



# Compositional Grammar for SNOMED CT Expressions in HL7 Version 3

External Draft for Trial Use

Date 20081223

Version 0.06



## Document Properties

<b>Filename:</b>	CompositionalGrammar 20081223.doc
<b>Title:</b>	Compositional Grammar for SNOMED CT Expressions in HL7 Version 3
<b>Creating Author:</b>	Kent Spackman, John Gutai
<b>Subject*:</b>	IHTSDO, Harmonization with HL7, Compositional Grammar

## Amendment History

Version	Date	Editor	Comments
0.01	20081017	Kent Spackman	Initial Draft
0.02	20081103	John Gutai	Added section 5 on possible future enhancements, grammar updates from David Hislop and footnote on term definition from Ed Cheetham.
0.03	20081112	John Gutai	Minor updates on grammar definition and changes to introduction.
0.04	20081126	John Gutai	Added examples, amended syntax definition and included guidance on language, in line with comments from Technical Committee, MOLF and Affiliates Forum.
0.05	20081126	John Gutai	Updated examples following initial review
0.06	20081223	John Gutai	Updated following comments received back from HL7 review.

© International Health Terminology Standards Development Organisation 2008. All rights reserved.

SNOMED CT® was originally created by the College of American Pathologists.

This document forms part of the International Release of SNOMED CT® distributed by the International Health Terminology Standards Development Organisation (IHTSDO), and is subject to the IHTSDO's SNOMED CT® Affiliate Licence. Details of the SNOMED CT® Affiliate Licence may be found at <http://www.ihtsdo.org/our-standards/licensing/>.

No part of this document may be reproduced or transmitted in any form or by any means, or stored in any kind of retrieval system, except by an Affiliate of the IHTSDO in accordance with the SNOMED CT® Affiliate Licence. Any modification of this document (including without limitation the removal or modification of this notice) is prohibited without the express written permission of the IHTSDO.

Any copy of this document that is not obtained directly from the IHTSDO [or a Member of the IHTSDO] is not controlled by the IHTSDO, and may have been modified and may be out of date. Any recipient of this document who has received it by other means is encouraged to obtain a copy directly from the IHTSDO [or a Member of the IHTSDO. Details of the Members of the IHTSDO may be found at <http://www.ihtsdo.org/members/>].



## Table of Contents

1 Introduction.....	4
1.1 Purpose of Document.....	4
1.2 HL7 Concept Descriptor (CD) data type requirement .....	5
1.3 Prior versions and status of revision.....	5
2 Compositional grammar: Normative specification .....	7
3 Informative comments .....	8
4 Examples of Grammar .....	10
5 Future Enhancements for Consideration .....	14
5.1 Addition of a Version Number to the Grammar .....	14
5.2 Relevance for use outside HL7 .....	14
5.3 Guidance on Syntactic and Semantic Validation of Expressions .....	15
5.4 General guidance on usage of grammar .....	15
5.5 Guidance on usage within HL7 V3 .....	16
5.6 Inclusion of an Expression Identifier.....	16
5.7 Guidance on the Canonical Form of an Expression.....	17
5.8 Inclusion of a mechanism to indicate whether expressions are primitive .....	17



# 1 Introduction

## 1.1 Purpose of Document

In May 2008, the HL7 Version 3 Standard “Data Types – Abstract Specification, Release 2” was released for Normative Ballot 2.

Among other things, this standard defines what can be carried in the Concept Description (CD) data type as “the plain code symbol defined by the code system, or an expression in a syntax defined by the code system which describes the concept.”

From January 2009, the R2 message format will be available for general use. Without an approved specification for SNOMED CT compositional grammar, there will be no guidance to system vendors on how SNOMED CT expressions should be transported in HL7 V3 messages. This may result in the inconsistent use of an unapproved SNOMED CT specification.

This document provides a technical specification of a grammar for composing SNOMED CT expressions that can be carried in HL7 version 3 messages, in the Concept Descriptor (CD) data type. In particular, the grammar is intended to replace the qualifier mechanism that formerly was in the HL7 Concept Descriptor data type (CD data type), and which was removed in the HL7 version 3 data types Release 2, necessitating an official syntax that HL7 specifications state must be published by the “code system”, meaning, in the case of SNOMED CT, published by IHTSDO.

The grammar definition is issued as *Draft for Trial Use*. Any comments relating to use of the grammar should be forwarded to John Gutai ([jgu@ihtsdo.org](mailto:jgu@ihtsdo.org)) for consideration in the next version. In addition, a number of proposed enhancements have been identified in Section 5 for consideration in the next version. Feedback on these and other suggestions should also be returned to IHTSDO at the above address.

An understanding of Augmented Backus-Naur Form (ABNF), and SNOMED CT concepts, terms, attributes and attribute groups are assumed in this document.



## 1.2 HL7 Concept Descriptor (CD) data type requirement

The following details are quoted from the HL7 Version 3 Standard: Data Types – Abstract Specification, Release 2, Normative Ballot 2 – May 2008 (HL7 V3 DT R2), section 4.5.1 “Code (code): ST.SIMPLE”:<sup>1</sup>

Code (code) : [ST.SIMPLE](#)

Definition: The plain code symbol defined by the code system, ***or an expression in a syntax defined by the code system which describes the concept.*** (emphasis added)

If provided, the code SHALL be an exact match to a plain code symbol or expression defined by the code System. If the code system defines a code or expression that includes whitespace, the code SHALL include the whitespace.

**An expression can only be used where the code System either defines an expression syntax, or there is a generally accepted syntax for the code System.** (emphasis added)

The syntax described herein is intended to satisfy the need for a “syntax defined by the code system” as stated above, when the “code System” is SNOMED CT.

## 1.3 Prior versions and status of revision

The SNOMED Composition Grammar is a lightweight syntax for representation of SNOMED CT expressions. An earlier version was distributed in the International Release of SNOMED CT in the file named “SNOMED\_CT\_Representational\_Forms\_20080731.pdf” and entitled “SNOMED Clinical Terms Abstract Logical Models and Representational Forms, External Draft for Comment Version 6b.” The Compositional Grammar is listed in section 3.2.1 of that document, on pages 54-57. This previous version has been used for demonstration purposes and has been proven to be both human readable and machine parsable.

The version provided in this document follows the prior version in most details, but differs from the prior version in the following ways:

- 1) The syntax of the grammar specification is now Augmented Backus-Naur Form (ABNF)<sup>2</sup> which provides a formal standards-based reference for the grammar’s structure.

---

<sup>1</sup> [http://www.hl7.org/v3ballot/html/infrastructure/datatypes\\_r2/datatypes\\_r2.htm](http://www.hl7.org/v3ballot/html/infrastructure/datatypes_r2/datatypes_r2.htm)

<sup>2</sup> ABNF as defined by Internet Standard 68, RFC 5234



- 2) The definition of <refinements> is corrected to allow one or more <attributeGroup>s without a preceding <attributeSet>.
- 3) Unnecessary whitespace designators, <ws>, were removed from several places in the grammar.
- 4) The maximum length constraint for SNOMED Clinical Terms Identifiers (SCTIDs) is added to this grammar. SCTIDs consist of sequences of digits, from a minimum of 6 to a maximum of 18 digits in length.
- 5) The hex code for carriage return (CR) was incorrectly given as '0C' in the previous version. It is corrected to '0D' in this version.
- 6) Detailed character encoding information for UTF-8 is added.
- 7) The definition of term has been amended to allow correct parsing by the APG parser generator.
- 8) <sctId> has been added, to enable concept identifiers to be defined in a single place in the grammar definition.



## 2 Compositional grammar: Normative specification

**Table 1: ABNF definition of the SNOMED CT Compositional Grammar**

```
expression = concept *("+" concept) [":" ws refinements ]
concept = ws conceptId ws ["|" ws term ws "|" ws]
conceptId = sctId
term = 1*nonwsnonpipe *( 1*SP 1*nonwsnonpipe )
refinements = ( attributeSet *attributeGroup ) / 1*attributeGroup
attributeGroup = "{" attributeSet "}" ws
attributeSet = attribute *("," attribute)
attribute = attributeName "=" attributeValue
attributeName = ws attributeNameId ws ["|" ws term ws "|" ws]
attributeValue = concept / (ws "(" expression ")" ws)
attributeNameId = sctId
sctId = digitNonZero 5*17( digit )
ws = *( SP / HTAB / CR / LF ) ; white space
SP = %x20
HTAB = %x09
CR = %x0D
LF = %x0A
digit = %x30-39
digitNonZero = %x31-39 ; digits 1 through 9, but excluding 0
nonwsnonpipe = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4
UTF8-2 = %xC2-DF UTF8-tail
UTF8-3 = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2( UTF8-tail ) /
        %xED %x80-9F UTF8-tail / %xEE-EF 2( UTF8-tail )
UTF8-4 = %xF0 %x90-BF 2( UTF8-tail ) / %xF1-F3 3( UTF8-tail ) /
        %xF4 %x80-8F 2( UTF8-tail )
UTF8-tail = %x80-BF
```



### 3 Informative comments

Table 2. BNF representation of Compositional Grammar (detail)

<code>expression = concept *( "+" concept ) [ ":" ws refinements ]</code>	
	An expression supports combinations of one or more <i>concepts</i> optionally refined by a set of refinements. The meaning of the expression is a subtype of all the <i>concepts</i> constrained by the set of refinements. Note that where there is a requirement for multiple separately qualified concepts to be present these are expressed in attribute groups within a refinement of a general concept such as "situation with explicit context".
<code>concept = ws conceptId ws [ " " ws term ws " " ws ]</code>	
	A <i>concept</i> is represented by a <i>conceptId</i> optionally followed by a term enclosed by a pair of " " characters. Whitespace before or after the <i>conceptId</i> is ignored as is any whitespace between the initial " " characters and the first non-whitespace character in the term or between the last non-whitespace character and before second " " character.
<code>conceptId = sctId</code>	
	The <i>conceptId</i> must be a valid SNOMED CT identifier for a <i>concept</i> . The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.
<code>term = 1*nonwsnonpipe *( 1*SP 1*nonwsnonpipe )</code>	
	The term must be the term from a SNOMED CT description that is associated with the <i>concept</i> identified by the preceding <i>concept</i> identifier. For example, the term could be the preferred description, or the preferred description associated with a particular translation. The term may include valid UTF-8 characters except for the pipe " " character <sup>3</sup> . The term begins with the first non-whitespace character following the starting " " character and ends with the last non-whitespace character preceding the next " " character.
<code>refinements = ( attributeSet *attributeGroup ) / 1*attributeGroup</code>	
	A refinement contains all the grouped and ungrouped attributes that refine the meaning of the containing expression. The ungrouped attributes, if any, are all listed first, followed by all the grouped attributes.
<code>attributeGroup = "{" attributeSet "}" ws</code>	
	An attribute group contains a collection of attributes that operate together as part of the refinement of the containing expression.
<code>attributeSet = attribute *( "," attribute )</code>	
	An attribute set contains one or more attribute name-value pairs expressing refinements. They are separated by commas.

<sup>3</sup> The specification for *term* should be comparable with the specification for the Concepts.FullySpecifiedName and Descriptions. Term fields in the release table structure (as described in SNOMED Clinical Terms Technical Reference Guide, July 2008, IHTSDO). The *nonpipe* constraint adds greater stringency to the Compositional Grammar specification.



<code>attribute = attributeName "=" attributeValue</code>	
	An attribute name-value pair expressing a single refinement of the containing expression.
<code>attributeName = ws attributeNameId ws [ " " ws term ws " " ws ]</code>	
	The name (or relationship type) of an attribute to which a value is applied to refine the meaning of a containing expression. The attribute name is represented by an appropriate <i>conceptId</i> optionally followed by a term enclosed by a pair of " " characters.  Whitespace before or after the <i>conceptId</i> is ignored as is any whitespace between the initial " " characters and the first non-whitespace character in the term or between the last non-whitespace character and before second " " character.
<code>attributeValue = concept / ( ws "(" expression ")" ws )</code>	
	A <i>concept</i> or expression representing the value of a named attribute which refines the meaning of a containing expression. If an expression is used this must be enclosed in brackets.
<code>attributeNameId = sctId</code>	
	The attribute name id must be the <i>conceptId</i> for a <i>concept</i> that is a subtype descendent of the SNOMED CT <i>concept</i> "attribute".
<code>sctId = digitNonZero 5*17( digit )</code>	
	A <i>sctId</i> is used for an attribute id or a concept id. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.
<code>ws = *( SP   HTAB   CR   LF )</code>	
	Whitespace characters (space, tab, linefeed and carriage return) are ignored everywhere in the expression except: <ul style="list-style-type: none"> <li>a. Whitespace within a <i>conceptId</i> or <i>attributeNameId</i> is an error. <ul style="list-style-type: none"> <li>o Note: whitespace before or after the last digit of a valid identifier is ignored.</li> </ul> </li> <li>b. Whitespace within a <i>term</i> is treated as a significant character of the term. <ul style="list-style-type: none"> <li>o Note whitespace before the first or after the last non-whitespace character of a term is ignored</li> </ul> </li> </ul>
<code>nonwsnonpipe = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4</code>	
	Non whitespace includes printable ASCII characters (these are also valid UTF8 characters encoded as one octet) and also includes all UTF8 characters encoded as 2- 3- or 4-octet sequences. It excludes space (which is %x20) and the pipe character " " (which is %x7C), and excludes CR, LF, HTAB and other ASCII control codes. See RFC 3629 (UTF-8, a transformation format of ISO 10646 authored by the Network Working Group).
<code>digitNonZero = %x31-39</code>	
	The first character of a <i>concept</i> identifier is constrained to a digit other than zero.
<code>digit = %x30-39</code>	
	Any digit 0 through 9



## 4 Examples of Grammar

The following examples build on each other and in complexity. They are primarily aimed at demonstrating the syntax of the expression grammar, although its meaning is also discussed in a number of places:

An expression may consist of a single concept, followed by a description associated with that concept. Which particular description to use is not mandated, but as a general rule, it may be preferable to use the preferred term in any particular dialect to achieve some level of consistency. However, such guidance is not strictly in the scope of this document, and may be given elsewhere.

```
297186008|motorcycle accident|
```

The syntax does not require a description to be associated with a particular concept, so the following is also a valid expression:

```
297186008
```

Two or more concepts may be combined to form a new concept by joining them with the “+” symbol. The resultant expression is the child of each of the concepts in the expression. The resultant expression below IS AN accident caused by a blizzard and also IS A motorcycle accident.

```
217724009|accident caused by blizzard|+297186008|motorcycle accident|
```

Although not stipulated by the syntax, note that two concepts joined in this way must be from the same top level hierarchy. The syntax does not mandate which concepts in the expression should have associated descriptions and which should not so it is valid, but not advisable, to mix and match. For example, the following syntax is valid:

```
217724009+297186008|motorcycle accident|
```

The syntax allows spaces, tabs and carriage returns in most places. For example, the following examples have identical meaning to the one above:

```
217724009 + 297186008|motorcycle accident|
```



```
217724009
+ 297186008
| motorcycle accident |
```

Using the “+” symbol is symmetrical and equivalent to starting with one of the concepts and adding an IS A refinement, with a value set to the other concept. For example, the following two expressions are equivalent to each other and to the preceding expression:

```
217724009|accident caused by blizzard|:
116680003|is a|=297186008|motorcycle accident|
```

```
297186008|motorcycle accident|:
116680003|is a|=217724009|accident caused by blizzard|
```

One or more refinements may be added to a concept to qualify it. This is done by putting the concept to be qualified before a colon and the qualifying expression after. The qualifying expression is of the form “attribute = value”. The example below describes an operation to remove an ovary using a laser.

```
83152002|oophorectomy|:
260686004|method|=257820006|laser excision – action|
```

Refinements may also be applied to a conjoined concept. For example, the following two expressions (building on a preceding example) are equivalent:

```
313056006|epiphysis of ulna|:
272741003|laterality|=7771000|left|

119189000|ulna part| + 312845000|epiphysis of upper limb|:
272741003|laterality|=7771000|left|
```

Note that there are no brackets round “119189000|ulna part| + 312845000|epiphysis of upper limb” in the above example.

Where more than one qualifying expression is required, these can be separated using a comma. The example below describes the removal of the right ovary using laser excision.

```
83152002|oophorectomy|:
260686004|method|=257820006|laser excision - action|,
363704007|procedure site|=20837000|structure of right ovary|
```

A further example, below, describes the removal of the left fallopian tube using diathermy excision:

```
120053002|Salpingectomy|:
260686004|method|=261519002|diathermy excision - action|,
363704007|procedure site|=113293009|structure of left fallopian tube|
```



Where a SNOMED CT concept *comprises* a number of other concepts or sub-expressions, it may be necessary to group qualifications applied to that concept in order to avoid ambiguity as to how they apply. An example of a SNOMED CT concept that *comprises* a number of other sub-expressions is:

```
116028008|salpingo-oophorectomy|
```

This procedure *comprises* two sub-procedures: the excision of part of all of the ovarian structure; and the excision of part or all of the fallopian tube structure. We should note at this point that there is a subtle difference between a *subsumptive* relationship and a *comprising* relationship:

A motorcycle accident caused by low visibility *is a* motorcycle accident AND *is an* accident caused by a blizzard.

A salpingo-oophorectomy *comprises* a fallopian tube excision and an oophorectomy.

This is demonstrated by the SNOMED CT normal form for salpingo-oophorectomy, as shown below:

```
71388002|procedure|:  
{260686004|method|=129304002|excision - action|,  
405813007|procedure site - Direct|=15497006|ovarian structure|}  
{260686004|method|=129304002|excision - action|,  
405813007|procedure site - Direct|=31435000|fallopian tube structure|}
```

Where it is necessary within a post-coordinated expression to unambiguously qualify individual components of a concept comprised of a number of other expressions (as in the above example), grouping may be used. The following example describes a salpingo-oophorectomy, with laser excision of right ovary and diathermy excision of left fallopian tube. Note that without the grouping, it would not be possible to tell on what structure the laser excision was used and on what structure the diathermy excision was used.

```
116028008|salpingo-oophorectomy|:  
{260686004|method|=257820006|laser excision - action|,  
363704007|procedure site|=20837000|structure of right ovary|}  
{260686004|method|=261519002|diathermy excision - action|,  
363704007|procedure site|=113293009|structure of left fallopian tube|}
```

A number of grouped qualifiers may be thus used to refine a concept. Note there is no comma between adjacent groups (as there are between adjacent expressions). Also note, the syntax does not limit the number of qualifiers in a group or the number of groups within an expression.

It is also possible to nest expressions, one inside the other. Any legal expression may be wrapped in a pair of brackets, and included in another expression in the same way as a concept would be. For



example, the following expression describes a fracture of the femur caused by a motorcycle accident in a blizzard:

```
71620000|fracture of femur|:
42752001|due to| = (217724009|accident caused by blizzard|+297186008|motorcycle accident|)
```

In the example above, note the use of “( )” brackets, to identify a nested expression, as opposed to “{ }” brackets, used elsewhere, to identify groups.

The following examples show how complex expressions may be build up from simple ones, a layer at a time. This first expression describes a left hip:

```
24136001|hip joint structure|:
272741003|laterality|=7771000|left|
```

This next uses the “left hip” expression to describe a procedure to replace it:

```
397956004|prosthetic arthroplasty of the hip|:
363704007|procedure site| = (24136001|hip joint structure|:272741003|laterality|=7771000|left|)
```

Applying a further grouped refinement to the above describes a procedure to replace a left hip by inserting a prosthesis. Note that this example mixes an ungrouped qualification and a grouped qualification. Where this is done, all ungrouped qualifications should appear before the groups. Note also that there is no comma between the last qualification and the first group.

```
397956004|prosthetic arthroplasty of the hip|:
363704007|procedure site| = (24136001|hip joint structure|:272741003|laterality|=7771000|left|)
{363699004|direct device|=304120007|total hip replacement prosthesis|,
260686004|method|=257867005|insertion - action|}
```

Finally, the above expression may be included within a contextual wrapper, to describe a procedure that has been performed on a patient to replace a left hip by inserting a prosthesis.

```
243796009|situation with explicit context|:
{363589002|associated procedure|
=(397956004|prosthetic arthroplasty of the hip|:
363704007|procedure site| = (24136001|hip joint structure|:272741003|laterality|=7771000|left|)
{363699004|direct device|=304120007|total hip replacement prosthesis|,
260686004|method|=257867005|insertion - action|}),
408730004|procedure context|=385658003|done|,
408731000|temporal context|=410512000|current or specified|,
408732007|subject relationship context|=410604004|subject of record|
}
```



## 5 Future Enhancements for Consideration

A number of additional enhancements to the grammar have been identified and are listed below. These enhancements are for consideration in a subsequent release of the grammar, and not for the initial release:

### 5.1 Addition of a Version Number to the Grammar

- It is proposed that a version number be added to the expression, identifying which version of the SNOMED compositional grammar the expression conforms to. This will facilitate backwards compatibility of implementations with previous versions of the grammar.
- Version numbering will start at v0.01 (implied for the initial draft version), working upwards through v0.02, v0.03, etc. for draft releases. The first accepted release will be v1.00.
- Where version numbering is not explicitly specified, it will be implied to be this version (v0.01).
- In order to avoid unnecessary overhead in the messaging, an expression comprising of a single concept will not be versioned, as it is anticipated that a stand-alone concept will form a valid expression within all future versions of the grammar.
- Sub-expressions within an expression will not be explicitly versioned.
- Based on the above, it is proposed that the grammar specification be amended as follows:

```
snomedExpression = concept / versionedExpression  
versionedExpression = version "::" expression  
version = "v" 1*(digit) "." 2(digit)
```

### 5.2 Relevance for use outside HL7

There is an immediate requirement for a compositional grammar to support SNOMED CT expressions used within HL7 Version 3. This draft specification is proposed to meet that requirement. However, it is recognized that this compositional grammar will have wider application outside the HL7 community. Therefore, it is expected that once comments have been received across the wider community, the



next revision of this specification will encompass the more general requirement for a SNOMED CT Compositional Grammar.

### 5.3 Guidance on Syntactic and Semantic Validation of Expressions

Syntactic validation of an expression may be performed with an ABNF parser using the grammar definition included in this specification.

Semantic validation of an expression is dependent on both the release of SNOMED that the expression was derived from and also on the context of the information record that the expression is embedded in. Although semantic validation is outside the scope of this document, it is expected that semantic validation tools may be produced commercially. Consideration will also be given to producing an open source semantic validation toolkit, based on the deliverables of the Machine Readable Concept Model (MRCM) project.

As well as identifying the version of the compositional grammar to which an expression conforms (see above), in order to support semantic validation, consideration should also be given as to whether an expression should identify the SNOMED release from which the expression was derived.

### 5.4 General guidance on usage of grammar

The next version of this document may also give further guidance on how the grammar should be used. Examples include:

- Guidance on when to include a description and which description to include.
- Further guidance on the use of the “+” operator and its meaning.
- Further guidance on the use of groups and their meaning, particularly when used to refine concepts that may be comprised of a number of separate elements.
- Guidance on how to create an expression consistently. The following examples show a number of different ways to formulate an expression for a fracture of the radius and ulna, all syntactically correct, not all of which may have the desired meaning or reduce to the same normalized form. Advice on how to consistently and semantically correctly create a single expression for such a concept may be useful:



125605004|fracture of bone|:  
{363698007|finding site| = 23416004 |bone structure of ulna|}  
{363698007|finding site| = 62413002 |bone structure of radius|} <sup>4</sup>

64572001|disease|:  
{116676008|associated morphology|=72704001|fracture|,  
363698007|finding site| =23416004|bone structure of ulna|}  
{116676008|associated morphology|=72704001|fracture|,  
363698007|finding site| =62413002|bone structure of radius|}

125605004|fracture of bone|:  
363698007|finding site|= 110535000|radius AND ulna, CS|<sup>5</sup>

64572001|disease|:  
363698007|finding site|=110535000|radius AND ulna, CS|  
{116676008|associated morphology|=72704001|fracture|,  
363698007|finding site|=272673000|bone structure|}

12676007|fracture of radius|+54556006|fracture of ulna|

## 5.5 Guidance on usage within HL7 V3

Guidance on how SNOMED CT expressions may be used in HL7 will be necessary, but falls outside the scope of future revisions of this document. It is likely that a new document will be required to cover these areas.

## 5.6 Inclusion of an Expression Identifier

In order to avoid time consuming activities and improve consistency (for example, when computing normal forms for expressions), it has been suggested in the “Translating SNOMED expressions to Normal forms” document that organizations keep a central table of expressions, with each expression being identified by a unique identifier (e.g.: a UUID).

---

<sup>4</sup> Use of the brackets in this expression made be redundant.

<sup>5</sup> Use of CS (combined site) concepts when building expressions may not be recommended.



In order to support this mechanism, it is proposed that the grammar support a mechanism to identify a centrally held expression as an alternative to an “in-record” defined expression.

It is therefore proposed that the ABNF specification be amended as follows:

```
snomedExpression = concept / versionedExpression / expressionId  
expressionId = UUID
```

## 5.7 Guidance on the Canonical Form of an Expression

In order to allow expression matching using textual comparison, the “Translating SNOMED expressions to Normal forms” document describes a method for translating a compositional grammar expression to its normal form, and from there to a canonical form.

Although the translation mechanism itself is outside the scope of this document, a reduced ABNF definition for canonical form may be both useful and within the scope of a future version of the grammar.

## 5.8 Inclusion of a mechanism to indicate whether expressions are primitive

Depending on their intended use, expressions could be regarded as either primitive or fully defined. Some mechanism to optionally indicate whether an expression is primitive or full defined may be useful, both for expressions and for sub-expressions (enclosed in brackets). However, at the present time, it is not clear whether such a mechanism would be useful, and whether it would require a textual description to be associated with the expression.

If such a mechanism were required, than a possible method of providing it may be to add a flag to the start of an expression, to indicate whether it was primitive (“P”) or full-defined (“D”), as below:

```
expression = ["P" / "D"] concept *("+" concept) [":" ws refinements ]
```