

eCH-0033: Beschreibung von XML Namespaces

Name	Beschreibung von XML Namespaces
Standard-Nummer	eCH-0033
Kategorie	Interoperabilitätsstandard; Verfahrensstandard; Musterlösung; Best Practice
Reifegrad	Definiert ; Experimentell; Implementiert; Verbreitet
Status	In Arbeit; Entwurf; Vorschlag ; Genehmigt; Ausser Kraft
Gültig seit	2006-11-29
Gültigkeitsdauer	
Änderungen	
Ersetzt Standard	
Basiert auf	
Version	1.0
Sprachen	Deutsch
Herausgeber / Vertrieb	Verein eCH, Laupenstrasse 18a, 3008 Bern T 031 560 00 20, F 031 560 00 25
Autoren	Erik Wilde, ETH Zürich (http://dret.net/netdret/)

Zusammenfassung

Das vorliegende Dokument beschreibt, in welcher Form XML Namespaces beschrieben werden sollten, so dass die Benutzer von XML-basierten Vokabularen im Rahmen von eCH eine einfache Quelle haben, über die sie Dokumentation zu einem XML Namespace finden können. Ausgehend von einem einfachen XML Schema können zu einer Definition eines XML Vokabulars auf diese Art auf eine einfache Art Beschreibungen zu einem XML Namespace erzeugt werden, aus denen sowohl menschen- als auch maschinenlesbare Information gewonnen werden kann.

Versionen

<i>Version</i>	<i>Datum</i>	<i>Name</i>	<i>Bemerkungen (geändert, geprüft, genehmigt) *(geplant)</i>
0.10	2005-01-15	Erik Wilde	erster Entwurf auf Grundlage eCH-Sitzung 25.11.04
0.20	2005-05-29	Erik Wilde	Einarbeitung der Kommentare vom 8.4.05 (eCH-Sitzung)
0.30	2005-08-14	Erik Wilde	Einarbeitung der Kommentare vom 3.6.05 (eCH-Sitzung)
0.31	2005-09-05	Erik Wilde	Einarbeitung der Kommentare von Willy Müller
0.40	2005-09-16	Erik Wilde	Einarbeitung der Kommentare vom 16.9.05 (eCH-Sitzung)
0.41	2005-09-24	Erik Wilde	Einarbeitung des Schemas und der Beispiele
0.90	2006-01-08	Erik Wilde	Einarbeitung der Kommentare vom 2.12.05 (eCH-Sitzung)
0.91	2006-02-08	Erik Wilde	Einarbeitung der Kommentare vom 2.2.06 (eCH-Sitzung)
1.0	2006-11-28	Erik Wilde	Definitive Version

Inhaltsverzeichnis

1	Status des Dokuments	6
1.1	Terminologie der Empfehlungen	6
2	Zusammenfassung der Empfehlungen.....	7
2.1	Benennung und Beschreibung der Namespaces.....	7
2.2	Beschreibung der Versionen.....	7
2.3	Aufbau eines Namespace Names mit Versionierung.....	8
2.4	Beschreibungstyp URI	8
2.5	Beschreibungsaspekt Namespace Root Name.....	9
2.6	Beschreibungsaspekt Namespace Name	9
2.7	Beschreibungsaspekt Titel.....	9
2.8	Beschreibungsaspekt Abstract.....	9
2.9	Beschreibungsaspekt Prefix.....	10
2.10	Beschreibungsaspekte für die Versionierung (Major/Minor Version)	10
2.11	Beschreibungsaspekt Schema.....	10
2.12	Beschreibungsaspekt Date	10
2.13	XML Beschreibung	10
2.14	XHTML Beschreibung	10
3	Einleitung.....	12
3.1	Überblick	12
3.2	Anwendungsgebiet.....	13
3.3	Vorteile	13
3.4	Schwerpunkte	14
4	Grundlagen von XML Namespaces	15
4.1	XML Namespaces	15
4.1.1	Versionen der Empfehlung	15
4.1.2	Definition von Namespaces	15
4.1.3	Deklaration von Namespaces.....	16
4.1.4	Verwendung von Namespaces	17
4.2	XML Schema und XML Namespaces	17
4.3	Beschreibung von XML Namespaces	18

4.3.1	Bestehende Technologien	19
4.3.1.1	Resource Directory Description Language (RDDL)	19
4.3.1.2	Gleaning Resource Descriptions from Dialects of Languages (GRDDL) ...	19
4.3.2	Gewählter Ansatz	20
5	Namespace Namen und Versionsangaben.....	21
5.1	Benennung und Beschreibung des Namespaces	21
5.1.1	Problemstellung	21
5.1.2	Empfehlungen.....	21
5.2	Beschreibung der Versionen.....	22
5.2.1	Problemstellung	22
5.2.2	Empfehlungen.....	22
5.3	Aufbau eines Namespace Names mit Versionierung.....	23
6	Aufbau von Namespace-Beschreibungen	25
6.1	Strukturierte Namespace-Beschreibungen	25
6.2	Unstrukturierte Namespace-Beschreibungen	27
6.3	Interne Struktur einer Beschreibung	27
6.3.1	Beschreibungsaspekte	27
6.3.2	Typen von Beschreibungsaspekten.....	27
6.3.2.1	Token	28
6.3.2.2	String.....	28
6.3.2.3	Date.....	28
6.3.2.4	Number.....	28
6.3.2.5	Texts	28
6.3.2.6	URI	28
6.4	Unterstützte Beschreibungsaspekte	29
6.4.1	Namespace Root Name.....	30
6.4.2	Namespace Name	31
6.4.3	Title	31
6.4.4	Abstract.....	31
6.4.5	Prefix.....	32
6.4.6	Versionierung.....	32
6.4.6.1	Major Version	32

6.4.6.2	Minor Version	33
6.4.6.3	Nickname	33
6.4.7	Dependency.....	33
6.4.8	Documentation.....	33
6.4.9	Schema.....	33
6.4.10	Tool.....	34
6.4.11	Test Data	34
6.4.12	Example Data	34
6.4.13	Owner	34
6.4.14	Date	35
6.4.15	Status.....	35
7	Format von Namespace-Beschreibungen	36
7.1	XML Beschreibung.....	36
7.2	XHTML Beschreibung	36
8	Sicherheitsüberlegungen	38
9	Hinweise auf Rechte Dritter	39
	Anhang A – Referenzen & Bibliographie	40
	Anhang B – Mitarbeit & Überprüfung.....	43
	Anhang C – Abkürzungen & Glossar	44
	Anhang E – Urheberrechte.....	45

1 Status des Dokuments

Das vorliegende Dokument ist *in Arbeit*. Es ist nur für den internen Gebrauch in der Fachgruppe gedacht.

Das vorliegende Dokument ist ein *Entwurf*. Es wurde veröffentlicht, damit es von interessierten Kreisen begutachtet und bei Bedarf versuchsweise umgesetzt werden kann. Feedback ist erwünscht und sollte an die Autoren gerichtet werden. Der Inhalt ist vom Expertenausschuss noch nicht genehmigt, er kann auf der Basis des eingehenden Feedbacks noch ändern und hat daher keine normative Kraft.

Das vorliegende Dokument enthält den endgültigen Text, der dem Expertenausschuss zur Genehmigung vorgeschlagen wird. Es wurde noch nicht offiziell genehmigt und hat daher noch keine normative Kraft.

Das vorliegende Dokument wurde vom Expertenausschuss *genehmigt*. Es hat für das definierte Einsatzgebiet im festgelegten Gültigkeitsbereich normative Kraft.

1.1 Terminologie der Empfehlungen

Richtlinien in diesem Dokument werden gemäss der Terminologie aus [RFC2119] angegeben, dabei kommen die folgenden Ausdrücke zur Anwendung, die durch **GROSSSCHREIBUNG** als Wörter mit den folgenden Bedeutungen kenntlich gemacht werden (Zitat aus [RFC2119]):

- **MUST**: This word, or the terms "**REQUIRED**" or "**SHALL**", mean that the definition is an absolute requirement of the specification.
- **MUST NOT**: This phrase, or the phrase "**SHALL NOT**", mean that that definition is an absolute prohibition of the specification.
- **SHOULD**: This word, or the adjective "**RECOMMENDED**", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- **SHOULD NOT**: This phrase, or the phrase "**NOT RECOMMENDED**" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- **MAY**: This word, or the adjective "**OPTIONAL**", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option **MUST** be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option **MUST** be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

2 Zusammenfassung der Empfehlungen

2.1 Benennung und Beschreibung der Namespaces

Empfehlungen aus Abschnitt 5.1

- **MUST NOT:** Namespace Namen müssen URIs [RFC3986] sein, d.h. insbesondere sind keine IRIs [RFC3987] erlaubt.
- **MUST:** eCH Namespace Namen verwenden das `http` URI Scheme [RFC2616].
- **MUST:** Der Domain Name ist der allgemein verwendete Web Server Name der definierenden Instanz (z.B. `www.ejpd.admin.ch`), mit oder ohne führendes `www`, je nachdem, wie dies beim Web-Server konfiguriert ist und bevorzugt verwendet wird. Auf diesem Server muss die Namespace-Beschreibung zugänglich gemacht werden, es ist also sicherzustellen, dass dies möglich ist.
- **SHOULD:** Der Namespace-Pfad sollte kurz und prägnant sein, idealerweise unter einem gemeinsamen Präfix für alle Namespace Namen, die Empfehlung lautet `xml ns`. Der Grund dafür ist, dass dieser Name über lange Zeit (die Benutzungsdauer der mit dem Namespace verbundenen Schemas und damit auch der Instanzen) gültig bleiben sollte, und für einfache Namen lässt dies einfacher erreichen als für komplizierte Pfade.
- **MUST:** Es muss einen Pfad (idealerweise einen Namen) geben für den Anwendungsfall, z.B. `http://www.ech.ch/xml ns/ech-0033`, der auf eine den allgemeinen Anwendungsfall beschreibende Ressource verweist.
- **SHOULD:** Die Namespace-Beschreibung, die unter der versionsunabhängigen URI `http://www.ech.ch/xml ns/ech-0033` verfügbar ist, sollte auf alle existierenden Versionen verweisen.
- **MAY:** Die obigen Empfehlungen dürfen ignoriert werden, falls der Namespace Name durch eine externe Instanz vorgegeben wird (z.B. einen Standard), aber trotzdem durch eine Namespace-Beschreibung beschrieben werden soll.

2.2 Beschreibung der Versionen

Empfehlungen aus Abschnitt 5.2

- **MUST:** Major und Minor Versions müssen jeweils mit ganzen positiven Zahlen nummeriert und diese in aufsteigender Reihenfolge vergeben werden. Führende Nullen (ausser für die 0 selber) und Vorzeichen sind nicht zulässig.
- **MUST:** Es muss auf den allgemeinen Pfad aufbauende Pfade für konkrete Versionen des Schemas geben, z.B. `http://www.ech.ch/xml ns/ech-0033/1` als die Beschreibung der ersten Version des Schemas.
- **MUST:** Die Benennung der Major Version muss dem XML Schema Typen `xs: token` gehorchen, darf aber keine Schrägstriche enthalten.

- **MUST:** Der gleiche Namespace Name darf nur für kompatible Schemas verwendet werden, d.h. für Schemas, bei denen die Validierung von Instanzen mit allen Versionen des Schemas erfolgreich möglich ist. Inkompatible Änderungen (d.h. neue Major Versions) müssen eine Versionierung des Namespace Namens zur Folge haben.
- **MUST:** Rückwärtskompatible Applikationen müssen die Richtlinien der Namespace Namen Versionierung respektieren und Dokumente mit älteren Namespace Namen akzeptieren.
- **MAY:** Die obigen Empfehlungen dürfen ignoriert werden, falls der Namespace Name durch eine externe Instanz vorgegeben wird (z.B. einen Standard), aber trotzdem durch eine Namespace-Beschreibung beschrieben werden soll.
- **SHOULD:** Ausser in begründeten Ausnahmefällen sollte immer mit Minor Versions gearbeitet werden. Das erste Release einer Major Version sollte auch gleich immer einer Minor Version Nummer zugeordnet werden (wie in Abschnitt 6.4.9 beschrieben verweist die Major Version Beschreibung dann auf kein Schema, sondern dieser Verweis findet sich lediglich in der Beschreibung der ersten Minor Version).

2.3 Aufbau eines Namespace Names mit Versionierung

Empfehlungen aus Abschnitt 5.3

- **MUST:** Für die Beschreibung der Minor Version eines Schemas muss die Beschreibung der Major Version existieren.
- **MUST:** Die Beschreibung der Minor Version eines Schemas muss unter der URI verfügbar sein, die sich aus der Konkatenation der Namespace URI des Schemas (diese enthält die Major Version), eines Schrägstriches, und der Minor Version ergibt. Dies ist die URI, unter der die Beschreibung der Minor Version zugänglich sein muss, die Namespace URI des durch die Minor Version beschriebenen Namespaces ist nach wie vor der Namespace Name der Major Version (ohne Schrägstriches und Minor Version)!
Beispiel der Minor Version Beschreibung: `http://www.ech.ch/xml ns/ech-0033/1/0`
Beispiel des beschriebenen Namespaces: `http://www.ech.ch/xml ns/ech-0033/1`

2.4 Beschreibungstyp URI

Empfehlungen aus Abschnitt 6.3.2.6

- **SHOULD:** URIs sollten auf so informative Ressourcen wie möglich zeigen. Gibt es z.B. eine Übersichtsseite eines Dokuments und dort Versionen in verschiedenen Sprachen, so sollte die URI auf die Übersichtsseite zeigen. Ist die durch die URI zu identifizierende Ressource durch einen XML Namespace beschrieben, sollte die URI den Namespace Name angeben und nicht die URI des Schemas oder der Dokumentation.

2.5 Beschreibungsaspekt Namespace Root Name

Empfehlungen aus Abschnitt 6.4.1

- **MUST:** Dies ist der Root Name des beschriebenen Namespaces (d.h. ohne Versionsangabe und ohne Schrägstrich am Ende, wie in Abschnitt 5.1 beschrieben), gemäss XML Namespaces muss es sich dabei um eine URI [RFC3986] handeln.
- **MUST:** Wird der Beschreibungsaspekt des Namespace Root Name in einer Major Version Beschreibung verwendet, so muss eine Namespace Root Beschreibung mit dem gleichen Namespace Root Name existieren (andernfalls kann die Major Version Beschreibung keiner Namespace Root Beschreibung zugeordnet werden und ist daher fehlerhaft).
- **MUST:** Wird der Beschreibungsaspekt des Namespace Root Name in einer Minor Version Beschreibung verwendet, so muss eine Major Version Beschreibung mit dem gleichen Namespace Root Name existieren (andernfalls kann die Minor Version Beschreibung keiner Major Version Beschreibung zugeordnet werden und ist daher fehlerhaft).
- **SHOULD:** Die Beschreibung des XML Namespaces sollte unter dieser URI verfügbar sein (in dem in Abschnitt 7.2 beschriebenen Format), dies kann jedoch nicht immer garantiert werden, z.B. bei Namespaces, deren Namespace Name vorgegeben ist und eine ausserhalb des eigenen Verantwortungsbereiches liegende Domain benutzt.

2.6 Beschreibungsaspekt Namespace Name

Empfehlung aus Abschnitt 6.4.2

- **MUST:** Dies ist der Name des beschriebenen Namespaces (dieser Beschreibungsaspekt kann nur für in Abschnitt 6.2 beschriebene Simple Namespaces verwendet werden), gemäss XML Namespaces muss es sich dabei um eine URI [RFC3986] handeln.

2.7 Beschreibungsaspekt Titel

Empfehlung aus Abschnitt 6.4.3

- **SHOULD:** Hierbei handelt es sich um den Titel des definierten Vokabulars, es sollte ein kurzer und beschreibender Name sein, mit dem das Vokabular üblicherweise bezeichnet wird.

2.8 Beschreibungsaspekt Abstract

Empfehlungen aus Abschnitt 6.4.4

- **SHOULD:** Idealerweise sollte (**SHOULD**) der Abstract mehrsprachig vorliegen.
- **MUST:** Liegt der Abstract mehrsprachig vor, so müssen die Versionen in den verschiedenen Sprachen jeweils mit einer Sprachmarkierung versehen werden.

2.9 Beschreibungsaspekt Prefix

Empfehlung aus Abschnitt 6.4.5

- **SHOULD:** Die Namespace-Beschreibung sollte einen Präfix empfehlen, der nur dann nicht verwendet werden sollte, wenn schwerwiegende Gründe dagegen sprechen, z.B. ein Namenskonflikt mit einem Präfix eines anderen verwendeten Vokabulars.

2.10 Beschreibungsaspekte für die Versionierung (Major/Minor Version)

Empfehlungen aus Abschnitt 6.4.6

- **MUST:** Minor Versionen müssen den gleichen Namespace Namen wie die zugrundeliegende Major Version verwenden.
- **SHOULD:** Die Minor Version wird nicht im Namespace Name verwendet, sie dient alleine der Markierung der Minor Version Beschreibung, und sollte, falls durch das Vokabular vorgesehen, als Angabe in Schemas und/oder Dokumenten erscheinen.
- **MUST:** Im Fall von XML Schema muss die Minor Version (und nur diese) im version Attribut des Schemas angegeben werden.

2.11 Beschreibungsaspekt Schema

Empfehlung aus Abschnitt 6.4.9

- **MUST NOT:** Wenn es zu Major Versions dazugehörige Minor Versions gibt, so darf die Namespace-Beschreibung der Major Version auf kein Schema verweisen.

2.12 Beschreibungsaspekt Date

Empfehlung aus Abschnitt 6.4.14

- **MUST:** Dieser Beschreibungsaspekt muss ein Datum gemäss dem XML Schema xs: date Typ enthalten, der ein *Calendar Date* gemäss [ISO8601] definiert (dies ist das etablierte YYYY-MM-DD Format).

2.13 XML Beschreibung

Empfehlung aus Abschnitt 7.1

- **SHOULD:** In Ergänzung zu den in Kapitel 6 definierten Namespace-Beschreibungen definiert eCH ein XML Schema und damit ein Format, das verwendet werden sollte, um Namespace-Beschreibungen zu repräsentieren

2.14 XHTML Beschreibung

Empfehlung aus Abschnitt 7.2

- **MUST:** Existiert unter dem Namespace Name eines Vokabulars (dies betrifft Namespace Root und Major Version Namespaces) eine HTML Seite, so muss diese den

hier beschriebenen Anforderungen genügen, d.h. sie muss XHTML benutzen, und sie muss mit einem `link` Element mit dem `ech-0033:xml-desc` Attributwert auf das XML verweisen, das gemäss Abschnitt 7.1 den Namespace beschreibt.

3 Einleitung

XML Namespaces [xmlns,xmlns11] sind ein Mechanismus, um eine Menge von Namen mit einer gemeinsamen Identifikation (dem Namespace Namen) zu versehen. XML Namespaces werden in verschiedenen Zusammenhängen und Anwendungen verwendet. Die häufigste Anwendung im Bereich der XML-Technologien ist, die durch ein Schema (also eine DTD oder ein XML Schema oder eine andere Art von Definition eines XML-Vokabulars) definierten Namen (besonders von Interesse sind hier Element- und/oder Attributnamen) einem Namespace zuzuordnen, und dadurch zu erreichen, dass sie eindeutig erkennbar verwendet werden können. Obwohl das vorliegende Dokument nicht ausschliesslich für den Anwendungsfall von durch ein Schema definierte Namen verwendbar ist, stellt dieser Anwendungsfall den Ausgangspunkt und den Schwerpunkt der Betrachtungen dar.

Während ein XML Namespace selber eine einfache Struktur hat (eine benannte und u.U. durch eine zusätzliche Beschreibung, z.B. ein Schema, strukturierte Menge von Namen), können die durch die Namen des XML Namespaces beschriebenen Konzepte sehr komplex sein. Für die Verwendung der Namen eines XML Namespaces sind daher nicht nur die Namen notwendig, sondern es muss ihre Bedeutung bekannt sein, so dass sie angemessen interpretiert und verwendet werden können. Aus diesem Grund definiert das vorliegende Dokument eine Richtlinie dafür, in welcher Weise die Namen eines XML Namespaces beschrieben werden sollten. Der besondere Schwerpunkt liegt dabei auf XML Namespaces, die Namen eines XML Schemas enthalten, die zur Verwendung in XML Dokumenten gedacht sind ([eCH-0018] enthält weitere Informationen zu XML und XML Schema).

3.1 Überblick

Kaum eine der verbreiteten XML-Technologien kommt ohne XML Namespaces aus, und für viele Technologien ist die Unterstützung von XML Namespaces essentiell. Eine der wichtigsten Anwendungen von XML Namespaces ist die Benennung eines durch ein Schema definierten Vokabulars. In allen Anwendungen, in denen Instanzen eines Schema von verschiedenen Anwendern verarbeitet werden, sollte das Schema einen Namespace definieren, durch dessen Name das im Schema definierte Vokabular weltweit eindeutig identifizierbar wird (in XML Schema erfolgt dies innerhalb der Schemadefinition mit dem `targetNamespace` Attribut).

In Ronald Bourrets "XML Namespaces FAQ" [bourret] findet sich eine exzellente Sammlung von Fragen und Antworten zum Thema XML Namespaces. Eine ebenfalls sehr gute Beschreibung von Namespaces im Zusammenhang mit der Frage nach der Beschreibung von Namespaces hat Tim Bray (einer der Autoren des XML-Standards) unter dem Titel "Architectural Theses on Namespaces and Namespace Documents" verfasst [bray].

Das vorliegende Dokument stellt keine generelle Erklärung von XML Namespaces dar, sondern setzt ihre grundlegende Kenntnis voraus. Ziel des vorliegenden Dokuments ist es, die Beschreibung von XML Namespaces so zu vereinheitlichen, dass für im Rahmen von eCH verwendete definierte Namespaces eine einheitliche Art von Beschreibungen verwendet wird, die sich zudem automatisiert interpretieren lassen, um ein Verzeichnis von eCH Namespaces erstellen zu können.

Das World Wide Web Consortium (W3C) stellt in Abschnitt 4.5.4 seiner "Architecture of the World Wide Web" Recommendation [webarch] unter dem Titel "Namespace Documents" als "Good Practice" fest: "The owner of an XML namespace name **SHOULD** make available material intended for people to read and material optimized for software agents in order to meet the needs of those who will use the namespace vocabulary."

Das vorliegende Dokument ist die Empfehlung, wie dieser "Good Practice" im Rahmen von eCH Aktivitäten nachgekommen werden kann und sollte. Der "Owner" eines XML Namespaces ist derjenige, der ihn (und natürlich vor allem das zugrundeliegende Vokabular) definiert, und der demzufolge auch die Beschreibung für den XML Namespace erstellen sollte.

3.2 Anwendungsgebiet

Anwendungsgebiet des vorliegenden Dokumentes sind alle Vokabulare, die als XML Namespace definiert und/oder verwendet werden. Schwerpunkt des vorliegenden Dokumentes sind Vokabulare, die durch eine Schemadefinition und insbesondere ein XML Schema definiert sind, jedoch ist das vorliegende Dokument nicht auf diesen Anwendungsfall begrenzt.

Obwohl andere Anwendungen möglich sind (im Prinzip jede XML-orientierte Anwendung, in der mehrere Vokabulare definiert und administriert werden sollen), sind diese jedoch nicht Ziel dieses Dokuments, und das Dokument konzentriert sich auf den Anwendungsfall, dass XML-Vokabulare beschrieben werden sollen, die im Rahmen von E-Government Aktivitäten definiert und verwendet werden.

3.3 Vorteile

Der Vorteil einer wohldefinierten und maschinenlesbaren Beschreibung eines XML Namespaces liegt in den folgenden Punkten:

- *Einheitliche Beschreibung:* Eine einheitliche Beschreibung von XML Namespaces macht es den Anwendern von XML Technologien leichter, die Informationen zu finden, die für die Verwendung eines Vokabulars wichtig sind. Vor allem in einem komplexen und nicht streng hierarchisch organisierten Umfeld ist dies von grosser praktischer Bedeutung.
- *Überprüfung der Beschreibung:* Durch die Überprüfung der Beschreibung kann sichergestellt werden, dass die Beschreibung (zumindest syntaktisch) korrekt ist und maschinell ausgewertet werden kann. Diese Überprüfung ist daher eine wichtige Qualitätskontrolle, um zumindest aus formaler Hinsicht sicherzustellen, dass Namespace-Beschreibungen korrekt verwendet werden.
- *Erstellung eines Verzeichnisses:* Ausgehend von der maschinellen Interpretation von Namespace-Beschreibungen ist es möglich, ein Verzeichnis beschriebener Namespaces zu erstellen, in dem zum einen alle relevante Information zu eCH Namespaces gesammelt ist, und in dem darüber hinaus durch Suchfunktionen nach bestehenden Vokabularen gesucht werden kann.

Das vorliegende Dokument soll die Vorteile einer einheitlichen Namespace-Beschreibung verdeutlichen, so dass die Verwendung von einheitlichen Namespace-Beschreibungen aus

eigenem Interesse der Anwender (sowohl der definierenden Instanzen und als auch der Benutzer von XML Schemas) geschieht.

3.4 Schwerpunkte

Das vorliegende Dokument hat einerseits den Schwerpunkt, für eine vordefinierte Menge an Beschreibungsaspekten eines XML Namespaces Kategorien zu definieren, so dass eine Beschreibung eines XML Namespaces anhand dieser Kategorien strukturiert werden kann (Abschnitt 5). Dabei können die Kategorien zum einen als nachträgliche Strukturierungshilfe benutzt werden, sie können aber ebenso bereits beim Erstellen der Beschreibung als Leitlinie dazu dienen, welche Aspekte einer Namespace-Beschreibung von Interesse sind und von daher Bestandteil einer Beschreibung sein sollten.

Der zweite Schwerpunkt des vorliegenden Dokuments liegt in der Definition eines einheitlichen Formates, die Namespace-Beschreibungen verwenden sollten. Dabei gibt es einerseits ein XML-Format, das die Namespace-Beschreibung in vordefinierten XML Elementen ermöglicht (Abschnitt 7.1), und zum anderen ein XHTML-orientiertes Format, das das eigentliche Publishing-Format einer Namespace-Beschreibung ist und durch maschinenlesbare Informationen ergänzt wird (Abschnitt 7.2).

4 Grundlagen von XML Namespaces

Die folgenden Abschnitte behandeln die Grundlagen von XML Namespaces, dies sind eine kurze Erläuterung des Standards selber (Abschnitt 4.1), eine Beschreibung der Verwendung des XML Namespaces Standards im Zusammenhang mit XML Schema (Abschnitt 4.2), und eine Behandlung existierender Ansätze und Konzepte, wie sich die Beschreibung von XML Namespaces realisieren lässt, sowie die Einordnung des eCH Ansatzes (Abschnitt 4.3).

4.1 XML Namespaces

In diesem Abschnitt werden kurz die wichtigsten Grundlagen des XML Namespaces Standards beschrieben, ohne auf XML Schema (näher beschrieben in Abschnitt 4.2) oder die Frage der Organisation von Namespace Namen (näher beschrieben in Abschnitt 4.3) einzugehen.

4.1.1 Versionen der Empfehlung

Die Verwendung eines Namespaces ist sehr eng mit XML verbunden, denn durch das Verwenden von xml ns Attributen und mit Präfix versehenen Namen werden XML Namespace Mechanismen in XML verwendet (die Details sind in den Abschnitten 4.1.3 und 4.1.4 beschrieben), ohne dass man dies in irgendeiner Form vorgängig konfigurieren müsste.

Das heisst, dass es nicht möglich ist, die Version des verwendeten XML Namespace Standards explizit auszuwählen, sondern diese ergibt sich implizit aus der verwendeten XML-Version (die wiederum aus der XML-Deklaration am Beginn des XML-Dokuments ersichtlich ist, z.B. `<?xml version="1.0" ?>`). Da es momentan zwei Versionen der XML Spezifikation gibt, gibt es dementsprechend auch zwei Versionen der XML Namespaces Spezifikation, die sich allerdings nur wenig voneinander unterscheiden:

- XML 1.0 [xml10third]: XML Namespaces 1.0 [xmlns]
- XML 1.1 [xml11]: XML Namespaces 1.1 [xmlns11]

XML 1.0 und XML 1.1 sind inkompatibel (ein reiner XML 1.0 Prozessor weist XML 1.1 Dokumente prinzipiell zurück), so dass sich ausser bei Fällen, in denen die Features von XML 1.1 unbedingt notwendig sind, die Verwendung von XML 1.1 (und damit XML Namespaces 1.1) nicht empfiehlt (siehe dazu auch die Empfehlung zur Verwendung verschiedener XML-Versionen in [eCH-0018]).

4.1.2 Definition von Namespaces

Wie der Name des Standards bereits zeigt, ist ein "Namespace" nichts anderes als ein "Namensraum", also eine Menge von Namen. Diese Namen können z.B. durch ein XML Schema definiert sein, es können aber auch andere Mechanismen verwendet werden wie DTDs oder auch nicht-formale Methoden, so dass die Namen des Namespaces nur durch eine textuelle Beschreibung definiert sind (dies ist z.B. bei XSLT der Fall, für das es weder eine DTD noch ein XML Schema noch irgendein anderes Schema gibt, sondern dessen Vokabular alleine durch eine textuelle Beschreibung der Elemente und Attribute definiert ist).

Diese Trennung des reinen "Namensraumes" und eines u.U. vorhandenen Schemas, das Regeln zur Verwendung und Kombination der Namen definiert, ist ein Kernaspekt der XML

Namespaces. XML Namespaces definieren nur, wie Namen als einem Namespace zugehörig gekennzeichnet (und damit bei der Interpretation erkannt) werden können, alle darüberhinausgehenden Mechanismen (wie bei Schemas z.B. die Validierung) haben mit XML Namespaces nichts mehr zu tun.

Ein Namespace wird also definiert durch eine beliebige Art von Definition, die die Aussage macht, sie definiere einen Namespace. Es gibt keine Festlegung, wie das zu geschehen hat, ob es also z.B. formal passiert oder durch eine textuelle Beschreibung in einem Dokument. Der Namespace Standard beschäftigt sich aus diesem Grund auch nicht mit der eigentlichen Definition von Namespaces, sondern mit deren Deklaration, also der Art und Weise, wie man in einem XML Dokument ausdrückt, dass man Namen aus einem bestimmten Namespace verwendet.

4.1.3 Deklaration von Namespaces

Die Deklaration von XML Namespaces ist einfach, sie geschieht durch xml ns Attribute beliebiger Elemente in XML Dokumenten. Diese Attribute deklarieren einen Namespace für die nachfolgende (d.h. innerhalb dieses Elementes vorgenommene) Verwendung. Ein Namespace wird entweder einem Präfix zugeordnet (xml ns: prefix="uri", wobei prefix ein beliebiger XML Name ohne Doppelpunkte sein darf und uri der Namespace Name ist), oder es ist der Default Namespace (xml ns="uri", wobei uri der Namespace Name ist). Die Wahl der Namens für den Präfix ist von rein lokaler Bedeutung und es können beliebige Präfixe gewählt werden (im Rahmen der Namenskonventionen), es empfiehlt sich aber, aus Gründen der Kompaktheit und Wiedererkennbarkeit Präfixe zu verwenden, die kurz und prägnant sind, die sich u.U. etabliert haben, und die ein einfaches Lesen des XML-Dokuments erlauben.

Deklarierte Namespaces werden auf das Element der Deklaration und darunterliegende (in der Baumsicht des XML-Dokuments) Inhalte angewendet und können durch ihren Präfix für Element- und Attributnamen verwendet werden, wobei anzumerken ist, dass der Default Namespace nur auf Elementnamen angewendet wird, nicht jedoch auf Attributnamen. Neuere XML-Technologien beginnen zudem zunehmend, Namespace Deklarationen auch in Inhalten (also Element-Inhalten oder Attributwerten) zur Interpretation von Namen zu verwenden, Beispiele dafür sind XML Schema und XSLT (wobei in XML Schema unprefixted QNames in Attributwerten als dem Default Namespace zugehörig betrachtet werden, in XSLT dagegen als keinem Namespace zugehörig). In diesen Fällen müssen Applikationen zur Interpretation von QNames in Element-Inhalten oder Attributwerten die an der Stelle ihres Auftretens bestehenden Namespace-Deklarationen kennen und berücksichtigen, was daher in neuen XML-orientierten Technologien gut unterstützt wird, z.B. durch die `in-scope-prefixes` Funktion in XSLT 2.0 oder durch die `lookupNamespaceURI` Methode des Node Interfaces in DOM3.

Neben der Deklaration von Namespaces kann man Deklarationen auch wieder aufheben. In Namespaces 1.0 ist dies ausschliesslich für den Default Namespace erlaubt, während in Namespaces 1.1 die Deklarationen von beliebigen Namespaces aufgehoben werden können. Das Aufheben von Deklarationen erfolgt dadurch, dass die aufzuhebende Deklaration mit einem leeren String als Namespace Name angegeben wird, soll also z.B. die Deklaration des Default Namespace aufgehoben werden, so geschieht dies mit der Attributzuweisung

xml ns="" , und ab dieser Stelle im Dokumentenbaum ist der Default Namespace dann nicht mehr deklariert. Das Aufheben von Deklarationen und vor allem das Wiederverwenden von Präfixen aufgehobener Deklarationen sollte mit Zurückhaltung praktiziert werden, da ein XML Dokument auf diese Weise leicht unübersichtlich werden kann.

4.1.4 Verwendung von Namespaces

Nach der Deklaration in einem Dokument können die deklarierten Namespaces verwendet werden. D.h., dass es keine Verwendung von Namespaces ohne vorherige Deklaration gibt. Der Grund dafür ist, dass die Verwendung eines deklarierten Namespaces immer über den Präfix geschieht (oder durch dessen Abwesenheit, beim Default Namespace), so dass ohne vorgängige Deklaration keine Verwendung möglich ist.

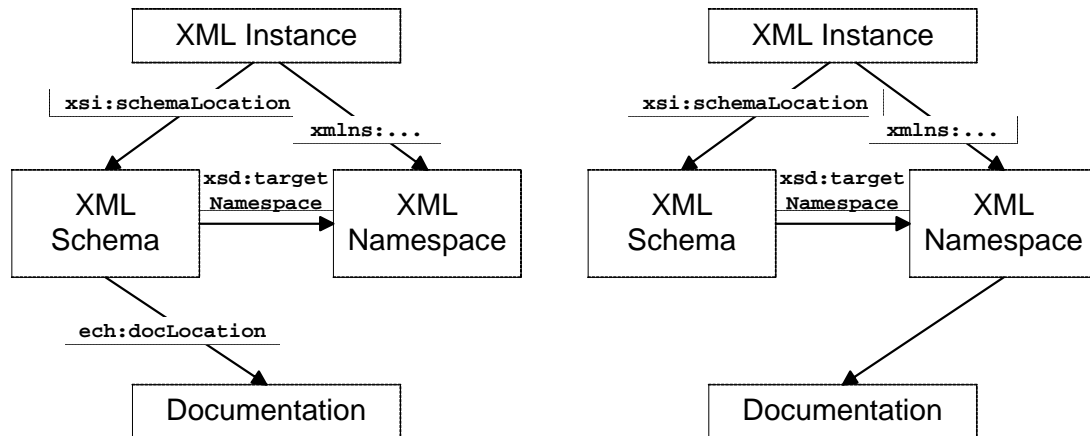
Die Verwendung eines deklarierten Namespaces geschieht durch die Verwendung des definierten Präfixes bzw. keines Präfixes (im Fall des Default Namespaces). Ein Präfix übernimmt also in etwa die Funktion einer abkürzenden Schreibweise, da man anstatt des kompletten Namespace Names nun jeweils nur den Präfix angeben muss, der üblicherweise wesentlich kompakter gewählt wird als der Namespace Name. Die Präfixe werden meistens auf Element- oder Attributnamen verwendet, in manchen Fällen (z.B. XML Schema und XSLT, wie oben erwähnt) jedoch auch in Element-Inhalten oder Attributwerten.

Gibt es einen Default Namespace, so werden ohne Präfix angegebene Namen unterschiedlich behandelt. Elementnamen werden als dem Default Namespace zugehörig interpretiert, während Attributnamen nach wie vor als keinem Namespace zugehörig interpretiert werden. Im Fall von in Element-Inhalten oder Attributwerten vorkommenden Namen ohne Präfix richtet sich die Interpretation nach der verwendeten Technologie, bei XML Schema werden sie als dem Default Namespace zugehörig betrachtet, während sie bei XSLT als keinem Namespace zugehörig betrachtet werden (bei XSLT 2.0 kann diese Interpretation dann durch das `xpath-default-namespace` Attribut auf dem `stylesheet` Element geändert werden).

4.2 XML Schema und XML Namespaces

XML Schema erlaubt es, in der Schema-Definition den mit diesem Schema definierten Namespace anzugeben, dies geschieht mit dem `targetNamespace` Attribut des `schema` Elements. Auf diese Weise kann einem XML Schema angesehen werden, welchen XML Namespace es definiert.

Die Instanz eines XML Schemas muss in jedem Fall diesen XML Namespace verwenden, da die Namen des XML Schemas (Elemente und Attribute) als diesem Namespace zugehörig referenziert werden müssen (andernfalls handelt es sich nicht durch die im XML Schema definierten Namen und die Validierung schlägt fehl). Zudem kann die Instanz auf das Schema Dokument mittels des `schemaLocation` Attributes des `Document` Elements verweisen. Abschnitt 11.1 in [eCH-0018] spricht jedoch als Empfehlung aus, dass XML Schema Instanzen nur das `targetNamespace` Attribut verwenden sollten, nicht jedoch das `schemaLocation` Attribut (der Grund dafür ist, dass das `schemaLocation` Attribut den Ort des Schema Dokuments als URI angibt, während der Namespace Namen eine abstraktere Kennzeichnung



ist, die sich auch nicht ändert, wenn das Schema Dokument an einen anderen Ort verschoben wird, also seine URI ändert).

Wie in obiger Abbildung dargestellt, kann es also zwei Verweise geben, mittels derer eine XML Schema Instanz auf die Herkunft der in ihr verwendeten Namen verweist. Die Frage, die sich grundsätzlich stellt, ist, ob eine zu einem XML Schema gehörige Dokumentation nun, wie in der linken Abbildung gezeigt, über das XML Schema zugänglich gemacht werden soll (z.B. durch ein darin eingebettetes Attribut hier beispielhaft als ech: docLocation angedeutet), oder, wie in der rechten Abbildung gezeigt, über den Namespace.

Obwohl beide vorgestellte Varianten technisch machbar sind, wird für die Namespace-Beschreibung für eCH lediglich die zweite Variante empfohlen. Der Entscheid, die Dokumentation über den Namespace zugänglich zu machen, und nicht über das Schema, hat zwei hauptsächliche Gründe:

- **Lokalisierung:** Da in der Instanz gemäss eCH Empfehlung [eCH-0018] kein Verweis auf das Schema verwendet werden sollte, ist das Schema über die Instanz nicht unmittelbar und direkt über eine URI auffindbar. Über den als targetNamespace deklarierten Namespace ist jedoch der Namespace immer lokalisierbar (und über die mit diesem verbundene Beschreibung dann auch das XML Schema). Dies setzt voraus, dass sich hinter der Namespace URI tatsächlich eine entsprechende Beschreibungs-Ressource verbirgt, was nicht zwingend notwendig ist, aber empfehlenswert.
- **Unabhängigkeit von XML Schema:** Durch die Verbindung der Beschreibung mit dem Namespace ist der im vorliegenden Dokument beschriebene Mechanismus unabhängig von XML Schema, und kann zur Beschreibung beliebiger Namespaces eingesetzt werden, seien diese durch andere Schemasprachen definiert (z.B. eine DTD), oder gänzlich andere Dinge als formale Schemas, z.B. nur informell durch ein textuelles Dokument.

Es wird zudem in der Literatur allgemein empfohlen, hinter Namespace Names nicht direkt Schemas zu hinterlegen, sondern Beschreibungen, die auf Schemas und andere Ressourcen verweisen.

4.3 Beschreibung von XML Namespaces

Das Thema der Beschreibung von Namespaces ist nach wie vor ein offenes Forschungsthema, es gibt keine etablierten Standards, und auch im W3C selber wird zwar anerkannt, dass dies ein relevantes Problem ist (im Dokument [webarch] unter dem Abschnitt "[Namespace Documents](#)"), bisher wird es aber nicht mit viel Engagement angegangen. Es gab allerdings doch schon Initiativen im W3C, die nachstehend beschrieben werden, von denen sich jedoch bisher keine etablieren konnte.

4.3.1 Bestehende Technologien

Die folgenden Namespace-Beschreibungen sind vom W3C veröffentlichte Dokumente, die jedoch keine weit verbreiteten Standards darstellen. Der Erfolg RDDL (Abschnitt 4.3.1.1) ist durch seine Verwendung des unpopulären XLink beeinträchtigt worden, und GRDDL (Abschnitt 4.3.1.2) ist noch zu neu, um einen möglichen Erfolg beurteilen zu können.

4.3.1.1 Resource Directory Description Language (RDDL)

Als ein erster Ansatz zur Beschreibung von XML Namespaces wurde die *Resource Directory Description Language (RDDL)* [rddl] entwickelt. RDDL ist eine sehr einfach definierte und zu verstehende Sprache, die auf der *Extensible Linking Language (XLink)* [xlink] basiert. XLink erlaubt es, Links zu mehr als einer Ressource zu machen (im Gegensatz zum 1:1 Link Modell von HTML), und Links sowie die verlinkten Ressourcen zu typisieren. RDDL definiert eine Reihe an vorgegebenen Link- und Ressourcen-Typen und verlangt, dass die zu einer Namespace-Beschreibung gehörenden Ressourcen auf diese Weise deklariert und in einer HTML-Seite eingebettet werden. Auf diese Weise stellt RDDL sicher, dass eine maschinenlesbare Beschreibung besteht, wie man von der Namespace-Beschreibung zu den den Namespace beschreibenden Ressourcen gelangt.

Wie sich nach der Definition von RDDL herausstellte, war XLink kein erfolgreicher Standard, so dass es wenig verfügbare Software zum Verarbeiten von XLink gab, und daher auf XLink aufbauende Technologien meist wenig erfolgreich waren. Aus diesem Grunde wurde beschlossen, RDDL nicht mehr auf XLink basieren zu lassen.

Im folgenden *Resource Directory Description Language (RDDL) 2.0* [rddl2] Standard wurde daher auf das *Resource Description Framework (RDF)* [rdf] anstatt auf XLink gesetzt, so dass die maschinenlesbare Beschreibung nicht mehr in XLink, sondern in RDF vorliegen muss. Auch diese Version von RDDL geht davon aus, dass diese Information in eine XHTML-Seite eingebettet wird, so dass der Anwender eine solche Seite erstellen muss.

Im Laufe der Arbeiten am (nicht abgeschlossenen) RDDL 2.0 zeigte sich, dass der Grossteil der Benutzer eher selbstdefinierte XML-basierte Namespace-Beschreibungen benutzen wollte (im Gegensatz zu der durch RDDL 2.0 erzwungenen RDF Syntax), von denen im Bedarfsfall dann menschen- und/oder maschinenlesbare Versionen erzeugt werden können. Aus diesem Grund wurde die Entwicklung an RDDL 2.0 eingestellt, und in der Folge an der im folgenden Abschnitt beschriebenen Sprache weitergearbeitet.

4.3.1.2 Gleaning Resource Descriptions from Dialects of Languages (GRDDL)

Die *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)* [grddl] Sprache beschreibt, wie eine Namespace-Beschreibung (die ein beliebiges XML-Dokument sein kann) auf eine Transformation (oftmals implementiert in XSLT) verweist, die dann verwendet

werden kann, um aus dem XML (oft wird es XHTML sein) eine Beschreibung zu extrahieren, die dann im Format des *Resource Description Framework (RDF)* [rdfconcepts] vorliegt.

4.3.2 Gewählter Ansatz

Der für eCH und im vorliegenden Dokument beschriebene Ansatz orientiert sich an GRDDL [wil05e,wil05l] und wählt damit die momentan vielversprechendste Technologie. Unabhängig von Erfolg oder Misserfolg von GRDDL als Standard des W3C bietet der Ansatz den Vorteil, dass sich eine einfache und benutzerfreundliche XML-Beschreibung für XML Namespaces definieren lässt, und diese dann durch den einfachen Einsatz von XML-Technologien in andere Repräsentationen transformiert werden kann, z.B. eine HTML Seite als Web-basierte Beschreibung eines Namespaces, oder ein anderes XML-Format, das sich dazu eignet, ein Verzeichnis aller bei eCH registrierter Namespaces zu führen.

5 Namespace Namen und Versionsangaben

Namespace Namen sind URIs, müssen jedoch nicht zwingend auf eine existierende Ressource verweisen. Um den Umgang mit Namespaces zu vereinfachen, verlangt das vorliegende Dokument jedoch, dass der Namespace Name auf eine Ressource verweisen muss (ausser bei von aussen vorgegebenen Namespace Namen), und regelt zudem, welches Format diese Ressource haben muss (dieses Format ist in Abschnitt 7.2 beschrieben).

Der Name eines Namespaces ist aus Sicht des XML Namespaces Standards aber lediglich irgendeine URI, kann also im Prinzip ein beliebiges URI Scheme (z.B. `http`, `ftp` oder `urn`) verwenden.

ACHTUNG: Die folgenden Abschnitte behandeln den Fall, dass Namespace Namen und Versionierungsstrategie gewählt werden können. Sind Namespace Namen oder Versionierungsstrategie gegeben (weil z.B. ein existierendes Schema verwendet wird, das von einer externen Instanz definiert und gepflegt wird), so kommen die nachfolgenden Betrachtungen nicht zur Anwendung.

5.1 Benennung und Beschreibung des Namespaces

5.1.1 Problemstellung

Die grosse Freiheit bei der Auswahl von Namespace Namen macht es schwierig, eine Auswahl zu treffen, in welcher Form Namespace Namen definiert werden sollten. Damit Namespace Namen im Kontext von eCH einem leicht nachvollziehbaren Schema folgen, wird mit den folgenden Richtlinien die Struktur von Namespace Namen näher geregelt.

5.1.2 Empfehlungen

- **MUST NOT:** Namespace Namen müssen URIs [RFC3986] sein, d.h. insbesondere sind keine IRIs [RFC3987] erlaubt.
- **MUST:** eCH Namespace Namen verwenden das `http` URI Scheme [RFC2616].
- **MUST:** Der Domain Name ist der allgemein verwendete Web Server Name der definierenden Instanz (z.B. `www.ejpd.admin.ch`), mit oder ohne führendes `www`, je nachdem, wie dies beim Web-Server konfiguriert ist und bevorzugt verwendet wird. Auf diesem Server muss die Namespace-Beschreibung zugänglich gemacht werden, es ist also sicherzustellen, dass dies möglich ist.
- **SHOULD:** Der Namespace-Pfad sollte kurz und prägnant sein, idealerweise unter einem gemeinsamen Präfix für alle Namespace Namen, die Empfehlung lautet `xml ns`. Der Grund dafür ist, dass dieser Name über lange Zeit (die Benutzungsdauer der mit dem Namespace verbundenen Schemas und damit auch der Instanzen) gültig bleiben sollte, und für einfache Namen lässt dies einfacher erreichen als für komplizierte Pfade.
- **MUST:** Es muss einen Pfad (idealerweise einen Namen) geben für den Anwendungsfall, z.B. `http://www.ech.ch/xml ns/ech-0033`, der auf eine den allgemeinen Anwendungsfall beschreibende Ressource verweist.

- **SHOULD:** Die Namespace-Beschreibung, die unter der versionsunabhängigen URI `http://www.ech.ch/xmlns/ech-0033` verfügbar ist, sollte auf alle existierenden Versionen verweisen.
- **MAY:** Die obigen Empfehlungen dürfen ignoriert werden, falls der Namespace Name durch eine externe Instanz vorgegeben wird (z.B. einen Standard), aber trotzdem durch eine Namespace-Beschreibung beschrieben werden soll.

5.2 Beschreibung der Versionen

Jede Software und Datenbeschreibung, die über eine gewisse Zeit hinweg existiert und benutzt wird, muss versioniert werden, Dies ist zum einen notwendig, weil u.U. Fehler erkannt werden, die korrigiert werden müssen, oder aus dem Grund, dass die Software oder Datenbeschreibung an neue Anforderungen angepasst werden muss, die sich auf Grund von Änderungen des Umfelds oder der Zielvorstellungen ergeben haben.

5.2.1 Problemstellung

Verschiedene Versionen eines Vokabulars können unterschiedliche Definitionen benutzen, die durch unterschiedliche Schemas beschrieben sind. Ändert sich ein Vokabular im Rahmen der Versionierung in nicht rückwärtskompatibler Weise, so sollte dies auch im Namespace Namen reflektiert werden, um unangemessene Interpretation gemäss einer veralteten Beschreibung zu vermeiden. Aus diesem Grund sollten die Namespace Namen die Versionierung widerspiegeln, sobald es sich um nicht rückwärtskompatible Versionierungen handelt (diese werden als Major Versions bezeichnet).

5.2.2 Empfehlungen

- **MUST:** Major und Minor Versions müssen jeweils mit ganzen positiven Zahlen nummeriert und diese in aufsteigender Reihenfolge vergeben werden. Führende Nullen (ausser für die 0 selber) und Vorzeichen sind nicht zulässig.
- **MUST:** Es muss auf den allgemeinen Pfad aufbauende Pfade für konkrete Versionen des Schemas geben, z.B. `http://www.ech.ch/xmlns/ech-0033/1` auf die Beschreibung der ersten Version des Schemas.
- **MUST:** Die Benennung der Major Version muss dem XML Schema Typen `xs:token` gehorchen, darf aber keine Schrägstriche enthalten.
- **MUST:** Der gleiche Namespace Name darf nur für kompatible Schemas verwendet werden, d.h. für Schemas, bei denen die Validierung von Instanzen mit allen Versionen des Schemas erfolgreich möglich ist. Inkompatible Änderungen (d.h. neue Major Versions) müssen eine Versionierung des Namespace Namens zur Folge haben.
- **MUST:** Rückwärtskompatible Applikationen müssen die Richtlinien der Namespace Name Versionierung respektieren und Dokumente mit älteren Namespace Namen akzeptieren.

- **MAY:** Die obigen Empfehlungen dürfen ignoriert werden, falls der Namespace Name durch eine externe Instanz vorgegeben wird (z.B. einen Standard), aber trotzdem durch eine Namespace-Beschreibung beschrieben werden soll.
- **SHOULD:** Ausser in begründeten Ausnahmefällen sollte immer mit Minor Versions gearbeitet werden. Das erste Release einer Major Version sollte auch gleich immer einer Minor Version Nummer zugeordnet werden (wie in Abschnitt 6.4.9 beschrieben verweist die Major Version Beschreibung dann auf kein Schema, sondern dieser Verweis findet sich lediglich in der Beschreibung der ersten Minor Version).

5.3 Aufbau eines Namespace Names mit Versionierung

Das Resultat der in den Abschnitten 5.1 und 5.2 aufgeführten Festlegungen ist ein Namespace Name, der schematisch betrachtet aus den folgenden Komponenten besteht (die nicht kommentierten Komponenten sind dabei Syntax, die zur Trennung der Komponenten verwendet wird):

http	://	www.ech.ch	/	xml ns	/	ech-0033	/	1
URI Scheme		Domain Name		allgemeiner Name für Namespace-Beschreibungen		Anwendungsfall		Major Version

Wie in Abschnitt 6.1 beschrieben, gibt es für die Beschreibung von Namespaces die Unterscheidung zwischen der Namespace Root und konkreten Versionen. Aus der Sicht des Namensaufbaus handelt es sich bei der Namespace Root um den Teil des Namespaces Names bis zum Anwendungsfall (ohne den folgenden Schrägstrich), und bei den Namespace Names der Versionen um die Namespace Names inklusive des trennenden Schrägstrichs und der Versionsidentifikation.

Die Minor Version eines Schemas wird nicht im Namespace Namen festgehalten, da dieser nur die Major Version Informationen enthält. Die Minor Version eines Schemas wird im minorVersion Attribut einer Instanz angegeben (dieses Attribut ist als Hilfskomponente für die Definition von XML Schemas in [eCH-0050] definiert und beschrieben), und ist des weiteren im version Attribut des Schemas anzugeben (die Major Version findet sich bereits in dessen targetNamespace), diese Markierung des Schemas ist in [eCH-0035] näher beschrieben. Durch diese beiden Angaben lässt sich sowohl im Schema als auch in Instanzen feststellen, zu welcher Minor Version eines Schemas eine Instanz und eine Schema gehören.

Da die Minor Version nicht im Namespace Namen festgehalten wird, kann es auch keine Namespace URI geben, unter der die Beschreibung der Minor Version verfügbar sein muss. Damit diese Beschreibungen aber auch möglichst einfach auffindbar sind, müssen für Beschreibungen von Minor Versions eines Schemas die folgenden Richtlinien eingehalten werden:

- **MUST:** Für die Beschreibung der Minor Version eines Schemas muss die Beschreibung der Major Version existieren.
- **MUST:** Die Beschreibung der Minor Version eines Schemas muss unter der URI verfügbar sein, die sich aus der Konkatenation der Namespace URI des Schemas (diese enthält die Major Version), eines Schrägstriches, und der Minor Version ergibt. Dies ist die URI, unter der die Beschreibung der Minor Version zugänglich sein muss, die Namespace URI des durch die Minor Version beschriebenen Namespaces ist nach wie vor der Namespace Name der Major Version (ohne Schrägstriches und Minor Version)!

Beispiel der Minor Version Beschreibung: `http://www.ech.ch/xmlns/ech-0033/1/0`

Beispiel des beschriebenen Namespaces: `http://www.ech.ch/xmlns/ech-0033/1`

6 Aufbau von Namespace-Beschreibungen

Die in den folgenden Abschnitten beschriebene Struktur und Aspekte einer Namespace-Beschreibung gehören zu den wiederkehrenden Beschreibungsaspekten, die bei vielen Namespaces nützlich sind. Aus diesem Grund werden sie hier explizit aufgeführt und beschrieben, und es stehen fest definierte Konstrukte zur Verfügung, mit denen sie repräsentiert werden können (das genaue Format dieser Konstrukte ist in Kapitel 7 beschrieben). Gibt es zu einem Namespace eine derartige Beschreibung, so sollte sie auch explizit als solche gekennzeichnet und entsprechend repräsentiert werden, wie in Kapitel 7 beschrieben.

Es kann natürlich zu einem Namespace noch weitere Beschreibungsaspekte geben, die nicht in der folgenden Liste definierter Beschreibungsaspekte aufgeführt sind, aber dennoch nützlich oder sogar notwendig sind für eine Beschreibung. Die in Abschnitt 6.2 beschriebene Kategorie ermöglicht in diesem Fall ohne weiteres, auch eine solche Beschreibung aufzunehmen. Deren Bedeutung ist dann zwar nicht formal definiert, zumindest besteht jedoch eine konzeptionelle und maschinenlesbare Verbindung zwischen der Namespace-Beschreibung und dem zusätzlichen Beschreibungsaspekt.

Wie bereits im vorangegangenen Abschnitt erwähnt, muss unterschieden werden zwischen im durch eCH vorgegebenen Rahmen definierten Namespaces, die den Benennungs- und Versionierungsregeln folgen sollten, und ausserhalb des durch eCH vorgegebenen Rahmen definierten Namespaces, die diesen Regeln höchstwahrscheinlich nicht folgen. Beide Arten von Namespaces können durch die hier definierten Namespace-Beschreibungen definiert werden, jedoch auf etwas unterschiedliche Weise. Abschnitt 6.1 bezieht sich auf im durch eCH vorgegebenen Rahmen definierte Namespaces und definiert aus diesem Grund auch gemäss der Namespace-Struktur strukturierte Beschreibungen. Abschnitt 6.2 dagegen bezieht sich auf ausserhalb des durch eCH vorgegebenen Rahmens definierte Namespaces und definiert daher die Beschreibung unstrukturierter Namespaces.

Abschnitt 6.3 beschreibt allgemein die interne Struktur einer Namespace-Beschreibung, und Abschnitt 6.4 schliesslich geht detailliert auf die einzelnen Aspekte ein, die in einer Namespace-Beschreibung enthalten sein können.

6.1 Strukturierte Namespace-Beschreibungen

Die folgende Betrachtung bezieht sich auf Namespace-Beschreibungen, die im Rahmen der in den Abschnitten 5.1 und 5.2 aufgestellten Empfehlungen verwendet werden. Handelt es sich bei dem zu beschreibenden Namespace um einen ausserhalb dieses Rahmens definierten Namespace, so ist Abschnitt 6.2 anzuwenden.

Gesamthaft betrachtet kann eine komplette Namespace-Beschreibung bestehen aus einer Beschreibung der Namespace Root, Beschreibungen von Major Versions, und Beschreibungen von Minor Versions. Dabei sind nicht alle diese Teile verpflichtend, einzig die Beschreibung der Namespace Root muss immer existieren. Die verschiedenen Arten von Namespace-Beschreibungen sind wie folgt definiert:

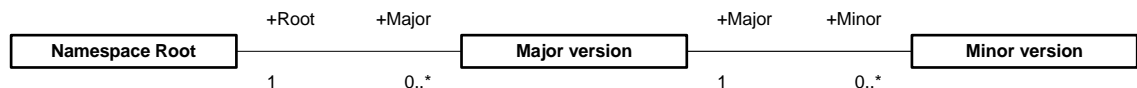
- *Namespace Root*: Dies ist die Beschreibung des Szenarios, für das dieser Namespace und damit verbundene Schemas gedacht sind. Bei der Namespace Root handelt es sich noch nicht um die Beschreibung eines konkreten Schemas, sondern um den

Kontext, innerhalb dessen dann ein oder mehrere Schema-Versionen (Major Version mit u.U. dazugehörigen Minor Versions) definiert sind. Man kann die Namespace-Root Beschreibung von daher auch als eine Beschreibung anderer Namespaces betrachten, nämlich der Major Version Namespaces.

- *Major Version:* Während die Namespace Root ein Szenario beschreibt, sind Major (wie auch Minor) Version die Beschreibung konkreter Vokabulare, die innerhalb dieses Szenarios definiert werden. Eine Major Version ist demnach einem Szenario zugeordnet, und sobald sich ein Vokabular in inkompatibler Weise verändert, muss eine neue Major Version des Vokabulars erstellt werden (siehe Abschnitt 5.2 zur Versionierung)
- *Minor Version:* Bei nicht inkompatiblen Änderungen eines Vokabulars kann eine neue Minor Version erstellt werden. Minor Versions sind generell Major Versions zugeordnet. Eine Minor Version kann von einer Namespace-Beschreibung beschrieben werden, sie hat jedoch keinen eignen zugeordneten Namespace Namen (gemäß Abschnitt 5.2 zur Versionierung wird der Namespace Name mit der Major Version gebildet, die Minor Version dagegen findet keinen Eingang in den Namespace Name). Anwendungen sind natürlich frei, die Minor Version über einen anderen Mechanismus zu markieren, z.B. durch eine Kennzeichnung der Minor Version innerhalb des Vokabulars (meistens wird dies durch ein Attribut geschehen wie das in [eCH-0050] vorgeschlagene Attribut zur Markierung von Minor Versions).

Diese drei Teile stehen in Beziehungen zueinander. Jeder Namespace muss durch genau eine Namespace Root beschrieben werden. Die Namespace Root selber beschreibt keine Namen eines konkreten XML-Vokabulars (dies machen die untergeordneten Major und Minor Version Beschreibungen), aber sie lässt sich als Beschreibung der Namespace Namen der Major Versions verstehen.

Der Namespace Root sind beliebig viele Major Versions zugeordnet, die sich durch die Versionsangabe der Major Version unterscheiden müssen. Jeder Major Version sind beliebig viele Minor Versions zugeordnet, die die gleiche Major Version haben müssen wie die Major Version, der sie zugeordnet sind, und die sich durch ihre Versionsangabe der Minor Version unterscheiden müssen. Diese Zusammenhänge lassen sich wie folgt darstellen:



Dabei bedeuten die jeweils rechts stehenden Beziehungen zwischen den verschiedenen Arten von Beschreibungen, dass es jeweils 0 bis beliebig viele zugeordnete Beschreibungen geben kann.

Bei allen drei Arten von Beschreibungen handelt es sich um Namespace-Beschreibungen, sie weisen daher eine Reihe von Gemeinsamkeiten auf. Jede der drei Arten von Beschreibungen erlaubt jedoch eine andere Kombination von Beschreibungsaspektes (die im Detail in Abschnitt 6.4 beschrieben sind).

6.2 Unstrukturierte Namespace-Beschreibungen

Handelt es sich bei den zu beschreibenden Namespaces um solche, die nicht gemäss den in den Abschnitten 5.1 und 5.2 aufgestellten Empfehlungen benannt und strukturiert sind, so existiert keine wie im vorangegangenen Abschnitt vorgestellte Struktur, sondern der Namespace wurde nach externen Vorgaben definiert und benannt und soll lediglich im Rahmen von eCH beschrieben werden. Aus diesem Grund handelt es sich in diesem Fall um eine andere Art von Beschreibung, die im Gegensatz zu den vorangegangenen Beschreibungen, die zueinander in Beziehung stehen (Namespace Root, Major Version, Minor Version), ohne Beziehungen zu anderen Namespace-Beschreibungen existiert. Aus diesem Grund wird diese Art der Namespace-Beschreibung als ein *Simple Namespace* bezeichnet.

6.3 Interne Struktur einer Beschreibung

Eine Namespace-Beschreibung kann gemäss der beiden vorangegangenen Abschnitte eine von vier möglichen Formen annehmen, und zwar Namespace Root, Major Version, Minor Version, oder Simple Namespace. Jede dieser Beschreibungen ist ähnlich aufgebaut, es handelt sich jeweils um einen Container, der eine Anzahl erlaubter Beschreibungsaspekte enthalten kann. Das genaue Format ist in Abschnitt 7 beschrieben, es handelt sich dabei um ein XML-basiertes Format.

Im folgenden soll kurz auf die verschiedenen Arten von Beschreibungsaspekten eingegangen werden, bevor sie in Abschnitt 6.4 aufgelistet werden.

6.3.1 Beschreibungsaspekte

Beschreibungsaspekte einer Namespace-Beschreibung sind diejenigen Aspekte, für die es (im einzelnen in Abschnitt 6.4 beschriebene) definierte Kategorien gibt. Ein typisches Beispiel eines solchen Beschreibungsaspektes ist das Datum, das beschreibt, wann eine Namespace-Beschreibung erstellt wurde.

Bevor die definierten Beschreibungsaspektes in Abschnitt 6.4 beschrieben werden, soll kurz auf die möglichen Typen (d.h. erlaubten Wertemengen) von Beschreibungsaspekten eingegangen werden, diese sind in Abschnitt 6.3.2 aufgelistet. Die Idee dieser Typen ist, dass es verschiedene Arten von Beschreibungsaspekten gibt.

6.3.2 Typen von Beschreibungsaspekten

eCH Namespace-Beschreibungen unterstützen verschiedene Beschreibungsaspekte, die unterschiedliche Typen haben können. Es werden dabei sechs Typen unterschieden, Token, Strings, Dates, Numbers, Texte mit Sprachangabe, und URIs als Verweise auf Web Ressourcen. In einigen Fällen wird eine Sprachmarkierung unterstützt, so dass verschiedene sprachige Versionen angegeben werden können.. Werden Sprachmarkierungen verwendet, so benutzen diese Sprachkennzeichnungen gemäss [RFC3066].

Alle unterstützten Datentypen können bei den konkreten Beschreibungsaspekten als wiederholbar markiert werden, in diesem Fall können mehrere Werte angegeben werden. Die Wiederholbarkeit ist in der Abbildung in Abschnitt 6.4 mit für die einzelnen Beschreibungsaspekte markiert. Ist angegeben, dass für diesen Beschreibungsaspekt mehrere Werte des jeweili-

gen Typs erlaubt sind, so kann der entsprechende Aspekt mehrfach auftreten, z.B. können verschiedene Tools angegeben werden.

6.3.2.1 Token

Token sind kurze Zeichenketten ohne Whitespace Zeichen (Space, Tab, NL, CR), sie können (abgesehen von Whitespace) alle Zeichen des Unicode Character Repertoires enthalten, es ist jedoch empfehlenswert, diese Angaben auf ASCII zu beschränken. Sie benutzen den `xs: token` Datentyp von XML Schema.

6.3.2.2 String

Strings sind Zeichenketten beliebiger Länge, sie können alle Zeichen des Unicode Character Repertoires enthalten. Sie benutzen den `xs: string` Datentyp von XML Schema.

6.3.2.3 Date

Dates sind Daten gemäss [ISO8601], dieser Standard beschreibt Daten in der bekannten YYYY-MM-DD Notation. Im XML Schema wird der `xs: date` Typ verwendet, wobei dieser zwar zusätzlich noch die Angabe einer Zeitzone erlaubt, diese Angabe wird jedoch für die hier beschriebene Verwendung nicht gestattet.

6.3.2.4 Number

Numbers sind nicht-negative ganze Zahlen, die keine führenden Nullen (ausser für die Null selber) und kein Leerzeichen aufweisen dürfen. Sie benutzen zu diesem Zweck den `xs: nonNegativeInteger` Datentyp von XML Schema mit zusätzlichen lexikalischen Einschränkungen.

6.3.2.5 Texts

Texts sind Mengen von Zeichenketten beliebiger Länge, die Zeichenketten können alle Zeichen des Unicode Character Repertoires enthalten. Texts kann mehrere Texte enthalten, die einzelnen Texte können mit einer Sprachmarkierung versehen werden (durch das `xml: lang` Attribut), diese sollte immer dann verwendet werden, wenn es sich bei dem Inhalt um mehrere sprachabhängige Texte handelt. Wird sprachabhängiger Text verwendet, so sollte dieser idealerweise auch in mehreren Sprachen angegeben werden, dies ist durch die Definition von Texts erlaubt und erwünscht.

6.3.2.6 URI

URIs sind Verweise auf Ressourcen, die über eine in der URI codierte Zugriffsmethode zugreifbar sind (HTTP und FTP sind die beiden populärsten Zugriffsmethoden). Bei den Ressourcen handelt es sich um Ressourcen beliebiger Medientypen, und erst durch den Zugriff auf die Ressource ist es möglich, den Medientyp festzustellen (das Interpretieren der Endung einer URI ist nicht zuverlässig, zudem gibt es viele URIs, die keine solche interpretierbare Endung aufweisen). Dabei wird es sich häufig um weitverbreitete Medientypen handeln, also z.B. HTML oder PDF Dokumente. Um den Zugriff auf und den automatisierten Umgang mit diesen Ressourcen zu erleichtern, sollten die Beschreibungen in einem plattformunabhängigen Format zur Verfügung gestellt werden (plain Text, HTML oder PDF). Um bei Ressourcen, die sprachabhängig sind, markieren zu können, in welcher Sprache diese Ressour-

cen vorliegen, können Sprachmarkierungen verwendet werden, die die für die Ressource massgebliche Sprache angeben.

Um die mittels URI identifizierten Ressourcen beschreiben zu können, kann eine URI optional mit einer Beschreibung versehen werden, die in einem kurzen Text beschreibt, worauf die URI verweist. Da diese Beschreibung der Ressource sprachabhängig ist, kann sie (analog den in Abschnitt 6.3.2.3 beschriebenen Texten) mit einer Sprachmarkierung versehen werden. Die Sprachmarkierung einer URI bezieht damit immer auf die identifizierte Ressource, während die Sprachmarkierung der die URI beschreibenden Texte die in der Beschreibung verwendete Sprache markiert.

URIs werden von den Beschreibungsaspekten in drei verschiedenen Arten verwendet, es handelt sich bei allen diesen Arten um URIs (die als Syntax den Typ `xs:anyURI` verwenden), jedoch unterscheiden sie sich in der Art der Ressource, auf die mit ihnen verwiesen werden sollte:

- *URI*: Dies ist eine normale URI die auf eine Ressource zeigt, die menschen- oder maschinenlesbare Daten enthalten kann. Es kann eine Sprachmarkierung geben.
- *URI-human*: In diesem Fall soll die URI auf eine Ressource zeigen, die menschenlesbare Informationen enthält. Es kann eine Sprachmarkierung geben.
- *URI-machine*: In diesem Fall soll die URI auf eine Ressource zeigen, die maschinenlesbare bzw. automatisch weiterverarbeitbare Informationen enthält. Es kann eine Sprachmarkierung geben, jedoch wird sich dies die Ausnahme sein, da die meisten maschinenlesbaren Ressourcen sprachunabhängig sein werden.

Im XML Schema wird bei den Wertemengen dieser drei URI Typen nicht unterschieden, welche Art von URI erwartet wird, aus der rein syntaktisch orientierten XML Schema Sicht handelt es sich in allen Angaben von URIs um Werte mit dem Typ `xs:anyURI`.

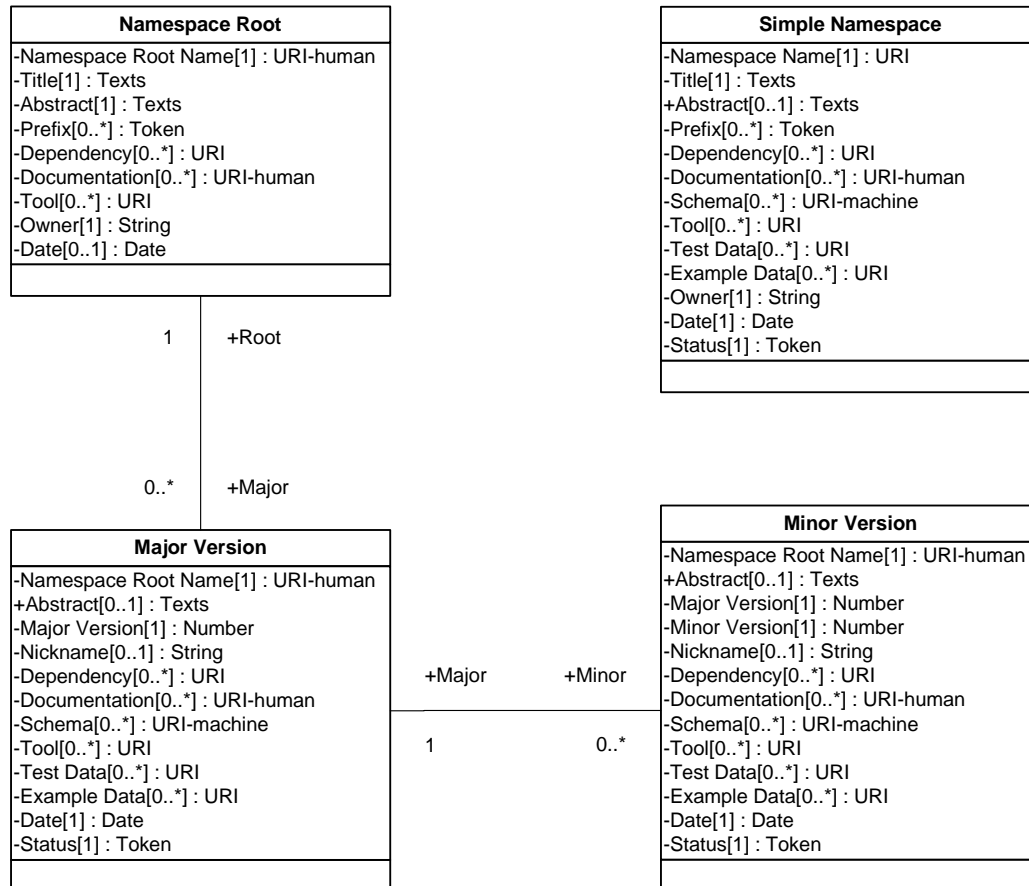
- **SHOULD**: URIs sollten auf so informative Ressourcen wie möglich zeigen. Gibt es z.B. eine Übersichtsseite eines Dokuments und dort Versionen in verschiedenen Sprachen, so sollte die URI auf die Übersichtsseite zeigen. Ist die durch die URI zu identifizierende Ressource durch einen XML Namespace beschrieben, sollte die URI den Namespace Name angeben und nicht die URI des Schemas oder der Dokumentation.

6.4 Unterstützte Beschreibungsaspekte

Um die in den folgenden Abschnitten aufgelisteten Beschreibungsaspekte zu illustrieren, wird als Beispiel jeweils angegeben, was für den entsprechenden Beschreibungsaspekt für die Beschreibung des XML Schemas angegeben wird, das in Abschnitt 7.1 definiert wird. Bei diesem Schema handelt es sich um das XML Schema für Namespace-Beschreibungen, die hier aufgeführten Beispiele zeigen also somit, wie sich der Namespace des eCH XML Schemas für eCH Namespace-Beschreibungen mit seinen eigenen Mechanismen beschreiben lässt.

Die folgende Abbildung definiert, welche Beschreibungsaspekte für welche Art von Namespace-Beschreibung (wie in den Abschnitten 6.1 und 6.2 beschrieben) verwendet werden kann.

Aus der Abbildung ist ersichtlich, dass eine Namespace Root Beschreibung etwas andere Beschreibungsaspekte zulässt als die Major und Minor Version Beschreibungen und als die Simple Namespace Beschreibung. Major und Minor Version Beschreibung unterscheiden sich nur dahingehend, dass bei einer Major Version Beschreibung die bei der Minor Version Beschreibung zwingend vorgeschriebene Angabe der Minor Version nicht erlaubt ist. Liegen



Beschreibungen verschiedener Beschreibungsaspekte mehrfach vor (z.B. Abstracts für die Namespace Root Beschreibung, eine dazugehörige Major Version Beschreibung, und eine dazugehörige Minor Version Beschreibung), so werden alle die logisch übergeordneten Beschreibungen ebenfalls als zutreffend interpretiert, in diesem Fall gäbe es für die Minor Version also Abstracts aus der Root, der Major Version, und der Minor Version selber.

6.4.1 Namespace Root Name

Dies ist der Root Name des beschriebenen Namespaces (d.h. ohne Versionsangabe und ohne Schrägstrich am Ende, wie in Abschnitt 5.1 beschrieben), gemäss XML Namespaces muss (**MUST**) es sich dabei um eine URI [RFC3986] handeln. Der Namespace Root Name ist nur anwendbar auf strukturierte Beschreibungen gemäss Abschnitt 6.1, bei unstrukturierten Namespace-Beschreibungen (Simple Namespace) kommt der in Abschnitt 6.4.2 beschriebene Namespace Name zur Anwendung.

- **MUST:** Wird der Beschreibungsaspekt des Namespace Root Name in einer Major Version Beschreibung verwendet, so muss eine Namespace Root Beschreibung mit dem gleichen Namespace Root Name existieren (andernfalls kann die Major Version Beschreibung keiner Namespace Root Beschreibung zugeordnet werden und ist daher fehlerhaft).
- **MUST:** Wird der Beschreibungsaspekt des Namespace Root Name in einer Minor Version Beschreibung verwendet, so muss eine Major Version Beschreibung mit dem gleichen Namespace Root Name existieren (andernfalls kann die Minor Version Beschreibung keiner Major Version Beschreibung zugeordnet werden und ist daher fehlerhaft).
- **SHOULD:** Die Beschreibung des XML Namespaces sollte unter dieser URI verfügbar sein (in dem in Abschnitt 7.2 beschriebenen Format), dies kann jedoch nicht immer garantiert werden, z.B. bei Namespaces, deren Namespace Name vorgegeben ist und eine ausserhalb des eigenen Verantwortungsbereiches liegende Domain benutzt.

Beispiel: "http://www.ech.ch/xmlns/ech-0033/1"

6.4.2 Namespace Name

Dies ist der Name des beschriebenen Namespaces (dieser Beschreibungsaspekt kann nur für Simple Namespaces verwendet werden), gemäss XML Namespaces muss (**MUST**) es sich dabei um eine URI [RFC3986] handeln. Der Namespace Name ist nur anwendbar auf unstrukturierte Beschreibungen gemäss Abschnitt 6.2, bei strukturierten Namespace-Beschreibungen kommt der in Abschnitt 6.4.1 beschriebene Namespace Root Name zur Anwendung.

Beispiel: "http://www.w3.org/XML/1998/namespace"

6.4.3 Title

Hierbei handelt es sich um den Titel des definierten Vokabulars, es sollte (**SHOULD**) ein kurzer und beschreibender Name sein, mit dem das Vokabular üblicherweise bezeichnet wird. Falls notwendig oder gewünscht, kann der Name mehrsprachig angegeben werden, in diesem Fall muss (**MUST**) die Sprachmarkierung verwendet werden.

Beispiel: "eCH Namespace description"

6.4.4 Abstract

Der Abstract ist ein kurzer beschreibender Text, der in wenigen Sätzen die Verwendung des Vokabulars beschreibt. Dieser Text sollte auch für jemanden verständlich sein, der sich noch nicht detailliert mit dem Anwendungsgebiet und dem Vokabular beschäftigt hat. Idealerweise sollte (**SHOULD**) der Abstract mehrsprachig vorliegen, in diesem Fall müssen (**MUST**) die Versionen in den verschiedenen Sprachen jeweils mit einer Sprachmarkierung versehen werden.

Beispiel: "This is the 'eCH Namespace description' of the 'eCH Namespace description', in other words, the 'eCH Namespace description' describing itself. The vocabulary of this namespace is used to describe XML Namespaces, defining a number of predefined descrip-

tion types. Namespace description can then be transformed to XHTML with embedded machine-readable information and subsequently published, read, harvested, and used for compiling namespace description directories."

6.4.5 Prefix

Namespaces Präfixe (wie in den Abschnitten 4.1.3 und 4.1.4 beschrieben) haben rein lokale Bedeutung, d.h. sie können im Prinzip in jedem XML Dokument frei gewählt werden. Trotzdem ist es für Benutzer einfacher und sehr hilfreich, wenn das gleiche Vokabular in XML Dokumenten immer mit dem gleichen Namespace Präfix verwendet wird. Aus diesem Grund sollte (**SHOULD**) die Namespace-Beschreibung einen Präfix empfehlen, der nur dann nicht verwendet werden sollte, wenn schwerwiegende Gründe dagegen sprechen, z.B. ein Namenskonflikt mit einem Präfix eines anderen verwendeten Vokabulars. Aus diesem Grund können auch mehrere Präfixe angegeben werden, um im Fall von Namenskonflikten einen alternativen Präfix zur Auswahl zu stellen. Da der Präfix in XML Dokumenten eingesetzt werden soll, muss er dem XML Schema Typ `xs:NCName` (*non-colonized name*, XML Namen ohne Doppelpunkt) gehorchen.

Beispiel: "echnsd"

6.4.6 Versionierung

Abschnitte 8.2 und 11.2 von [eCH-0018] und Abschnitt 5.2 des vorliegenden Dokuments empfehlen, dass die Versionierung von Schemas nur bei Major Versionen den Namespace Namen ändern soll. Minor Versionen müssen (**MUST**) den gleichen Namespace Namen wie die zugrundeliegende Major Version verwenden, und sollten (**SHOULD**) die Minor Version des Schemas im Schema selber und über ein Attribut auch in den Dokumenten signalisieren. Aus diesem Grunde gilt das gleiche auch für die Namespace-Beschreibungen, die demnach für Major Versionen andere sind (da sie unter einem anderen Namespace Namen verfügbar sind), für Minor Versions jedoch dem gleichen Namespace Name zugeordnet sind (ein Namespace Name für alle Minor Versionen einer Major Version eines Schemas).

Für gemäss Abschnitt 6.1 strukturierte Namespace-Beschreibungen ergibt sich der effektive Namespace Name für Major und Minor Version Beschreibungen aus dem in Abschnitt 6.4.1 beschriebenen Namespace Root Name, einem Schrägstrich als Separator, und dem daran angefügten Major Version Wert.

6.4.6.1 Major Version

Dieser Aspekt bezeichnet die Major Version des beschriebenen Vokabulars. Diese Versionsangabe wird zur Bildung des Namespace Name der Major Version Beschreibung mit einem Schrägstrich getrennt an den in Abschnitt 6.4.1 beschriebenen Namespace Root Name angehängt. Unter der resultierenden URI sollte die Major Version Beschreibung verfügbar sein, denn dies ist der effektive Namespace Name des beschriebenen Namespaces.

Wird die Major Version Nummer in einer Minor Version Beschreibung verwendet, so muss eine Major Version Beschreibung mit dieser Nummer existieren (andernfalls kann die Minor Version Beschreibung keiner Major Version Beschreibung zugeordnet werden und ist daher fehlerhaft).

Beispiel: "1"

6.4.6.2 Minor Version

Dieser Aspekt bezeichnet die Minor Version(en) des beschriebenen Vokabulars. Diese Minor Versionen können, falls nötig, benutzt werden, um andere Beschreibungsaspekte als spezifisch für bestimmte Minor Versionen zu qualifizieren. Die Minor Version wird nicht im Namespace Name verwendet, sie dient alleine der Markierung der Minor Version Beschreibung, und sollte (**SHOULD**), falls durch das Vokabular vorgesehen, als Angabe in Schemas und/oder Dokumenten erscheinen (wie in [eCH-0050] beschrieben), dies wird jedoch nicht im Namespace Name ersichtlich, der lediglich die Major Version identifiziert.

Im Fall von XML Schema muss (**MUST**) die Minor Version (und nur diese) im version Attribut des Schemas angegeben werden. Die Major Version ist dort bereits über den target-namespace des Schemas ersichtlich.

Beispiel: "0"

6.4.6.3 Nickname

Gemäss Abschnitt 5.2 müssen Major und Minor Version mit ganzen Zahlen numeriert werden. Um die in manchen Bereichen oder Projekten anderen Versionierungsbenennungen ebenfalls in der Beschreibung festhalten zu können, kann im Nickname der "Name" einer Major oder Minor Version festgehalten werden, falls dieser anderen Richtlinien folgt als der von eCH vorgegebenen numerierenden Versionierungsbenennung.

Beispiel: "Vista"

6.4.7 Dependency

Abhängigkeiten halten fest, inwiefern das beschriebene Vokabular von anderen Vokabularen oder Standards abhängig ist. Dies können einerseits frühere Major Versionen des Vokabulars sein, oder andere Standards oder Technologien, auf denen das Vokabular aufbaut.

Beispiel: "http://www.ech.ch/xmlns/ech-0033-rdf/1" "The 'eCH directory information' depends on the 'RDF for eCH Namespace description' schema, because this schema defines the description facets which are simply reformulated as XML in the 'eCH directory information' schema."

6.4.8 Documentation

Die Namespace-Beschreibung eines Vokabulars stellt keine umfassende Dokumentation dar, sondern ist eine kurze und maschinenlesbare Zusammenfassung der wichtigsten Aspekte einer Namespace-Beschreibung. Die eigentliche Dokumentation liegt in der Regel in einem umfassenderen Dokument vor, dessen URI als Verweis auf die Dokumentation angegeben wird.

Beispiel: "http://dret.net/netdret/publications#wil05h"

6.4.9 Schema

Viele durch eine Namespace-Beschreibung beschriebene Vokabulare sind durch ein Schema beschrieben. Vokabulare können allerdings auch durch kein Schema oder durch mehrere (einander ergänzende) Schemas beschrieben sein.

Im Normalfall wird es ein Schema geben, und oft wird es sich dabei um ein XML Schema [xmlschema1sec,xmlschema2sec] handeln. Dabei ist zu beachten, dass XML Schema unterscheidet zwischen dem Schema, das die logische Einheit ist, und Schema-Dokumenten, die einzelne Files sind aber u.U. Referenzen auf andere Schema-Dokumente enthalten, die aufgelöst werden müssen, um das Schema als ganzes interpretieren zu können. Im Fall von XML Schema verweist dieser Beschreibungsaspekt also auf ein (gemäss XML Schema Terminologie) Schema-Dokument.

Wenn es (wie in Abschnitt 5.2 empfohlen) zu Major Versions dazugehörige Minor Versions gibt, so darf die Namespace-Beschreibung der Major Version auf kein Schema verweisen (**MUST NOT**).

Beispiel: "http://www.ech.ch/xmlns/ech-0033/1/ech-0033-1-0.xsd"

6.4.10 Tool

Es kann verschiedene Werkzeuge oder Programme geben, mit Hilfe derer der Umgang mit dem Vokabular möglich wird. Dabei kann es sich um Programme zur Überprüfung handeln, um Programme zum Erstellen von Daten, oder um Programme, mit denen Daten des Vokabulars in irgendeiner Form verarbeitet werden können.

Beispiel: "http://www.ech.ch/xmlns/ech-0033/1/ech-0033-1-0-2grddl.xslt"

6.4.11 Test Data

Bei komplizierteren Vokabularen ist es oft hilfreich, Daten zur Verfügung zu haben, die als Testgrundlage dienen können. Idealerweise sind diese Daten in einer Form gestaltet, dass sie möglichst vielfältig die Möglichkeiten des Vokabulars ausnutzen. Üblicherweise wird es sich bei Testdaten um einzelne Files oder Archive handeln, anhand derer der Umgang mit dem Vokabular getestet werden kann. Damit dies leichter fällt, sollten die Testdaten gut dokumentiert sein.

Beispiel: "http://www.ech.ch/xmlns/ech-0033/1", denn in diesem Fall kann die Namespace-Beschreibung sich selbst als Testdaten angeben, da sie sich selbst beschreibt und alle Beschreibungsaspekte verwendet.

6.4.12 Example Data

Der Einstieg die Verwendung eines Vokabulars fällt oft wesentlich leichter, wenn anhand vorgegebener Beispiele nachvollzogen werden kann, wie dieses Vokabular verwendet werden sollte. Üblicherweise wird es sich bei Beispielen um einzelne Files oder Archive handeln, anhand derer das Vokabular erlernt werden kann. Damit dies leichter fällt, sollten die Beispiele gut dokumentiert sein.

Beispiel: "http://www.ech.ch/xmlns/ech-0033/1", denn in diesem Fall kann die Namespace-Beschreibung sich selbst als Beispiel angeben, da sie sich selbst beschreibt und ausführlich dokumentiert ist.

6.4.13 Owner

Um im Fall von Rückfragen oder Problemen hinsichtlich einer Namespace-Beschreibung möglichst einfach eine Ansprechperson oder -stelle zu finden, kann Kontaktinformation als

Bestandteil der Namespace-Beschreibung angegeben werden. Werden die Informationen dort mit URIs angegeben, so ist dies ebenfalls maschinenlesbare Information, die es ermöglicht, je nach angegebenem URI Schema (z.B. `http`, `mailto`, `tel`, `fax`) Kontakt aufzunehmen.

Beispiel: "eCH; Laupenstrasse 18a; 3008 Bern; tel:+41-31-5600020; fax:+41-31-5600025; mailto:info@ech.ch"

6.4.14 Date

Dieser Beschreibungsaspekt muss (**MUST**) ein Datum gemäss dem XML Schema `xs:date` Typ enthalten, der ein *Calendar Date* gemäss [ISO8601] definiert (dies ist das etablierte YYYY-MM-DD Format). Dieses Datum sollte das Datum der Erstellung oder der letzten Aktualisierung der Namespace-Beschreibung angeben (also nicht das Erstellungs-Datum des Schemas oder anderer mit dem Namespace assoziierter Ressourcen).

Beispiel: "2006-01-06"

6.4.15 Status

Der Status eines Namespaces beschreibt den Entwicklungsstand des damit beschriebenen Vokabulars und drei Werte annehmen. Ist eine feinere Unterteilung oder genauere Beschreibung weiterer Zustände gewünscht, so kann dies in den Beschreibungen mittels der Kategorie "Anderes" (im folgenden Abschnitt beschrieben) erreicht werden. Die drei definierten Werte des Status sind die folgenden:

- `workinprogress`: In diesem Fall handelt es sich um eine laufende Entwicklung, und es kann nicht davon ausgegangen werden, dass das beschriebene Vokabular stabil ist. Bevor Applikationen damit arbeiten, sollten sie auf den Status `released` warten, oder durch Kontakte zu den Owners des Vokabulars sicherstellen, dass ein ausreichender Grad an Stabilität erreicht ist.
- `released`: Hier handelt es sich um eine ausgereiftes und für die Benutzung freigegebenes Vokabular, das bedenkenlos im produktiven Betrieb benutzt werden kann.
- `deprecated`: In diesem Fall ist das Vokabular veraltet und von einer Benutzung wird abgeraten. Dies kann der Fall sein, falls neuere Versionen des Vokabulars existieren, oder falls komplett neue Vokabulare existieren, die anstatt dessen verwendet werden sollten

Beispiel: "released".

7 Format von Namespace-Beschreibungen

Namespace-Beschreibungen gemäss dem vorliegenden Dokument sollten die in Abschnitt 6 beschriebene Information enthalten. In welcher Form diese Information vorliegt, ist vor allem unter zwei Gesichtspunkten wichtig:

- *Erstellung und Verwaltung:* Um die Information zu erstellen und zu verwalten, wird am besten ein Format verwendet, das die Beschreibungen auf einfache und gut überschaubare Weise strukturiert. Am einfachsten geschieht dies durch einen speziellen XML Dokumententyp, der durch ein Schema definiert wird (dies ist in Abschnitt 7.1 beschrieben).
- *Publizieren:* Um Namespace-Beschreibungen zu publizieren, sollte ein Format verwendet werden, das die Beschreibungen in menschen- und maschinenlesbarer Form enthält. Am einfachsten geschieht dies durch ein (X)HTML-Format, das durch zusätzliche Informationen ergänzt wird (dies ist in Abschnitt 7.2 beschrieben).

Auch wenn diese beiden Formate an sich die gleiche Information enthalten, ist es doch sinnvoll, für die beiden aufgeführten Anwendungsbereiche unterschiedliche Formate zu verwenden. Um den Umgang mit beiden Formaten zu vereinfachen, und weil es sich bei beiden Formaten um XML-basierte Formate handelt, werden von eCH Stylesheets in XSLT 2.0 zur [xslt20] Verfügung gestellt, die eine Transformation des XML-Formats in die XHTML Darstellung erlauben.

7.1 XML Beschreibung

In Ergänzung zu den in Kapitel 6 definierten Namespace-Beschreibungen definiert eCH ein XML Schema und damit ein Format, das verwendet werden sollte (**SHOULD**), um Namespace-Beschreibungen zu repräsentieren. Dieses Schema orientiert sich hauptsächlich daran, eine einfache XML Repräsentation von XML Namespace-Beschreibungen zu ermöglichen, so dass diese auch ohne Tool-Unterstützung erstellt, geändert, und verstanden werden können. Um Anwender zu unterstützen, stellt eCH darüberhinaus Tools zur Verfügung, mit denen aus solch einer XML-strukturierten Namespace-Beschreibung XHTML (wie in Abschnitt 7.2 beschrieben) erzeugt werden kann.

Das XML Schema definiert für jede der vier Namespace-Klassen (Namespace Root, Major Version, Minor Version, Simple Namespace) ein eigenes Element, jedes dieser Elemente kann als Document Element einer Namespace-Beschreibung verwendet werden. Die einzelnen Beschreibungsaspekte werden als Kind-Elemente des jeweiligen Document Elements repräsentiert, die Typen der Beschreibungsaspekte werden durch XML Schema Typen repräsentiert. Details sind dem XML Schema selber zu entnehmen.

7.2 XHTML Beschreibung

Ziel des vorliegenden Dokuments ist nicht nur ein Format für Beschreibungen von XML namespaces zu definieren, sondern auch, diese Beschreibungen über das Web verfügbar zu machen. Aus diesem Grund wird davon ausgegangen, dass das XML Format als Ausgangspunkt verwendet wird, um eine HTML-Version einer Beschreibung zu erzeugen. Diese

HTML-Version verwenden XHTML 1.0 [xhtml10] in seiner *Transitional* Variante, die viele Dinge erlaubt, die in der eingeschränkteren *Strict* Variante nicht zulässig sind. HTML [html401] ist nicht zulässig, weil es nicht mit XML-Tools interpretiert und validiert werden kann und deswegen wesentlich schlechter geeignet ist, Ressourcen zu beschreiben, die maschinell weiterverarbeitbar sein sollen.

Um von den generierten XHTML Beschreibungen automatisch zur XML-Version zu finden, muss sich im XHTML ein Verweis auf die XML-Version finden. Dieser wird im head Element des html Elements angebracht, wo alle Informationen zu einem HTML-Dokument zu finden sind. Innerhalb des head Elementes gibt es link Elemente, die auf mit dem HTML-Dokument verbundene Ressourcen zeigen (eine typische Anwendung dieses Mechanismus' ist der Verweis auf ein externes Stylesheet). Im link Element beschreibt das rel Attribut die Art der Assoziation zwischen dem HTML-Dokument und der angegebenen Ressource. Aus dem XHTML, das unter der URI eines Namespaces vorgefunden wird, muss mit dem Attributwert ech-0033: xml -desc auf die XML-Version der Beschreibung verwiesen werden. Ein Verweis hat im XHTML also folgendes Aussehen (in diesem Fall befindet sich die XML-Version der Beschreibung im gleichen Verzeichnis wie das XHTML-Dokument):

```
<html xml ns="http://www.w3.org/1999/xhtml" >
  <head>
    . . . .
    <link rel="ech0033: xml -desc" href="description.xml" />
  </head>
  <body>
    . . . .
  </body>
</html >
```

Existiert unter dem Namespace Name eines Vokabulars (dies betrifft Namespace Root und Major Version Namespaces) eine HTML Seite, so muss (**MUST**) diese den hier beschriebenen Anforderungen genügen, d.h. sie muss XHTML benutzen, und sie muss mit einem link Element mit dem ech-0033: xml -desc Attributwert auf das XML verweisen, das gemäss Abschnitt 7.1 den Namespace beschreibt.

8 Sicherheitsüberlegungen

Da es sich bei den hier definierten Aspekten lediglich um Beschreibungen existierender Vokabulare existiert, werden durch diese Beschreibungen keine sicherheitsrelevanten Fragen neu gestellt. Allerdings sollte beachtet werden, dass die Beschreibungen der Namespaces allgemein zugänglich und nicht speziell zugangskontrolliert sind, und falls dies sicherheitsrelevante Fragen berührt, sollte überdacht werden, ob der Namespace durch eine öffentlich zugängliche Beschreibung dokumentiert werden sollte.

9 Hinweise auf Rechte Dritter

Im vorliegenden Dokument sind keine Rechte Dritter betroffen, es handelt sich um eine Definition der Beschreibung von XML Vokabularen, die im Rahmen von eCH entwickelt wurde und daher keinen Einschränkungen hinsichtlich ihrer Benutzung unterliegt.

Anhang A – Referenzen & Bibliographie

- [bourret] Ronald Bourret, *XML Namespaces FAQ*,
<http://www.rpbouret.com/xml/NamespacesFAQ.htm>
- [bray] Tim Bray, *Architectural Theses on Namespaces and Namespace Documents*, <http://www.textuality.com/tag/Issue8.html>
- [eCH-0018] Erik Wilde, Hanspeter Salvisberg, Alexander Pina, *XML Best Practices*, eCH, Berne, Switzerland, eCH-0018, August 2005.
- [eCH-0035] Erik Wilde, *Design von XML Schemas*, eCH, Berne, Switzerland, eCH-0035, 2006.
- [eCH-0050] Erik Wilde, *Hilfskomponenten zur Konstruktion von XML Schemas*, eCH, Berne, Switzerland, eCH-0050, 2006.
- [grddl] Dan Connolly, *Gleaning Resource Descriptions from Dialects of Languages (GRDDL)*, World Wide Web Consortium, W3C Working Draft WD-grddl-20061024, October 2006. <http://www.w3.org/TR/2006/WD-grddl-20061024/>
- [html401] Dave Raggett, Arnaud Le Hors, Ian Jacobs, *HTML 4.01 Specification*, World Wide Web Consortium, Recommendation REC-html401-19991224, December 1999. <http://www.w3.org/TR/1999/REC-html401-19991224>
- [ISO8601] International Organization for Standardization, *Data Elements and Interchange Formats — Information Interchange — Representation of Dates and Times*, ISO 8601, December 2004.
<http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874>
- [RFC2119] Scott O. Bradner, *Key Words for use in RFCs to Indicate Requirement Levels*, Internet RFC 2119, March 1997.
<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2616] Roy T. Fielding, Jim Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, Tim Berners-Lee, *Hypertext Transfer Protocol — HTTP/1.1*, Internet RFC 2616, June 1999.
<http://www.ietf.org/rfc/rfc2616.txt>
- [RFC3066] Harald Tveit Alvestrand, *Tags for the Identification of Languages*, Internet RFC 3066, January 2001. <http://www.ietf.org/rfc/rfc3066.txt>
- [RFC3986] Tim Berners-Lee, Roy Fielding, Larry Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, Internet RFC 3986, January 2005.
<http://www.ietf.org/rfc/rfc3986.txt>
- [RFC3987] Martin Dürst, Michel Suignard, *Internationalized Resource Identifiers (IRIs)*, Internet RFC 3987, January 2005.
<http://www.ietf.org/rfc/rfc3987.txt>

- [rddl] Jonathan Borden, Tim Bray, *Resource Directory Description Language (RDDL)*, February 2002.
<http://www.openhealth.org/RDDL/20020218/rddl-20020218.html>
- [rddl2] Jonathan Borden, Tim Bray, *Resource Directory Description Language (RDDL) 2.0*, January 2004.
<http://www.openhealth.org/RDDL/20040118/rddl-20040118.html>
- [rdfconcepts] Graham Klyne, Jeremy J. Carroll, *Resource Description Framework (RDF): Concepts and Abstract Syntax*, World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210, February 2004.
<http://www.w3.org/TR/2004/REC-rdf-concepts-20040210>
- [webarch] Ian Jacobs, Norman Walsh, *Architecture of the World Wide Web, Volume One*, World Wide Web Consortium, Recommendation REC-webarch-20041215, December 2004. <http://www.w3.org/TR/2004/REC-webarch-20041215>
- [wil05e] Erik Wilde, Willy Müller, *Organizing Federal E-Government Schemas*, Technical Report TIK Report No. 212, Computer Engineering and Networks Laboratory (TIK), ETH Zürich, February 2005.
<http://dret.net/netdret/publications#wil05e>
- [wil05l] Erik Wilde, *Describing Namespaces with GRDDL*, In: *Poster Proceedings of the Fourteenth International World Wide Web Conference*, Chiba, Japan, May 2005.
<http://dret.net/netdret/publications#wil05l>
- [xhtml10] Steven Pemberton, *XHTML 1.0: The Extensible HyperText Markup Language (Second Edition)*, World Wide Web Consortium, Recommendation REC-xhtml1-20020801, August 2002.
<http://www.w3.org/TR/2002/REC-xhtml1-20020801>
- [xlink] Steven J. DeRose, Eve Maler, David Orchard, *XML Linking Language (XLink) Version 1.0*, World Wide Web Consortium, Recommendation REC-xlink-20010627, June 2001.
<http://www.w3.org/TR/2001/REC-xlink-20010627>
- [xml10third] Tim Bray, Jean Paoli, C. Michael Sperberg-McQueen, Eve Maler, François Yergeau, *Extensible Markup Language (XML) 1.0 (Third Edition)*, World Wide Web Consortium, Recommendation REC-xml-20040204, February 2004. <http://www.w3.org/TR/2004/REC-xml-20040204>
- [xml11] Tim Bray, Jean Paoli, C. Michael Sperberg-McQueen, Eve Maler, François Yergeau, John Cowan, *Extensible Markup Language (XML) 1.1*, World Wide Web Consortium, Recommendation REC-xml11-20040204, February 2004. <http://www.w3.org/TR/2004/REC-xml11-20040204>

- [xmlns] [Tim Bray](#), [Dave Hollander](#), [Andrew Layman](#), *Namespaces in XML*, World Wide Web Consortium, Recommendation REC-xml-names-19990114, January 1999. <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- [xmlns11] Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin, *Namespaces in XML 1.1*, World Wide Web Consortium, Recommendation REC-xml-names11-20040204, February 2004. <http://www.w3.org/TR/2004/REC-xml-names11-20040204>
- [xmlschema1sec] Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, *XML Schema Part 1: Structures Second Edition*, World Wide Web Consortium, Recommendation REC-xmlschema-1-20041028, October 2004. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>
- [xmlschema2sec] Paul V. Biron, Ashok Malhotra, *XML Schema Part 2: Datatypes Second Edition*, World Wide Web Consortium, Recommendation REC-xmlschema-2-20041028, October 2004. <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>
- [xslt20] Michael Kay, *XSL Transformations (XSLT) Version 2.0*, World Wide Web Consortium, Working Draft WD-xslt20-20050915, September 2005. <http://www.w3.org/TR/2005/WD-xslt20-20050915>

Anhang B – Mitarbeit & Überprüfung

Remo Dick, ISC EJPD

Claude Eisenhut, Eisenhut Informatik

Urs Gähler, VRSG

Jürg Hotz, Kanton Thurgau

Willy Müller, ISB

Hubert Münt, Data Factory

Alexander Pina, Unisys (Schweiz) AG

Hanspeter Salvisberg, Unisys (Schweiz) AG

Hans Ulrich Wiedmer, KOGIS - LT

Erik Wilde, ETH Zürich

Anhang C – Abkürzungen & Glossar

Ein kommentiertes, mit weiterführenden Links und Literaturverweisen versehenes und wesentlich ausführlicheres Abkürzungsverzeichnis und Glossar findet sich auf dem Web unter <http://dret.net/glossary/>.

DTD	Document Type Definition
GRDDL	Gleaning Resource Descriptions from Dialects of Languages
HTML	Hypertext Markup Language
IRI	Internationalized Resource Identifier
RDDL	Resource Directory Description Language
RDF	Resource Description Framework
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
XHTML	Extensible Hypertext Markup Language
XLink	Extensible Linking Language
XML	Extensible Markup Language
XSLT	XSL Transformations

Anhang E – Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende mittels spezieller, schriftlicher Vereinbarung, sein betreffendes geistiges Eigentum oder seine Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

eCH-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.