVersion : String
415         implies(other : CS) : BL
416         implies(other : CD) : BL
417
418     class Decimal extends Real
419
420     class Compression Enum("DF", "GZ", "ZL", "Z", "BZ", "Z7")
421
422     class Code extends String
423
424     class IntegrityCheckAlgorithm Enum("SHA1", "SHA256")
425
426     class ED extends ANY
427         charset: Code
428         compression: Compression
429         data: String
430         description: ST
431         integrityCheck: String
432         integrityCheckAlgorithm: IntegrityCheckAlgorithm
433         language: Code
434         mediaType: String
435         reference: TEL
436         thumbnail: ED
437         translation: Sequence(ED)
438         value: String
439         xml: Any
440

```

```

441     class TimingEvent Enum("HS", "WAKE", "AC", "ACM", "ACD", "ACV", "IC", "ICM",
442 "ICD", "ICV", "PC", "PCM", "PCD", "PCV", "C", "CM", "CD", "CV")
443
444     class EIVL_TS extends QSET_TS
445         event: TimingEvent
446         offset: IVL_PQ
447
448     class EntityNamePartQualifier Enum("LS", "AC", "NB", "PR", "HON", "BR",
449 "AD", "SP", "MID", "CL", "IN", "PFX", "SFX")
450
451     class EntityNamePartType Enum("FAM", "GIV", "TITLE", "DEL")
452
453     class ENXP extends XP
454         qualifier: Set(EntityNamePartQualifier)
455         type: EntityNamePartType
456
457     class EntityNameUse Enum("ABC", "SYL", "IDE", "C", "OR", "T", "I", "P",
458 "ANON", "A", "R", "OLD", "DN", "M", "SRCH", "PHON")
459
460     class EN extends ANY
461         part: Sequence(ENXP)
462         use: set_EntityNameUse
463         --operations
464         canonical() : EN
465
466     class GLIST_PQ extends ANY
467         denominator: Integer
468         head: PQ
469         increment: QTY
470         period: Integer
471
472     class GLIST_REAL extends ANY
473         denominator: Integer
474         head: REAL
475         increment: QTY
476         period: Integer
477
478     class GLIST_TS extends ANY
479         denominator: Integer
480         head: TS
481         increment: QTY
482         period: Integer
483
484     class IdentifierReliability Enum("ISS", "VRF", "UNV")
485
486     class IdentifierScope Enum("BUSN", "OBJ", "VER", "VW")
487
488     class II extends ANY
489         displayable: Boolean
490         extension: String
491         identifierName: String
492         reliability: IdentifierReliability
493         root: Uid
494         scope: IdentifierScope
495
496     class INT extends QTY
497         value: Integer
498         --operations
499         "+"(other : INT) : INT
500         "-"(other : INT) : INT
501         "*" (other : INT) : INT
502         "/"(other : INT) : REAL
503         "/"(other : REAL) : REAL
504         abs() : INT
505         "div"(other : INT) : INT
506         "mod"(other : INT) : INT

```

```

507         "min"(other : INT) : INT
508         "max"(other : INT) : INT
509
510     class IVL_CO extends QSET_CO
511         any: CO
512         high: CO
513         highClosed: Boolean
514         low: CO
515         lowClosed: Boolean
516         width: QTY
517
518     class IVL_INT extends QSET_INT
519         any: INT
520         high: INT
521         highClosed: Boolean
522         low: INT
523         lowClosed: Boolean
524         width: QTY
525
526     class IVL_MO extends QSET_MO
527         any: MO
528         high: MO
529         highClosed: Boolean
530         low: MO
531         lowClosed: Boolean
532         width: QTY
533
534     class IVL_PQ extends QSET_PQ
535         any: PQ
536         high: PQ
537         highClosed: Boolean
538         low: PQ
539         lowClosed: Boolean
540         width: QTY
541
542     class IVL_QTY extends QSET_QTY
543         any: QTY
544         high: QTY
545         highClosed: Boolean
546         low: QTY
547         lowClosed: Boolean
548         width: QTY
549
550     class IVL_REAL extends QSET_REAL
551         any: REAL
552         high: REAL
553         highClosed: Boolean
554         low: REAL
555         lowClosed: Boolean
556         width: QTY
557
558     class IVL_TS extends QSET_TS
559         any: TS
560         high: TS
561         highClosed: Boolean
562         low: TS
563         lowClosed: Boolean
564         width: QTY
565
566     class MO extends QTY
567         currency: Code
568         value: Decimal
569         --operations
570         "+"(other : MO) : MO
571         "-"(other : MO) : MO
572         "*" (other : REAL) : MO

```

```

573         "/"(other : REAL) : MO
574         "min"(other : MO) : MO
575         "max"(other : MO) : MO
576
577     class CalendarCycle Enum("CY", "MY", "CM", "CW", "WM", "WY", "DM", "CD",
578 "DY", "DW", "HD", "CH", "NH", "CN", "SN", "CS")
579
580     class PIVL_TS extends QSET_TS
581     alignment: CalendarCycle
582     count: INT
583     frequency: RTO
584     isFlexible: Boolean
585     period: PQ
586     phase: IVL_TS
587
588     class PQ extends QTY
589     codingRationale: CodingRationale
590     translation: Sequence(PQR)
591     unit: Code
592     value: Decimal
593     --operations
594     canonical : PQ
595     "+"(other : PQ) : PQ
596     "-"(other : PQ) : PQ
597     "-"() : PQ
598     "*" (other : PQ) : PQ
599     "*" (other : Real) : PQ
600     "*" (other : Real) : PQ
601     "*" (other : Integer) : PQ
602     "/" (other : PQ) : PQ
603     "/" (other : Real) : PQ
604     "/" (other : Real) : PQ
605     "/" (other : Integer) : PQ
606     inverted() : PQ
607     abs() : PQ
608     "min"(other : PQ) : PQ
609     "max"(other : PQ) : PQ
610     toInterval() : IVL_PQ
611     convert(unit: Code):PQ
612
613     class PQR extends CD
614     value: Decimal
615
616
617
618     class QSC_TS extends QSET_TS
619     code: CD
620
621     class QSD_PQ extends QSET_PQ
622     minuend: QSET_PQ
623     subtrahend: QSET_PQ
624
625     class QSD_TS extends QSET_TS
626     minuend: QSET_TS
627     subtrahend: QSET_TS
628
629     class QSET_CO extends ANY
630     originalText: ED
631
632     class QSET_INT extends ANY
633     originalText: ED
634
635     class QSET_MO extends ANY
636     originalText: ED
637
638     class QSET_PQ extends ANY

```

```

639         originalText: ED
640
641     class QSET_QTY extends ANY
642         originalText: ED
643
644     class QSET_REAL extends ANY
645         originalText: ED
646
647     class QSET_TS extends ANY
648         originalText: ED
649
650     class QSI_PQ extends QSET_PQ
651         term: Sequence(QSET_PQ)
652
653     class QSI_TS extends QSET_TS
654         term: Sequence(QSET_TS)
655
656     class QSP_PQ extends QSET_PQ
657         high: QSET_PQ
658         low: QSET_PQ
659
660     class QSP_TS extends QSET_TS
661         high: QSET_TS
662         low: QSET_TS
663
664     class QSS_PQ extends QSET_PQ
665         term: Sequence(PQ)
666
667     class QSS_TS extends QSET_TS
668         term: Sequence(TS)
669
670     class QSU_PQ extends QSET_PQ
671         term: Sequence(QSET_PQ)
672
673     class QSU_TS extends QSET_TS
674         term: Sequence(QSET_TS)
675
676     class UncertaintyType Enum("U", "N", "LN", "G", "E", "X2", "T", "F", "B")
677
678     class QTY extends ANY
679         expression: ED
680         originalText: ED
681         uncertainRange: IVL_QTY
682         uncertainty: QTY
683         uncertaintyType: UncertaintyType
684         --operations
685         "<"(other : QTY) : BL
686         "<="(other : QTY) : BL
687         ">="(other : QTY) : BL
688         ">"(other : QTY) : BL
689         "-"(other : QTY) : QTY
690         "+"(other : QTY) : QTY
691         comparable(other : QTY) : Boolean
692         isDifference(other : QTY) : Boolean
693
694     class REAL extends QTY
695         value: Decimal
696         --operations
697         "+"(other : REAL) : REAL
698         "-"(other : REAL) : REAL
699         "*" (other : REAL) : REAL
700         "/"(other : REAL) : REAL
701         "min"(other : REAL) : REAL
702         "max"(other : REAL) : REAL
703         "-"() : REAL
704         abs() : REAL

```

```

705         ceiling() : INT
706         floor() : INT
707         round() : INT
708         inverted() : REAL
709         power(other : REAL) : REAL
710         toInterval() : IVL_REAL
711
712     class RTO extends QTY
713         denominator: QTY
714         numerator: QTY
715
716     class SC extends ST
717         code: CD
718
719     class SD_TEXT extends ANY
720         ID: String
721         language: Code
722         mediaType: String
723         styleCode: set_Code
724
725     class SD_TITLE extends ANY
726         ID: String
727         language: Code
728         mediaType: String
729         styleCode: set_Code
730
731     class SLIST_INT extends ANY
732         digit: Sequence(INT)
733         origin: INT
734         scale: QTY
735
736     class SLIST_PQ extends ANY
737         digit: Sequence(INT)
738         origin: PQ
739         scale: QTY
740
741     class SLIST_REAL extends ANY
742         digit: Sequence(INT)
743         origin: REAL
744         scale: QTY
745
746     class SLIST_TS extends ANY
747         digit: Sequence(INT)
748         origin: TS
749         scale: QTY
750
751     class ST extends ANY
752         language: Code
753         translation: Sequence(ST)
754         value: String
755         --operations
756         mediaType() : String
757         size() : Integer
758         concat(other : ST) : ST
759         substring(lower : INT, upper : INT) : ST
760         toInteger() : INT
761         toReal() : REAL
762         toTimestamp() : TS
763
764     class TelecommunicationCapability Enum("voice", "fax", "data", "tty", "sms")
765
766     class TelecommunicationAddressUse Enum("H", "HP", "HV", "WP", "DIR", "PUB",
767 "BAD", "TMP", "AS", "EC", "MC", "PG")
768
769     class TEL extends ANY
770         capabilities: Set(TelecommunicationCapability)

```

```

771         use: Set(TelecommunicationAddressUse)
772         useablePeriod: QSET_TS
773         value: String
774         --operations
775         canonical() : TEL
776
777     class TS extends QTY
778         value: String
779         --operations
780         "+"(other : PQ) : TS
781         "-"(other : PQ) : TS
782         "-"(other : TS) : PQ
783         "min"(other : TS) : TS
784         "max"(other : TS) : TS
785         toInterval() : IVL_TS
786         precision() : Integer
787
788
789     class UVP_AD extends ANY
790         probability: Decimal
791         value: AD
792
793     class UVP_BL extends ANY
794         probability: Decimal
795         value: BL
796
797     class UVP_CD extends ANY
798         probability: Decimal
799         value: CD
800
801     class UVP_CO extends ANY
802         probability: Decimal
803         value: CO
804
805     class UVP_CS extends ANY
806         probability: Decimal
807         value: CS
808
809     class UVP_ED extends ANY
810         probability: Decimal
811         value: ED
812
813     class UVP_EN extends ANY
814         probability: Decimal
815         value: EN
816
817     class UVP_II extends ANY
818         probability: Decimal
819         value: II
820
821     class UVP_INT extends ANY
822         probability: Decimal
823         value: INT
824
825     class UVP_MO extends ANY
826         probability: Decimal
827         value: MO
828
829     class UVP_PQ extends ANY
830         probability: Decimal
831         value: PQ
832
833     class UVP_REAL extends ANY
834         probability: Decimal
835         value: REAL
836

```



```

837     class UVP_RTO extends ANY
838         probability: Decimal
839         value: RTO
840
841     class UVP_SC extends ANY
842         probability: Decimal
843         value: SC
844
845     class UVP_ST extends ANY
846         probability: Decimal
847         value: ST
848
849     class UVP_TEL extends ANY
850         probability: Decimal
851         value: TEL
852
853     class UVP_TS extends ANY
854         probability: Decimal
855         value: TS
856
857     class Uid extends String
858
859     class Uri extends String
860
861     class XReference
862         xref: String
863
864     class XmlID extends String
865
866     class XmlIDREF extends String
867
868     class set_Code
869         value: Set(Code)
870
871     class set_CodingRationale
872         value: Set(CodingRationale)
873
874     class set_EntityNamePartQualifier
875         value: Set(EntityNamePartQualifier)
876
877     class set_EntityNameUse
878         value: Set(EntityNameUse)
879
880     class set_IDREF
881         value: Set(XmlIDREF)
882
883     class set_TelecommunicationAddressUse
884         value: Set(TelecommunicationAddressUse)
885
886     class set_TelecommunicationCapability
887         value: Set(TelecommunicationCapability)
888
889     class COLL_AD extends ANY
890
891     class COLL_BL extends ANY
892
893     class COLL_CD extends ANY
894
895     class COLL_CO extends ANY
896
897     class COLL_CS extends ANY
898
899     class COLL_ED extends ANY
900
901     class COLL_EN extends ANY
902

```

```

903     class COLL_II extends ANY
904
905     class COLL_INT extends ANY
906
907     class COLL_MO extends ANY
908
909     class COLL_PQ extends ANY
910
911     class COLL_REAL extends ANY
912
913     class COLL_RTO extends ANY
914
915     class COLL_SC extends ANY
916
917     class COLL_ST extends ANY
918
919     class COLL_TEL extends ANY
920
921     class COLL_TS extends ANY
922
923     class BAG_AD extends COLL_AD
924         value: Bag(AD)
925
926     class BAG_BL extends COLL_BL
927         value: Bag(BL)
928
929     class BAG_CD extends COLL_CD
930         value: Bag(CD)
931
932     class BAG_CO extends COLL_CO
933         value: Bag(CO)
934
935     class BAG_CS extends COLL_CS
936         value: Bag(CS)
937
938     class BAG_ED extends COLL_ED
939         value: Bag(ED)
940
941     class BAG_EN extends COLL_EN
942         value: Bag(EN)
943
944     class BAG_II extends COLL_II
945         value: Bag(II)
946
947     class BAG_INT extends COLL_INT
948         value: Bag(INT)
949
950     class BAG_MO extends COLL_MO
951         value: Bag(MO)
952
953     class BAG_PQ extends COLL_PQ
954         value: Bag(PQ)
955
956     class BAG_REAL extends COLL_REAL
957         value: Bag(REAL)
958
959     class BAG_RTO extends COLL_RTO
960         value: Bag(RTO)
961
962     class BAG_SC extends COLL_SC
963         value: Bag(SC)
964
965     class BAG_ST extends COLL_ST
966         value: Bag(ST)
967
968     class BAG_TEL extends COLL_TEL

```

```

969         value: Bag(TEL)
970
971     class BAG_TS extends COLL_TS
972         value: Bag(TS)
973
974     class DSET_AD extends COLL_AD
975         value: Sequence(AD)
976
977     class DSET_BL extends COLL_BL
978         value: Sequence(BL)
979
980     class DSET_CD extends COLL_CD
981         value: Sequence(CD)
982
983     class DSET_CO extends COLL_CO
984         value: Sequence(CO)
985
986     class DSET_CS extends COLL_CS
987         value: Sequence(CS)
988
989     class DSET_ED extends COLL_ED
990         value: Sequence(ED)
991
992     class DSET_EN extends COLL_EN
993         value: Sequence(EN)
994
995     class DSET_II extends COLL_II
996         value: Sequence(II)
997
998     class DSET_INT extends COLL_INT
999         value: Sequence(INT)
1000
1001     class DSET_MO extends COLL_MO
1002         value: Sequence(MO)
1003
1004     class DSET_PQ extends COLL_PQ
1005         value: Sequence(PQ)
1006
1007     class DSET_REAL extends COLL_REAL
1008         value: Sequence(REAL)
1009
1010     class DSET_RTO extends COLL_RTO
1011         value: Sequence(RTO)
1012
1013     class DSET_SC extends COLL_SC
1014         value: Sequence(SC)
1015
1016     class DSET_ST extends COLL_ST
1017         value: Sequence(ST)
1018
1019     class DSET_TEL extends COLL_TEL
1020         value: Sequence(TEL)
1021
1022     class DSET_TS extends COLL_TS
1023         value: Sequence(TS)
1024
1025     class HIST_AD extends LIST_AD
1026
1027     class HIST_BL extends LIST_BL
1028
1029     class HIST_CD extends LIST_CD
1030
1031     class HIST_CO extends LIST_CO
1032
1033     class HIST_CS extends LIST_CS
1034

```

```

1035      class HIST_ED extends LIST_ED
1036
1037      class HIST_EN extends LIST_EN
1038
1039      class HIST_II extends LIST_II
1040
1041      class HIST_INT extends LIST_INT
1042
1043      class HIST_MO extends LIST_MO
1044
1045      class HIST_PQ extends LIST_PQ
1046
1047      class HIST_REAL extends LIST_REAL
1048
1049      class HIST_RTO extends LIST_RTO
1050
1051      class HIST_SC extends LIST_SC
1052
1053      class HIST_ST extends LIST_ST
1054
1055      class HIST_TEL extends LIST_TEL
1056
1057      class HIST_TS extends LIST_TS
1058
1059      class LIST_AD extends COLL_AD
1060         value: Sequence(AD)
1061
1062      class LIST_BL extends COLL_BL
1063         value: Sequence(BL)
1064
1065      class LIST_CD extends COLL_CD
1066         value: Sequence(CD)
1067
1068      class LIST_CO extends COLL_CO
1069         value: Sequence(CO)
1070
1071      class LIST_CS extends COLL_CS
1072         value: Sequence(CS)
1073
1074      class LIST_ED extends COLL_ED
1075         value: Sequence(ED)
1076
1077      class LIST_EN extends COLL_EN
1078         value: Sequence(EN)
1079
1080      class LIST_II extends COLL_II
1081         value: Sequence(II)
1082
1083      class LIST_INT extends COLL_INT
1084         value: Sequence(INT)
1085
1086      class LIST_MO extends COLL_MO
1087         value: Sequence(MO)
1088
1089      class LIST_PQ extends COLL_PQ
1090         value: Sequence(PQ)
1091
1092      class LIST_REAL extends COLL_REAL
1093         value: Sequence(REAL)
1094
1095      class LIST_RTO extends COLL_RTO
1096         value: Sequence(RTO)
1097
1098      class LIST_SC extends COLL_SC
1099         value: Sequence(SC)
1100

```

```

1101     class LIST_ST extends COLL_ST
1102         value: Sequence(ST)
1103
1104     class LIST_TEL extends COLL_TEL
1105         value: Sequence(TEL)
1106
1107     class LIST_TS extends COLL_TS
1108         value: Sequence(TS)
1109
1110         class NPPD_AD extends ANY
1111             value: Sequence(UVP_AD)
1112
1113     class NPPD_BL extends ANY
1114         value: Sequence(UVP_BL)
1115
1116     class NPPD_CD extends ANY
1117         value: Sequence(UVP_CD)
1118
1119     class NPPD_CO extends ANY
1120         value: Sequence(UVP_CO)
1121
1122     class NPPD_CS extends ANY
1123         value: Sequence(UVP_CS)
1124
1125     class NPPD_ED extends ANY
1126         value: Sequence(UVP_ED)
1127
1128     class NPPD_EN extends ANY
1129         value: Sequence(UVP_EN)
1130
1131     class NPPD_II extends ANY
1132         value: Sequence(UVP_II)
1133
1134     class NPPD_INT extends ANY
1135         value: Sequence(UVP_INT)
1136
1137     class NPPD_MO extends ANY
1138         value: Sequence(UVP_MO)
1139
1140     class NPPD_PQ extends ANY
1141         value: Sequence(UVP_PQ)
1142
1143     class NPPD_REAL extends ANY
1144         value: Sequence(UVP_REAL)
1145
1146     class NPPD_RTO extends ANY
1147         value: Sequence(UVP_RTO)
1148
1149     class NPPD_SC extends ANY
1150         value: Sequence(UVP_SC)
1151
1152     class NPPD_ST extends ANY
1153         value: Sequence(UVP_ST)
1154
1155     class NPPD_TEL extends ANY
1156         value: Sequence(UVP_TEL)
1157
1158     class NPPD_TS extends ANY
1159         value: Sequence(UVP_TS)
1160
1161     class StrucDoc_Align Enum("left", "center", "right", "justify", "char")
1162
1163     class StrucDoc_Br
1164
1165     class StrucDoc_CMTtitle
1166         br: StrucDoc_Br

```

```

1167         content: StrucDoc_CMTitle
1168         footnote: StrucDoc_TitleFootnote
1169         footnoteRef: StrucDoc_FootnoteRef
1170         linkHtml: StrucDoc_LinkHtml
1171         sub: StrucDoc_Sub
1172         sup: StrucDoc_Sup
1173
1174     class StrucDoc_Caption
1175
1176     class StrucDoc_Captioned
1177         caption: StrucDoc_Caption
1178
1179     class StrucDoc_CellScope Enum("row", "col", "rowgroup", "colgroup")
1180
1181     class StrucDoc_Col extends StrucDoc_ColItem
1182
1183     class StrucDoc_ColGroup extends StrucDoc_ColItem
1184         col: Sequence(StrucDoc_Col)
1185
1186     class StrucDoc_ColItem extends StrucDoc_TableItem
1187         span: Integer
1188         width: StrucDoc_Length
1189
1190     class StrucDoc_Content
1191         revised: StrucDoc_Revised
1192
1193     class StrucDoc_Footnote
1194
1195     class StrucDoc_FootnoteRef
1196         IDREF: String
1197
1198     class StrucDoc_Frame Enum("void", "above", "below", "hsides", "lhs", "rhs",
1199 "vsides", "box", "border")
1200
1201     class StrucDoc_Item extends StrucDoc_Captioned
1202
1203     class StrucDoc_Length extends String
1204
1205     class StrucDoc_LinkHtml
1206         href: String
1207         name: String
1208         rel: String
1209         rev: String
1210         title: String
1211
1212     class StrucDoc_List extends StrucDoc_Captioned
1213         value: Sequence(StrucDoc_Item)
1214         listType: StrucDoc_ListType
1215
1216     class StrucDoc_ListType Enum("ordered", "unordered")
1217
1218     class StrucDoc_Paragraph extends StrucDoc_Captioned
1219
1220     class StrucDoc_RenderMultiMedia
1221         caption: StrucDoc_Caption
1222         referencedObject: set_IDREF
1223
1224     class StrucDoc_Revised Enum("insert", "delete")
1225
1226     class StrucDoc_Rules Enum("none", "groups", "rows", "cols", "all")
1227
1228     class StrucDoc_Sub
1229
1230     class StrucDoc_Sup
1231
1232     class StrucDoc_TCell extends StrucDoc_TableItem

```

```

1233     abbr: String
1234     axis: String
1235     colspan: Integer
1236     headers: set_IDREF
1237     rowspan: Integer
1238     scope: StrucDoc_CellScope
1239
1240     class StrucDoc_TRow extends StrucDoc_TableItem
1241
1242     class StrucDoc_TRowGroup extends StrucDoc_TableItem
1243         tr: Sequence(StrucDoc_TRow)
1244
1245     class StrucDoc_Table extends StrucDoc_Captioned
1246         border: StrucDoc_Length
1247         cellpadding: StrucDoc_Length
1248         cellspacing: StrucDoc_Length
1249         col: Sequence(StrucDoc_Col)
1250         colgroup: Sequence(StrucDoc_ColGroup)
1251         frame: StrucDoc_Frame
1252         rules: StrucDoc_Rules
1253         summary: String
1254         tbody: Sequence(StrucDoc_TRowGroup)
1255         tfoot: StrucDoc_TRowGroup
1256         thead: StrucDoc_TRowGroup
1257         width: StrucDoc_Length
1258
1259     class StrucDoc_TableItem
1260         align: StrucDoc_Align
1261         char: String
1262         charoff: StrucDoc_Length
1263         valign: StrucDoc_VAlign
1264
1265     class StrucDoc_TitleFootnote
1266
1267     class StrucDoc_VAlign Enum("top", "middle", "bottom", "baseline")
1268
1269
1270 EndPackage
1271
1272

```

2.4 vMR Definition example (Informative)

1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334

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
  documentCustodian: 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)
```



```

1335     class confidentialityType Enum("N", "R", "V")
1336
1337     class contactType
1338         contactAddress: Sequence(addr)
1339         contactMothersMaidenName: onm
1340         contactName: Sequence(pnm)
1341         contactTelecom: Sequence(tele)
1342         relationship: relationshipType
1343         type: supportType
1344
1345     class currencyType Enum("ARS", "AUD", "BRL", "CAD", "CHF", "CLF", "CNY",
1346 "DEM", "ESP", "EUR", "FIM", "FRF", "GBP", "ILS", "INR", "JPY", "KRW", "MXN",
1347 "NLG", "NZD", "PHP", "RUR", "THB", "TRL", "TWD", "USD", "ZAR")
1348
1349     class custodian
1350         custodianAddress: addr
1351         custodianID: Sequence(II)
1352         custodianName: onm
1353         custodianTelecom: tele
1354
1355     class doseRestriction
1356         denominator: PQ
1357         numerator: PQ
1358
1359     class encounter
1360         asAct: Act
1361         admissionType: CD
1362         dischargeDisposition: String
1363         encounterDateTime: IVL_TS
1364         encounterID: Sequence(II)
1365         encounterProvider: Sequence(organizationInformation)
1366         encounterType: encounterType
1367         facilityLocation: facilityLocation
1368         reasonForVisit: reasonForVisit
1369
1370     class encounterType extends CD
1371
1372     class facilityLocation
1373         locationDuration: IVL_TS
1374
1375     class FamilyHistory extends Any
1376         ClinicalGenomicChoice: Sequence(ClinicalGenomicChoice)
1377         Informant: informationSource
1378         PatientID: ST
1379         PedigreeAnalysisResults: Sequence(result)
1380         Relatives: Sequence(Relative)
1381
1382     class ClinicalGenomicChoice extends Any
1383         ClinicalObservation: ClinicalObservation
1384         GeneticLoci: Sequence(result)
1385         Relative: Relative
1386
1387     class ClinicalObservation extends Any
1388         causeOfDeath: BL
1389         code: CD
1390         DataEstimatedAge: IVL_REAL
1391         negationInd: BL
1392
1393     class fillNumber
1394         nullFlavor: nullFlavorType
1395         value: String
1396
1397     class fills
1398         value: String
1399
1400     class fillStatus

```

```

1401         value: statusType
1402
1403     class fulfillmentHistory
1404         dispenseDate: IVL_TS
1405         dispensingPharmacyLocation: addr
1406         fillNumber: fillNumber
1407         fillStatus: fillStatus
1408         prescriptionNumber: II
1409         provider: organizationInformation
1410         quantityDispensed: PQ
1411
1412     class guarantorInformation
1413         partyAddress: Sequence(addr)
1414         partyName: pnm
1415         partyPhone: Sequence(tele)
1416         responsibilityEffectiveDate: IVL_TS
1417
1418     class header
1419         confidentiality: CD
1420         custodian: custodian
1421         documentID: II
1422         documentTimestamp: TS
1423         healthcareProviders: healthcareProviders
1424         informationSource: informationSource
1425         language: CD
1426         languagesSpoken: Sequence(languageSpoken)
1427         personalInformation: personalInformation
1428         supports: Sequence(support)
1429         title: String
1430         version: version
1431
1432     class healthcareProvider
1433         dateRange: IVL_TS
1434         patientID: II
1435         providerEntity: providerInformation
1436         role: CD
1437
1438     class healthcareProviders
1439         careProvisionDateRange: IVL_TS
1440         healthcareProvider: Sequence(healthcareProvider)
1441
1442     class immunization
1443         administeredDate: TS
1444         medicationInformation: Sequence(medicationInformation)
1445         medicationSeriesNumber: PQ
1446         performer: organizationInformation
1447         reaction: Sequence(CD)
1448         refusalInd: String
1449         refusalReason: String
1450
1451     class indication extends CD
1452         freeTextRef: String
1453         status: statusType
1454
1455     class informationSource
1456         author: author
1457         informationSourceName: informationSourceName
1458
1459     class informationSourceName
1460         organizationName: onm
1461         personName: pnm
1462         relatedPersonName: pnm
1463
1464     class ContextClass extends Any
1465         Factory: Factory
1466         dataRange: IVL_TS

```

```

1467         advanceDirectives: Sequence(advanceDirective)
1468         allergies: Sequence(allergy)
1469         conditions: Sequence(condition)
1470         encounters: Sequence(encounter)
1471         immunizations: Sequence(immunization)
1472         informationSource: informationSource
1473         insuranceProviders: Sequence(insuranceProvider)
1474         medications: Sequence(medication)
1475         planOfCare: planOfCare
1476         pregnancies: Sequence(pregnancy)
1477         procedures: Sequence(procedure)
1478         provider: organizationInformation
1479         qualifier: qualifier
1480         results: results
1481         vitalSigns: vitalSignsOrganizer
1482
1483     class SinglePatient extends ContextClass
1484         FamilyHistory: FamilyHistory
1485         Patient: personInformation
1486
1487     class Population extends ContextClass
1488         FamilyHistory: Sequence(FamilyHistory)
1489         Patients: Sequence(SinglePatient)
1490
1491     class Template extends SinglePatient
1492         TemplateID: String
1493         RootAct: Act
1494         FocusAct: Act
1495         LocalVariable(VarName: String): Any
1496
1497     class eMeasure extends Template
1498
1499
1500     class insuranceProvider
1501         financialResponsibilityPartyType: CD
1502         guarantorInformation: Sequence(guarantorInformation)
1503         healthInsuranceType: CD
1504         healthPlanName: String
1505         memberInformation: memberInformation
1506         payer: organizationInformation
1507         resultID: II
1508         subscriberInformation: subscriberInformation
1509
1510
1511     class languageSpoken
1512         languageCode: CS
1513         modeCode: CE
1514         preferenceInd: preferenceInd
1515
1516     class medication
1517         admissionTiming: Sequence(admissionTiming)
1518         deliveryMethod: Sequence(CD)
1519         dose: PQ
1520         doseIndicator: String
1521         doseRestriction: doseRestriction
1522         fulfillmentHistory: Sequence(fulfillmentHistory)
1523         fulfillmentInstructions: String
1524         indicateMedicationStopped: IVL_TS
1525         indication: Sequence(indication)
1526         medicationInformation: Sequence(medicationInformation)
1527         orderInformation: Sequence(orderInformation)
1528         patientInstructions: String
1529         productForm: CD
1530         reaction: CD
1531         route: CD
1532         site: Sequence(CD)

```

```

1533         statusOfMedication: statusOfMedication
1534         typeOfMedication: CD
1535         vehicle: Sequence(vehicle)
1536
1537     class medicationInformation
1538         codedBrandName: CD
1539         codedProductName: CD
1540         drugManufacturer: organizationInformation
1541
1542     class memberInformation
1543         healthPlanCoverageDates: IVL_TS
1544         memberID: II
1545         patientAddress: Sequence(addr)
1546         patientDateOfBirth: TS
1547         patientName: pnm
1548         patientPhone: Sequence(tele)
1549         patientRelationshipToSubscriber: CD
1550
1551     class nullFlavorType Enum("NP", "NI", "MSK", "NA", "OTH", "NINF", "PINF",
1552 "UNK", "NASK", "TRC", "ASKU", "NAV")
1553
1554     class onm
1555         nullFlavor: nullFlavorType
1556
1557     class orderInformation
1558         fills: fills
1559         orderDateTime: IVL_TS
1560         orderExpirationDateTime: TS
1561         orderNumber: II
1562         quantityOrdered: PQ
1563
1564     class organizationInformation
1565         organizationAddress: Sequence(addr)
1566         organizationID: Sequence(II)
1567         organizationName: onm
1568         organizationTelecom: Sequence(tele)
1569
1570     class patientInformation
1571         personAddress: Sequence(addr)
1572         personID: II
1573         personInformation: Sequence(personInformation)
1574         personPhone: Sequence(tele)
1575
1576     class personalInformation
1577         patientInformation: patientInformation
1578
1579     class personInformation
1580         birthPlace: birthPlace
1581         ethnicity: CD
1582         gender: CD
1583         maritalStatus: Sequence(CD)
1584         personDateOfBirth: TS
1585         personName: Sequence(pnm)
1586         race: Sequence(CD)
1587         religiousAffiliation: CD
1588
1589     class pivlTs
1590         alignment: String
1591         institutionSpecified: Boolean
1592         nullFlavor: nullFlavorType
1593         operator: String
1594         period: PQ
1595         phase: IVL_TS
1596         value: String
1597
1598     class plannedEvent

```

```

1599     planFreeText: String
1600     planId: II
1601     plannedTime: IVL_TS
1602     planType: CD
1603
1604     class planOfCare
1605         plannedAct: plannedEvent
1606         plannedObservation: plannedEvent
1607         plannedProcedure: plannedEvent
1608
1609     class pnm
1610         family: String
1611         given: String
1612         nullFlavor: nullFlavorType
1613         prefix: String
1614         suffix: String
1615
1616
1617     class preferenceInd
1618         nullFlavor: nullFlavorType
1619         value: String
1620
1621     class pregnancy extends CD
1622
1623     class procedure
1624         bodySite: CD
1625         procedureDateTime: IVL_TS
1626         procedureFreeTextType: String
1627         procedureID: Sequence(II)
1628         procedureProvider: Sequence(organizationInformation)
1629         procedureType: CD
1630
1631     class providerInformation
1632         providerAddress: Sequence(addr)
1633         providerID: II
1634         providerName: pnm
1635         providerOrganizationName: onm
1636         providerTelecom: Sequence(tele)
1637         providerType: CD
1638
1639     class qualifier
1640         name: CD
1641         value: ANY
1642
1643     class reasonForVisit
1644         reason: CD
1645
1646     class reference
1647         referenceDocumentID: II
1648         referenceDocumentURL: String
1649
1650     class relationshipType Enum("CHILD", "SON", "HUSB")
1651
1652     class Relative extends Any
1653         ClinicalGenomicChoice: Sequence(ClinicalGenomicChoice)
1654         deceasedEstimatedAge: REAL
1655         livingEstimatedAge: REAL
1656         naturalFatherID: ST
1657         naturalMotherID: ST
1658         patientID: ST
1659         person: pnm
1660         Relationship: CD
1661
1662     class result
1663         resultDateTime: IVL_TS
1664         resultID: Sequence(II)

```

```

1665         resultInterpretation: CD
1666         resultReferenceRange: Sequence(resultReferenceRange)
1667         resultStatus: CS
1668         resultType: CD
1669         resultValue: ANY
1670
1671     class resultReferenceRange
1672
1673     class results
1674         resultsOrganizer: resultsOrganizer
1675
1676     class resultsOrganizer
1677         code: CD
1678         effectiveTime: IVL_TS
1679         result: Sequence(result)
1680
1681     class simpleCode
1682         code: String
1683         codeSystem: String
1684         codeSystemName: String
1685         displayName: String
1686         nullFlavor: nullFlavorType
1687
1688     class statusOfMedication extends CD
1689         status: statusType
1690
1691     class statusType Enum("normal", "active", "cancelled", "pending",
1692 "suspended", "terminated", "nullified", "completed")
1693
1694
1695     class subscriberInformation
1696         subscriberAddress: addr
1697         subscriberDateOfBirth: TS
1698         subscriberID: II
1699         subscriberName: pnm
1700         subscriberPhone: Sequence(tele)
1701
1702     class support
1703         contact: Sequence(contactType)
1704         date: IVL_TS
1705         guardian: contactType
1706
1707     class supportType Enum("GUARD", "AGNT")
1708
1709     class tele
1710         nullFlavor: nullFlavorType
1711         use: String
1712         useablePeriod: Sequence(useablePeriod)
1713         value: String
1714
1715     class translation
1716         code: String
1717         codeSystem: String
1718         codeSystemName: String
1719         codeSystemVersion: String
1720         displayName: String
1721         nullFlavor: nullFlavorType
1722         originalText: ED
1723         translation: String
1724         value: String
1725
1726     class treatingProvider
1727         treatingProviderID: II
1728
1729     class useablePeriod
1730         nullFlavor: nullFlavorType

```

```
1731         operator: String
1732         value: String
1733
1734     class valueType Enum("CE", "BL", "BN", "BIN", "CR", "UUID", "II", "URL",
1735 "AD", "BL", "EN", "QTY", "PQR", "RTO")
1736
1737     class vehicle extends CD
1738         name: String
1739
1740     class version
1741         number: String
1742         setID: II
1743
1744     class vitalSignsOrganizer
1745         code: CD
1746         effectiveTime: IVL_TS
1747         vitalSign: Sequence(result)
1748
1749
1750 EndPackage
```

2.5 Detailed Clinical Model Example (Informative)

1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810

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
```