INTERNATIONAL STANDARD

ISO 13120

Second edition
2019-04

Health informatics — Syntax to represent the content of healthcare classification systems — Classification Markup Language (ClaML)

Informatique de santé — Syntaxe de représentation du contenu des systèmes de classification des soins de santé — Langage de balisage de la classification (ClaML)

Reference number
ISO 13120:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 215, *Health informatics*.

This second edition cancels and replaces the first edition (ISO 13120:2013), which has been technically revised. The main changes compared to the previous edition are as follows:

— Alignment of the Classification Markup Language (ClaML) to HTML;

— Replacement of the format 'Document Type Definition' (DTD) by an 'XML Schema Definition' (XSD);

— Provision of XSD files and further informative Annexes on the ISO Standards Maintenance Portal;

— Addition of a new Annex C on 'Different ways of modification';

— Complete editorial revision.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Healthcare classification systems are developed and distributed in a variety of informal formats, such as MS Word, with little consistency in approach between developers. Exchanging data from these systems or attempting to parse the informal text into a more formal structure, say for publishing purposes, presents many challenges because mistakes are easily made and difficult to detect. For example, the accidental deletion of a tab can transform a sibling rubric into a parent. Text files with comma-separated value fields are another mechanism widely used for storing and transferring data, but as a solution here are limited by insufficient formal structuring capabilities.

In the interests of safely exchanging and distributing the content and hierarchical structure of healthcare classification systems, this document presents a simple XML specification, ClaML, for exchange and distribution of healthcare classification systems. XML is the chosen format for this syntax as: a) XML provides the necessary structuring elements, and b) there are many readily available XML parsers in existence.

This document builds on EN 14463:2008 and ISO 13120:2013.

In this version of the syntax representing ClaML 3.0.0 there had been performed a lot of structural and content-related changes to solve known problems with ClaML 2.0.0 according to experiences out of practical use and to serve additional demands of classification developers and end users. Major changes are the replacement of the DTD (Document Type Definition) by an XSD (XML Schema Definition) and the alignment with HTML by inclusion of XHTML 1.1.

ClaML is intended to serve as the core representation from which all publication forms can be derived. It contains information of a depth sufficient to uniquely identify and describe the structure and relevant elements of healthcare classification systems. This document does not intend to prescribe to developers how healthcare classification systems should be structured, nor does it define or explain the meaning of the structuring elements. ClaML is not meant to be a direct format for viewing or printing the content of a healthcare classification system. Views and prints are to be derived from this representation by post processing.

ClaML is targeted at:

a) developers of first generation[1] healthcare classification systems to assist in the construction, maintenance and publication (both in paper and electronic formats) of their particular healthcare classification systems;

b) developers of information systems to assist in the inclusion of mechanisms for unambiguous loading of healthcare classification systems into their applications;

c) organizations responsible for updating healthcare classification systems;

d) institutions receiving updated healthcare classification systems.

Health informatics — Syntax to represent the content of healthcare classification systems — Classification Markup Language (ClaML)

1 Scope

The main purpose of ClaML is to formally represent the content and hierarchical structure of healthcare classification systems in a markup language for the safe exchange and distribution of data and structure between organizations and dissimilar software products.

The scope of healthcare classification systems covered by this document encompasses terminologies, and is constrained to traditional paper-based systems (like ICD-10) and systems built according to categorial structures and a cross thesaurus (like ICNP)[2]. ClaML is intended for representation of healthcare classification systems in which classes have textual definitions, hierarchical ordering, named hierarchical levels (such as “chapter”, “section”), inclusion and exclusion criteria, and codes. It is not intended to cover any formal representation, neither for definition or composition of concepts, nor for specification of classification rules. Systems with such formal specifications can at best be partially represented using ClaML, and are hence out of scope. Most of the notes and examples in this document relate to ICD. This is because ICD is the most common classification system in the scope of this document. As a highly complex classification system it is an inexhaustible source for examples of nearly any kind. But all these notes and examples represent also other similar classification systems, if applicable, which are usually less complex. An overview of currently known classification systems using ClaML is provided in a separate document which is electronically available (see 7.3).

This document is not intended to:

a) provide a normative syntax on how a healthcare classification system is to be constructed;

b) define link types between elements in a healthcare classification system (this is left to the developers of healthcare classification systems);

c) provide a representation for direct viewing or printing.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 639-1, *Codes for the representation of names of languages — Part 1: Alpha-2 code*

3 Terms and definitions

No terms and definitions are listed in this document.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

4 Abbreviated terms

ClaML	Classification Markup Language
DTD	Document Type Definition
HTML	HyperText Markup Language[4][5][6]
ICD	International Classification of Diseases
ICF	International Classification of Functioning, Disability and Health
ICNP	International Classification for Nursing Practice
OPS	“Operationen und Prozedurenschlüssel”, the German procedure classification
WHO	World Health Organization
XHTML	Extensible HyperText Markup Language[4][5][7]
XML	Extensible Markup Language 1.0[3]
XSD	XML Schema Definition

5 Conformance

The normative part of this document is written in the form of an XML Schema Definition (XSD). Many commercially available XML tools provide facilities to test the conformance of an XML document with an XSD. Users of ClaML are encouraged to perform such a test before distributing their healthcare classification systems in the format of this document.

6 Conventions

The font Courier New is used to denote XSD or XML content of ClaML. XML examples in this document are only a partial representation of a ClaML file. For reasons of readability parts are left out at “...”.

In descriptive parts of this document **bold** text is used to denote elements and attributes defined in the XSD. For names of elements Upper CamelCase is used (i.e. a single string, consisting of multiple words without spaces, each starting with a capital); for names of attributes lowercase is used. Italic text in quotation marks is used to denote attribute values.

NOTE Names of externally defined XHTML 1.1 elements are in lowercase.

7 Classification markup language

7.1 Basis of the syntax

The basis of the syntax is to represent the content of healthcare classification systems. The syntax defined in this document is called Classification Markup Language. It is defined here in the form of an XSD. The reference to this syntax will be headed to ClaML in this document. The version of ClaML described in this document is version 3.0.0.

7.2 HTML inclusion

In the previous ClaML 2.0.0 version internally defined elements were used for representation of textual content of a healthcare classification system. These elements originated from the DocBook standard. Experiences with this version raised the wish to align ClaML to HTML for various reasons (e.g. HTML is more common, it enables easier transformation between different formats and it contains in ClaML so far missing features for accessibility demands). Therefore for this version these elements have been removed and replaced by inclusion of externally defined equivalent content of HTML. The module-based XHTML 1.1 definition[5][7] appeared to be the most suitable candidate for integration as it allows the inclusion of the model (xhtml11-model-1.xsd) and modules (xhtml11-modules-1.xsd) schema files into the same namespace of ClaML. This enables the use of these elements in a ClaML file without a prefix. Extensions by redefinition of XHTML modules allow keeping ClaML specific functions. This XHTML 1.1 inclusion should as well cover functionality of the HTML 5 specification[6] in its basics, which is not available as XML Schema Definition file. However, this partial inclusion does not meet the official criteria for XHTML Host Language Document Type Conformance or XHTML Integration Set Conformance. ClaML may therefore not be called “XHTML Host Language Conforming” or “XHTML Integration Set Conforming”[7].

For further information on the specific use of XHTML 1.1 elements in ClaML refer to the notes on the **Label** element in 7.7.25.1 and the extended XHTML anchor element (**a**) in 7.7.26.

7.3 Electronic inserts

The following electronic inserts are located in the ISO Standards Maintenance Portal 13120 folder: http://standards.iso.org/iso/13120/ed-2/en.

In the versions of ClaML published so far (EN 11463:2008, ISO 13120:2013), the DTD was only a regular part of the text document. In this version 3.0.0 of the syntax, the XML Schema Definition as given in 7.6 is also provided as electronic insert. The ClaML3.0.0.xsd file has been uploaded to the ISO database and is electronically available.

Direct link: http://standards.iso.org/iso/13120/ed-2/en/ClaML3.0.0.xsd

Also a DTD and an XSD file of the previous version 2.0.0 have been generated to support users upgrading to the new version. In the versions of ClaML published so far (EN 11463:2008, ISO 13120:2013), the specification was only defined as DTD. So for this revision the old DTD needed to be translated into an XSD first as basis for any further changes. Changes are much more comprehensible when comparing the new ClaML 3.0.0 XSD to this file instead of the old DTD. This XSD file might as well be used as replacement for the old DTD when dealing with ClaML 2.0.0-based healthcare classification systems. The files are stored in the ed-1-en.zip file. File names: ClaML2.0.0.dtd and ClaML2.0.0.xsd.

Direct link: http://standards.iso.org/iso/13120/ed-2/en/ed-1-en.zip

7.4 Informative addenda

The following informative addenda are located in the ISO Standards Maintenance Portal 13120 folder: http://standards.iso.org/iso/13120/ed-2/en.

According to the substantial changes from the previous ClaML version 2.0.0 to this version 3.0.0 some informative documents have been generated. These documents will help to better understand the changes and facilitate the transfer of healthcare classification systems to the new version.

The “ClaML_XSD_2.0.0_TO_3.0.0” document illustrates the changes of the specification in detail and shall help users to better understand them and facilitate an update to the new version. All changes based on the ClaML 2.0.0 XSD are yellow highlighted. Insertions are displayed as red and underlined text. Deletions are displayed as blue and striked-through text. The ClaML_XSD_2.0.0_TO_3.0.0.pdf file has been uploaded to the ISO database and is electronically available.

Direct link: http://standards.iso.org/iso/13120/ed-2/en/ClaML_XSD_2.0.0_TO_3.0.0.pdf

The “Inventory of classification systems using ClaML” document gives an overview of currently known healthcare classification systems using ClaML. These might serve as example healthcare classification systems when considering representing a new healthcare classification system in ClaML and gives a better understanding of the scope of the syntax. The file is stored in the ed-1-en.zip file. File name: Inventory_of_classification_systems_using_ClaML_2016-02-15.pdf.

Direct link: http://standards.iso.org/iso/13120/ed-2/en/ed-1-en.zip

The “ClaML 2.0.0 structure” document displays the ClaML 2.0.0 structure in a diagram. The file is stored in the ed-1-en.zip file. File name: ClaML2.0.0_structure.pdf.

Direct link: http://standards.iso.org/iso/13120/ed-2/en/ed-1-en.zip

The “ClaML 3.0.0 structure” document displays the ClaML 3.0.0 structure in a diagram. The ClaML3.0.0_structure.pdf file has been uploaded to the ISO database and is electronically available.

Direct link: http://standards.iso.org/iso/13120/ed-2/en/ClaML3.0.0_structure.pdf

7.5 ClaML implementation profile

ClaML offers a wide range of possibilities for the representation of classification content. The defined XML structure partially allows tree structures of endless depth. This generates serious problems for end users when importing ClaML files. As healthcare classification systems nowadays contain a huge amount of data, so do the resulting ClaML files. Therefore it is difficult for end users to oversee what needs to be addressed in their import routines so that no important information of the files is overlooked.

Hence classification system developers are highly encouraged to provide an implementation profile with their ClaML files. Such an implementation profile should at a minimum address the following questions:

- which elements and attributes are used in the current ClaML file;
- maximum expectable depth of the tree structure regarding potentially endless branches;
- subclassificiation structure (e.g. maximum level, use of multiple modifiers);
- notes on further necessary post-processing for generation of output formats.

7.6 XML Schema Definition

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--ClaML ver 3.0.0 -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:xml="http://www.w3.org/XML/1998/namespace"
 elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>

   <xs:include schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml11-model-1.xsd"/>
   <xs:redefine schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml11-modules-1.xsd">
      <xs:attributeGroup name="xhtml.a.attlist">
         <xs:attributeGroup ref="xhtml.a.attlist"/>
         <xs:attribute name="modifier" type="xs:string" use="optional"/>
         <xs:attribute name="code" type="xs:string" use="optional"/>
         <xs:attribute name="variants" type="xs:NMTOKEN"
          use="optional"/>
      </xs:attributeGroup>
      <xs:group name="xhtml.a.content">
         <xs:choice>
            <xs:group ref="xhtml.a.content"/>
            <xs:element ref="Usage"/>
         </xs:choice>
      </xs:group>
   </xs:redefine>

   <xs:group name="rubric.simple">
      <xs:choice>
         <xs:group ref="xhtml.Anchor.class"/>
         <xs:group ref="xhtml.InlPres.class"/>
         <xs:group ref="xhtml.InlPhras.class"/>
   </xs:choice>
</xs:group>

<xs:group name="rubric.complex">
   <xs:choice>
      <xs:group ref="rubric.simple"/>
      <xs:group ref="xhtml.BlkStruct.class"/>
      <xs:element ref="Include"/>
      <xs:element ref="IncludeDescendants"/>
      <xs:element ref="Fragment"/>
      <xs:group ref="xhtml.List.class"/>
      <xs:group ref="xhtml.Table.class"/>
   </xs:choice>
</xs:group>

<xs:element name="ClaML">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="Classification" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="xs:string" use="required"/>
   </xs:complexType>
</xs:element>

<xs:element name="Classification">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="Meta" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element ref="Identifier" minOccurs="0"
          maxOccurs="unbounded"/>
         <xs:element ref="Title" minOccurs="1"
          maxOccurs="unbounded"/>
         <xs:element ref="Authors" minOccurs="0" maxOccurs="1"/>
         <xs:element ref="Variants" minOccurs="0" maxOccurs="1"/>
         <xs:element ref="ClassKinds" minOccurs="1" maxOccurs="1"/>
         <xs:element ref="UsageKinds" minOccurs="0" maxOccurs="1"/>
         <xs:element ref="RubricKinds" minOccurs="1" maxOccurs="1"/>
         <xs:element ref="Modifier" minOccurs="0"
          maxOccurs="unbounded"/>
         <xs:element ref="ModifierClass" minOccurs="0"
          maxOccurs="unbounded"/>
         <xs:element ref="Class" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute ref="xml:lang" use="required"/>
      <xs:attribute ref="xml:space" use="optional"
       default="default"/>
   </xs:complexType>

   <xs:key name="Class-code-key">
      <xs:selector xpath="./Class"/>
      <xs:field xpath="@code"/>
   </xs:key>

   <xs:keyref name="SubClass-code-keyref" refer="Class-code-key">
      <xs:selector xpath="./Class/SubClass"/>
      <xs:field xpath="@code"/>
   </xs:keyref>

   <xs:keyref name="SuperClass-code-keyref" refer="Class-code-key">
      <xs:selector xpath="./Class/SuperClass"/>
      <xs:field xpath="@code"/>
   </xs:keyref>

   <xs:keyref name="IncludeDescendants-code-keyref"
    refer="Class-code-key">
      <xs:selector xpath=".//IncludeDescendants"/>
      <xs:field xpath="@code"/>
   </xs:keyref>
```

```
<xs:key name="Modifier-code-key">
   <xs:selector xpath="./Modifier"/>
   <xs:field xpath="@code"/>
</xs:key>

<xs:keyref name="ModifierClass-modifier-keyref"
 refer="Modifier-code-key">
   <xs:selector xpath="./ModifierClass"/>
   <xs:field xpath="@modifier"/>
</xs:keyref>

<xs:keyref name="ModifiedBy-code-keyref"
 refer="Modifier-code-key">
   <xs:selector xpath=".//ModifiedBy"/>
   <xs:field xpath="@code"/>
</xs:keyref>

<xs:keyref name="ExcludeModifier-code-keyref"
 refer="Modifier-code-key">
   <xs:selector xpath=".//ExcludeModifier"/>
   <xs:field xpath="@code"/>
</xs:keyref>

<xs:key name="Variant-name-key">
   <xs:selector xpath="./Variants/Variant"/>
   <xs:field xpath="@name"/>
</xs:key>

<xs:keyref name="variants-keyref" refer="Variant-name-key">
   <xs:selector xpath=".//*"/>
   <xs:field xpath="@variants"/>
</xs:keyref>

<xs:key name="ClassKind-name-key">
   <xs:selector xpath="./ClassKinds/ClassKind"/>
   <xs:field xpath="@name"/>
</xs:key>

<xs:keyref name="Class-kind-keyref" refer="ClassKind-name-key">
   <xs:selector xpath="./Class"/>
   <xs:field xpath="@kind"/>
</xs:keyref>

<xs:keyref name="IncludeDescendants-kind-keyref"
 refer="ClassKind-name-key">
   <xs:selector xpath=".//IncludeDescendants"/>
   <xs:field xpath="@kind"/>
</xs:keyref>

<xs:key name="UsageKind-name-key">
   <xs:selector xpath="./UsageKinds/UsageKind"/>
   <xs:field xpath="@name"/>
</xs:key>

<xs:keyref name="Usage-kind-keyref" refer="UsageKind-name-key">
   <xs:selector xpath=".//Usage"/>
   <xs:field xpath="@kind"/>
</xs:keyref>

<xs:key name="RubricKind-name-key">
   <xs:selector xpath="./RubricKinds/RubricKind"/>
   <xs:field xpath="@name"/>
</xs:key>

<xs:keyref name="Rubric-kind-keyref" refer="RubricKind-name-key">
   <xs:selector xpath="./*/Rubric"/>
   <xs:field xpath="@kind"/>
</xs:keyref>

<xs:key name="Author-name-key">
   <xs:selector xpath="./Authors/Author"/>
```

```
      <xs:field xpath="@name"/>
   </xs:key>

   <xs:keyref name="History-author-keyref" refer="Author-name-key">
      <xs:selector xpath=".//History"/>
      <xs:field xpath="@author"/>
   </xs:keyref>

</xs:element>

<xs:element name="Variants">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="Variant" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>

<xs:element name="Variant">
   <xs:complexType mixed="true">
      <xs:attribute name="name" type="xs:NMTOKEN" use="required"/>
   </xs:complexType>
</xs:element>

<xs:element name="Meta">
   <xs:complexType mixed="false">
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="value" type="xs:string" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
      <xs:attribute name="kind" type="xs:NMTOKEN" use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="Identifier">
   <xs:complexType mixed="false">
      <xs:attribute name="authority" type="xs:NMTOKEN"
       use="optional"/>
      <xs:attribute name="uid" type="xs:string" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
      <xs:attribute name="date" type="xs:dateTime" use="optional"/>
      <xs:attribute name="effectivedate" type="xs:dateTime"
       use="optional"/>
      <xs:attribute name="expirationdate" type="xs:dateTime"
       use="optional"/>
      <xs:attribute name="status" type="xs:string" use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="Title">
   <xs:complexType mixed="true">
      <xs:attribute name="name" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="version" type="xs:string" use="optional"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="Authors">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="Author" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>
```

```
<xs:element name="Author">
   <xs:complexType mixed="true">
      <xs:attribute name="name" type="xs:NMTOKEN" use="required"/>
   <xs:attribute name="variants" type="xs:NMTOKEN" use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="ClassKinds">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="ClassKind" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>

<xs:element name="UsageKinds">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="UsageKind" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="RubricKinds">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="RubricKind" minOccurs="1"
          maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>

<xs:element name="ClassKind">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="Display" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="UsageKind">
   <xs:complexType mixed="false">
      <xs:attribute name="name" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="mark" type="xs:string" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="Usage">
   <xs:complexType mixed="false">
      <xs:attribute name="kind" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="RubricKind">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="Display" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
```

```
         <xs:attribute name="name" type="xs:NMTOKEN" use="required"/>
         <xs:attribute name="inherited" type="xs:boolean"
          use="optional" default="true"/>
         <xs:attribute name="variants" type="xs:NMTOKEN"
          use="optional"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="Display">
      <xs:complexType mixed="true">
         <xs:attribute ref="xml:lang" use="optional"/>
         <xs:attribute name="variants" type="xs:NMTOKEN"
          use="optional"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="Modifier">
      <xs:complexType mixed="false">
         <xs:sequence>
            <xs:element ref="Meta" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="SubClass" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="Rubric" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="History" minOccurs="0"
             maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="code" type="xs:string" use="required"/>
         <xs:attribute name="kind" type="xs:NMTOKEN" use="optional"/>
         <xs:attribute name="variants" type="xs:NMTOKEN"
          use="optional"/>
         <xs:attribute name="version" type="xs:string" use="optional"/>
         <xs:attribute name="effectivedate" type="xs:dateTime"
          use="optional"/>
         <xs:attribute name="expirationdate" type="xs:dateTime"
          use="optional"/>
         <xs:attribute name="status" type="xs:string" use="optional"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="ModifierClass">
      <xs:complexType mixed="false">
         <xs:sequence>
            <xs:element ref="Usage" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="Meta" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="SuperClass" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="SubClass" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="Rubric" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="History" minOccurs="0"
             maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="modifier" type="xs:string" use="optional"/>
         <xs:attribute name="code" type="xs:string" use="required"/>
         <xs:attribute name="kind" type="xs:NMTOKEN" use="optional"/>
         <xs:attribute name="variants" type="xs:NMTOKEN"
          use="optional"/>
         <xs:attribute name="version" type="xs:string" use="optional"/>
         <xs:attribute name="effectivedate" type="xs:dateTime"
          use="optional"/>
         <xs:attribute name="expirationdate" type="xs:dateTime"
          use="optional"/>
         <xs:attribute name="status" type="xs:string" use="optional"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="Class">
      <xs:complexType mixed="false">
```

```
         <xs:sequence>
            <xs:element ref="Usage" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="Meta" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="SuperClass" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="SubClass" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="ModifiedBy" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="ValidModifierClass" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="ExcludeModifier" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="Rubric" minOccurs="0"
             maxOccurs="unbounded"/>
            <xs:element ref="History" minOccurs="0"
             maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="code" type="xs:string" use="required"/>
         <xs:attribute name="kind" type="xs:NMTOKEN" use="required"/>
         <xs:attribute name="variants" type="xs:NMTOKEN" use="optional"/>
         <xs:attribute name="version" type="xs:string" use="optional"/>
         <xs:attribute name="effectivedate" type="xs:dateTime"
          use="optional"/>
         <xs:attribute name="expirationdate" type="xs:dateTime"
          use="optional"/>
         <xs:attribute name="status" type="xs:string" use="optional"/>
      </xs:complexType>

      <xs:unique name="ModifiedBy-position-key">
         <xs:selector xpath="./ModifiedBy"/>
         <xs:field xpath="@position"/>
      </xs:unique>

      <xs:keyref name="ValidModifierClass-position-keyref"
       refer="ModifiedBy-position-key">
         <xs:selector xpath=".//ValidModifierClass"/>
         <xs:field xpath="@position"/>
      </xs:keyref>

   </xs:element>

   <xs:element name="ModifiedBy">
      <xs:complexType mixed="false">
         <xs:sequence>
            <xs:element ref="Meta" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
         <xs:attribute name="code" type="xs:string" use="required"/>
         <xs:attribute name="position" type="xs:integer"
          use="optional"/>
         <xs:attribute name="variants" type="xs:NMTOKEN"
          use="optional"/>
         <xs:attribute name="optionalmodifier" type="xs:boolean"
          use="optional" default="false"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="ExcludeModifier">
      <xs:complexType mixed="false">
         <xs:attribute name="code" type="xs:string" use="required"/>
         <xs:attribute name="variants" type="xs:NMTOKEN"
          use="optional"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="ValidModifierClass">
      <xs:complexType mixed="false">
         <xs:sequence>
            <xs:element ref="Meta" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="ValidModifierClass" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="code" type="xs:string" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
      <xs:attribute name="position" type="xs:integer"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="Rubric">
   <xs:complexType mixed="false">
      <xs:sequence>
         <xs:element ref="Usage" minOccurs="0"
          maxOccurs="unbounded"/>
         <xs:element ref="Label" minOccurs="1"
         maxOccurs="unbounded"/>
         <xs:element ref="History" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="id" type="xs:ID" use="optional"/>
      <xs:attribute name="kind" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="Label">
   <xs:complexType mixed="true">
      <xs:group ref="rubric.complex" minOccurs="0"
       maxOccurs="unbounded"/>
      <xs:attribute ref="xml:lang" use="optional"/>
      <xs:attribute ref="xml:space" use="optional"
       default="default"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="History">
   <xs:complexType mixed="true">
      <xs:attribute name="author" type="xs:NMTOKEN" use="required"/>
      <xs:attribute name="date" type="xs:dateTime" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="SuperClass">
   <xs:complexType mixed="false">
      <xs:attribute name="code" type="xs:string" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="SubClass">
   <xs:complexType mixed="false">
      <xs:attribute name="code" type="xs:string" use="required"/>
      <xs:attribute name="variants" type="xs:NMTOKEN"
       use="optional"/>
   </xs:complexType>
</xs:element>

<xs:element name="Fragment">
   <xs:complexType mixed="true">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
         <xs:group ref="rubric.simple"/>
         <xs:element ref="Usage"/>
      </xs:choice>
      <xs:attribute name="class" type="xs:string" use="optional"/>
```

```
         <xs:attribute name="type" use="optional" default="item">
            <xs:simpleType>
               <xs:restriction base="xs:NMTOKEN">
                  <xs:enumeration value="item"/>
                  <xs:enumeration value="list"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:attribute>
      </xs:complexType>
   </xs:element>

   <xs:element name="Include">
      <xs:complexType mixed="false">
         <xs:attribute name="class" type="xs:string" use="optional"/>
         <xs:attribute name="rubric" type="xs:IDREF" use="required"/>
      </xs:complexType>
   </xs:element>

   <xs:element name="IncludeDescendants">
      <xs:complexType mixed="false">
         <xs:attribute name="code" type="xs:string" use="required"/>
         <xs:attribute name="kind" type="xs:NMTOKEN" use="required"/>
      </xs:complexType>
   </xs:element>

</xs:schema>
```

7.7 Semantic description of the Classification Markup Language

7.7.1 ClaML

7.7.1.1 General

The element **ClaML** identifies a Classification Markup Language file.

7.7.1.2 Contents

The element **ClaML** contains one or more **Classification** elements.

7.7.1.3 Required attributes

The attribute **version** shall specify the version of ClaML used in the remaining document. The value to indicate the current version shall be *"3.0.0"*.

7.7.1.4 Optional attributes

The element **ClaML** has no optional attributes.

7.7.1.5 Examples

No examples in this clause.

7.7.2 Classification

7.7.2.1 General

The element **Classification** identifies a classification in the Classification Markup Language file.

7.7.2.2 Contents

The element **Classification** shall contain

- an optional number of **Meta** elements,
- an optional number of **Identifier** elements,
- one or more **Title** elements,
- one optional **Authors** element,
- one optional **Variants** element,
- one **ClassKinds** element,
- one optional **UsageKinds** element,
- one **RubricKinds** element,
- an optional number of **Modifier** elements,
- an optional number of **ModifierClass** elements,
- an optional number of **Class** elements.

7.7.2.3 Required attributes

The attribute **xml:lang**[3] defines the language of the content of the classification. The attribute value of **xml:lang** shall follow ISO 639-1 and ISO 3166-1 compliant to RFC 5646[8] (see 7.7.2.5 for examples).

The attribute **xml:lang** should be defined once in the **Classification** element and only on changes in a descendant **Display** or **Label** element to avoid redundancy.

7.7.2.4 Optional attributes

The attribute **xml:space**[3] is used to indicate that white space (spaces, carriage returns, line feeds, tabs) shall be preserved within the classification. The default value for this attribute is *"default"*, which indicates that white space may be ignored. The value *"preserve"* indicates that white space shall be preserved.

The attribute **xml:space** should be defined once at the **Classification** element and only on changes in a descendant **Label** element to avoid redundancy.

7.7.2.5 Examples

EXAMPLE

```
xml:lang="en" (English – no region assigned)
xml:lang="en-GB" (English – UK)
xml:lang="de-DE" (German – Germany)
xml:lang="de-AT" (German – Austria)
xml:lang="nl-NL" (Dutch – Netherlands)
xml:lang="de-CH" (German – Switzerland)
xml:lang="fr-CH" (French – Switzerland)
xml:lang="it-CH" (Italian – Switzerland)
```

Above are listed some examples for language tags.

7.7.3 Meta

7.7.3.1 General

The element **Meta** shall be used to define meta information about a class or the classification.

Meta elements subordinate to **Class** elements shall be inherited by all resulting classes when modifying a **Class** by **ModifierClass** (or modifying **Class**) element if not defined at the **ModifierClass** (or modifying **Class**) element itself.

Meta elements subordinate to **ModifiedBy** or **ValidModifierClass** elements shall be used to define metadata of the modified class depending on which **Class** element the **ModifierClass** (or modifying **Class**) elements are combined with.

Meta elements subordinate to **ModifiedBy** elements shall be used to define metadata to be inherited by all resulting classes when modifying a specific **Class** element by **ModifierClass** (or modifying **Class**) elements. Metadata defined in **ModifiedBy** elements shall override metadata of all associated **ModifierClass** (or modifying **Class**) elements.

Meta elements subordinate to **ValidModifierClass** elements shall be used to define metadata to be inherited by the resulting specific class when modifying a specific **Class** element by a specific **ModifierClass** (or specific modifying **Class**) element referenced in the **code** attribute of the **ValidModifierClass** element. Metadata defined in **ValidModifierClass** elements shall override metadata of the associated **ModifierClass** (or modifying **Class**) elements.

NOTE 1 This method of metadata definition becomes necessary when it is intended that a modified class has different metadata than the superordinate **Class** element but according to the multiple usage of the referred **Modifier** element it is not possible to attach this metadata to the **ModifierClass** (or modifying **Class**) element (see 7.7.3.5 for examples).

NOTE 2 For the development of unambiguous data definitions of both metadata names and values, documents like ISO/IEC 11179-4 can be referenced.

7.7.3.2 Contents

The element **Meta** has no content.

7.7.3.3 Required attributes

The attribute **name** defines the name for the meta information.

The attribute **value** defines the content of the meta information.

7.7.3.4 Optional attributes

The attribute **variants** defines the variants in which this **Meta** element is valid. When the attribute is absent the **Meta** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in a classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **kind** may be used to differentiate between different types of metadata.

7.7.3.5 Examples

EXAMPLE 1

```
<Meta name="MortL1Code" value="1-002"/>
```

The example is taken from ICD-10. It indicates the relation to the mortality list 1.

EXAMPLE 2

```
<ModifierClass code="0" modifier="mod1">
…
</ModifierClass>

<Class code="C88.0" kind="category">
   <Meta name="colour" value="green"/>
   …
   <ModifiedBy code="mod1" position="5"/>
      <Meta name="colour" value="red"/>
   …
</Class>
```

In this example all modified classes of C88.0, for example C88.00, have the metadata *“colour”* set to *“red”* instead of *“green”*.

EXAMPLE 3

```
<ModifierClass code=".0" modifier="S18R83">
   …
</ModifierClass>

<ModifierClass code=".1" modifier="S18R83">
   …
</ModifierClass>

<Class code="R86" kind="category">
   <Meta name="AgeReject" value="9"/>
   …
   <ModifiedBy code="S18R83" position="4"/>
   <ValidModifierClass code=".0"/>
   <ValidModifierClass code=".1">
      <Meta name="AgeReject" value="K"/>
   </ValidModifierClass>
   …
</Class>
```

In this example only the **ModifierClass** element with code *“.1”* inherits the metadata *“K”* when combined with *“R86”*. If the **ModifierClass** element has its own metadata *“AgeReject”*, this content would be substituted by *“K”*.

EXAMPLE 4

```
<ModifierClass code=".1" modifier="mod1">
   <Meta name="meta1" value="nullified modifier metadata"/>
   …
</ModifierClass>

<Class code="Q66" kind="category">
   <Meta name="meta1" value="real metadata"/>
   …
   <ModifiedBy code="mod1" position="4"/>
      <Meta name="meta1" value=""/>
   …
</Class>
```

In this example the value of the **Meta** element of the **ModifierClass** element is nullified by an empty value “” of the **Meta** element subordinate to the **ModifiedBy** element.

7.7.4 Identifier

7.7.4.1 General

The element **Identifier** defines an issuing authority and the unique identifier for the classification defined by that authority.

7.7.4.2 Contents

The element **Identifier** has no content.

7.7.4.3 Required attributes

The attribute **uid** is required and defines the unique identifier for the classification.

7.7.4.4 Optional attributes

The optional attribute **authority** identifies the authority that issued the uid.

The attribute **variants** defines the variants in which this **Identifier** element is valid. When the attribute is absent the **Identifier** element is valid in all **variants**. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **date** defines the date and time of publication of the classification.

The attribute **effectivedate** defines the date and time, when the classification becomes effective.

The attribute **expirationdate** defines the date and time, when the classification expires.

The attribute **status** defines the status of the classification.

For date and time defining attributes (**date**, **effectivedate**, **expirationdate**) a string with the format "YYYY-MM-DDThh:mm:ss" conforming to the "xs:dateTime" restriction shall be used. If there is no specific time "YYYY-MM-DDT00:00:00" shall be used.

7.7.4.5 Examples

EXAMPLE 1

```
<Identifier authority="HL7" uid="2.16.840.1.113883.6.3"
date="2005-11-15T00:00:00"/>
```

The example shows a HL7 registered identifier specifying that the ClaML-file is containing ICD-10 data.

EXAMPLE 2

```
<Identifier authority="DIMDI" uid="1.2.276.0.76.5.430"
date="2015-09-25T00:00:00" effectivedate="2016-01-01T00:00:00"
expirationdate="2016-12-31T00:00:00" status="deprecated"/>
```

The example shows a DIMDI registered identifier specifying that the ClaML-file is containing ICD-10-GM 2016 data. The optional attribute **status** can be used to indicate that the classification is not any more effective as it passed its expiration date (yearly revision cycle).

7.7.5 Title

7.7.5.1 General

The element **Title** defines the title for the classification.

7.7.5.2 Contents

The content of the element **Title** is limited to text.

7.7.5.3 Required attributes

The attribute **name** defines a short name for the classification.

7.7.5.4 Optional attributes

The attribute **version** defines the version of the classification. It is recommended to use the well-known major.minor.patch version numbering scheme.

7.7.5.5 Examples

EXAMPLE

```
<Title name="ICD-10" version="10.2006.13">International Classification
of Diseases, 10th revision</Title>
<Title name="OPS" version="2010"/>
```

The first example shows a **Title** element for the ICD-10. In the second example a **Title** element is given for the OPS just holding the short name.

7.7.6 Authors

7.7.6.1 General

The element **Authors** defines the authors of the classification. This can, for example, be an organization issuing the file or a person creating a healthcare classification system.

7.7.6.2 Contents

The element **Authors** contains one or more **Author** elements.

7.7.6.3 Required attributes

The element **Authors** has no required attributes.

7.7.6.4 Optional attributes

The attribute **variants** defines the variants in which this **Authors** element is valid. When the attribute is absent the **Authors** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.6.5 Examples

See 7.7.7.5 for examples.

7.7.7 Author

7.7.7.1 General

The element **Author** defines an author of the classification. In case multiple organizations work on the classification they can all be named as authors (see example below).

7.7.7.2 Contents

The content of the element **Author** is limited to text.

7.7.7.3 Required attributes

The attribute **name** defines the name for the author and uniquely identifies the author in the classification. This attribute is referenced in the **author** attribute of the **History** element (7.7.27.3).

7.7.7.4 Optional attributes

The attribute **variants** defines the variants in which this **Author** element is valid. When the attribute is absent the **Author** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.7.5 Examples

EXAMPLE

```
<Authors>
   <Author name="who">World Health Organization</Author>
   <Author name="fic_nl">Dutch WHO-FIC</Author>
</Authors>
```

7.7.8 Variants

7.7.8.1 General

Optionally, ClaML supports multiple variants of a classification within the same classification subtree of a ClaML file. The element **Variants** lists the variants (if any) that are contained in a classfication.

NOTE The idea of variants is to represent high congruent classifications in a single data structure. This approach avoids redundant data storage and enables the use of synergies by mutual maintenance of these classifications, e.g. cross-checks, plausibility checks, shared metadata and other.

7.7.8.2 Contents

The element **Variants** contains one or more **Variant** elements.

7.7.8.3 Required attributes

The element **Variants** has no required attributes.

7.7.8.4 Optional attributes

The element **Variants** has no optional attributes.

7.7.8.5 Examples

See 7.7.9.5 for examples.

7.7.9 Variant

7.7.9.1 General

The element **Variant** uniquely identifies a variant in the classification.

7.7.9.2 Contents

The content of the element **Variant** is limited to text.

7.7.9.3 Required attributes

The attribute **name** defines the name for the variant and uniquely identifies the variant in the classification. This attribute is referenced in the **variants** attribute of various elements.

7.7.9.4 Optional attributes

The element **Variant** has no optional attributes.

7.7.9.5 Examples

EXAMPLE

```
<Variants>
   <Variant name="GM"/>
   <Variant name="WHO"/>
</Variants>
```

Above are listed two variants of ICD-10 in the example of the German diagnosis classifications. The WHO variant represents the unmodified German translation of the international ICD-10 maintained by WHO. The GM variant represents the German Modification which is derived from ICD-10 of WHO with national extensions e.g. for reimbursement purposes.

7.7.10 ClassKinds

7.7.10.1 General

The element **ClassKinds** lists the kinds of classes, which are present in the classification.

7.7.10.2 Contents

The element **ClassKinds** contains one or more **ClassKind** elements.

7.7.10.3 Required attributes

The element **ClassKinds** has no required attributes.

7.7.10.4 Optional attributes

The element **ClassKinds** has no optional attributes.

7.7.10.5 Examples

See 7.7.11.5 for examples.

7.7.11 ClassKind

7.7.11.1 General

The element **ClassKind** defines a class kind in the classification.

7.7.11.2 Contents

The element **ClassKind** contains an optional number of **Display** elements.

7.7.11.3 Required attributes

The attribute **name** defines the name for the class kind and uniquely identifies the class kind in the classification. This attribute is referenced in the **kind** attribute of the elements **Class** and **IncludeDescendants** (7.7.20.3, 7.7.32.3).

7.7.11.4 Optional attributes

The attribute **variants** defines the variants in which this **ClassKind** element is valid. When the attribute is absent the **ClassKind** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.11.5 Examples

EXAMPLE

```
<ClassKinds>
   <ClassKind name="chapter"/>
   <ClassKind name="block"/>
   <ClassKind name="category"/>
</ClassKinds>
```

In the example, the three common kinds of classes are listed. This example ClaML section applies to ICD-10 and other similar classifications.

7.7.12 UsageKinds

7.7.12.1 General

The element **UsageKinds** lists the kinds of usage of classes, which are present in the classification.

7.7.12.2 Contents

The element **UsageKinds** contains one or more **UsageKind** elements.

7.7.12.3 Required attributes

The element **UsageKinds** has no required attributes.

7.7.12.4 Optional attributes

The attribute **variants** defines the variants in which this **UsageKinds** element is valid. When the attribute is absent the **UsageKinds** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.12.5 Examples

See 7.7.14.5 for examples.

7.7.13 UsageKind

7.7.13.1 General

The element **UsageKind** defines a usage kind in the classification.

7.7.13.2 Contents

The element **UsageKind** has no content.

7.7.13.3 Required attributes

The attribute **name** defines the name for the usage kind and uniquely identifies the usage kind in the classification. This attribute is referenced in the **kind** attribute of the **Usage** element (7.7.14.3).

The attribute **mark** specifies how the code of a class with a specific usage kind shall be marked.

7.7.13.4 Optional attributes

The attribute **variants** defines the variants in which this **UsageKind** element is valid. When the attribute is absent the **UsageKind** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.13.5 Examples

See 7.7.14.5 for examples.

7.7.14 Usage

7.7.14.1 General

The element **Usage** is used to refer to a usage kind and to specify the usage of a modifier class, a class, a rubric, or a reference.

7.7.14.2 Contents

The element **Usage** has no content.

7.7.14.3 Required attributes

The attribute **kind** refers to a usage kind, e.g. “*etiology*”, “*manifestation*”, etc (for further information see Table B.2). A **kind** attribute shall only refer to an existing **name** attribute of a **UsageKind** element in the classification. The usage kinds are listed in the element **UsageKinds** (7.7.12)

7.7.14.4 Optional attributes

The attribute **variants** defines the variants in which this **Usage** element is valid. When the attribute is absent the **Usage** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.14.5 Examples

EXAMPLE

```
<UsageKinds>
   <UsageKind name="etiology" mark="†"/>
   <UsageKind name="manifestation" mark="*"/>
</UsageKinds>
```

The example shows the usage kinds that are used in ICD-10. The primary code is for the underlying disease and is marked with a dagger (†); an additional optional code for the manifestation is marked with an asterisk (*)[9].

7.7.15 RubricKinds

7.7.15.1 General

The element **RubricKinds** lists the kinds of rubrics, which are present in the classification.

7.7.15.2 Contents

The element **RubricKinds** contains one or more **RubricKind** elements.

7.7.15.3 Required attributes

The element **RubricKinds** has no required attributes.

7.7.15.4 Optional attributes

The element **RubricKinds** has no optional attributes.

7.7.15.5 Examples

See 7.7.16.5 for examples.

7.7.16 RubricKind

7.7.16.1 General

The element **RubricKind** defines a rubric kind in the classification.

7.7.16.2 Contents

The element **RubricKind** contains an optional number of **Display** elements.

7.7.16.3 Required attributes

The attribute **name** defines the name for the **RubricKind** and uniquely identifies the rubric kind in the classification. This attribute is referenced in the **kind** attribute of the **Rubric** element (7.7.24.3).

7.7.16.4 Optional attributes

The attribute **inherited** specifies if rubrics with this **RubricKind** are inherited by subclasses. The default value for this attribute is *“true”*.

The attribute **variants** defines the variants in which this **RubricKind** element is valid. When the attribute is absent the **RubricKind** element is valid in all **variants**. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.16.5 Examples

EXAMPLE

```
<RubricKinds>
   <RubricKind inherited="false" name="exclusion"/>
   <RubricKind inherited="false" name="inclusion"/>
   <RubricKind inherited="false" name="note"/>
   <RubricKind inherited="false" name="preferred"/>
   <RubricKind inherited="false" name="preferredLong"/>
</RubricKinds>
```

This section of ClaML applies to ICD-10, ICF and similar classfications in the same way. An explanation of the recommended kinds of rubrics is given in Table B.3.

7.7.17 Display

7.7.17.1 General

The element **Display** defines how a **ClassKind** or **RubricKind** element is to be displayed in a specific language.

7.7.17.2 Contents

The content of the element **Display** is limited to text.

7.7.17.3 Required attributes

The element **Display** has no required attributes.

7.7.17.4 Optional attributes

The attribute **variants** defines the variants in which this **Display** element is used. When the attribute is absent the **Display** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **xml:lang** defines the language of the content of the element. The attribute values of **xml:lang** shall follow ISO 639-1 and ISO 3166-1 compliant to RFC 5646[8] (see 7.7.2.5 for examples).

The attribute **xml:lang** should be defined once in the **Classification** element and only on changes in a descendant **Display** (or **Label**) element to avoid redundancy.

7.7.17.5 Examples

EXAMPLE

```
<Classification xml:lang="en">
…
   <ClassKinds>
      <ClassKind name="chapter">
         <Display>Chapter</Display>
      </ClassKind>
      <ClassKind name="block">
         <Display>Section</Display>
      </ClassKind>
      <ClassKind name="category">
      </ClassKind>
   </ClassKinds>

   <RubricKinds>
      <RubricKind name="inclusion">
         <Display xml:lang="de-DE">Inklusivum</Display>
      </RubricKind>
   </RubricKinds>
…
</Classification>
```

7.7.18 Modifier

7.7.18.1 General

The element **Modifier** defines a modifier in the classification.

NOTE **Class** elements can also be used for modification of other **Class** elements.

7.7.18.2 Contents

The element **Modifier** contains:

— an optional number of **Meta** elements;

— an optional number of **SubClass** elements;

— an optional number of **Rubric** elements;

— an optional number of **History** elements.

7.7.18.3 Required attributes

The attribute **code** defines the code for the modifier and uniquely identifies the modifier in the classification. The value is case sensitive.

7.7.18.4 Optional attributes

The attribute **kind** can be used to differentiate between different types of modifiers.

The attribute **variants** defines the variants in which this **Modifier** element is valid. When the attribute is absent the **Modifier** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **version** defines the version of the modifier. It is recommended to use the well-known major.minor.patch version numbering scheme.

The attribute **effectivedate** defines the date and time, when the modifier becomes effective.

The attribute **expirationdate** defines the date and time, when the modifier expires.

The attribute **status** defines the status of the modifier.

For date and time defining attributes (**date**, **effectivedate**, **expirationdate**) a string with the format "YYYY-MM-DD**T**hh:mm:ss" conforming to the "xs:dateTime" restriction shall be used. If there is no specific time "YYYY-MM-DDT00:00:00" shall be used.

7.7.18.5 Examples

EXAMPLE

```
<Modifier code="Md1" status="active">
   <SubClass code="0"/>
   <SubClass code="1"/>
   <SubClass code="2" variants="CM"/>
</Modifier>
```

In this **Modifier** element the **SubClass** element with the code 2 is valid for the variant ICD-10-CM only whereas the other **SubClass** elements are valid for all variants contained in the classification.

The optional attribute **status** is used to indicate that this is an active modifier.

7.7.19 ModifierClass

7.7.19.1 General

The element **ModifierClass** defines a modifier class in the classification.

7.7.19.2 Contents

The element **ModifierClass** contains:

- an optional number of **Usage** elements.
- an optional number of **Meta** elements;
- an optional number of **SuperClass** elements;
- an optional number of **SubClass** elements;
- an optional number of **Rubric** elements;
- an optional number of **History** elements.

NOTE If there are no **Meta** elements present, metadata are inherited from the associated **Class, ModifiedBy** or **ValidModifierClass** element.

7.7.19.3 Required attributes

The attribute **code** defines the code of the modifier class. The value is case sensitive.

7.7.19.4 Optional attributes

The attribute **modifier** references the code of the modifier the modifier class belongs to. A **modifier** attribute shall only refer to an existing **code** attribute of a **Modifier** element in the classification. The value is case sensitive.

The attribute **kind** can be used to differentiate between different types of modifier classes.

The attribute **variants** defines the variants in which this **ModifierClass** element is valid. When the attribute is absent the **ModifierClass** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **version** defines the version of the modifier classes. It is recommended to use the well-known major.minor.patch version numbering scheme.

The attribute **effectivedate** defines the date and time, when the modifier class becomes effective.

The attribute **expirationdate** defines the date and time, when the modifier class expires.

The attribute **status** defines the status of the modifier class.

For date and time defining attributes (**date**, **effectivedate**, **expirationdate**) a string with the format "YYYY-MM-DD**T**hh:mm:ss" conforming to the "xs:dateTime" restriction shall be used. If there is no specific time "YYYY-MM-DDT00:00:00" shall be used.

7.7.19.5 Examples

EXAMPLE

```
<ModifierClass modifier="Md1" code="0" status="active">
   <SuperClass code="Md1"/>
   <SubClass code="00"/>
   <SubClass code="01"/>
   <SubClass code="02"/>
</ModifierClass>
```

The attribute **modifier** is used if the **ModifierClass** element is assigned to one single **Modifier** element. If the **ModifierClass** element is used in many different **Modifier** elements (like Value Sets) then this attribute is omitted and the relation to the **Modifier** elements is expressed by **SubClass** and/or **SuperClass** elements only (see 7.7.21).

The optional attribute **status** can be used to indicate that this is an active modifier class.

7.7.20 Class

7.7.20.1 General

The element **Class** defines a class in the classification.

7.7.20.2 Contents

The element **Class** contains:

- an optional number of **Usage** elements;
- an optional number of **Meta** elements;
- an optional number of **SuperClass** elements;
- an optional number of **SubClass** elements;
- an optional number of **ModifiedBy** elements;
- an optional number of **ValidModifierClass** elements;
- an optional number of **ExcludeModifier** elements;
- an optional number of **Rubric** elements;
- an optional number of **History** elements.

7.7.20.3 Required attributes

The attribute **code** defines the code for the class and uniquely identifies the class in the classification. The value is case sensitive.

The attribute **kind** references the class kind of the class, e.g. *"chapter"*, *"block"*, *"category"*, etc (for further information see Table B.1). A **kind** attribute shall only refer to an existing **name** attribute of a **ClassKind** element in the classification. The class kinds are listed in the element **ClassKinds** (7.7.10).

7.7.20.4 Optional attributes

The attribute **variants** defines the variants in which this **Class** is valid. When the attribute is absent the **Class** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **version** defines the version of the class. It is recommended to use the well-known major.minor.patch version numbering scheme.

The attribute **effectivedate** defines the date and time, when the class becomes effective.

The attribute **expirationdate** defines the date and time, when the class expires.

The attribute **status** defines the status of the class.

For date and time defining attributes (**date**, **effectivedate**, **expirationdate**) a string with the format "YYYY-MM-DDThh:mm:ss" conforming to the "xs:dateTime" restriction shall be used. If there is no specific time "YYYY-MM-DDT00:00:00" shall be used.

7.7.20.5 Examples

EXAMPLE 1

```
<Class code="A00" kind="category">
   <SuperClass code="A00-A09"/>
   <SubClass code="A00.0"/>
   <SubClass code="A00.1"/>
   <SubClass code="A00.9"/>
   <Rubric kind="preferred">
      <Label>Cholera</Label>
   </Rubric>
</Class>
```

This **Class** element with the ICD-10 Code A00 is of the kind *"category"*. Its superclass is the block A00-A09, its subclasses are A00.0, A00.1 and A00.9. The preferred term is *"Cholera"*. The example shows a simple **Class** element which would look much the same in ICF or similar classifications.

EXAMPLE 2

```
<Class code="A90" kind="category" status="deprecated">
   <SuperClass code="A90-A99"/>
   <Rubric kind="preferred">
      <Label>Dengue fever [classical dengue]</Label>
   </Rubric>
</Class>
```

In this **Class** element the optional attribute **status** is used to indicate that this is a deprecated class. This class has been removed and replaced by classes A97.0, A97.1, A97.9 with publication of ICD-10 in 2016. Using this function the deprecated class can be kept as inactive class in the new ClaML file.

NOTE According to the Class-code-key restriction it is not possible to have identical codes even with different status in one classification subtree of a ClaML file, which would be for instance the case when a code is reused.

7.7.21 ModifiedBy

7.7.21.1 General

The element **ModifiedBy** refers to the code of a **Modifier** element, which modifies the class and its descendants. It is assumed that all subclasses of a **Modifier** element are also regarded for modification, even though not directly referred in the **Modifier** element.

In case the modifier shall not modify a descendant of the class the element **ExcludeModifier** shall be defined at that specific descendant (see 7.7.22).

Meta elements subordinate to **ModifiedBy** elements shall be used to define metadata which shall be applied to all resulting classes when modifying a specific **Class** element by **ModifierClass** (or modifying **Class**) elements. Metadata defined in **ModifiedBy** elements shall override metadata of all associated **ModifierClass** (or modifying **Class**) elements (for further information see 7.7.3.1). If this metadata shall be restricted only to one or more specific **ModifierClass** (or modifying **Class**) elements then **Meta** elements subordinate to **ValidModifierClass** elements shall be used instead (see 7.7.23).

This version of ClaML enables two possibilities to design modification of codes as described below. For additional information on this and examples refer to Annex C.

— The "classical" way of modification

  Modifier and **ModifierClass** (or modifying **Class**) elements build modifier lists which are used by many different **Class** elements. **ModifierClass** (or modifying **Class**) elements used for modification may refer only to one **Modifier** element. If **ModifierClass** elements are used the attribute **modifier** shall be used. **ValidModifierClass** elements are used to express valid combinations of a class with its modifier classes.

— The "Value Set" way of modification

  Modifier elements are used to group **ModifierClass** (or modifying **Class**) elements to "Value Sets" expressing the valid combinations of a class with its modifier classes. The use of **ValidModifierClass** elements becomes unnecessary. **ModifierClass** (or modifying **Class**) elements used for modification may be referred in multiple **Modifier** elements. If **ModifierClass** elements are used the attribute **modifier** shall not be used (this requires the use of unambiguous codes for **ModifierClass** elements though). The relation to **ModifierClass** (or modifying **Class**) elements shall be expressed by multiple **SubClass** and/or **SuperClass** elements subordinate to **Modifier** and/or **ModifierClass** (or modifying **Class**) elements instead.

NOTE **Class** elements can also be used for modification of other **Class** elements.

7.7.21.2 Contents

The element **ModifiedBy** contains an optional number of **Meta** elements.

7.7.21.3 Required attributes

The attribute **code** references the code of the modifier. A **code** attribute shall only refer to an existing **code** attribute of a **Modifier** element in the classification. The value is case sensitive.

7.7.21.4 Optional attributes

The attribute **position** specifies the position for the code of the **ModifierClass** element when the modifier is used to generate the subclasses of the modified class and shall be a numeric value. The first position shall be specified by the number '1'. The value shall be unique in a **Class** element.

NOTE 1 The **position** attribute can be used with classifications where the modifier code is attached to the primary code as suffix (e.g. ICD-10). It has not been set as a required attribute as there are classifications where a position of the attached code is not applicable (e.g. modification by combination of multiple primary codes, cluster coding in ICD-11).

NOTE 2 The **position** attribute allows applying **ModifierClass** elements to any particular position (see 7.7.21.5 for examples).

The attribute **variants** defines the variants in which this **ModifiedBy** element is valid. When the attribute is absent this **ModifiedBy** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **optionalmodifier** is used to indicate that the referenced modifier can be used optionally. Otherwise the referenced modifier is to be used mandatorily. The default value of the attribute **optionalmodifier** is *"false"*.

7.7.21.5 Examples

EXAMPLE 1 There are some cases in ICD-10 where a modifier should be applied to the 5th character position of the code but the code itself is only three characters long. According to the ICD-10 Volume 2 Instruction Manual (2016, section 2.4.5 and 2.4.6) in these cases the letter 'X' shall be used to fill the 4th character position[9].

```
   <Class code="T08" kind="category">
      …
      <ModifiedBy code="S19T08" position="5" optionalmodifier="true"/>
      …
   </Class>
```

EXAMPLE 2 If a classification is created which allows limited addition of a 5th digit, the modification is specified as follows:

```
<Class code="C88" kind="category">
…
   <SubClass code="C88.0"/>
   <SubClass code="C88.1"/>
   <ModifiedBy code="Md1" position="5"/>
</Class>
<Class code="C88.0" kind="category">
   <SuperClass code="C88"/>
</Class>
<Class code="C88.1" kind="category">
   <SuperClass code="C88"/>
   <ExcludeModifier code="Md1"/>
</Class>
```

In the example, both **Class** C88 and its descendant C88.0 are modified by **Modifier** element Md1. At the descendant **Class** element C88.1 the modifier is excluded, i.e. **Class** element C88.1 is not modified by **Modifier** element Md1.

7.7.22 ExcludeModifier

7.7.22.1 General

The element **ExcludeModifier** refers to the code of a modifier which is not to be used for this class and its descendants.

7.7.22.2 Contents

The element **ExcludeModifier** has no content.

7.7.22.3 Required attributes

The attribute **code** references the code of the excluded modifier. A **code** attribute shall only refer to an existing **code** attribute of a **Modifier** element in the classification. The value is case sensitive.

7.7.22.4 Optional attributes

The attribute **variants** defines the variants in which this **ExcludeModifier** element is valid. When the attribute is absent this **ExcludeModifier** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.22.5 Examples

See 7.7.21.5 for examples.

7.7.23 ValidModifierClass

7.7.23.1 General

The element **ValidModifierClass** refers to the code of a **ModifierClass** (or modifying **Class**) element that is valid for this class. Only valid modifier classes may be used to modify the **Class** element. If no **ValidModifierClass** element is listed, then all **ModifierClass** (or modifying **Class**) elements are valid for this **Class** element.

Multiple cascading levels of **ValidModifierClass** elements shall be used to express valid combinations of a class with its modifiers when attaching multiple modifier levels.

Meta elements subordinate to **ValidModifierClass** elements shall be used to define metadata which shall be applied to the resulting specific class when modifying a specific **Class** element by a specific **ModifierClass** (or specific modifying **Class**) element referenced in the **code** attribute of the **ValidModifierClass** element. Metadata defined in **ValidModifierClass** elements shall override metadata of the referred **ModifierClass** (or modifying **Class**) elements (for further information, see 7.7.3.1). If this metada shall be applied to all resulting classes then **Meta** elements subordinate to **ModifiedBy** elements shall be used instead (see 7.7.21).

7.7.23.2 Contents

The element **ValidModifierClass** contains:

- an optional number of **Meta** elements;
- an optional number of **ValidModifierClass** elements.

7.7.23.3 Required attributes

The attribute **code** defines the code of the valid **ModifierClass** (or modifying **Class**) element. The value is case sensitive.

7.7.23.4 Optional attributes

The attribute **variants** defines the variants in which this **ValidModifierClass** element is valid. When the attribute is absent this **ValidModifierClass** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **position** refers to the equivalent attribute of the **ModifiedBy** element. It may be used when dealing with multiple modifier levels to indicate the current level of the ValidModifierClass. It shall be a numeric value. The first position shall be specified by the number '1'. The attribute shall only refer to an existing **position** attribute of a **ModifiedBy** element subordinate to a **Class** element.

7.7.23.5 Examples

EXAMPLE 1

```
<Class code="C88" kind="digit3">
   <ModifiedBy code="Md1"/>
   <ValidModifierClass code=".0"/>
</Class>
```

C88.0 would be a valid class.

EXAMPLE 2

```
<Class code="E10" kind="category">

   <ModifiedBy code="S04E10_4" position="4"/>
   <ModifiedBy code="S04E10_5" position="5"/>

   <ValidModifierClass code=".7" position="4">
      <ValidModifierClass code="2" position="5"/>
      <ValidModifierClass code="3" position="5"/>
      <ValidModifierClass code="4" position="5"/>
      <ValidModifierClass code="5" position="5"/>
   </ValidModifierClass>

   <ValidModifierClass code=".8" position="4">
      <ValidModifierClass code="0" position="5"/>
      <ValidModifierClass code="1" position="5"/>
   </ValidModifierClass>

</Class>
```

This example illustrates cascading **ValidModifierClass** elements to express valid combinations of multiple modifier levels. The **position** attribute is used to sustain clarity about the current modifier level.

E10.72, E10.73, E10.74, E10.75, E10.80, E10.81 would be valid classes.

7.7.24 Rubric

7.7.24.1 General

The element **Rubric** defines the labels that belong to a **Class**, **Modifier**, or **ModifierClass** element.

7.7.24.2 Contents

The element **Rubric** contains:

- an optional number of **Usage** elements.
- one or more **Label** elements;
- an optional number of **History** elements.

7.7.24.3 Required attributes

The attribute **kind** refers to the rubric kind of a rubric: *"preferred"*, *"inclusion"*, *"exclusion"*, etc (for further information see Table B.3). A **kind** attribute shall only refer to an existing **name** attribute of a **RubricKind** element in the classification. The rubric kinds are listed in the element **RubricKinds** (7.7.15)

7.7.24.4 Optional attributes

The attribute **id** uniquely identifies the rubric. The first character of the attribute **id** shall be a letter, underscore or colon.

The attribute **variants** defines the variants in which this **Rubric** is valid. When the attribute is absent the **Rubric** is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.24.5 Examples

See 7.7.25.5 for examples.

7.7.25 Label

7.7.25.1 General

The element **Label** defines a piece of text.

NOTE 1 In previous ClaML versions, child elements of **Label** (for representation of textual content) were defined internally by a set of elements, which originated from the DocBook standard. For reason of HTML alignment and easier transformation between different formats, in this version, these elements have been replaced by externally referenced XHTML 1.1 inclusion.

For general information on the XHTML inclusion in ClaML see 7.2.

As these elements are defined by an externally defined schema, they will not be further explained in this document. Table 1 provides an overview of which XHTML elements replace the corresponding deprecated ClaML 2.0.0 DocBook elements. For further explanation of the use of XHTML elements and attributes refer to the XHTML 1.1 definition available online[5] or to other appropriate tutorials or literature on the use of HTML.

Table 1 — Overview of which XHTML elements replace corresponding deprecated ClaML 2.0.0 DocBook elements

ClaML 2.0.0 elements	XHTML 1.1. defined elements in ClaML 3.0.0	Notes
Reference	**a**	This element shall be used to define a reference within a rubric to another class either in the ClaML file or in some external classification.
Term	**em, strong, dfn, code, samp, kbd, var, cite, abbr, acronym, q, tt, i, b, big, small, sub, sup**	
Para	**p**	This element shall be used to define a paragraph within a rubric.
List	**ul, ol, dl**	These elements contain at least one and optionally more list item elements.
ListItem	**li**	This element contains a piece of text that shall be formatted as an item in a list.
Table	**table**	This element defines a table
Caption	**caption**	This element defines the caption of a table.
THead	**thead**	This element defines the headings of a table.
TBody	**tbody**	This element defines the body of a table.
TFoot	**tfoot**	This element defines the footer of a table.
Row	**tr**	This element defines a row within a table.
Cell	**td, th**	These elements define a cell within a row of a table.

NOTE 2 The purpose of the **Label** element is to group all text and text labelling elements subordinate to one element. The basic idea behind the apparent redundancy of the **Rubric** and **Label** elements is to offer a possibility to group different translations of the same text in one rubric (see A.2). But multiple **Label** elements subordinate to a **Rubric** element can also be used for different variants or even for text fragments of a rubric. The actual structure of **Rubric** elements in regards of **Label** elements in a ClaML file can be described in an implementation profile (see 7.5).

7.7.25.2 Contents

The element **Label** contains text and the following ClaML and XHTML elements defined by XHTML 1.1 content model and modules:

- an optional number of XHTML elements of the Anchor.class group (**a**);
- an optional number of XHTML elements of the InlPres.class or InlPhras.class groups (**em**, **strong**, **dfn**, **code**, **samp**, **kbd**, **var**, **cite**, **abbr**, **acronym**, **q**, **tt**, **i**, **b**, **big**, **small**, **sub**, **sup**);
- an optional number of XHTML elements of the BlkStruct.class group (**p**, **div**);
- an optional number of **Include** elements;
- an optional number of **IncludeDescendants** elements;
- an optional number of **Fragment** elements;
- an optional number of XHTML elements of the List.class group (**ul**, **ol**, **dl**);
- an optional number of XHTML elements of the Table.class group (**table**).

7.7.25.3 Required attributes

The element **Label** has no required attributes.

7.7.25.4 Optional attributes

The attribute **variants** defines the variants in which this **Label** is valid. When the attribute is absent the **Label** is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

The attribute **xml:lang**[3] defines the language of the content of the element. The attribute values of **xml:lang** shall follow ISO 639-1 and ISO 3166-1 compliant to RFC 5646[8] (see 7.7.2.5 for examples).

The attribute **xml:space**[3] is used to indicate that white space (spaces, carriage returns, line feeds, tabs) shall be preserved within the rubric. The default value for this attribute is *"default"*, which indicates that white space may be ignored. The value *"preserve"* indicates that white space shall be preserved.

The attributes **xml:lang** and **xml:space** should be defined once in the **Classification** element and only on changes in a descendant **Label** (or **Display**) element to avoid redundancy.

7.7.25.5 Examples

EXAMPLE

```
<Rubric id="r1234" kind="preferred">
   <Label xml:lang="en">Cholera</Label>
</Rubric>
```

7.7.26 Extended XHTML anchor element (a)

7.7.26.1 General

The XHTML element **a** is defined by XHTML 1.1 content model and modules. It is used to define a reference within a **Label** to another **Class** either in the ClaML file or in some external classification. The element is extended by ClaML specific element and attributes.

7.7.26.2 Contents

The XHTML element **a** contains text and:

- an optional number of XHTML elements of the a.content group;
- an optional number of **Usage** elements.

7.7.26.3 Required attributes

The attributes of the XHTML element **a** are defined by XHTML a.attlist attribute group.

7.7.26.4 Optional attributes

The attributes of the XHTML element **a** are defined by XHTML a.attlist attribute group. The optional attributes are extended by the following ClaML specific attributes:

The attribute **modifier** defines the code of the modifier the referenced **ModifierClass** element belongs to. A single **ModifierClass** element can be referenced by the combined use of the attributes **modifier** and **code** in the case **modifier** attributes are used (see 7.7.21). The value is case sensitive.

The attribute **code** defines the code of the referenced **Class**, **Modifier** or **ModifierClass** element. In the case this attribute is absent the contents of the XHTML element **a** define the code of the referenced **Class**, **Modifier** or **ModifierClass** element. The value is case sensitive.

The attribute **variants** defines the variants in which this XHTML **a** element is valid. When the attribute is absent the XHTML **a** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

For transformation to HTML-based output formats the ClaML specific elements and attributes of the XHTML element **a** need to be removed or replaced by suitable replacement to conform to HTML restrictions:

- If the **code** and/or **modifier** attributes are used, they need to be removed or transformed into an XHTML **href** attribute.
- If the **Usage** element is used, it needs to be removed or replaced by the text of the mark as text content.
- As different variants of the classification are assumed to be split when transformed into output formats the **variants** attribute is expected to be omitted. If the variants are not intended to be split the attribute needs to be replaced by adequate information in the output text.

7.7.26.5 Examples

EXAMPLE 1

```
<a>A00.0</a>
<a code="A00">A00-B99</a>
<a><Usage kind="etiology"/>A17</a>
```

These are simple references to codes of **Class** elements in the example of ICD-10.

EXAMPLE 2

```
<a code="S04E10_4">[See before E10 for subdivisions]</a>
```

This is a reference to the code of a **Modifier** element in the example of ICD-10.

EXAMPLE 3

```
<a modifier="ST8836"code="h">h</a>
```

This is a reference to a single **ModifierClass** element in the example of the OPS. This example does not apply to cases where no **modifier** attributes are used (see 7.7.21). In these cases unambiguous identification of the referenced **ModifierClass** element should be possible by use of the **code** attribute only.

7.7.27 History

7.7.27.1 General

The element **History** contains notes on changes to a **Modifier**, **ModifierClass**, **Class** or **Rubric** element. It is intended only for basic historical information to be conveyed with the classification. For more comprehensive historical information, e.g. an audit trail or a separate history capture mechanism is recommended. In case the classification is updated in a regular sequence and the changes to the classification done in that specific period have to be recorded and exchanged together with the classification in a ClaML file, the **History** element as described in this subclause can be used. It will basically specify when the change was done and who changed it and an explanation can be given in text.

7.7.27.2 Contents

The content of the element **History** is limited to text.

7.7.27.3 Required attributes

The attribute **author** references the unique identifier of the **Author** element. An **author** attribute shall only refer to an existing **name** attribute of an **Author** element in the classification. The authors are defined in the element **Authors** (7.7.6)

The attribute **date** gives the date of the change.

For the attribute **date** a string with the format "YYYY-MM-DD**T**hh:mm:ss" conforming to the "xs: dateTime" restriction shall be used. If there is no specific time "YYYY-MM-DDT00:00:00" shall be used.

7.7.27.4 Optional attributes

The attribute **variants** defines the variants in which this **History** element is valid. When the attribute is absent the **History** element is valid in all **variants**. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

7.7.27.5 Examples

EXAMPLE

```
<Rubric id="r123" kind="preferred">
   <Label xml:lang="en">Cholera, unspecified</Label>
   <History author="a234" date="2005-11-15T00:00:00">add unspecified</History>
</Rubric>
```

7.7.28 SuperClass

7.7.28.1 General

The element **SuperClass** defines a parent class of a **ModifierClass** or **Class** element.

7.7.28.2 Contents

The element **SuperClass** has no content.

7.7.28.3 Required attributes

The attribute **code** references the code of the parent class. A **code** attribute shall only refer to an existing **code** attribute of a **Class** element in the classification in the case of a **SuperClass** element subordinate to a **Class** element. There are no such restrictions in the case of a **SuperClass** element subordinate to a **ModiferClass** element. The value is case sensitive.

7.7.28.4 Optional attributes

The attribute **variants** defines the variants in which this **SuperClass** element is valid. When the attribute is absent the **SuperClass** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

NOTE As both **SuperClass** and **SubClass** elements are optional it is up to the classification developer whether they regard it as suitable for their classification to use both kinds of these hierarchical elements or only one kind of them to express the hierarchies between the classes.

7.7.28.5 Examples

See 7.7.19.5 for examples.

7.7.29 SubClass

7.7.29.1 General

The element **SubClass** defines a child class of a **Modifier**, **ModifierClass** or **Class** element.

7.7.29.2 Contents

The element **SubClass** has no content.

7.7.29.3 Required attributes

The attribute **code** references the code of the child class. A **code** attribute shall only refer to an existing **code** attribute of a **Class** element in the classification in the case of a **SubClass** element subordinate to a **Class** element. There are no such restrictions in the case of **SubClass** elements subordinate to **Modifier** or **ModiferClass** elements. The value is case sensitive.

7.7.29.4 Optional attributes

The attribute **variants** defines the variants in which this **SubClass** is valid. When the attribute is absent the **SubClass** element is valid in all variants. A **variants** attribute shall only refer to an existing **name** attribute of a **Variant** element in the classification. The variants are listed in the element **Variants** (7.7.8).

NOTE 1 The **SubClass** element has been introduced in addition to the **SuperClass** element. This apparent redundancy is motivated by the need to represent the order of subclasses in case of multi-hierarchies. An added advantage is that the **Class** element can describe the complete definition of a class, and be communicated individually.

NOTE 2 As both **SubClass** and **SuperClass** elements are optional it is up to the classification developer whether they regard it as suitable for their classification to use both kinds of these hierarchical elements or only one kind of them to express the hierarchies between the classes.

7.7.29.5 Examples

EXAMPLE

```
<Modifier code="ST5780">
   <SubClass code="0"/>
   <SubClass code="1"/>
…
   <SubClass code="v"/>
   <SubClass code="w"/>
   <SubClass code="z"/>
   <SubClass code="x"/>
</Modifier>
```

This example was taken from OPS. The subclasses of this modifier have to be ordered in a specific way, which is expressed in the order of the subclasses. Otherwise the non-alphabetical order of the subclasses cannot be displayed correctly and i.e. output mechanisms will not be able to arrange them in the order specified by the editor of the classification.

7.7.30 Fragment

7.7.30.1 General

The element **Fragment** defines a fragment of text within a **Label** element.

NOTE The idea of the **Fragment** element in ClaML is to provide a fully defined expression of the item lists used in ICD-10 where every item of each column has to be combined with the items of the other column(s)[1]. In order to reduce redundancy classification developers can represent this kind of list by XHTML tables. Bullet point lists can also be represented by XHTML unstructured lists (**ul** element). For further information and examples on the use of the **Fragment** element in ClaML refer to A.3.

7.7.30.2 Contents

The element **Fragment** contains text and the following ClaML and XHTML elements and their attributes defined by XHTML 1.1 content model and modules:

- an optional number of XHTML elements of the Anchor.class group (**a**);
- an optional number of XHTML elements of the InlPres.class or InlPhras.class groups (**em**, **strong**, **dfn**, **code**, **samp**, **kbd**, **var**, **cite**, **abbr**, **acronym**, **q**, **tt**, **i**, **b**, **big**, **small**, **sub**, **sup**);
- an optional number of **Usage** elements.

NOTE The child elements of the Fragment element (except the **Usage** element) are defined by XHTML 1.1. For further information on XHTML elements in CLaML see the notes on the **Label** element in 7.7.25.1.

7.7.30.3 Required attributes

The element **Fragment** has no required attributes.

7.7.30.4 Optional attributes

The attribute **class** is used to assign a class name to a **Fragment** element.

The attribute **type** defines the type of a **Fragment** element. Possible values are *"item"* and *"list"*. The default value is *"item"*.

[1] Known as "curly braces lists" in earlier ICD versions.

7.7.30.5 Examples

EXAMPLE

```
<Rubric kind="inclusion">
   <Label>
      <Fragment type="list">Tuberculosis:</Fragment>
      <Fragment type="list">disseminated</Fragment>
   </Label>
</Rubric>
<Rubric kind="inclusion">
   <Label>
      <Fragment type="list">Tuberculosis:</Fragment>
      <Fragment type="list">generalized</Fragment>
   </Label>
</Rubric>
```

displayed as:

Tuberculosis:

— disseminated

— generalized

```
<Rubric kind="inclusion">
   <Label>
      <Fragment type="item">Leukorrhoea (vaginalis)</Fragment>
      <Fragment type="item">due to Trichomonas (vaginalis)</Fragment>
   </Label>
</Rubric>
<Rubric kind="inclusion">
   <Label>
      <Fragment type="item"><Usage
       kind="etiology"/>Prostatitis</Fragment>
      <Fragment type="item">due to Trichomonas (vaginalis)</Fragment>
   </Label>
</Rubric>
```

displayed as:

Leukorrhoea (vaginalis)	due to Trichomonas (vaginalis)
Prostatitis†	

7.7.31 Include

7.7.31.1 General

The element **Include** references a **Rubric** element, which shall be included in the current rubric.

7.7.31.2 Contents

The element **Include** has no content.

7.7.31.3 Required attributes

The attribute **rubric** contains the unique identifier of the **Rubric** element that shall be included. A **rubric** attribute shall only refer to an existing **id** attribute of a **Rubric** element in the classification.

7.7.31.4 Optional attributes

The attribute **class** is used to assign a class name to an **Include** element.

7.7.31.5 Examples

EXAMPLE

```
<Class code="A00">
   <Rubric id="r123" kind="preferred">
      <Label xml:lang="en">Incision of ear</Label>
   </Rubric>
</Class>
<Class code="A00.0">
   <Rubric kind="preferred">
      <Label xml:lang="en">
         <Include rubric="r123"/>external ear
      </Label>
   </Rubric>
</Class>
```

displayed as:

A00 Incision of ear

A00.0 Incision of ear: external ear

7.7.32 IncludeDescendants

7.7.32.1 General

The element **IncludeDescendants** references a **Class** element**.** The code and preferred rubrics of the descendants of the referenced **Class** element shall be included in the rubric.

7.7.32.2 Contents

The element **IncludeDescendants** has no content.

7.7.32.3 Required attributes

The attribute **code** references the code of the class. A **code** attribute shall only refer to an existing **code** attribute of a **Class** element in the classification. The value is case sensitive.

The attribute **kind** defines the class kind of the descendants that shall be included. A **kind** attribute shall only refer to an existing **name** attribute of a **ClassKind** element in the classification. The class kinds are listed in the element **ClassKinds** (7.7.10)

7.7.32.4 Optional attributes

The element **IncludeDescendants** has no optional attributes.

7.7.32.5 Examples

EXAMPLE

```
<Class code="I" kind="chapter">
   <Rubric kind="contents">
      <Label xml:lang="en">This chapter contains the following
       blocks:<IncludeDescendants code="I" kind="block"/>
      </Label>
   </Rubric>
</Class>
```

Annex A
(informative)

Examples of usage of ClaML

A.1 Representing the dagger and asterisk system of ICD

A feature of the ICD-9 and ICD-10 is the dagger (etiology) and asterisk (manifestation) system, which is used as an alternative method to classify diagnostic statements including a general disease and a manifestation in a particular organ or site. This system can be represented in ClaML by defining a usage kind for asterisk codes and one for dagger codes.

```
<UsageKinds>
   <UsageKind name="etiology" mark="†"/>
   <UsageKind name="manifestation" mark="*"/>
</UsageKinds>
...
<Class code="A17.0" kind="digit4">
   <Usage kind="etiology"/>
   <SuperClass code="A17"/>
   <Rubric kind="preferred">
      <Label>
         Tuberculous meningitis <a>G01</a>
      </Label>
   </Rubric>
</Class>
...
<Class code="G01" kind="digit3">
   <Usage kind="manifestation"/>
   <SuperClass code="G00-G09"/>
   <Rubric kind="preferred">
      <Label>
         Meningitis in bacterial diseases classified elsewhere
      </Label>
   </Rubric>
</Class>
```

Theoretically any code in ICD-10 (except for asterisk codes) can be referenced as a dagger code. For example at G01 the code A22.8 is referenced as a dagger code, although A22.8 itself is not marked as a dagger code.

```
<Class code="A22.8" kind="digit4">
   <SuperClass code="A22"/>
   <Rubric kind="preferred">
      <Label>Other forms of anthrax</Label>
   </Rubric>
   ...
</Class>
...
<Class code="G01" kind="digit3">
   <SuperClass code="G00-G09"/>
   ...
   <Rubric kind="inclusion">
      <Label>
         Meningitis in anthrax <a>A22.8<Usage kind="etiology"/></a>
      </Label>
   </Rubric>
</Class>
```

In ICD-10 a dagger can also be added to a piece of text, as for example at A59.0 where a dagger is added to the text fragment Prostatitis. The element **Usage** subordinate to the element **Fragment** is used in such a case.

```
<Class code="A59.0" kind="digit4">
   <Rubric kind="inclusion">
      <Label>
         <Fragment type="item">Leukorrhoea (vaginalis)</Fragment>
         <Fragment type="item">due to Trichomonas (vaginalis)</Fragment>
      </Label>
   </Rubric>
   <Rubric kind="inclusion">
      <Label>
         <Fragment type="item"><Usage kind="etiology"/>
            Prostatitis(<a>N51.0<Usage
            kind="manifestation"/></a>)</Fragment>
      <Fragment type="item">due to Trichomonas (vaginalis)</Fragment>
      </Label>
   </Rubric>
</Class>
```

A.2 References to different languages

The next piece of ClaML demonstrates how the same rubric in different languages would be represented.

```
<ClaML version="3.0.0">
   <Classification xml:lang="en-GB">
      <Title name="ICD" version="10.0.0">
         International Classification of Diseases, 10th revision
      </Title>
      <ClassKinds>
         <ClassKind name="chapter">
            <Display xml:lang="en-GB">Chapter</Display>
         </ClassKind>
      </ClassKinds>
      <Class code="I" kind="chapter">
         <Rubric id="r1234567890" kind="preferred">
            <Label>
               Certain infectious and parasitic diseases
            </Label>
            <Label xml:lang="nl-NL">
               Bepaalde infectieziekten en parasitaire aandoeningen
            </Label>
            <Label xml:lang="de-DE">
               Bestimmte infektiöse und parasitäre Krankheiten
            </Label>
         </Rubric>
      </Class>
   </Classification>
</ClaML>
```

A.3 Text containing repetition and layout

In books a particular kind of layout (see below) is often used to prevent repetition of the same texts in consecutive lines. This type of layout can be represented in ClaML using the element **Fragment**. Each **Rubric** element contains every part of the text, but identical parts of the text are displayed only once in output formats.

A16.0 Tuberculosis of lung, bacteriologically and histologically negative

Tuberculous:

- — bronchiectasis
- — fibrosis of lung
- — pneumonia
- — pneumothorax

bacteriologically and histologically negative

```
<Class code="A16.0" kind="category">
   <SuperClass code="A15-A19"/>
   <Rubric kind="preferred">
      <Label>
         Tuberculosis of lung, bacteriologically and histologically
         negative
      </Label>
   </Rubric>
   <Rubric kind="text">
      <Label>
         <Fragment type="list">
            Tuberculous:
         </Fragment>
         <Fragment type="list">
            bronchiectasis
         </Fragment>
         <Fragment type="item">
            bacteriologically and histologically negative
         </Fragment>
      </Label>
   </Rubric>
   <Rubric kind="text">
      <Label>
         <Fragment type="list">
            Tuberculous:
         </Fragment>
         <Fragment type="list">
            fibrosis of lung
         </Fragment>
         <Fragment type="item">
            bacteriologically and histologically negative
         </Fragment>
      </Label>
   </Rubric>
   <Rubric kind="text">
      <Label>
         <Fragment type="list">
            Tuberculous:
         </Fragment>
         <Fragment type="list">
            pneumonia
         </Fragment>
         <Fragment type="item">
            bacteriologically and histologically negative
         </Fragment>
      </Label>
   </Rubric>
   <Rubric kind="text">
      <Label>
         <Fragment type="list">
            Tuberculous:
         </Fragment>
         <Fragment type="list">
            pneumothorax
         </Fragment>
         <Fragment type="item">
            bacteriologically and histologically negative
         </Fragment>
      </Label>
   </Rubric>
</Class>
```

Annex B
(informative)

Suggested usage of ClaML attribute values

B.1 General

In principle there is no need to standardize on attribute values in this document. Because all entities are unequivocally identified, bulk changes are a trivial though sometimes time-consuming task. This list is provided for trial purposes. It might help in more easy interchange within certain communities.

B.2 Class kind attribute values

Table B.1 — Class kind attribute values

Value	Usage
chapter	A chapter is a self-contained block of sections dealing with a specific content (e.g. in ICD-10 chapter 1 represents infectious diseases, in OPS chapter 8 represents non-surgical-therapeutic procedures).
block	A block is a self-contained block of codes dealing with a specific sub-content of a chapter (e.g. in ICD-10 section A00-A09 represents intestinal infectious diseases — a specific group of infectious diseases).
category	A category is an entity that describes a specific concept (e.g. in ICD-10 A00 represents Cholera – a specific infectious disease).

B.3 Usage kind attribute values

The basic cause or underlying disease process is assigned a code marked with a dagger (†), and its clinical manifestation another, marked with an asterisk (*), with the two used jointly. An example of this is the coding of tuberculosis of the spinal column, which is coded as A18.0† (Chapter I – Certain infectious and parasitic diseases) as the basic cause and as M49.0* (Chapter XIII – Diseases of the musculoskeletal system and connective tissue) as its clinical manifestation.

Table B.2 — Usage kind attribute values

Value	Usage
etiology	The basic cause or underlying disease process is assigned a code marked with a dagger (†).
manifestation	The clinical manifestation is assigned a code marked with an asterisk (*).

B.4 Rubric kind attribute values

Table B.3 — Rubric kind attribute values

Value	Usage
preferred	The attribute **kind**=*"preferred"* defines a specific unique term that identifies the meaning of a class.
inclusion	The attribute **kind**=*"inclusion"* shall be used for additional terms that can be used within a class.
exclusion	The attribute **kind**=*"exclusion"* shall be used for terms that are excluded from a class.
coding-hint	Coding instructions
definition	Otherwise unspecified texts added to rubrics. Should be used for a descriptive phrase for a given concept in a healthcare classification system.
note	General remark
text	e.g. a text for a modifier
title	A title for a text rubric
introduction	A long text at the beginning of a chapter.
footnote	As in the printed versions of ICD.