

DIN EN ISO 13120

ICS 35.240.80

Entwurf

Einsprüche bis 2017-11-22
Vorgesehen als Ersatz für
DIN EN ISO 13120:2013-11

**Medizinische Informatik –
Syntax zur Darstellung des Inhalts medizinischer Klassifikationssysteme –
Klassifikations-Auszeichnungssprache (ClaML) (ISO/DIS 13120:2017);
Deutsche und Englische Fassung prEN ISO 13120:2017**

Health informatics –
Syntax to represent the content of healthcare classification systems –
Classification Markup Language (ClaML) (ISO/DIS 13120:2017);
German and English version prEN ISO 13120:2017

Informatique de santé –
Syntaxe de représentation du contenu des systèmes de classification des soins de santé –
Langage de marquage de la classification (ClaML) (ISO/DIS 13120:2017);
Version allemande et anglaise prEN ISO 13120:2017

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2017-09-22 wird der Öffentlichkeit zur Prüfung und
Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs
besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an named@din.de möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an den DIN-Normenausschuss Medizin (NAMED), 10772 Berlin, Burggrafenstr. 6, 10787 Berlin.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten
Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 131 Seiten

DIN-Normenausschuss Medizin (NAMED)
DIN-Normenausschuss Informationstechnik und Anwendungen (NIA)

Nationales Vorwort

Dieses Dokument (prEN ISO 13120:2017) wurde vom Technischen Komitee ISO/TC 215 „Health informatics“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 251 „Medizinische Informatik“ erarbeitet, dessen Sekretariat von NEN (Niederlande) gehalten wird.

Das zuständige nationale Gremium ist der Arbeitsausschuss NA 063-07-03 AA „Terminologie“ im DIN-Normenausschuss Medizin (NAMed).

Um Zweifelsfälle in der Übersetzung auszuschließen, ist die englische Originalfassung beigelegt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen Text.

Für die in diesem Dokument zitierten internationalen Dokumente wird im Folgenden auf die entsprechenden deutschen Dokumente hingewiesen:

ISO 3166-1 siehe DIN EN ISO 3166-1

Änderungen

Gegenüber DIN EN ISO 13120:2013-11 wurden folgende Änderungen vorgenommen:

- a) Anpassung der Klassifikations-Auszeichnungssprache (ClAML) an HTML;
- b) Ablösung des Formats „Document Type Definition“ (DTD) durch „XML Schema Definition“ (XSD);
- c) Bereitstellung von XSD-Dateien und weiteren informativen Anhängen über die ISO-Webseite;
- d) Aufnahme eines neuen Anhangs C „Verschiedene Modifikationswege“;
- e) komplette redaktionelle Überarbeitung.

Nationaler Anhang NA (informativ)

Literaturhinweise

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*

Medizinische Informatik — Syntax zur Darstellung des Inhalts medizinischer Klassifikationssysteme — Klassifikations- Auszeichnungssprache (ClaML) (ISO/DIS 13120:2017)

*Informatique de santé — Syntaxe de représentation du contenu des systèmes de classification des soins de santé —
Langage de marquage de la classification (ClaML) (ISO/DIS 13120:2017)*

*Health informatics — Syntax to represent the content of healthcare classification systems — Classification Markup
Language (ClaML) (ISO/DIS 13120:2017)*

ICS:

Deskriptoren:

Dokument-Typ: Europäische Norm
Dokument-Untertyp:
Dokumentstufe: parallele Umfrage
Dokumentsprache: D

STD Version 2.8l

Inhalt

	Seite
Europäisches Vorwort	4
Vorwort	5
Einleitung	6
1 Anwendungsbereich	7
1.1 Hauptzwecke	7
1.2 Themenstellungen, die außerhalb des Anwendungsbereichs der Norm liegen	7
2 Normative Verweisungen	7
3 Abkürzungen	8
4 Konformität	8
5 Konventionen	8
6 Klassifikations-Auszeichnungssprache	8
6.1 Grundlage der Syntax	8
6.2 Einbindung von HTML	8
6.3 Elektronische Beilagen	9
6.4 Informativer Nachtrag	9
6.5 ClaML-Implementierungsprofil	10
6.6 XML Schema Definition	10
6.7 Semantische Beschreibung der Klassifizierungs-Auszeichnungssprache	19
6.7.1 ClaML	19
6.7.2 Klassifikation (en: Classification)	20
6.7.3 Meta	21
6.7.4 Kennung (en: Identifier)	23
6.7.5 Überschrift (en: Title)	24
6.7.6 Autoren (en: Authors)	24
6.7.7 Autor (en: Author)	25
6.7.8 Varianten (en: Variants)	25
6.7.9 Variante	26
6.7.10 Klassenarten (en: ClassKinds)	26
6.7.11 Klassenart (en: ClassKind)	27
6.7.12 Nutzungsarten (en: UsageKinds)	27
6.7.13 Nutzungsart (en: UsageKind)	27
6.7.14 Nutzung (en: Usage)	28
6.7.15 Rubrikenarten (en: RubricKinds)	28
6.7.16 Rubrikenart (en: RubricKind)	29
6.7.17 Sichtanzeige (en: Display)	29
6.7.18 Modifizierung (en: Modifier)	30
6.7.19 Modifizierungsklasse (en: ModifierClass)	32
6.7.20 Klasse	33
6.7.21 ModifiziertDurch (en: ModifiedBy)	35
6.7.22 ExcludeModifier	37
6.7.23 ValidModifierClass	37
6.7.24 Rubrik (en: Rubric)	39
6.7.25 Kennzeichnung	39
6.7.26 Erweitertes XHTML Ankerelement (a)	41

6.7.27	Historie (en: History)	43
6.7.28	SuperKlasse (en: SuperClass)	44
6.7.29	Unterklasse (en: SubClass).....	44
6.7.30	Fragment.....	45
6.7.31	Include	46
6.7.32	IncludeDescendants	47
Anhang A (informativ) Beispiele zur Anwendung dieser Norm.....		48
A.1	Darstellung des Kreuz-Stern-Systems der ICD	48
A.2	Verweisungen auf die verschiedenen Sprachen	49
A.3	Text, der Wiederholungen und Layoutinformationen beinhaltet.....	50
Anhang B (informativ) Vorgeschlagene Anwendung der ClaML-Attribute.....		52
B.1	Allgemeines	52
B.2	ClassKind-Attribute	52
B.3	UsageKind-Attribute	52
B.4	RubricKind-Attribute.....	53
Anhang C (informativ) Unterschiedliche Möglichkeiten der Modifikation		54
C.1	Allgemeines	54
C.2	Die „klassische“ Methode der Modifikation.....	54
C.2.1	Schematisches Beispiel (klassische Methode).....	55
C.2.2	XML Beispiel (klassische Methode)	56
C.2.3	Beispiel einer HTML Ausgabe (klassische Methode)	58
C.2.4	Beispiel redundanter Modifizierungslisten (klassische Methode).....	59
C.3	Die Modifikation auf Basis des „Value Set“	60
C.3.1	Schematisches Beispiel (Value Set-Methode)	61
C.3.2	XML Beispiel (Value Set-Methode).....	62
C.3.3	Beispiel einer HTML Ausgabe (Value Set-Methode).....	64
Literaturhinweise.....		65

Europäisches Vorwort

Dieses Dokument (prEN ISO 13120:2017) wurde vom Technischen Komitee ISO/TC 215 „Health informatics“ in Zusammenarbeit mit dem Technischen Komitee CEN/TC 251 „Medizinische Informatik“ erarbeitet, dessen Sekretariat von NEN gehalten wird.

Dieses Dokument ist derzeit zur parallelen Umfrage vorgelegt.

Anerkennungsnotiz

Der Text von ISO/DIS 13120:2017 wurde von CEN als prEN ISO 13120:2017 ohne irgendeine Abänderung genehmigt.

Vorwort

ISO (die Internationale Organisation für Normung) ist eine weltweite Vereinigung nationaler Normungsorganisationen (ISO-Mitgliedsorganisationen). Die Erstellung von Internationalen Normen wird üblicherweise von ISO Technischen Komitees durchgeführt. Jede Mitgliedsorganisation, die Interesse an einem Thema hat, für welches ein Technisches Komitee gegründet wurde, hat das Recht, in diesem Komitee vertreten zu sein. Internationale staatliche und nichtstaatliche Organisationen, die in engem Kontakt mit ISO stehen, nehmen ebenfalls an der Arbeit teil. ISO arbeitet bei allen elektrotechnischen Themen eng mit der Internationalen Elektrotechnischen Kommission (IEC) zusammen.

Internationale Normen werden in Übereinstimmung mit den Gestaltungsregeln der ISO/IEC-Direktiven, Teil 2 erarbeitet.

Die Hauptaufgabe der Technischen Komitees besteht in der Erarbeitung Internationaler Normen. Die von den Technischen Komitees angenommenen Norm-Entwürfe werden den Mitgliedsorganisationen zur Umfrage zur Verfügung gestellt. Für eine Veröffentlichung als Internationale Norm wird eine Zustimmung von mindestens 75 % der abstimmenden Mitgliedsländer benötigt.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. ISO ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

ISO 13120 wurde vom Technischen Komitee ISO/TC 215, *Health informatics* erarbeitet.

Einleitung

Medizinische Klassifikationssysteme werden in einer Vielzahl verschiedener, informaler Formate entwickelt und vertrieben, wie z. B. MS Word, ohne dass dabei von Seiten der Entwickler auf formale Konsistenz geachtet wird. Der Datenaustausch mit diesen Systemen oder der Versuch, den informalen Text in eine formellere Struktur zu zergliedern, z. B. zum Zweck der Veröffentlichung, stellt eine große Herausforderung dar, da sich ungewollte Fehler leicht einschleichen und schwer zu erkennen sind. Zum Beispiel das versehentliche Löschen eines Tabulators kann eine Schwester- Rubrik in eine Eltern-Rubrik verwandeln. ASCII Dateien im CSV Format stellen eine weitere, weit verbreitete Methode dar, Daten zu speichern und zu übertragen, sind jedoch als Lösung in diesem Fall aufgrund ungenügender formeller Strukturierungsmöglichkeiten nur bedingt zu gebrauchen.

Im Interesse des einfachen Austauschs und der einfachen Verbreitung des Inhalts sowie der hierarchischen Struktur von medizinischen Klassifikationssystemen stellt dieser internationale Standard eine einfache XML-basierte Auszeichnungssprache – ClaML – für den Austausch und die Verbreitung von medizinischen Klassifikationssystemen dar. XML wurde für diesen Standard gewählt, da a) XML die benötigten Strukturierungselemente bereitstellt und b) eine Reihe verschiedener XML Parser bereits verfügbar sind.

Diese Internationale Norm baut auf EN 14463:2008 und ISO EN 13120:2013 auf. In dieser Version der Norm, die ClaML 3.0 repräsentiert, wurden zahlreiche strukturelle und inhaltliche Änderungen durchgeführt, um bekannte, sich im praktischen Gebrauch zeigende Probleme in Verbindung mit ClaML 2.0 zu lösen und zusätzliche Anforderungen von Entwicklern der Klassifikationen und Endnutzern zu erfüllen. Zu den wichtigsten Änderungen gehören die Ersetzung der DTD (Document Type Definition) durch eine XSD (XML Schema Definition) und die Angleichung an HTML durch Einbindung von XHTML 1.1.

Diese Internationale Norm dient als Kerndarstellung, von der alle Veröffentlichungsarten abgeleitet werden können. Er beinhaltet ausreichend tiefe Informationen um die Struktur und wichtige Elemente von medizinischen Klassifikationssystemen eindeutig zu identifizieren und zu beschreiben. Dieser Standard versucht nicht den Entwicklern vorzuschreiben, wie medizinischen Klassifikationssysteme aufgebaut sein sollten noch definiert oder erklärt er die Bedeutung der Strukturierungselemente. Dieser Standard soll keine direkte Vorlage zum Ausdruck oder der Einsichtnahme von medizinischen Klassifikationssystemen darstellen. Einsichtnahme und Ausdrucke sollen von dieser Darstellung durch Nachbearbeitung abgeleitet werden.

Diese Internationale Norm gilt für:

- a) Entwickler der ersten und zweiten Generation(1) von medizinischen Klassifikationssystemen, welche beim Entwurf, der Überwachung und Veröffentlichung (sowohl elektronisch als auch Papier-basiert) eines bestimmten Systems beteiligt sind;
- b) Entwickler von Informationssystemen, welche an der Einbringung von Mechanismen zum eindeutigen Laden von medizinischen Klassifizierungssystemen in ihre Anwendungen beteiligt sind;
- c) Organisationen, welche für die Aktualisierung von medizinischen Klassifizierungssystemen verantwortlich sind;
- d) Einrichtungen, die aktualisierte medizinischen Klassifizierungssysteme erhalten.

1 Anwendungsbereich

1.1 Hauptzwecke

Der Hauptzweck dieser Internationalen Norm ist die formelle Darstellung des Inhalts und der hierarchischen Struktur von medizinischen Klassifikationssystemen in einer Auszeichnungssprache (en: markup language), um den sicheren Austausch und die Verteilung von Daten und Strukturen zwischen Organisationen und verschiedenen Softwareprodukten zu ermöglichen.

Der Anwendungsbereich medizinischer Klassifikationssysteme, welcher durch diese Internationale Norm abgedeckt wird, umfasst Terminologien und ist an traditionellen, auf Papier basierenden Systemen (wie z. B. ICD-10) und kompositorischen Systemen, welche nach kategorischen Strukturen und Thesaurus aufgebaut sind (wie z. B. ICNP), angelehnt. (2) Diese Internationale Norm soll medizinische Klassifikationssysteme darstellen, deren Klassen Textdefinitionen, eine als hierarchische Ebenen bezeichnete hierarchische Ordnung (wie z. B. „Kapitel“, „Abschnitt“), Einschluss- bzw. Ausschlusskriterien und Kodierungen enthalten. Er soll nicht alle formalen Darstellungen abdecken, weder zur Definition von Konzepten noch zur Festlegung von Klassifizierungsregeln. Systeme mit derartigen formalen Festlegungen können bestenfalls teilweise durch diesen Standard dargestellt werden und gehen über den Umfang des Standards hinaus. Der Leser dieses Dokuments wird feststellen, dass sich die meisten Anmerkungen und Beispiele in dieser Norm auf ICD-10 beziehen. Dies liegt daran, dass ICD-10 das gängigste Klassifikationssystem im Anwendungsbereich dieser Norm ist. Als hochkomplexes Klassifikationssystem ist es eine unerschöpfliche Quelle für Beispiele fast jeglicher Art. All diese Anmerkungen und Beispiele gelten jedoch auch für alle anderen vergleichbaren Klassifikationssysteme, sofern anwendbar, die meist weniger komplex sind. Ein Überblick über derzeit bekannte Klassifikationssysteme auf Basis von ClaML wird in einem gesonderten Dokument gegeben, das der Norm im Abschnitt „Ressourcen“ beigelegt ist.

1.2 Themenstellungen, die außerhalb des Anwendungsbereichs der Norm liegen

Dieser Standard soll nicht:

- a) eine normative Syntax hinsichtlich des Aufbaus eines medizinischen Klassifikationssystems vorschreiben;
- b) Zusammenhänge zwischen Elementen von medizinischen Klassifikationssystemen darstellen; dies obliegt den Entwicklern von medizinischen Klassifikationssystemen;
- c) eine Darstellung zur direkten Einsichtnahme oder Druck darstellen.

2 Normative Verweisungen

Die folgenden zitierten Dokumente sind für die Anwendung dieses Dokuments erforderlich. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

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 Abkürzungen

ClAML	Klassifikations-Auszeichnungssprache (en: classification markup Language)
XML	eXtensible Markup Language 1.0 (3)
DRG	Diagnosis-Related Group
DTD	Document Type Definition
IANA	Internet Assigned Numbers Authority
ICD	International Classification of Diseases
ICF	International Classification of Functioning, disability and health
ICNP	International Classification for Nursing Practice
HTML	HyperText Markup Language (4) (5) (6)
OPS	„Operationen und Prozedurenschlüssel“ (en: German procedure classification)
WHO	World Health Organization
XHTML	Extensible Hypertext Markup Language (4) (5) (7)

4 Konformität

Der normative Teil dieser Internationalen Norm ist in Form einer XML Schema Definition (XSD) geschrieben. Viele am Markt verfügbare XML Tools bieten die Möglichkeit, die Konformität eines XML Dokuments mit einer XSD zu überprüfen. Benutzern dieses Standards wird geraten, ihr medizinisches Klassifikationssystem einem derartigen Test zu unterziehen, bevor sie ihr medizinisches Klassifikationssystem im Format dieses Standards verbreiten.

5 Konventionen

Die Schriftart Arial wird verwendet, um XSD oder XML Inhalte des ClAML zu kennzeichnen.

Der Schriftschnitt **Fett** wird verwendet, um in der XSD definierte Elemente und Attribute zu kennzeichnen. Für die Kennzeichnung der Namen der Elemente wird der CamelCase verwendet (z. B. Aufreihung, keine Leerzeichen, mehrere Wörter, jedes Wort, das mit einem Großbuchstaben beginnt; für die Kennzeichnung der Namen der Attribute wird das lowercase verwendet.

ANMERKUNG Namen extern definierter XHTML 1.1 Elemente werden in Kleinbuchstaben angegeben.

6 Klassifikations-Auszeichnungssprache

6.1 Grundlage der Syntax

Die Grundlage der Syntax soll den Inhalt eines medizinischen Klassifikationssystems darstellen. Die in diesem Standard definierte Syntax heißt Klassifikations-Auszeichnungssprache. Sie wird innerhalb dieser Norm in Form einer XSD definiert. Innerhalb dieses Dokuments wird die Syntax mit ClAML abgekürzt. Die in diesem Dokument beschriebene Version von ClAML ist 3.0.0.

6.2 Einbindung von HTML

In der Vorgängerversion von ClAML 2.0.0 wurden zur Darstellung von Textinhalten des Klassifikationssystems intern definierte Elemente verwendet. Diese Elemente gingen auf den DocBook Standard zurück. Aufgrund der Erfahrungen mit dieser Version entstand der Wunsch, ClAML an HTML anzugleichen - aus verschiedenen Gründen (HTML wird häufiger eingesetzt, die Transformation von Formaten wird vereinfacht, fehlende Funktionen für Anforderungen an die Zugänglichkeit und mehr). Deshalb wurden diese Elemente ab dieser Version entfernt und durch Einbindung von extern definierten, äquivalenten HTML Inhalten ersetzt. Die modulbasierte XHTML 1.1 (5) (7) Definition schien die am besten

geeignete Integrationslösung, da sie die Einbeziehung der Schemadateien des Modells (xhtml11-model-1.xsd) und der Module (xhtml11-modules-1.xsd) in den gleichen ClaML Namensraum unterstützt. So können dieser Elemente in einer ClaML Datei ohne Präfix eingesetzt werden. Erweiterungen durch die Neudefinitionen von XHTML Modulen ermöglichen die Beibehaltung ClaML-spezifischer Funktionen. Diese XHTML 1.1 Einbindung sollte auch die HTML 5 (6) Funktionalität, die nicht als XML Schema-Datei verfügbar ist, in ihren Grundlagen abdecken. Diese partielle Einbindung erfüllt jedoch nicht die offiziellen Kriterien für die Konformität eines Dokumenttyps als XHTML Host Language oder die Konformität als XHTML Integration Set. ClaML darf deshalb weder „konform als XHTML Host Language“ noch „konform als XHTML Integration Set“ (7) genannt werden.

Weitere Informationen zur spezifischen Nutzung von XHTML 1.1 Elementen in ClaML finden Sie in den Anmerkungen zum **Label**-Element in Abschnitt 6.7.25.1 und zum erweiterten XHTML anchor-Element (**a**) in Abschnitt 6.7.26.

6.3 Elektronische Beilagen

Die folgenden elektronischen Beilagen werden auf dem Wartungsportal für ISO-Normen 13120 in diesem Ordner bereitgestellt: <http://standards.iso.org/iso/13120/ed-2/en>

In den bisher veröffentlichten Versionen von ClaML (EN 11463:2008, ISO 13120:2013) war die DTD nur ein regulärer Teil des Textdokuments. In dieser Version 3.0.0 der Norm muss die XML Schema Definition, wie in Abschnitt 6.6 angegeben, auch als eine elektronische Beilage bereitgestellt werden. Die ClaML3.0.0.xsd Datei wurde in die ISO-Datenbank hochgeladen und ist elektronisch verfügbar.

Direkte Verknüpfung: <http://standards.iso.org/iso/13120/ed-2/en/ClaML3.0.0.xsd>

Darüber hinaus wurden eine DTD- und eine XSD-Datei der Vorversion 2.0.0 erstellt, um Benutzer zu unterstützen, die auf die neue Version aktualisieren. In den bisher veröffentlichten Versionen von ClaML (EN 11463:2008, ISO 13120:2013) war die Spezifikation nur als DTD definiert. Für diese Revision musste somit die alte DTD als Basis für alle weiteren Änderungen zunächst in XSD übertragen werden. Änderungen sind viel besser begreifbar, wenn die neue ClaML 3.0 XSD mit dieser Datei anstatt mit der alten DTD verglichen wird. Diese XSD-Datei könnte genauso gut als Ersatz für die alte DTD verwendet werden, wenn es um Klassifikationen auf Basis von ClaML 2.0 geht. Die Dateien werden in der ed-1-en.zip-Datei gespeichert. Dateinamen: ClaML2.0.0.dtd und ClaML2.0.0.xsd

Direkte Verknüpfung: <http://standards.iso.org/iso/13120/ed-2/en/ed-1-en.zip>

6.4 Informativer Nachtrag

Die folgenden informativen Nachträge befinden sich im Wartungsportal für ISO-Normen im Ordner 13120: <http://standards.iso.org/iso/13120/ed-2/en>

Im Hinblick auf die wesentlichen Änderungen in dieser Version 3.0.0 gegenüber der Vorgängerversion ClaML 2.0.0 wurden mehrere informative Dokumente generiert. Diese Dokumente sollen einen besseren Einblick in die Änderungen geben und den Transfer von Klassifikationssystemen auf die neue Version vereinfachen.

Die Änderungen der Auszeichnungssprache werden zum besseren Verständnis für den Benutzer und zur Vereinfachung der Aktualisierung auf die neue Version in Dokument „ClaML_XSD_2.0.0_TO_3.0.0“ detailliert beschrieben. Alle auf der ClaML 2.0 XSD basierenden Änderungen sind gelb hervorgehoben. Einfügungen werden als rot unterstrichener Text dargestellt. Löschungen werden als blau durchgestrichener Text dargestellt. Die ClaML_XSD_2.0.0_TO_3.0.0.pdf-Datei wurde in die ISO-Datenbank hochgeladen und ist elektronisch verfügbar.

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

Das Dokument „Inventory of classification systems using ClaML“ gibt einen Überblick über die zurzeit bekannten Klassifikationen auf Basis von ClaML. Wenn eine neue Klassifikation in ClaML dargestellt werden soll, können diese Klassifikationen als Beispielklassifikationen dienen und zu einem besseren Verständnis des Anwendungsbereichs der Norm beitragen. Die Datei wird in der ed-1-en.zip-Datei gespeichert. Dateiname: Inventory_of_classification_systems_using_ClaML_2016-02-15.pdf

Direkte Verknüpfung: <http://standards.iso.org/iso/13120/ed-2/en/ed-1-en.zip>

Im Dokument „ClaML 2.0.0 structure“ wird die ClaML 2.0.0-Struktur in einem Diagramm dargestellt. Die Datei wird in der ed-1-en.zip-Datei gespeichert. Dateiname: ClaML2.0.0_structure.pdf

Direkte Verknüpfung: <http://standards.iso.org/iso/13120/ed-2/en/ed-1-en.zip>

Im Dokument „ClaML 3.0.0 structure“ wird die ClaML 3.0.0-Struktur in einem Diagramm dargestellt. Die ClaML 3.0.0 structure.pdf-Datei wurde in die ISO-Datenbank hochgeladen und ist elektronisch verfügbar.

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

6.5 ClaML-Implementierungsprofil

ClaML bietet zahlreiche Möglichkeiten, Klassifikationsinhalte darzustellen. Die definierte XML-Struktur unterstützt teilweise Baumstrukturierungen mit endloser Tiefe. Dies bereitet Endnutzern gravierende Probleme beim Importieren von ClaML-Dateien. Da Klassifikationssysteme heutzutage eine enorme Menge von Daten enthalten, ist auch das Datenvolumen der resultierenden ClaML-Datei entsprechend groß. Für Endnutzer lässt sich folglich nur schwer einschätzen, welchen Faktoren sie in ihren Importroutinen Rechnung tragen müssen, um sicherzustellen, dass keine wichtigen Informationen übersehen werden.

Entwicklern von Klassifikationssystemen wird deshalb nachdrücklich empfohlen, ihren ClaML-Dateien ein Implementierungsprofil beizufügen. Ein solches Implementierungsprofil sollte mindestens folgenden Fragen Rechnung tragen:

- Welche Elemente und Attribute werden in der aktuellen ClaML-Datei verwendet?
- Maximal zu erwartende Tiefe der Baumstruktur im Hinblick auf potenzielle Endlosverzweigungen
- Anmerkungen zum weiteren Nachbearbeitungsbedarf zur Erzeugung von Ausgabeformaten
- Struktur der Subklassifikationen (z. B. maximale Ebene, Verwendung mehrerer Modifizierungen)

6.6 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:attributeGroup>
  </xs:redefine>
</xs:schema>
```

```
<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:sequence>
  </xs:complexType>
</xs:element>
```

```
<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:complexType>
</xs:element>
```

```
<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:complexType>
</xs:element>
```

```
<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>
```

6.7 Semantische Beschreibung der Klassifizierungs-Auszeichnungssprache

6.7.1 ClaML

6.7.1.1 Allgemeines

Das Element **ClaML** identifiziert eine Datei mit Klassifizierungs-Auszeichnungssprache.

6.7.1.2 Inhalt

Das Element **ClaML** beinhaltet ein oder mehrere **Classification**-Elemente;

6.7.1.3 Obligatorische Attribute

Das Attribut **version** gibt die im Dokument benutzte ClaML-Version an. Um die aktuelle Version anzugeben, muss der Wert des Attributs „3.0.0“ sein.

6.7.1.4 Optionale Attribute

Das Element **ClaML** hat keine optionalen Attribute.

6.7.2 Klassifikation (en: Classification)

6.7.2.1 Allgemeines

Das Element **Classification** identifiziert eine Klassifikation in der Datei mit Klassifizierungsauszeichnungssprache.

6.7.2.2 Inhalt

Das Element **Classification** muss beinhalten:

- eine beliebige Anzahl des Elements **Meta**;
- eine beliebige Anzahl des Elements **Identifier**;
- ein oder mehrere **Title**-Elemente;
- ein optionales **Authors**-Element;
- ein optionales **Variants**-Element;
- ein **ClassKinds**-Element;
- ein optionales **UsageKinds**-Element;
- ein **RubricKinds**-Element;
- eine beliebige Anzahl des Elements **Modifizier**;
- eine beliebige Anzahl des Elements **ModifizierClass**;
- eine beliebige Anzahl des Elements **Class**.

6.7.2.3 Obligatorische Attribute

Das Attribut **xml:lang** (3) definiert die Sprache des Inhalts der Klassifikation. Die Attributwerte **xml:lang** müssen sich an ISO 639-1 und ISO 3166-1 in Einklang mit RFC 3066 (8) orientieren. Sprachenidentifikatoren, die bei der IANA registriert sind, ist das Prefix „I-“ oder „i-“ voranzustellen. Für privat genutzte Codes ist das Prefix „X-“ oder „x-“ zu verwenden.

Mit dem Attribut **xml:space** (3) wird darauf hingewiesen, dass Leerraum (Leerzeichen, Zeilenumschaltung, Zeilenvorschub, Tabulatoren) innerhalb der Klassifikation beizubehalten ist. Der Standardwert für dieses Attribut ist **default**, was angibt, dass white space ignoriert werden darf. Der Wert **preserve** gibt an, dass white space erhalten bleiben muss.

ANMERKUNG Die Attribute **xml:lang** und **xml:space** sollten einmal auf der Ebene des Dokumentstamms und nur bei Änderungen eines Nachkommens definiert werden, um Redundanzen zu vermeiden (siehe 6.7.17, 6.7.25.4).

BEISPIEL

```
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)
```

ANMERKUNG Einige Beispiele möglicher Sprachtags sind weiter oben aufgeführt.

6.7.2.4 Optionale Attribute

Das Element **Classification** hat keine optionalen Attribute.

6.7.3 Meta

6.7.3.1 Allgemeines

Das Element **Meta** wird benutzt, um Metainformationen über eine Klasse oder Klassifikation zu definieren.

Class-Elementen untergeordnete **Meta**-Elemente müssen beim Verändern einer **Class** durch **ModifierClass** (oder modifizierende **Class**) an alle resultierenden Klassen vererbt werden, wenn sie nicht auf der Ebene der **ModifierClass** (oder modifizierende **Class**) selbst definiert werden.

ModifiedBy- oder **ValidModifierClass**-Elementen untergeordnete **Meta**-Elemente werden benutzt, um Metadaten der modifizierten Klasse zu definieren, je nachdem, mit welchem **Class**-Element die **ModifierClass**- (oder modifizierenden **Class**)-Elemente kombiniert werden.

ModifiedBy-Elementen untergeordnete **Meta**-Elemente werden benutzt, um Metadaten zu definieren, die beim Verändern einer bestimmten **Class** durch **ModifierClass** (oder modifizierende **Class**) an alle resultierenden Klassen zu vererben sind. In **ModifiedBy**-Elementen definierte Metadaten müssen die Metadaten von allen beigefügten **ModifierClass**- (oder modifizierenden **Class**)-Elementen überschreiben.

ValidModifierClass-Elementen untergeordnete **Meta**-Elemente werden verwendet, um Metadaten zu definieren, die bei Änderung einer bestimmten **Class** durch eine bestimmte **ModifierClass** (oder bestimmte modifizierende **Class**), auf die im **code**-Attribut der **ValidModifierClass** Bezug genommen wird, an die resultierende Klasse vererbt werden müssen. In **ValidModifierClass**-Elementen definierte Metadaten müssen Metadaten der in Bezug genommenen **ModifierClass**-(oder modifizierenden **Class**)-Elemente überschreiben.

ANMERKUNG Diese Methode zur Definition von Metadaten wird erforderlich, wenn sich die Metadaten einer modifizierten Klasse von denen der untergeordneten **Class**-Elements unterscheiden, es aber gemäß der mehrfachen Verwendung des in Bezug genommenen **Modifier** nicht möglich ist, diese Metadaten mit dem **ModifierClass**-(oder modifizierende **Class**)-Element zu verknüpfen.

BEISPIEL 1

```
<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>
```

ANMERKUNG In diesem Beispiel muss für alle modifizierten Klassen von C88.0, z.B. C88.00, die „Farbe“ der Metadaten auf „rot“ anstatt „grün“ eingestellt sein.

BEISPIEL 2

```
<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>
    ...
  </ModifiedBy >
</Class>
```

ANMERKUNG In diesem Beispiel erbt nur die **ModifierClass** mit Code “.1” die „K“-Metadaten, wenn mit „R86“ kombiniert. Wenn die **ModifierClass** über eigene „AgeReject“-Metadaten verfügt, würde dieser Inhalt ignoriert und durch „K“ ersetzt.

BEISPIEL 2

```
<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>
```

ANMERKUNG In diesem Beispiel wird der **Meta**-Wert des **ModifierClass**-Elements durch einen leeren Wert "" des **ModifiedBy Meta**-Elements aufgehoben.

ANMERKUNG Für die Entwicklung eindeutiger Datendefinitionen sowohl für Metadatenamen als auch Metadatenwerte wird empfohlen, auf Normen wie ISO 11179-4 zu verweisen.

6.7.3.2 Inhalt

Das Element **Meta** hat keinen Inhalt.

6.7.3.3 Obligatorische Attribute

Das Attribut **name** legt den Namen der Metainformation fest.

Das Attribut **value** bestimmt den Inhalt der Metainformation.

6.7.3.4 Optionale Attribute

Das Attribut **variants** legt die Varianten fest, in denen das Element **Meta** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, besitzt das Element **Meta** in allen **variants** Gültigkeit. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

Das Attribut **kind** darf zur Unterscheidung zwischen unterschiedlichen Arten von Metadaten verwendet werden.

BEISPIEL

```
<Meta name="MortBCode" value="001"/>
```

ANMERKUNG Das Beispiel entstammt ICD-10. Es gibt den Bezug zur Morbiditätsliste an.

6.7.4 Kennung (en: Identifier)

6.7.4.1 Allgemeines

Das optionale Element **Identifier** darf mehrfach vorkommen. Es definiert eine herausgebende Behörde/Stelle und den eindeutigen Identifikator zur Klassifizierung, welcher von dieser Behörde/Stelle festgelegt wurde.

6.7.4.2 Inhalt

Das Element **Identifier** hat keinen Inhalt.

6.7.4.3 Obligatorische Attribute

Das Attribut **uid** ist obligatorisch und legt den eindeutigen Identifikator für die Klassifikation fest.

6.7.4.4 Optionale Attribute

Das optionale Attribut **authority** gibt die Behörde/Stelle an, welche die uid ausgegeben hat.

Das Attribut **variants** definiert die Varianten, in denen das Element **Identifier** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, besitzt das Element **Identifier** in allen **variants** Gültigkeit. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

Das **date**-Attribut definiert Datum und Uhrzeit der Veröffentlichung.

Das Attribut **effectivedate** definiert Datum und Uhrzeit der Wirksamkeit der Klassifikation.

Das Attribut **expirationdate** definiert Datum und Uhrzeit des Verfalls der Klassifikation.

Das **Status**-Attribut definiert den Status der Klassifikation.

ANMERKUNG

Für Attribute, die Datum und Uhrzeit definieren (**date**, **effectivedate**, **expirationdate**), muss ein String im Format „JJJJ-MM-TTThh:mm:ss“ verwendet werden, der die Anforderungen der „xs:dateTime“-Beschränkung erfüllt. Wenn es keine spezifische Zeit gibt, ist „YYYY-MM-DDT00:00:00“ zu verwenden.

BEISPIEL 1

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

ANMERKUNG Das Beispiel zeigt einen HL7 registrierten Identifikator, der angibt, dass die ClaML Datei ICD-10 Daten enthält.

BEISPIEL 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"/>
```

ANMERKUNG Das Beispiel zeigt einen DIMDI registrierten Identifikator, der angibt, dass die ClaML Datei ICD-10-GM 2016 Daten enthält. Das optionale **status**-Attribut darf verwendet werden, um darauf hinzuweisen, dass die Klassifikation nicht mehr gültig ist, da ihr Gültigkeitsdatum (jährlicher Revisionszyklus) überschritten wurde.

6.7.5 Überschrift (en: Title)

6.7.5.1 Allgemeines

Das Element **Title** definiert den Titel der Klassifikation.

6.7.5.2 Inhalt

Der Inhalt des Elements **Title** ist auf Text beschränkt.

6.7.5.3 Obligatorische Attribute

Das Attribut **name** definiert eine Abkürzung für die Klassifizierung.

6.7.5.4 Optionale Attribute

Das Attribut **version** legt die Version der Klassifikation fest. Es wird empfohlen, das weit verbreitete major.minor.patch Schema zur Versionsangabe zu verwenden.

BEISPIEL

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

ANMERKUNG Das erste Beispiel zeigt ein title- Element für ICD-10. Im zweiten Beispiel ist ein title- Element für die OPS mit Abkürzung angegeben.

6.7.6 Autoren (en: Authors)

6.7.6.1 Allgemeines

Das Element **Authors** definiert die Autoren der ClaML Datei. Hier kann z. B. eine Organisation oder eine Person angegeben sein, welche eine medizinische Klassifikation erstellt hat. Für den Fall, dass mehrere Organisationen an einer medizinischen Klassifizierung arbeiten, können diese alle als Autoren angegeben werden (siehe Beispiel unten).

6.7.6.2 Inhalt

Das Element **Authors** beinhaltet ein oder mehrere **Author**-Elemente.

6.7.6.3 Obligatorische Attribute

Das Element **Authors** besitzt keine obligatorischen Attribute.

6.7.6.4 Optionale Attribute

Das Attribut **variants** definiert die Varianten, in denen das Element **Authors** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, besitzt das Element **Authors** in allen **variants** Gültigkeit. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

6.7.7 Autor (en: Author)

6.7.7.1 Allgemeines

Das Element **Author** repräsentiert den Namen eines Autors der Klassifikation.

6.7.7.2 Inhalt

Der Inhalt des Elements **Author** ist auf Text beschränkt.

6.7.7.3 Obligatorische Attribute

Das Attribut **name** identifiziert den Autor eindeutig. Auf das Attribut wird im Element **History** Bezug genommen.

BEISPIEL

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

6.7.7.4 Optionale Attribute

Das Attribut **variants** definiert die Varianten, in denen das Element **Authors** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, besitzt das Element **Authors** in allen **variants** Gültigkeit. Die Varianten werden im Element **Variants** festgelegt (6.7.8).

6.7.8 Varianten (en: Variants)

6.7.8.1 Allgemeines

Optional unterstützt der Standard mehrere Varianten einer Klassifikation in der gleichen ClAML Datei. Das Element **Variants** definiert die Varianten (falls zutreffend), die in einer ClAML Datei enthalten sind.

ANMERKUNG Hinter dem Konzept der Varianten steht die Idee, Klassifikationssysteme mit hoher Kongruenz in einer einzelnen Datei darzustellen. Dieser Ansatz vermeidet eine redundante Datenspeicherung und ermöglicht die Nutzung von Synergien durch die wechselseitige Pflege dieser Klassifikationen, beispielsweise in Form von Gegenproben, Plausibilitätskontrollen, gemeinsam genutzten Metadaten und mehr.

6.7.8.2 Inhalt

Das Element **Variants** beinhaltet ein oder mehrere **Variant**-Element(e).

6.7.8.3 Obligatorische Attribute

Das Element **Variants** besitzt keine obligatorischen Attribute.

6.7.8.4 Optionale Attribute

Das Element **Variants** besitzt keine optionalen Attribute.

BEISPIEL

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

ANMERKUNG In diesem Beispiel sind in der deutschen Diagnoseklassifikationen zwei Varianten der ICD-10 aufgelistet. Die WHO-Variante ist die ungeänderte deutsche Übersetzung der internationalen ICD-10, die von der Weltgesundheitsorganisation (WHO) herausgegeben wird. Die GM-Variante ist die deutsche Modifikation (German Modification), die mit länderspezifischen Erweiterungen aus der ICD-10-WHO abgeleitet wurde, beispielsweise zur Kostenerstattung.

6.7.9 Variante

6.7.9.1 Allgemeines

Das Element **Variant** identifiziert eine Variante innerhalb einer ClaML Datei eindeutig.

6.7.9.2 Inhalt

Der Inhalt des Elements **Variant** ist auf Text beschränkt.

6.7.9.3 Obligatorische Attribute

Das Attribut **name** bestimmt die Variante im restlichen Teil der ClaML Datei eindeutig.

6.7.9.4 Optionale Attribute

Das Element **Variant** besitzt keine optionalen Attribute.

6.7.10 Klassenarten (en: ClassKinds)

6.7.10.1 Allgemeines

Das Element **ClassKinds** listet die Klassenarten, die in der Klassifikation vorhanden sind.

6.7.10.2 Inhalt

Das Element **ClassKinds** beinhaltet ein oder mehrere **ClassKind**-Elemente.

BEISPIEL

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

ANMERKUNG Im Beispiel sind die drei verbreiteten Klassenarten aufgelistet. Dieser ClaML-Abschnitt entspricht ICD10, ICF und anderen vergleichbaren Klassifikationen.

6.7.10.3 Obligatorische Attribute

Das Element **ClassKinds** besitzt keine obligatorischen Attribute.

6.7.10.4 Optionale Attribute

Das Element **ClassKinds** besitzt keine optionalen Attribute.

6.7.11 Klassenart (en: ClassKind)

6.7.11.1 Allgemeines

Das Element **ClassKind** gibt den Namen einer **Class** an.

6.7.11.2 Inhalt

Das Element **ClassKind** beinhaltet eine beliebige Anzahl des Elements **Display**.

6.7.11.3 Obligatorische Attribute

Das Attribut **name** gibt den Namen für **ClassKind** an und identifiziert **ClassKind** eindeutig.

6.7.11.4 Optionale Attribute

Das Attribut **variants** definiert die Varianten, in denen das Element **ClassKind** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, besitzt das Element **ClassKind** in allen **variants** Gültigkeit. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

6.7.12 Nutzungsarten (en: UsageKinds)

6.7.12.1 Allgemeines

Das optionale Element **UsageKinds** listet die Nutzungsarten einer Klasse, die in einer Klassifikation vorhanden sind.

6.7.12.2 Inhalt

Das Element **UsageKinds** beinhaltet ein oder mehrere Elemente **UsageKind**.

6.7.12.3 Obligatorische Attribute

Das Element **UsageKinds** besitzt keine obligatorischen Attribute.

6.7.12.4 Optionale Attribute

Das **variants**-Attribut definiert die Varianten, in denen das Element **UsageKinds** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, ist das Element **UsageKinds** in allen **variants** gültig. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

6.7.13 Nutzungsart (en: UsageKind)

6.7.13.1 Allgemeines

Das Element **UsageKind** gibt den Namen von **UsageKind** an.

6.7.13.2 Inhalt

Das Element **UsageKind** ist leer.

6.7.13.3 Obligatorische Attribute

Das Attribut **name** legt den Namen einer **UsageKind** und identifiziert diese eindeutig.

Das Attribut **mark** beschreibt wie der Code einer **Class** mit einer bestimmten **UsageKind** gekennzeichnet werden soll.

BEISPIEL

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

ANMERKUNG Das Beispiel gibt die Nutzungsarten an, die in ICD-10 verwendet werden. Der Primärcode gibt die zugrundeliegende Krankheit an und wird mit einem Kreuz (+) gekennzeichnet; ein optionaler Zusatzcode steht für die Manifestation einer Erkrankung wird mit einem Sternchen (*) gekennzeichnet (9).

6.7.13.4 Optionale Attribute

Das **variants**-Attribut definiert die Varianten, in denen das Element **UsageKind** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, ist das Element **UsageKind** in allen **variants** gültig. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

6.7.14 Nutzung (en: Usage)

6.7.14.1 Allgemeines

Das Element **Usage** nimmt Bezug auf **UsageKind** und bestimmt die Nutzung der **ModifierClass** oder **Class**.

6.7.14.2 Inhalt

Das Element **Usage** besitzt keinen Inhalt.

6.7.14.3 Obligatorische Attribute

Das Attribut **kind** bezieht sich auf eine **UsageKind**. Ein **kind**-Attribut kann sich nur auf einen vorhandenen Namen einer **UsageKind** in einer Klassifikation beziehen.

6.7.14.4 Optionale Attribute

Das **variants**-Attribut definiert die Varianten, in denen das Element **Usage** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, ist das Element **Usage** in allen **variants** gültig. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

6.7.15 Rubrikenarten (en: RubricKinds)

6.7.15.1 Allgemeines

Das Element **RubricKinds** listet die Rubrikenarten auf, die in der Klassifikation vorhanden sind.

6.7.15.2 Inhalt

Das Element **RubricKinds** beinhaltet ein oder mehrere **RubricKind**- Elemente.

6.7.15.3 Obligatorische Attribute

Das Element **RubricKinds** besitzt keine obligatorischen Attribute.

6.7.15.4 Optionale Attribute

Das Element **RubricKinds** besitzt keine optionalen Attribute.

BEISPIEL

```
<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>
```

ANMERKUNG Dieser ClaML-Abschnitt entspricht wieder ICD-10, ICF und vergleichbaren Klassifikationen. Eine Erklärung der verschiedenen, empfohlenen Rubrikenarten ist in Tabelle B.3 angegeben.

6.7.16 Rubrikenart (en: RubricKind)

6.7.16.1 Allgemeines

Das Element **RubricKind** beschreibt den Namen einer **RubricKind**.

6.7.16.2 Inhalt

Das Element **RubricKind** beinhaltet eine beliebige Anzahl des **Display**- Elements.

6.7.16.3 Obligatorische Attribute

Das Attribut **name** gibt den Namen für eine **RubricKind** an und identifiziert diese eindeutig.

6.7.16.4 Optionale Attribute

Das Attribut **inherited** gibt an, ob Rubriken mit dieser **RubricKind** an Unterklassen vererbt werden. Der Standardwert für dieses Attribut ist **false**.

Das **variants**-Attribut definiert die Varianten, in denen das Element **RubricKind** Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, ist das Element **RubricKind** in allen **variants** gültig. Die Varianten werden im Element **Variants** festgelegt (6.3.7).

6.7.17 Sichtanzeige (en: Display)

6.7.17.1 Allgemeines

Das Element **Display** beschreibt wie eine **ClassKind** oder **RubricKind** in einer bestimmten Sprache angezeigt werden soll.

6.7.17.2 Inhalt

Der Inhalt des Elements **Display** ist auf Text beschränkt.

6.7.17.3 Obligatorische Attribute

Das Element **Display** besitzt keine obligatorischen Attribute.

6.7.17.4 Optionale Attribute

Das Attribut **variants** definiert die Varianten, in denen dieses **Display**-Element benutzt wird. Falls das Attribut nicht angegeben ist, wird das **Display**-Element in allen Varianten benutzt.

Das Attribut **xml:lang** definiert die Sprache des Inhalts des Elements. Die Attributwerte **xml:lang** müssen sich an ISO 639-1 und ISO 3166-1 in Einklang mit RFC 3066 (8) orientieren. Sprachenidentifikatoren, die bei der IANA registriert sind, ist das Prefix „I-“ oder „i-“ voranzustellen. Für privat genutzte Codes ist das Prefix „X-“ oder „x-“ zu verwenden.

ANMERKUNG Das Attribut **xml:lang** wird im Dokumentstamm definiert (siehe 6.7.2.3) und darf nur bei Änderungen eines Nachkommens definiert werden, um Redundanzen zu vermeiden.

BEISPIEL

```
<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">Inklusiva</Display>
    </RubricKind>
  </RubricKinds>

</Classification>
```

6.7.18 Modifizierung (en: Modifier)

6.7.18.1 Allgemeines

Das Element **Modifier** beschreibt eine Modifizierung innerhalb einer Klassifikation.

ANMERKUNG **Class**-Elemente dürfen auch für die Änderung anderer **Class**-Elemente verwendet werden.

6.7.18.2 Inhalt

Das Element **Modifier** beinhaltet:

- eine beliebige Anzahl des Elements **Meta**;
- eine beliebige Anzahl des Elements **SubClass**;
- eine beliebige Anzahl des Elements **Rubric**;
- eine beliebige Anzahl des Elements **History**.

6.7.18.3 Obligatorische Attribute

Das Attribut **Code** gibt den Namen für den Modifier an und identifiziert den **Modifier** eindeutig. Bei diesem Wert ist die Groß-/Kleinschreibung relevant.

6.7.18.4 Optionale Attribute

Das Attribut **kind** kann verwendet werden, um unterschiedliche Arten von **Modifier** voneinander zu unterscheiden.

Das Attribut **variants** beschreibt die Variantenliste, in welcher dieser **Modifier** Gültigkeit besitzt. Wenn das Attribut nicht angegeben ist, besitzt der **Modifier** in allen Varianten Gültigkeit.

Das Attribut **version** legt die Version des **Modifier** fest. Es wird empfohlen, das weit verbreitete major.minor.patch Schema zur Versionsangabe zu verwenden.

Das Attribut **effectivedate** definiert das Wirksamkeitsdatum des **Modifier**.

Das Attribut **expirationdate** definiert das Ablaufdatum des **Modifier**.

Das Attribut **Status** definiert den Status des **Modifier**.

ANMERKUNG

Für Attribute, die Datum und Uhrzeit definieren (**date**, **effectivedate**, **expirationdate**), muss ein String im Format „JJJJ-MM-TTThh:mm:ss“ verwendet werden, der die Anforderungen der „xs:dateTime“-Beschränkung erfüllt. Wenn es keine spezifische Zeit gibt, ist „YYYY-MM-DDT00:00:00“ zu verwenden.

BEISPIEL 1

```
<Modifier code="Md1">
  <SubClass code="0"/>
  <SubClass code="1"/>
  <SubClass code="2" variants="cm" />
</Modifier>
```

ANMERKUNG In diesem **Modifier** ist die **SubClass** mit Code 2 nur für ICD-10-CM gültig, wobei die anderen **SubClass(es)** für alle Modifizierungen innerhalb eines Klassifikationsbaums Gültigkeit besitzen.

BEISPIEL 2

```
<Modifier code="Md2" status="active">
  <SubClass code="0"/>
  <SubClass code="1"/>
  <SubClass code="2"/>
</Modifier>
```

ANMERKUNG In diesem **Modifier** wird mit dem optionalen Attribut **status** angegeben, dass es sich um einen aktiven Modifier handelt.

6.7.19 Modifizierungsklasse (en: **ModifierClass**)

6.7.19.1 Allgemeines

Das Element **ModifierClass** beschreibt eine Modifizierungsklasse in der Klassifikation.

6.7.19.2 Inhalt

Das Element **ModifierClass** beinhaltet:

- eine beliebige Anzahl des Elements **Usage**;
- eine beliebige Anzahl des Elements **Meta**;
- eine beliebige Anzahl des Elements **SuperClass**;
- eine beliebige Anzahl des Elements **SubClass**;
- eine beliebige Anzahl des Elements **Rubric**;
- eine beliebige Anzahl des Elements **History**.

ANMERKUNG Wenn keine **Meta**-Elemente vorhanden sind, müssen Metadaten von **Class**- oder **ModifiedBy**-Elementen übernommen werden.

6.7.19.3 Obligatorische Attribute

Das Attribut **code** definiert den Code einer **ModifierClass**. Bei diesem Wert ist die Groß-/Kleinschreibung relevant.

6.7.19.4 Optionale Attribute

Das Attribut **modifier** nimmt Bezug auf den Code der Modifizierung, der die **ModifierClass** angehört. Ein **Modifier** kann sich nur auf einen vorhandenen **Modifier**-Code in einer Klassifikation beziehen. Bei diesem Wert ist die Groß-/Kleinschreibung relevant.

Das Attribut **kind** kann verwendet werden, um zwischen unterschiedliche Arten der **ModifierClass** zu unterscheiden.

Das Attribut **variants** definiert die Variantenliste in welcher diese **ModifierClass** gültig ist. Falls kein Attribut angegeben wird, gilt die **ModifierClass** in allen Varianten.

Das Attribut **version** legt die Version der **ModifierClass** fest. Es wird empfohlen, das weit verbreitete major.minor.patch Schema zur Versionsangabe zu verwenden.

Das Attribut **effectivedate** definiert das Wirksamkeitsdatum der **ModifierClass**.

Das Attribut **expirationdate** definiert das Ablaufdatum der **ModifierClass**.

Das Attribut **Status** definiert den Status des **ModifierClass**.

ANMERKUNG

Für Attribute, die Datum und Uhrzeit definieren (**date**, **effectivedate**, **expirationdate**), muss ein String im Format „JJJJ-MM-TTThh:mm:ss“ verwendet werden, der die Anforderungen der „xs:dateTime“-Beschränkung erfüllt. Wenn es keine spezifische Zeit gibt, ist „YYYY-MM-DDT00:00:00“ zu verwenden.

BEISPIEL 1

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

ANMERKUNG Das Attribut **modifier** muss verwendet werden, wenn die **ModifierClass** einem einzelnen **Modifier** zugeordnet ist. Wenn die **ModifierClass** in vielen unterschiedlichen **Modifier**-Elementen (z. B. Value Sets) verwendet wird, dann muss dieses Attribut weggelassen werden und die Beziehung zu den **Modifier**-Elementen durch **SuperClass**-Elemente ausgedrückt werden (siehe 6.7.21).

BEISPIEL 2

```
<ModifierClass modifier="Md2" code="0" status="active">
  <SuperClass code="Md2"/>
  <SubClass code="00"/>
  <SubClass code="01"/>
  <SubClass code="02"/>
</ModifierClass>
```

ANMERKUNG In dieser **ModifierClass** wird mit dem optionalen Attribut **status** angegeben, dass es sich um einen aktiven Modifier handelt.

6.7.20 Klasse

6.7.20.1 Allgemeines

Das Element **Class** beschreibt eine Klasse in der Klassifikation.

6.7.20.2 Inhalt

Das Element **Class** beinhaltet:

- eine beliebige Anzahl des Elements **Usage**.
- eine beliebige Anzahl des Elements **Meta**;
- eine beliebige Anzahl des Elements **SuperClass**;
- eine beliebige Anzahl des Elements **SubClass**;
- eine beliebige Anzahl des Elements **ModifiedBy**;
- eine beliebige Anzahl des Elements **ValidModifierClass**.
- eine beliebige Anzahl des Elements **ExcludeModifier**;
- eine beliebige Anzahl des Elements **Rubric**;
- eine beliebige Anzahl des Elements **History**.

6.7.20.3 Obligatorische Attribute

Das Attribut **Code** definiert den Code für die Klasse und identifiziert **Class** eindeutig. Bei diesem Wert ist die Groß-/Kleinschreibung relevant.

Das Attribut **kind** nimmt Bezug auf die **ClassKind** der Klasse, z. B. chapter, block, category, usw. Eine Liste ist in Tabelle B.1 angegeben. Ein **Kind**-Attribut kann sich nur auf einen vorhandenen **ClassKind**-Namen in einer Klassifikation beziehen.

6.7.20.4 Optionale Attribute

Das Attribut **variants** beschreibt eine Variantenliste, in der die **Class** gültig ist. Falls kein Attribut angegeben ist, besitzt **Class** in allen Varianten Gültigkeit. Eine Liste ist in Tabelle B.2 angegeben.

Das Attribut **version** legt die Version der **Class** fest. Es wird empfohlen, das weit verbreitete major.minor.patch Schema zur Versionsangabe zu verwenden.

Das Attribut **effectivedate** definiert das Wirksamkeitsdatum der **Class**.

Das Attribut **expirationdate** definiert das Ablaufdatum der **Class**.

Das Attribut **Status** definiert den Status der **Class**.

ANMERKUNG

Für Attribute, die Datum und Uhrzeit definieren (**date**, **effectivedate**, **expirationdate**), muss ein String im Format „JJJJ-MM-TTThh:mm:ss“ verwendet werden, der die Anforderungen der „xs:dateTime“-Beschränkung erfüllt. Wenn es keine spezifische Zeit gibt, ist „YYYY-MM-DDT00:00:00“ zu verwenden.

BEISPIEL 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>
```

ANMERKUNG Die Klasse mit dem Code A00 ist eine Kategorie. Ihre übergeordnete Klasse ist der Block A00-A09, ihre Unterklassen sind A00.0, A00.1 und A00.9. Der bevorzugte Begriff ist Cholera. Das Beispiel zeigt eine einfache Klasse, die in ICF oder vergleichbaren Klassifikationssystemen sehr ähnlich aussehen würde.

BEISPIEL 2

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

ANMERKUNG In dieser Klasse wird mit dem optionalen Attribut **status** angegeben, dass es sich um eine veraltete Klasse handelt. Diese Klasse wurde gelöscht und mit Veröffentlichung von ICD-10, Version 2016, durch die Klassen A97.0, A97.1, A97.9 ersetzt. Mit dieser Funktion darf die veraltete Klasse als inaktive Klasse in der neuen ClaML Datei beibehalten werden.

ANMERKUNG In Übereinstimmung mit der Einschränkung des Klassencode-Schlüssels darf es im **Klassifikationsbaum** einer ClaML Datei keine identischen veralteten Codes geben, was jedoch bei Wiederverwendung eines Codes der Fall wäre.

6.7.21 ModifiziertDurch (en: ModifiedBy)

6.7.21.1 Allgemeines

Das Element **ModifiedBy** bezeichnet den Code eines **Modifier** (oder einer **Class**), welcher die Klasse und deren Nachkommen verändert. Es wird angenommen, dass alle Unterklassen eines **Modifier** (oder einer **Class**) ebenfalls für die Modifizierung angesprochen werden, obwohl im Modifier-Element nicht direkt auf sie Bezug genommen wird.

Falls der **Modifier** einen Nachkommen der **Class** nicht ansprechen soll, muss das Element **ExcludeModifier** für diesen bestimmten Nachkommen angegeben werden (siehe 6.7.22).

ModifiedBy-Elementen untergeordnete **Meta**-Elemente werden zum Definieren von Metadaten verwendet, die beim Modifizieren einer bestimmten **Class** durch **ModifierClass** (oder modifizierende **Class**) auf alle resultierenden Klassen anzuwenden sind. In **ModifiedBy**-Elementen definierte Metadaten müssen die Metadaten von allen verknüpften **ModifierClass**- (oder modifizierenden **Class**)-Elementen überschreiben (weitere Informationen erhalten Sie unter 6.7.3.1). Wenn diese Metadaten nur auf ein oder mehrere spezifische **ModifierClass**- (oder modifizierende **Class**)-Elemente begrenzt werden sollen, müssen stattdessen **ValidModifierClass**-Elementen untergeordnete **Meta**-Elemente verwendet werden (siehe 6.7.23).

ANMERKUNG Beginnend mit dieser Version dürfen **Class**-Elemente auch zur Modifikation von anderen **Class**-Elementen verwendet werden.

ANMERKUNG Diese Version von ClaML unterstützt, wie nachstehend beschreiben, zwei Möglichkeiten, Code-Modifikationen zu entwickeln. Zusätzliche Informationen zu diesem Thema und illustrative Beispiele sind Anhang C zu entnehmen.

- Das „klassische“ Modifikationsverfahren für **Modifier**- und **ModifierClass** (oder modifizierende **Class**)-Elemente besteht darin, Modifizierungslisten zu entwickeln, die von vielen verschiedenen **Class**-Elementen verwendet werden. Für die Modifikation verwendete **ModifierClass**- (oder modifizierende **Class**)-Elemente dürfen sich nur auf einen **Modifier** beziehen. Bei Verwendung von **ModifierClass**-Elementen muss das Attribut **modifier** verwendet werden. Gültige Kombinationen dieser Modifizierungslisten werden durch **ValidModifierClass**-Elemente ausgedrückt.
- Bei einem „Value Set“-basierten Änderungsverfahren werden **Modifier**-Elemente verwendet, um **ModifierClass**- (oder modifizierende **Class**)-Elemente in „Value Sets“ zu gruppieren, die gleichzeitig gültige Modifizierungsklassen definieren. Die Verwendung von **ValidModifierClass**-Elementen wird überflüssig. Auf die für die Modifikation verwendeten **ModifierClass**- (oder modifizierenden **Class**)-Elemente darf in mehreren **Modifier**-Elementen Bezug genommen werden. Bei Verwendung von **ModifierClass**-Elementen darf das Attribut **modifier** nicht verwendet werden (dies erfordert jedoch die Verwendung von eindeutigen Codes für **ModifierClass**-Elemente). Die Beziehung zu **ModifierClass**- (oder modifizierenden **Class**)-Elementen muss stattdessen durch mehrere, den **Modifier**- und/oder **ModifierClass**- (oder modifizierenden **Class**)-Elementen untergeordnete **SubClass**- und/oder **SuperClass**-Elemente ausgedrückt werden.

6.7.21.2 Inhalt

Das Element **ModifiedBy** beinhaltet:

- eine beliebige Anzahl des Elements **Meta**;

6.7.21.3 Obligatorische Attribute

Das Attribut **code** nimmt Bezug auf den Code des **Modifier**. Ein **Code** kann sich nur auf einen vorhandenen **Modifier**-Code in einer Klassifikation beziehen. Bei diesem Wert ist die Groß-/Kleinschreibung relevant.

6.7.21.4 Optionale Attribute

Das Attribut **position** beschreibt die Position des Codes der **ModifierClass**, wenn der **Modifier** dazu benutzt wird, die Unterklassen der modifizierten **Class** zu generieren; das Attribut muss ein Zahlenwert sein. Die erste Position wird durch die Nummer '1' angegeben. Der Wert muss in einer **Class** eindeutig sein.

ANMERKUNG Es wird empfohlen, das Attribut **position** mit Klassifikationen zu verwenden, bei denen der Modifier-Code dem primären Code als Suffix beigefügt ist (z. B. ICD-10). Es wurde nicht als obligatorisches Attribut festgelegt, da es Klassifikationen gibt, bei denen die Position des beigefügten Codes nicht anwendbar ist (z. B. Modifikation durch Kombinieren von mehreren primären Codes, Cluster-Coding in ICD-11).

ANMERKUNG Das Attribut **position** ermöglicht die Anwendung von **ModifierClass**-Elementen auf jede betreffende Position.

BEISPIEL In einigen Fällen muss in ICD-10 ein Modifier auf die 5. Zeichenposition des Codes angewendet werden, obwohl der Code selbst nur drei Zeichen lang ist. Laut „ICD-10 Volume 2 Instruction Manual (2010, Abschnitt 2.4.5 and 2.4.6)“ muss in diesen Fällen die vierte Position mit dem Buchstaben „X“ besetzt werden. (9)

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

Das Attribut **variants** beschreibt die Variantenliste, in der dieses **ModifiedBy**-Element Gültigkeit besitzt. Falls das Attribut nicht angegeben wird, besitzt dieses **ModifiedBy** in allen Varianten Gültigkeit.

Mit dem Attribut **optionalmodifier** wird angegeben, dass der in Bezug genommene **Modifier** optional verwendet werden kann. Andernfalls ist der in Bezug genommene **Modifier** verpflichtend zu verwenden. Der Standardwert des Attributs **optionalmodifier** ist „falsch“.

BEISPIEL

Wenn eine klinische/medizinische Veränderung der Klassifizierung geschaffen wird, welches die begrenzte Addition der fünften Stelle erlaubt, wird diese Modifikation wie folgt angegeben:

```
<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>
```

ANMERKUNG Im Beispiel werden sowohl **Class** C88 als auch deren Nachkomme C88.0 durch den **Modifier** Md1 angesprochen. Für den Nachkommen **Class** C88.1 wurde der Modifier ausgeschlossen, d. h. **Class** C88.1 wird durch den **Modifier** Md1 nicht angesprochen.

6.7.22 ExcludeModifier

6.7.22.1 Allgemeines

Das Element **ExcludeModifier** nimmt Bezug auf den Code eines Modifier, der nicht auf diese Klasse angewendet werden soll.

6.7.22.2 Inhalt

Das Element **ExcludeModifier** besitzt keinen Inhalt.

6.7.22.3 Obligatorische Attribute

Das Attribut **code** bezieht sich auf den Code der exkludierten Modifizierung. Ein **Code** kann sich nur auf einen vorhandenen **Modifier**-Code in einer Klassifikation beziehen. Bei diesem Wert ist die Groß-/Kleinschreibung relevant.

6.7.22.4 Optionale Attribute

Das Attribut **variants** definiert eine Variantenliste, in welcher dieses Element **ExcludeModifier** Gültigkeit besitzt. Wenn das Attribut nicht angegeben wird, besitzt dieses Element **ExcludeModifier** in allen Varianten Gültigkeit.

6.7.23 ValidModifierClass

6.7.23.1 Allgemeines

Das Element **ValidModifierClass** nimmt Bezug auf den Code einer **ModifierClass** (oder modifizierenden **Class**), welche für diese **Class** Gültigkeit besitzt. Nur gültige Modifizierungsklassen dürfen dazu benutzt werden, eine **Class** zu verändern. Wenn kein **ValidModifierClass**-Element aufgeführt ist, besitzen alle **ModifierClass**- (oder modifizierenden **Class**)-Elemente für diese **Class** Gültigkeit.

Es müssen mehrere kaskadierende Ebenen mit **ValidModifierClass**-Elementen verwendet werden, um beim Verknüpfen mehrerer Modifizierungsebenen gültige Modifier-Kombinationen darzustellen.

ValidModifierClass-Elementen untergeordnete **Meta** -Elemente werden zum Definieren von Metadaten verwendet, die beim Modifizieren einer spezifischen **Class** durch eine bestimmte **ModifierClass** (oder eine bestimmte modifizierende **Class**), auf die im Attribut **Code** der **ValidModifierClass** Bezug genommen wird, auf die resultierende Klasse angewendet werden. In **ValidModifierClass**-Elementen definierte Metadaten müssen Metadaten der **ModifierClass**- (oder modifizierenden **Class**)-Elemente, auf die Bezug genommen wird, außer Kraft setzen (weitere Informationen sind 6.7.3.1 zu entnehmen). Wenn diese Metadaten auf alle resultierenden Klassen anzuwenden sind, müssen stattdessen die den **ModifiedBy**-Elementen untergeordneten **Meta** -Elemente verwendet werden (siehe 6.7.21).

6.7.23.2 Inhalt

Das Element **ValidModifierClass** beinhaltet:

- eine beliebige Anzahl des Elements **Meta**;
- eine beliebige Anzahl des Elements **ValidModifierClass**.

6.7.23.3 Obligatorische Attribute

Das Attribut **code** definiert den Code einer gültigen **ModifierClass** (oder modifizierenden **Class**). Bei diesem Wert ist die Groß-/Kleinschreibung relevant.

6.7.23.4 Optionale Attribute

Das Attribut **variants** definiert die Variantenliste, in der dieses **ValidModifierClass**-Element Gültigkeit besitzt. Falls das Attribut nicht angegeben ist, besitzt dieses **ValidModifierClass**-Element in allen Varianten Gültigkeit.

Das Attribut **position** nimmt Bezug auf das äquivalente Attribut des Elements **ModifierBy**. Es darf im Umgang mit mehreren Modifizierungsebenen dazu benutzt werden, die aktuelle Ebene der **ValidModifierClass** anzugeben. Es muss einem Zahlenwert entsprechen. Die erste Position wird durch die Nummer „1“ angegeben. Das Attribut darf nur auf eine vorhandene **Position** eines **ModifiedBy**-Elements in einer **Class** Bezug nehmen.

BEISPIEL 1

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

ANMERKUNG C88.0 wäre eine gültige Klasse.

BEISPIEL 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>
```

ANMERKUNG Dieses Beispiel veranschaulicht kaskadierende **ValidModifierClass**-Elemente, die dazu verwendet werden, gültige Kombinationen aus mehreren Modifizierungsebenen darzustellen. Das Positionsattribut dient dazu, Klarheit bezüglich der aktuellen Modifizierungsebene zu bewahren.

ANMERKUNG E10.72, E10.73, E10.74, E10.75, E10.80, E10.81 wären gültige Klassen.