eCH-0050: Hilfskomponenten zur Konstruktion von XML Schemas

Name	Hilfskomponenten zur Konstruktion von XML Schemas
Standard-Nummer	eCH-0050
Kategorie	Hilfsmittel
Reifegrad	Experimentell
Version	1.0
Status	Aufgehoben
Beschluss am	
Ausgabedatum	
Ersetzt Standard	
Sprachen	Deutsch
Autoren	Fachgruppe XML Erik Wilde, ETH Zürich (http://dret.net/netdret/)
Herausgeber / Vertrieb	Verein eCH, Amthausgasse 18, 3011 Bern T 031 560 00 20, F 031 560 00 25

Zusammenfassung

Das vorliegende Dokument wurde in den eCH-0018 ab Version 2.0 integriert.

Inhaltsverzeichnis

1 Status des Dokuments

Aufgehoben: Das Dokument wurde von eCH zurückgezogen. Es darf nicht mehr genutzt werden.

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 Einleitung

Bei der Definition von XML Schemas gibt es häufig Teile des Schemas, die wiederkehrende Komponenten betreffen, die für verschiedene Schemas verwendet werden können. Das vorliegende Dokument beschreibt die von eCH zur Verfügung gestellte Sammlung solcher Komponenten und soll dabei helfen, wiederverwendbare Komponenten zu verwenden.

2.1 Überblick

Viele der XML-Strukturen, die in verschiedenen Anwendungsszenarien verwendet werden, sind nicht einzigartig, sondern repräsentieren Konzepte, die in anderen Szenarien ebenfalls verwendet werden. Aus diesem Grund wäre es unnötig aufwendig, bei jedem XML Schema komplett neu zu beginnen. Anstatt dessen empfiehlt es sich, nach Möglichkeiten für Wiederverwendung zu suchen, so dass eventuell bereits existierende Schemas oder Teile von Schemas verwendet werden können. Die Vorteile dieses Vorgehens sind in Abschnitt 2.3 beschrieben.

- **SHOULD**: Werden in einem Schema Konzepte verwendet, für die sich bereits existierende Schema-Komponenten wiederverwenden lassen, so sollten diese Komponenten verwendet werden. Dies reduziert den Aufwand der Schemadefinition, und erleichtert das Verständnis des Schemas für Anwender, die die Komponenten bereits kennen.

2.2 Anwendungsgebiet

Das Anwendungsgebiet des vorliegenden Dokuments ist das Design von XML Schemas, bei dem, soweit möglich, bestehende Schema-Komponenten verwendet werden sollten. Generelle Richtlinien zum Design von XML Schemas sind im Dokument [eCH-0035] beschrieben, das vorliegende Dokument beschreibt lediglich eine Sammlung vorgegebener Komponenten.

2.3 Vorteile

Der Nutzen der Wiederverwendung von Schema-Komponenten (im Gegensatz dazu, für jedes XML Schema das gesamte Schema komplett selber zu definieren) besteht aus vier hauptsächlichen Aspekten:

- *Schemaerstellung*: Durch die Verwendung wiederverwendbarer Komponenten kann bei der Erstellung von Schemas Aufwand eingespart werden.
- *Schemaverwendung*: Durch die Verwendung wiederverwendbarer Komponenten kann der Benutzer ein Schema einfacher verstehen, weil dadurch wiederkehrende Konzepte auf gleiche Art in verschiedenen Schemas realisiert sind. Dies wird besonders dann nützlich, wenn es etablierte Komponenten gibt, die an den passenden Stellen immer wieder eingesetzt werden. Auf diese Weise fällt es den Benutzern eines Schemas sehr viel leichter, sich in einer Umgebung mit verschiedenen Schemas zu orientieren.

- *Instanzverarbeitung*: Durch die Verwendung wiederverwendbarer Komponenten kann ein wiederverwendetes Konzept bei der Verarbeitung einer Instanz wiedererkannt werden, und es kann z.B. entsprechender Code wiederverwendet werden, mit dessen Hilfe die Information verarbeitet werden kann.
- *Abbildung von Konzepten*: In einer Umgebung mit verschiedenen Schemas und verschiedenen Schema-basierten Diensten (z.B. Web Services) ist es häufig notwendig, verschiedene Schemas aufeinander abzubilden, z.B. um das Resultat eines Web Services in den Aufruf des nächsten Web Services einzusetzen. Werden für gewisse Teile der Schemas die gleichen Komponenten benutzt, so ist die Abbildung trivial. Andernfalls muss die Abbildung nicht nur implementiert werden als Abbildung zwischen zwei unterschiedlichen Schemas, sondern es muss auch noch detalliert untersucht werden, ob die abzubildenden Konzepte in der Tat semantisch gleichwertig sind.

Vor allem der letzte Punkt zeigt deutlich, dass die Wiederverwendung von Schema-Komponenten weit über das einfache Wiederverwenden von Markup hinausgeht. Viel wichtiger ist, dass auf diese Weise deutlich wird, dass es sich bei den wiederverwendeten Strukturen um die semantisch gleichen Konzepte handelt, so dass kein Aufwand geleistet werden muss, um neue Konzepte zu verstehen (und u.U. dann später einfach zu erkennen, dass es sich um ein neues Schema für bereits mit einem anderen Schema erfasste Konzepte handelt).

2.4 Schwerpunkte

Natürlich kann sich die Wiederverwendung von Schema-Komponenten nicht zum Ziel setzen, alle nur möglichen Konzepte zu erfassen und in Form von Schemas zu beschreiben. Das vorrangige Ziel des vorliegenden Dokuments ist es daher, diejenigen Konzepte und dazugehörige Schemas zu erfassen, bei denen eine sehr grosse Wahrscheinlichkeit für ihre Wiederverwendung besteht, z.B. Konzepte wie Sprachmarkierungen oder Postanschriften. Das Anwendungsgebiet sind, wie durch eCH vorgegeben, eGovernment Aktivitäten.

Ziel dieses Dokuments ist es aber auch, bestehende Lücken zu beseitigen und eine wachsende Sammlung an Schema-Komponenten bereitzustellen. Aus diesem Grund wird dieses Dokument periodisch erweitert und neu veröffentlicht werden, um jeweils neue Schema-Komponenten aufnehmen und zur Wiederverwendung anbieten zu können.

3 Verwendung der Hilfskomponenten

Die in diesem Dokument aufgeführten Hilfskomponenten können auf zwei verschiedene Arten benutzt werden:

1. Jede Hilfskomponente oder Gruppe von Hilfskomponenten ist durch ein eigenes Schema beschrieben, das einen eigenen Namespace [XMLns,XMLns11] definiert. Auf diese Weise kann ausschliesslich dieses Schema importiert werden in ein zu definierendes Schema, und dort kann die Hilfskomponente verwendet werden.
2. Es gibt ein umfassendes ech-0050 Schema, das sämtliche Schemas der Hilfskomponenten importiert. Auf diese Weise können, indem das ech-0050 Schema importiert wird, sämtliche Hilfskomponenten importiert werden.

Welche dieser beiden Arten benutzt wird, wird nicht von eCH vorgegeben. Es empfiehlt sich am ehesten, bei der Verwendung nur einer Hilfskomponente das dazugehörige Schema direkt zu verwenden, während sich bei der Verwendung einer grösseren Zahl von Hilfskomponenten eher das ech-0050 Schema anbietet.

Eine wichtige Bemerkung ist, dass das vorliegende Dokument die Idee der sogenannten „Chamäleon Schemas“ nicht unterstützt. Chamäleon Schemas sind Schemas, die ohne Namespace definiert werden und deren Komponenten erst durch einen Import in ein anderes Schema einen Namespace zugewiesen erhalten (sie nehmen dann, „wie ein Chamäleon“, den Namespace des Schemas an, in das sie importiert wurden). Chamäleon Schemas sollten bei der Verwendung von XML Schema ohnehin nicht verwendet werden, da sie es unmöglich machen, die ursprüngliche Gemeinsamkeit von Komponenten erkennbar zu machen (diese wird eben gerade über den Namespace erkannt). Aus diesem Grund haben alle in diesem Dokument aufgeführten Schemas einen eigenen Namespace.

Auf Grund der verschiedenartigen Eigenschaften von Hilfskomponenten und deren zu erwartender Änderungshäufigkeit wird zwischen zwei verschiedenen Arten von Hilfskomponenten unterschieden.

3.1 Stabile Hilfskomponenten

Stabile Hilfskomponenten sind Hilfskomponenten, bei denen die Wahrscheinlichkeit einer Änderung sehr gering ist, weil sie z.B. sehr einfache oder seit langem stabile Konzepte beschreiben. Ein Beispiel für diese Art von Hilfskomponente ist die Beschreibung von MIME Types, die sich an einem etablierten Standard orientiert, der in voraussehbarer Zeit nicht geändert werden wird.

Weil die Stabilität dieser Hilfskomponenten gross ist, werden sie in einem eigenen Namespace zusammengefasst, durch den sie identifiziert werden. Das Hinzufügen neuer Hilfskomponenten benötigt dabei jeweils nur eine neue Minor Version des Namespaces, weil es zu keinen inkompatiblen Änderungen des Namespaces kommt (mehr zur Benennung und Versionierung von XML Namespaces findet sich in [eCH-0033]). Es wird bei stabilen Hilfskomponenten also davon ausgegangen, das sie keinen Änderungen unterzogen werden, oder zumindest keinen nicht-rückwärtskompatiblen Änderungen.

Handelt es sich bei den Hilfskomponenten um Schemas für Elemente und/oder Attribute, deren Namespace Name vorgegeben ist, so können sie natürlich nicht den allgemeinen Namespace Name für stabile eCH Hilfskomponenten benutzen, sondern müssen den vorgegebenen Namespace Namen benutzen. Beispiele für solche Schemas mit vorgegebenen Namespace Namen sind das Schema für Sprachmarkierungen (Abschnitt 5) und das Schema für XLink (Abschnitt 7).

3.2 Instabile Hilfskomponenten

Instabile Hilfskomponenten sind Hilfskomponenten, bei denen die Wahrscheinlichkeit einer Änderung nicht sehr gering ist, weil sie z.B. noch in Entwicklung sind oder weil der Status und der Reifegrad der zugrundeliegenden Technologie unbekannt ist. Um die (möglicherweise inkompatiblen und daher eine neue Major Version des Namespaces benötigenden) Änderungen dieser Hilfskomponenten in ihren Auswirkungen zu isolieren, sind diese Hilfskomponenten jeweils mit einem eigenen Namespace beschrieben, so dass sich eine Änderung dieser Hilfskomponenten jeweils nur auf sie selber auswirkt.

Aus diesem Grund gibt es keinen gemeinsamen Namespace für die instabilen Hilfskomponenten, sondern sie müssen jeweils aus ihrem eigenen Namespace referenziert werden.

Dies heisst aber nicht, dass diese Hilfskomponenten nicht auch über das in Abschnitt 3 beschriebene ech-0050 Schema importiert werden können, denn dieses Schema importiert sowohl die stabilen als auch die instabilen Hilfskomponenten.

4 Markierung von Minor Versions

Gemäss [eCH-0033] ist die Major Versions eines Schemas im Namespace Name enthalten, sie muss deshalb in der Instanz nicht an anderer Stelle markiert werden. Minor Versions sind dagegen nicht im Namespace Name codiert, es muss deshalb auf eine andere Weise mitgeteilt werden können, zu welcher Minor Version eines Schemas eine Instanz gehört. Um dies zu tun, wird das hier beschriebene Attribut verwendet, das die Minor Version eines Schemas identifiziert.

Dieses Attribut muss auf dem *Document Element* (oftmals auch als *Root Element* bezeichnet) eines Dokuments verwendet werden, besteht ein Dokument aus Elementen verschiedener Namespaces (sogenannte *Compound Documents*), so kann das Attribut aber auch dort vorkommen, wo eine Unterbaum des Dokuments, der einen neuen Namespace verwendet, beginnt. Generell sollte das Minor Version Attribut also immer dort benutzt werden, wo aus hierarchischer Sicht mit der Interpretation von Elementen eines Schemas begonnen wird.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://www.ech.ch/xmlns/ech-0050/1"
   version="0">
  <xs:annotation>
   <xs:documentation>eCH XML Schema for a minor version attribute for
indicating the minor version of a XML Schema instance in a standardized
way.</xs:documentation>
  </xs:annotation>
  <xs:attribute name="minorVersion" type="xs:integer">
   <xs:annotation>
    <xs:documentation>This attribute is used for indicating the minor version of
an XML Schema instance in a standardized way. It SHOULD only appear on the
document element of XML Schema instances.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
</xs:schema>
```

Die Markierung muss mit einem ganzzahligen positiven Wert vorgenommen werden, damit Minor Versions eine wohldefinierte Ordnung haben. Diese Art der Versionsmarkierung für Minor Versions ist in [eCH-0033] definiert worden.

Da die Markierung der Minor Version eines verwendeten Schemas also in einem Attribut vorgenommen wird, das auf dem hierarchisch obersten Element dieses Schemas definiert sein muss, kann diese Markierung nur für Elemente benutzt werden. Dies bedeutet, dass dieser Mechanismus nicht für die Minor Version Markierung von Schemas benutzt werden kann, die nur Attribute definieren. Diese Einschränkung muss (**MUST**) beachtet werden, wenn solche Schemas definiert werden, und diese Schemas müssen dann eine eigene

Methode definieren und dokumentieren, wie die Minor Version Information in Instanzen codiert werden kann.

5 Sprachmarkierungen

Sprachmarkierungen werden häufig verwendet, um bei sprachabhängigen Texten die verwendete Sprache zu identifizieren. Da dies vor allem im Dokumentenbereich eine häufig wiederkehrende Anwendung ist, wird ein Mechanismus dafür in Abschnitt 2.12 von XML 1.0 [XML10third] bzw. Abschnitt 2.12 von XML 1.1 [XML11] festgelegt. Es handelt sich dabei um ein Attribut, mit Hilfe dessen die Sprache des Inhalts des Elementes angegeben werden kann. Das Format dieser Sprachmarkierung muss dem in [RFC3066] definierten Format gehorchen.

Dieses Attribut wird oftmals so interpretiert (und die XML Spezifikationen 1.0 und 1.1 definieren das auch so im oben erwähnten Abschnitt 2.12), dass es sich innerhalb des XML Dokuments vererbt auf hierarchisch tieferliegende Elemente. Ist also z.B. auf dem Document Element ein solches Attribut definiert, so legt dies die Sprache für alle sprachabhängigen Inhalte im gesamten Dokument fest, es sei denn, es würde durch eine erneute Angabe des Attributs der weiter oben angegebene Wert überschrieben oder gelöscht.

Der XML Standard definiert zwar dieses Attribut mit seinem Namen, seiner Bedeutung und Interpretation, aber es muss dennoch explizit in einem Schema zugelassen werden, damit es in einer Instanz vorkommen kann. Aus diesem Grund wird das hier beschriebene XML Schema benötigt, das als targetNamespace den XML Namespace selber definiert.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://www.w3.org/XML/1998/namespace"
   version="0">
 <xs:attribute name="lang" type="xs:language"/>
</xs:schema>
```

Die XML Spezifikationen definieren zwar den Namen des xml:lang Attributes und seine Bedeutung, aber kein Schema dafür. Das W3C veröffentlicht jedoch auf seiner Web Site XML Schemas, die alle für XML grundlegenden Attribute zusammenfassen. Dies ist in einer älteren Version das xml:lang Attribut zusammen mit xml:base und xml:space, dieses Schema ist unter http://www.w3.org/2001/03/xml.xsd zu finden. In einer kontinuierlich aufdatierten Version ist in jüngster Zeit noch das xml:id Attribut aus [XMLid] hinzugekommen, dieses Schema findet sich unter http://www.w3.org/2001/xml.xsd.

Das xml:lang Attribut dient einfachen Sprachmarkierungen von Inhalten in XML Dokumenten. Für komplexere Anwendungen kann es notwendig sein, bereits im Schema zu markieren, für welche Elemente und Attribute sprachabhängige Inhalte zu erwarten sind, so dass z.B. bei Übersetzungen von Instanzen anhand des Schemas entschieden werden kann, welche Teile übersetzt werden müssen. Für solche komplexeren Anwendungen ist das xml:lang Attribut nicht ausreichend, das W3C entwickelt mit dem *Internationalization Tag Set* [ITS] ein Konzept, dass diesen Anforderungen Rechnung trägt.

6 MIME Types

Einer der wichtigsten Mechanismen für viele Internet-Technologien sind MIME Types [RFC4288], mit denen markiert wird, was für einen Medientyp eine Ressource hat. Typische Beispiele sind Email und das Web, wo der Email Client oder der Browser durch den MIME Type erkennen, um was für einen Medientyp es sich bei einer bestimmten Ressource handelt, wenn sie diese erhalten und anschliessend verarbeiten wollen.

MIME Types sind zweiteilig aufgebaut, es gibt *Content Types* und *Subtypes*. Content Types identifizieren den eigentlichen Medientyp (z.B. Text oder Bild), während Subtypes das verwendete Format identifizieren (z.B. HTML im Fall von Texten oder JPEG im Fall von Bildern). Die erlaubten Werte für diese beiden Angaben sind in der *MIME Type Registry* [MIMEreg] der *Internet Assigned Numbers Authority (IANA)* definiert. Diese Registry wird kontinuierlich geändert, aus diesem Grund kann keine feste Liste der Werte angegeben werden, sondern die aktuell zugelassenen Werte müssen in der Registry nachgesehen werden.

Die Content Types ändern sich allerdings nur sehr selten, aus diesem Grund sind sie im Schema als feste Werteliste angegeben (text, multipart, message, application, image, audio, video, model). Die Subtypes dagegen sind kontinuierlicher Veränderung unterworfen, aus diesem Grund werden für sie keine Werte aufgezählt.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="http://www.ech.ch/xmlns/ech-0050/1"
   xmlns:mime="http://www.ech.ch/xmlns/ech-0050/1" version="0">
  <xs:annotation>
   <xs:documentation>eCH XML Schema for a minor version attribute for
indicating the minor version of a XML Schema instance in a standardized
way.</xs:documentation>
  </xs:annotation>
  <xs:simpleType name="MIMEtype">
   <xs:restriction base="xs:token">
    <xs:pattern value="(text|multipart|message|application|image|audio|video|
model)/\p{IsBasicLatin}+">
     <xs:annotation>
      <xs:documentation>The pattern hardcodes the existing MIME Content
Types, but the Subtypes are not defined here. The actual list of allowed Subtypes
is available from the IANA MIME Type Registry at
http://www.iana.org/assignments/media-types/.</xs:documentation>
     </xs:annotation>
    </xs:pattern>
   </xs:restriction>
  </xs:simpleType>
  <xs:attribute name="MIME" type="mime:MIMEtype"/>
  <xs:element name="MIME" type="mime:MIMEtype"/>
</xs:schema>
```

Das Schema definiert sowohl eine Attribut als auch ein Element, dass den MIME Type enthalten kann, und es kann natürlich auch nur der Typ wiederverwendet werden, falls das Element oder Attribut einen anderen Namen tragen soll.

7 Extensible Linking Language (XLink)

Die XML Linking Language [XLink] definiert ein Vokabular, mit Hilfe dessen Hyperlinks in XML benutzt werden können. XLink geht in seiner Funktionalität weit über das hinaus, was mit den einfachen Links in HTML realisiert werden kann, so erlaubt XLink z.B. Links mit mehr als zwei Endpunkten, und es erlaubt Links, die nicht in die Ressourcen eingebettet werden müssen, die sie verlinken. Weitergehende Informationen zu XLink finden sich z.B. in [Wilde02].

XLink definiert zwar die Namen und deren Bedeutung, aber kein XML Schema. Will man XLink also als XML Schema Komponente verwenden, so ist man darauf angewiesen, ein eigenes Schema zu erstellen oder eines von einer dritten Stelle zu nehmen. Das folgende Schema ist eine einfache Definition der durch XLink definierten Attribute, die jeweils als Attributgruppen verwendet werden können.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xlink="http://www.w3.org/1999/xlink"
	xmlns:xs="http://www.w3.org/2001/XMLSchema"
	targetNamespace="http://www.w3.org/1999/xlink">
	<!-- type attribute -->
	<xs:attribute name="type">
		<xs:simpleType>
			<xs:restriction base="xs:token">
				<xs:enumeration value="simple"/>
				<xs:enumeration value="extended"/>
				<xs:enumeration value="locator"/>
				<xs:enumeration value="arc"/>
				<xs:enumeration value="resource"/>
				<xs:enumeration value="title"/>
				<xs:enumeration value="none"/>
			</xs:restriction>
		</xs:simpleType>
	</xs:attribute>
	<!-- locator attribute -->
	<xs:attribute name="href" type="xs:anyURI"/>
	<!-- semantic attributes -->
	<xs:attribute name="role" type="xs:anyURI"/>
	<xs:attribute name="arcrole" type="xs:anyURI"/>
	<xs:attribute name="title" type="xs:string"/>
	<!-- behavior attributes -->
	<xs:attribute name="show">
		<xs:simpleType>
			<xs:restriction base="xs:token">
				<xs:enumeration value="new"/>
				<xs:enumeration value="replace"/>
				<xs:enumeration value="embed"/>
				<xs:enumeration value="other"/>
				<xs:enumeration value="none"/>
			</xs:restriction>
		</xs:simpleType>
	</xs:attribute>
```

```
        <xs:attribute name="actuate">
                <xs:simpleType>
                        <xs:restriction base="xs:token">
                                <xs:enumeration value="onLoad"/>
                                <xs:enumeration value="onRequest"/>
                                <xs:enumeration value="other"/>
                                <xs:enumeration value="none"/>
                        </xs:restriction>
                </xs:simpleType>
        </xs:attribute>
        <!-- traversal attributes -->
        <xs:attribute name="label" type="xs:string"/>
        <xs:attribute name="from" type="xs:string"/>
        <xs:attribute name="to" type="xs:string"/>
        <!-- attribute groups for xlink elements -->
        <xs:attributeGroup name="simpleLink">
                <xs:attribute ref="xlink:type" fixed="simple"/>
                <xs:attribute ref="xlink:href" use="optional"/>
                <xs:attribute ref="xlink:role" use="optional"/>
                <xs:attribute ref="xlink:arcrole" use="optional"/>
                <xs:attribute ref="xlink:title" use="optional"/>
                <xs:attribute ref="xlink:show" use="optional"/>
                <xs:attribute ref="xlink:actuate" use="optional"/>
        </xs:attributeGroup>
        <xs:attributeGroup name="extendedLink">
                <xs:attribute ref="xlink:type" fixed="extended"/>
                <xs:attribute ref="xlink:role" use="optional"/>
                <xs:attribute ref="xlink:title" use="optional"/>
        </xs:attributeGroup>
        <xs:attributeGroup name="locatorLink">
                <xs:attribute ref="xlink:type" fixed="locator"/>
                <xs:attribute ref="xlink:href" use="required"/>
                <xs:attribute ref="xlink:role" use="optional"/>
                <xs:attribute ref="xlink:title" use="optional"/>
                <xs:attribute ref="xlink:label" use="optional"/>
        </xs:attributeGroup>
        <xs:attributeGroup name="arcLink">
                <xs:attribute ref="xlink:type" fixed="arc"/>
                <xs:attribute ref="xlink:arcrole" use="optional"/>
                <xs:attribute ref="xlink:title" use="optional"/>
                <xs:attribute ref="xlink:show" use="optional"/>
                <xs:attribute ref="xlink:actuate" use="optional"/>
                <xs:attribute ref="xlink:from" use="optional"/>
                <xs:attribute ref="xlink:to" use="optional"/>
        </xs:attributeGroup>
        <xs:attributeGroup name="resourceLink">
                <xs:attribute ref="xlink:type" fixed="resource"/>
                <xs:attribute ref="xlink:role" use="optional"/>
                <xs:attribute ref="xlink:title" use="optional"/>
                <xs:attribute ref="xlink:label" use="optional"/>
        </xs:attributeGroup>
        <xs:attributeGroup name="titleLink">
                <xs:attribute ref="xlink:type" fixed="title"/>
```

```
        </xs:attributeGroup>
        <xs:attributeGroup name="noLink">
                <xs:attribute ref="xlink:type" fixed="none"/>
        </xs:attributeGroup>
</xs:schema>
```

8 Sicherheitsüberlegungen

Die im vorliegenden Dokument aufgelisteten Hilfsmittel haben keine sicherheitskritischen Auswirkungen.

9 Haftungsausschluss/Hinweise auf Rechte Dritter

Die in diesem Dokument enthaltenen Inhalte berühren keine Rechte Dritter.

eCH-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellt, oder welche **eCH** referenziert, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

10 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende 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.

Anhang A – Referenzen & Bibliographie

[eCH-0033] Erik Wilde, *Beschreibung von XML Namespaces*, eCH, Berne, Switzerland, eCH-0033, 2006.

[eCH-0035] Erik Wilde, *Design von XML Schemas*, eCH, Berne, Switzerland, eCH-0035, 2006.

[ITS] Christian Lieske, Felix Sasaki, *Internationalization Tag Set (ITS)*, World Wide Web Consortium, Working Draft WD-its-20060414, April 2006. http://www.w3.org/TR/2005/WD-its-20060414/

[MIMEreg] Internet Assigned Numbers Authority (IANA), *MIME Media Types*. http://www.iana.org/assignments/media-types/

[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

[RFC3066] Harald Tveit Alvestrand, *Tags for the Identification of Languages*, Internet RFC 3066, January 2001. http://www.ietf.org/rfc/rfc3066.txt

[RFC4288] Ned Freed, John C. Klensin, *Media Type Specifications and Registration Procedures*, Internet RFC 4288, December 2005. http://www.ietf.org/rfc/rfc4288.txt

[Wilde02] Erik Wilde, David Lowe, *XPath, XLink, XPointer, and XML: A Practical Guide to Web Hyperlinking and Transclusion*, Addison Wesley, Reading, Massachusetts, July 2002, ISBN 0201703440.

[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

[XMLid] Jonathan Marsh, Daniel Veillard, Norman Walsh, *xml:id Version 1.0*, World Wide Web Consortium, Recommendation REC-xml-id-20050909, September 2005. http://www.w3.org/TR/2005/REC-xml-id-20050909

[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

Anhang B – Mitarbeit & Überprüfung

Hans-Ulrich Bucher, Avataris AG

Remo Dick, ISC EJPD

Claude Eisenhut, Eisenhut Informatik

Urs Gähler, VRSG

Jürg Hotz, Kanton Thurgau

Adrian K. Keller, SAG Software Systems AG

Willy Müller, ISB

Hubert Münst, Data Factory

Patrick Ostertag, Etat de Fribourg

Alexander Pina, Unisys (Schweiz) AG

Fabian Probst, Fachhochschule Solothurn Nordwestschweiz

Hanspeter Salvisberg, Unisys (Schweiz) AG

Verena Sieber, T-Systems

Hans Ulrich Wiedmer, KOGIS - LT

Hansruedi Vock, BIT

Erik Wilde, ETH Zürich

Anhang C – Abkürzungen & Glossar

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

MIME	Multipurpose Internet Mail Extensions
W3C	World Wide Web Consortium
XLink	XML Linking Language
XML	Extensible Markup Language