

XRXEditor – ein generischer schemabasierter XForms- Editor

Dokumentation

Inhaltsverzeichnis

1	Einführung	3
1.1	XRXEditor.....	3
1.2	XForms-Anwendungen als XML-Editoren	3
1.3	Generische XForms-XML-Editoren.....	4
1.4	Gemischter Inhalt durch Textannotation	4
1.5	Entwicklungshintergrund.....	6
2	XRXEditor verwenden.....	9
2.1	eXist installieren	9
2.2	XRXEditor herunterladen und einbinden	9
2.2.1	Das Paket rxeditor einbinden	9
2.2.2	betterFORM-Erweiterung einbinden.....	12
2.3	Editor in XQuery-Dokument einbinden	12
2.3.1	XRXEditor Modul einbinden	12
2.3.2	Die Funktion rxxe:editor()	13
2.3.3	Das verwendete Schema.....	18
2.3.4	Das verwendete XML-Dokument	18
3	Konfigurationsmöglichkeiten.....	19
3.1	Konfigurationsattribute.....	19
3.2	Konfigurationsparameter	21
3.3	Weitere Konfigurationsvariablen	24
4	XSD mit Informationen für XRXEditor anreichern	25
4.1	Zusätzliche Informationen für den Editor	25

1 Einführung

1.1 XRXEditor

XRXEditor ist ein Softwaremodul, das aus einer XML-Schema-Definition (XSD) einen XML-Editor generiert, der in Webseiten eingebettet werden kann. Der resultierende Editor erlaubt es XML-Dokumente nach den Regeln des zugrundeliegenden Schemas zu bearbeiten, ohne dass der Benutzer Kenntnisse über das verwendete XML-Vokabular oder über XML im Allgemeinen besitzen muss. Das Softwaremodul besteht aus XQuery-Modulen, die das verwendete Schema einlesen und anhand dessen Regeln eine XForms-Anwendung generieren. Der Editor kann durch einen einzigen XQuery-Funktionsaufruf in eine Webseite eingebettet werden. Die erzeugte XForms-Anwendung generiert die passenden Steuerungselemente, die es dem Benutzer ermöglichen das XML-Dokument zu bearbeiten. XRXEditor kann in der XRX-Umgebung eXist¹ verwendet werden. Durch eine Erweiterung des XForms-Prozessors bietet XRXEditor die Möglichkeit zur Erzeugung und Bearbeitung von gemischem Inhalt. Zusätzlich bietet XRXEditor eine Methodik das verwendete Schema durch zusätzliche Informationen anzureichern, die über die Möglichkeiten einer XML-Schema-Definition hinausgehen. Diese Informationen werden bei der Generierung des Editors verwendet. Es können bspw. menschenlesbarere Benennungen oder Hinweise zur Bearbeitung von Knoten angegeben werden.

1.2 XForms-Anwendungen als XML-Editoren

XForms ist ein W3C Standard und eine Weiterentwicklung der HTML-Formulare, der versucht die Unzulänglichkeiten seines Vorgängers zu beheben. Einer der Hauptunterschiede von XForms zu HTML-Formularen ist die Entkoppelung der Daten von ihrer Darstellung. Die Trennung der Daten und der Benutzeroberfläche wird in XForms durch die Verwendung des Model-View-Controller-Architekturmuster erreicht. Der Model-Teil besteht dabei aus einem oder mehreren Modellen. Ein Modell beschreibt die Daten, deren Eigenschaften, Einschränkungen und deren Versendung. Der View-Teil beschreibt die Steuerungselemente, wie diese gruppiert werden sollen und deren Erscheinung. Die Steuerungselemente des View-Teils können an Daten des Models gebunden sein oder diese referenzieren.

¹ <http://www.exist-db.org>

Ein wichtiger Aspekt von XForms ist, dass alle Daten als XML-Instanzen ausgedrückt werden. Um die Daten einer Instanz zu initialisieren kann einfach ein externes XML-Dokument geladen werden. Die beinhalteten Daten des Dokuments werden in den, mit der Instanz verknüpften Steuerungselementen sichtbar und bearbeitbar. Ebenso können die (möglicherweise über die Steuerungselemente veränderten) Instanzen als XML an Server gesendet werden. Aus diesen Gründen eignet sich XForms sehr gut als Technologie zur Beschreibung von XML-Editoren.

Die Werte der XML-Knoten können über passende Steuerungselemente bearbeitet werden. Knoten können über das Betätigen von Schaltflächen in die XML-Instanz eingefügt oder wieder entfernt werden. Die Wohlgeformtheit des XML-Dokuments kann anders als in einem Texteditor nicht zerstört werden. Die Validität eines XML-Dokuments hängt von der verwendeten XSD ab. Dieses beschreibt welche Knoten in welche eingefügt werden dürfen, wie oft diese maximal vorkommen dürfen und minimal vorkommen müssen, und welche Werte diese besitzen können. Der Editor sollte den Benutzer so gut wie möglich dabei unterstützen, damit die Validität sichergestellt wird. Dies passiert durch passendes Ein- und Ausblenden der Einfüge- und Lösch-Schaltflächen, durch eine Auswahl eines passenden Steuerungselements anhand der Knoteneigenschaften und dessen Typs, sowie durch die Validierung der Knotenwerte.

1.3 Generische XForms-XML-Editoren

Prinzipiell muss für jedes Schema eine eigene, passende XForms-Anwendung geschrieben werden um einen XML-Editor zu realisieren, der die Validität für Dokumente gegen dieses Schema unterstützt. Änderungen im Schema müssen ständig in die XForms-Anwendung eingepflegt werden. Dafür bedarf es eines Entwicklers, der zum einen den XForms-Standard kennt, zum anderen die jeweilige XML-Anwendung. Sollen mehrere Schemata unterstützt werden müssen mehrere XForms-Anwendungen implementiert werden. Die Idee von XRXEditor ist, die XForms-Anwendung nicht hard coded zu schreiben, sondern dynamisch und automatisch aus dem Schema zu erstellen. Dazu dient die XQuery-Anwendung innerhalb XRXEditor, die das Schema einliest und eine dafür passende XForms-Anwendung generiert.

1.4 Gemischter Inhalt durch Textannotation

Obwohl sich XForms, wie bereits erwähnt, prinzipiell gut für XML-Editoren eignet, besitzt der Standard eine Einschränkung, die ihn für viele XML-Anwendungen nur bedingt nutzbar macht. XForms unterstützt keinen gemischten Inhalt. In XSDs ist es möglich Elemente so zu

definieren, dass diese sowohl Textinhalt als auch andere Elemente als Inhalt besitzen dürfen (gemischter Inhalt). XForms verlangt jedoch, dass jeder Knoten, auf den sich ein Steuerungselement bezieht, nur aus einfachem Inhalt wie Text besteht. Gemischter Inhalt wird nicht unterstützt, und kann weder angezeigt noch bearbeitet werden. XRXEditor bietet eine Möglichkeit dieses Problem zu umgehen, indem es durch eine Erweiterung des XForms-Prozessors ein spezielles Steuerungselement bietet, das es nicht nur erlaubt gemischten Inhalt anzuzeigen, sondern diesen auch zu bearbeiten und zu erzeugen. Gemischter Inhalt wird in einem Textannotationstextfeld angezeigt. Es kann gemischten Inhalt als annotierten Text anzeigen, es erlaubt weitere Elemente als Annotationen hinzuzufügen oder bestehende Elemente zu entfernen.

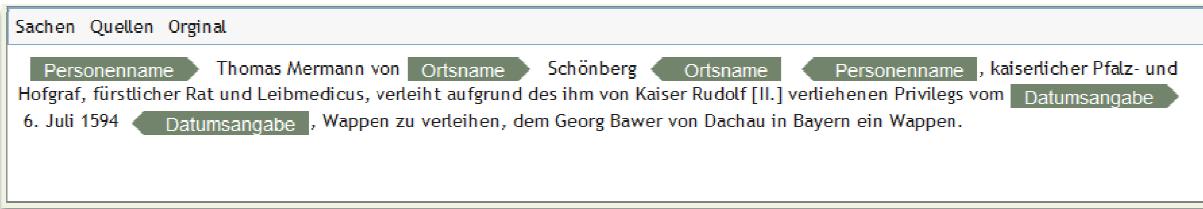


Abbildung 1 - Gemischter Inhalt in einem Textannotationsfeld

Innerhalb dieses Textannotationstextfeldes kann der Benutzer Text mit der Maus selektieren. Kontextabhängig für diese Auswahl wird das Menu des Steuerungselements erzeugt und bietet dem Benutzer eine Auswahl an zulässiger Annotationen (XML-Elemente).

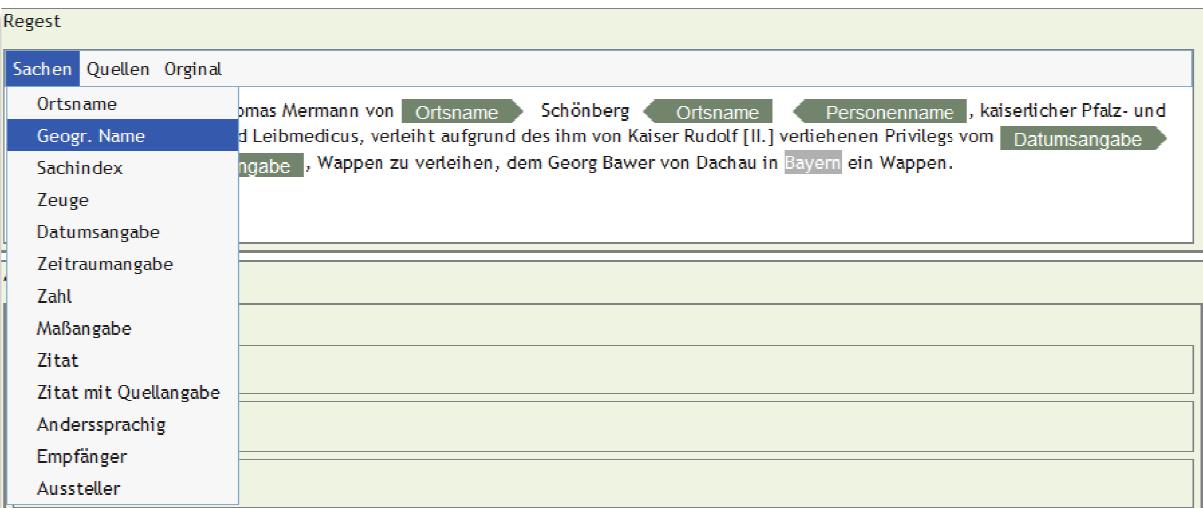


Abbildung 2 – Einfügen von Annotationen/Elementen

Durch Klicken auf die Annotationsbegrenzer (grüne Pfeile) erscheint ein Dialog, der es erlaubt die Annotationen wieder zu entfernen oder die Attribute des Elements zu bearbeiten, sie hinzuzufügen oder zu entfernen.

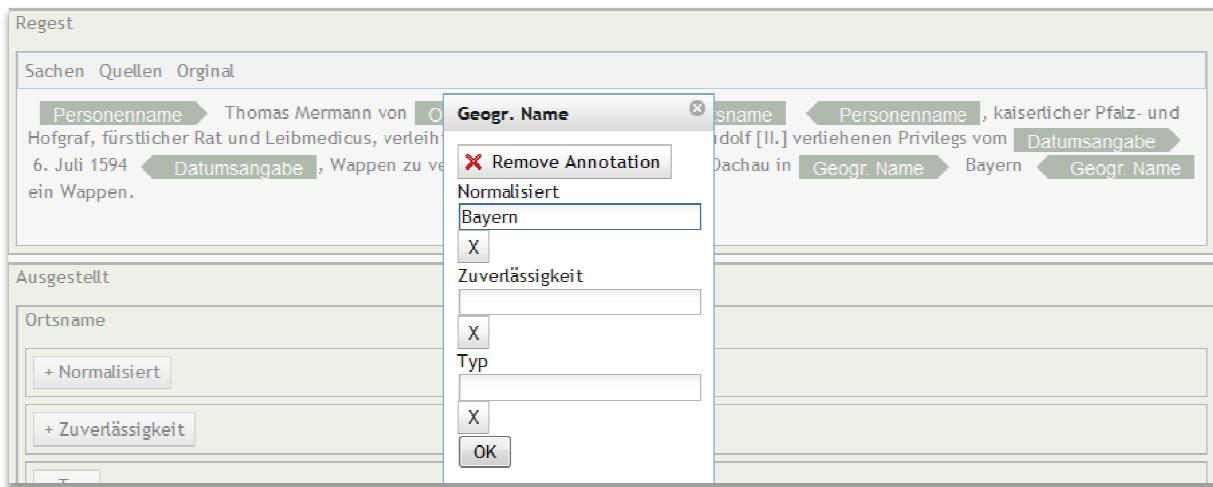


Abbildung 3 – Einfügen Löschen von Annotationen und Bearbeiten von Attributen

1.5 Entwicklungshintergrund

XRXEditor wurde innerhalb des Projektes „Virtuelles deutsches Urkundennetzwerk“ der Deutschen Forschungsgesellschaft entwickelt und ist eine Komponente des VdU Virtual Research Environment Toolkits (VdU/VRET), die Software einer Virtuellen Forschungsumgebung für mittelalterliche Urkunden.

Diese Software basiert auf der Softwarearchitektur XRX, die ausschließlich XML-Technologien verwendet. Mit XQuery wird auf der Server-Seite die Programmlogik implementiert, sowie die native XML-Datenbank abgefragt. In XHTML eingebettetes XForms beschreibt die Darstellung und Logik auf der Client-Seite. Auf allen Ebenen wird XML als Datenformat verwendet. Diese Software wird u.a. für die Web-Portale „Virtuelles Deutsches Urkundennetzwerk“² und Monasterium Collaborative Archiv³ verwendet. Für diese erzeugt XRXEditor einen Urkundeneditor und einen Bestandseditor. Diese sind Werkzeuge zur kollaborativen Bearbeitung von Metadaten.

Der Urkundeneditor basiert auf dem Schema der CEI (Charter Encoding Initiative)⁴, einem XML-Standard zur Beschreibung mittelalterlicher Urkunden. Er erlaubt es Wissenschaftlern, Archivaren und anderen Benutzern die Informationen der Urkunden zu bearbeiten und anzureichern, ohne Kenntnisse von CEI oder XML besitzen zu müssen.

² <http://www.vdu.uni-koeln.de>

³ <http://www.mom-ca.uni-koeln.de>

⁴ <http://www.cei.lmu.de>

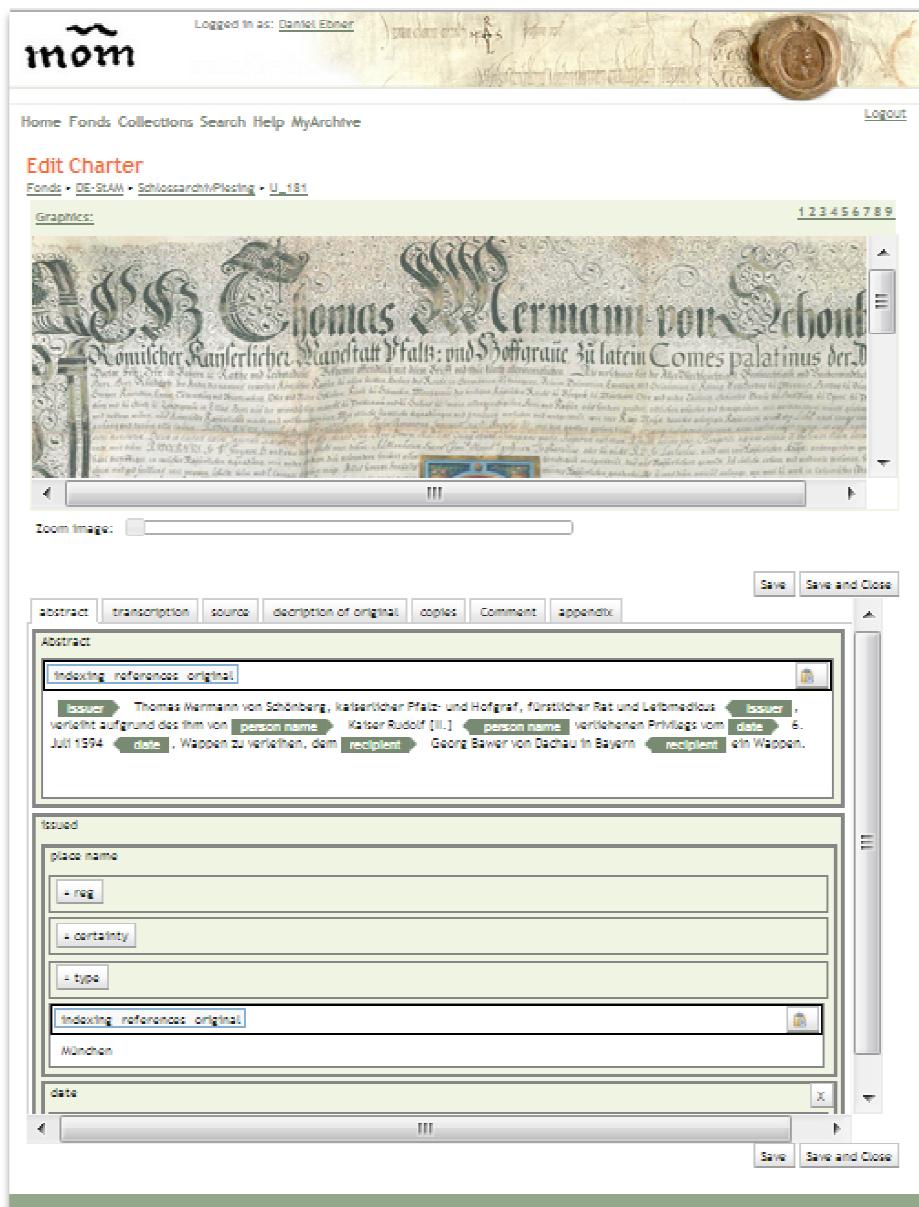


Abbildung 4 – Urkundeneditor

Der Bestandseditor erlaubt es Archivaren die Beschreibung der Urkundenbestände zu bearbeiten. Dieser basiert auf dem Schema der EAD (Encoded Archival Description)⁵, einem XML-Standard zur Beschreibung archivarischer Findmittel.

⁵ <http://www.loc.gov/ead/>

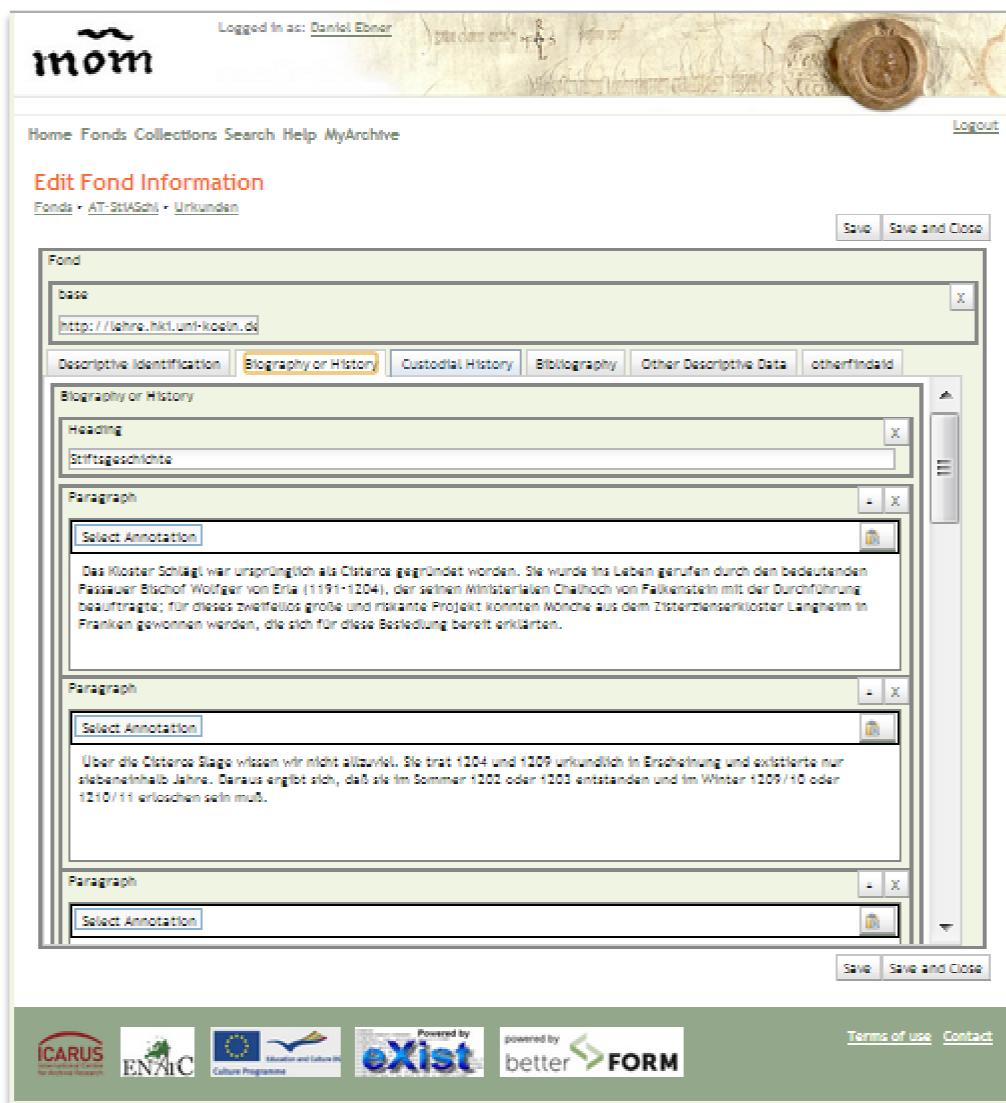


Abbildung 5– Bestandseditor

Obwohl XRXEdition als Komponente des VdU/VRET entwickelt wurde, ist es bewusst so entworfen und implementiert worden, dass es auch unabhängig von diesem Toolkit für weitere Editoren in anderen Projekten in XRХ-Umgebungen, für beliebige XSDs wiederverwendet werden kann.

2 XRXEditor verwenden

2.1 eXist installieren

eXist⁶ war ursprünglich eine reine native XML-Datenbank. Durch die Implementierung einer REST-Schnittstelle und dem Hinzufügen eines integrierten Webservers sowie eines XForms-Prozessors (betterFORM⁷) kann eXist mittlerweile als kompletter XRX-Anwendungsserver gesehen werden. Die aktuellste Version (Stand 08.03.2013) ist eXist 2.0. Diese kann unter <http://sourceforge.net/projects/exist/files/Stable/2.0/> heruntergeladen werden. Informationen zur Installation und zum Starten von eXist kann man unter <http://exist-db.org/exist/apps/doc/quickstart.xml> finden. Die folgenden Beschreibungen beziehen sich immer auf die Standardinstallation auf einem lokalen Rechner (localhost) mit dem Standardport 8080 und dem Kontextpfad „/exist“. Bei abweichender Installation müssen die URLs entsprechend angepasst werden.

2.2 XRXEditor herunterladen und einbinden

XRXEditor besteht aus zwei Teilen. Der erste Teil besteht aus dem Paket xrxeditor, welches XQuery-Modulen und Dateien beinhaltet. Dieses Paket ist verantwortlich das Schema abzufragen und anhand dessen Regeln die XForms-Anwendung zu generieren. Der zweite Teil ist eine Erweiterung des XForms-Prozessors betterFORM. Diese bietet das zusätzliche Steuerungselement zur Textannotation.

2.2.1 Das Paket xrxeditor einbinden

Das Paket xrxeditor-0.1.xar kann unter (<https://subversion.rrz.uni-koeln.de/trac/eXist-A/export/2418/trunk/my/XRX/src/core/app/editor/xrxeditor-0.1.xar>) heruntergeladen werden. Es handelt sich bei diesem Paket um eine eXist-App, die über den Paket-Manager, der über das Dashboard (<http://localhost:8080/exist/apps/dashboard/index.html>) zu erreichen ist, per Drag-And-Drop in die Datenbank importiert werden kann.

⁶ <http://exist-db.org>

⁷ <http://betterform.de>

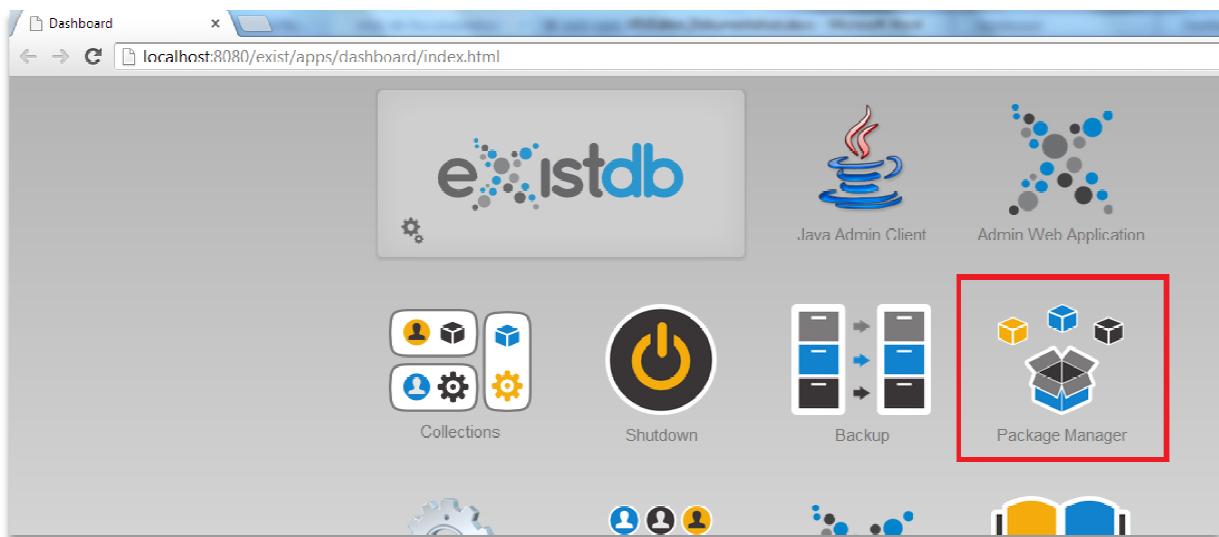


Abbildung 6 - Aufruf des Packge Managers über das Dashboard der lokalen eXist-Installation

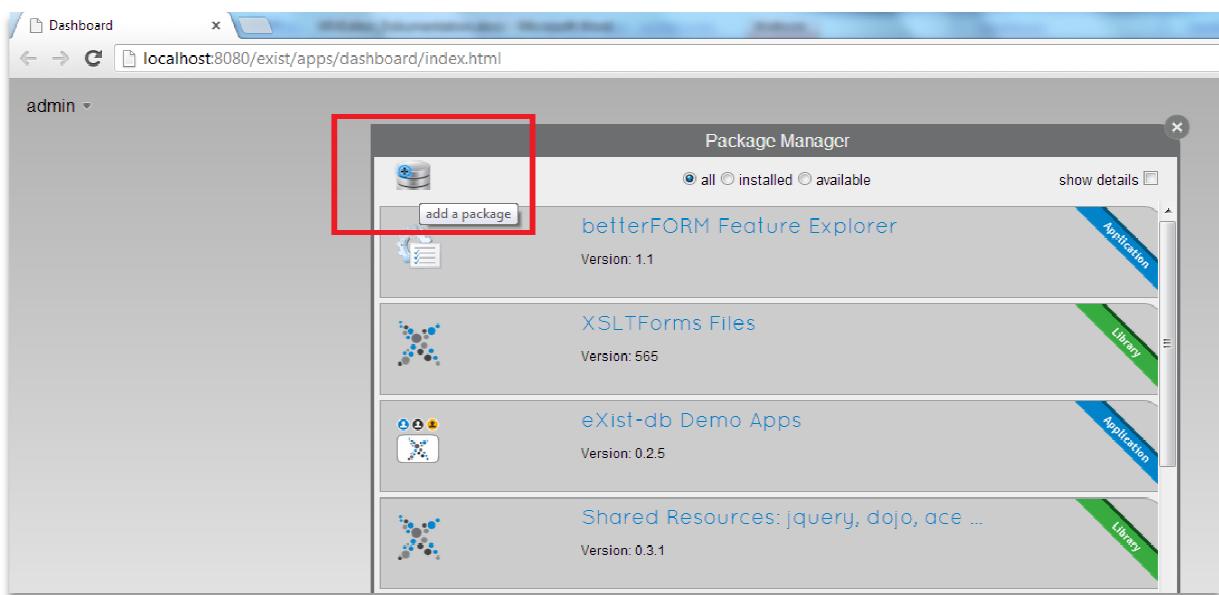


Abbildung 7 – Installation einer App durch Hinzufügen eines Pakets

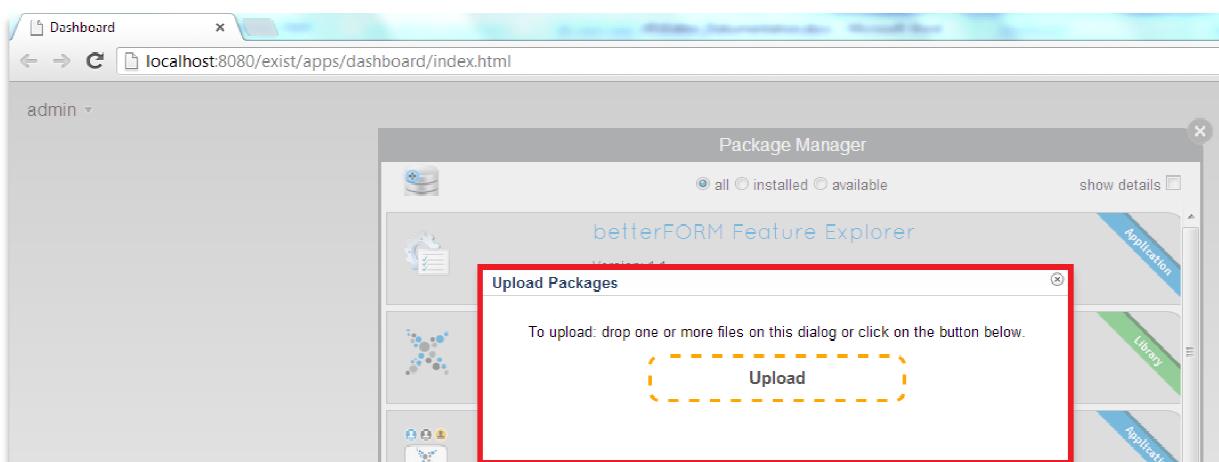


Abbildung 8– Hochladen des Pakets xrxeditor-0.1.xar durch Drag-And-Drop

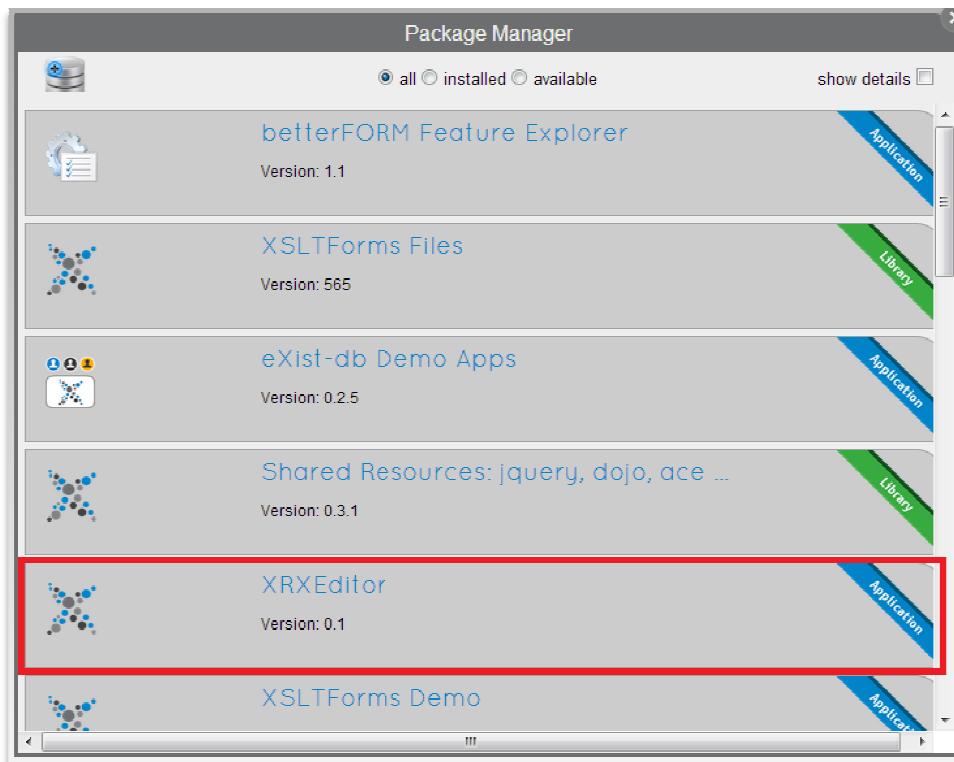


Abbildung 9– Nach erfolgreicher Installation ist XRXEditor im Package Manager aufgeführt

Über den „Collection Browser“ (zu erreichen über das Collections-Item auf dem Dashboard) müssten unter /db/apps/xrxeditor nun folgende Dokumente zu sehen sein.

Collection Browser					
Name	Permissions	Owner	Group	Last-modified	
..					
examples	crwxr-xr-x	admin	dba	February 28 2013 13:27:41	
build.xml	-rw-r--r--	admin	dba	February 28 2013 13:16:01	
expath-pkg.xml	-rwxrwxr-x	admin	dba	February 28 2013 13:16:01	
qrxra.xqm	-rw-r--r--	admin	dba	February 28 2013 13:27:18	
qrxre.xqm	-rw-r--r--	admin	dba	February 28 2013 13:25:48	
qxsd.xqm	-rw-r--r--	admin	dba	February 28 2013 13:26:32	
repo.xml	-rwxrwxr-x	admin	dba	February 28 2013 13:16:01	
upd.xqm	-rw-r--r--	admin	dba	February 28 2013 13:16:41	
xrxe-conf.xqm	-rw-r--r--	admin	dba	February 28 2013 13:19:59	
xrxe-services.xql	-rw-r--r--	admin	dba	February 28 2013 13:25:11	
xrxe.xqm	-rw-r--r--	admin	dba	February 28 2013 13:19:01	

Abbildung 10 – Die Datenbank beinhaltet die Collection /db /apps/xrxeditor mit allen benötigten XQuery-Skripten

2.2.2 betterFORM-Erweiterung einbinden

Der zweite Teil der XRXEeditor-Software ist die Datei betterform.jar, die unter <https://subversion.rzz.uni-koeln.de/trac/eXist-A/export/2379/trunk/my/eXist/extensions/betterform/main/lib/betterform.jar> heruntergeladen werden kann. Um diese Erweiterung verwenden zu können muss die ursprüngliche Datei unter `EXIST_HOME8/extensions/betterform/main/lib/betterform.jar` durch die heruntergeladene Datei ersetzt werden. Beim Austausch der Dateien sollte eXist nicht laufen und erst danach wieder gestartet werden.

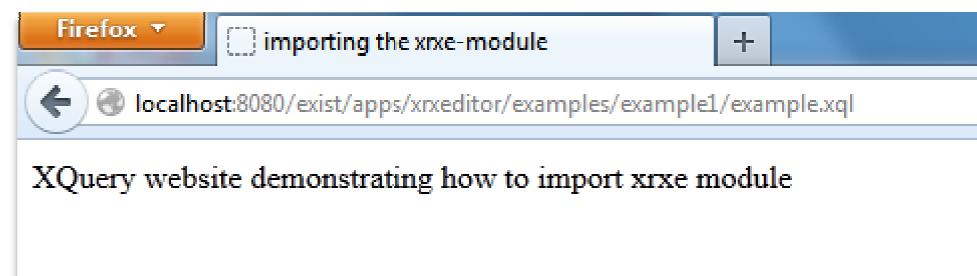
2.3 Editor in XQuery-Dokument einbinden

2.3.1 XRXEeditor Modul einbinden

Ein durch XRXEeditor generierter Editor kann in jede XQuery-Seite eingebunden werden, die das Modul `xrxe.xqm` importiert. Das Modul besitzt den Namensraum `http://www.monasterium.net/NS/xrxe` und kann durch den Befehl `import module namespace xrxe='http://www.monasterium.net/NS/xrxe' at '././xrxe.xqm';` eingebunden werden. Der relative oder absolute Datenbankpfad zum Modul `xrxe.xqm` muss eventuell angepasst werden.

```
xquery version "3.0";
import module namespace xrxe='http://www.monasterium.net/NS/xrxe' at '././xrxe.xqm';
<html xmlns="http://www.w3.org/1999/xhtml">
  <head><title>importing the xrxe-module</title></head>
  <body>XQuery website demonstrating how to import xrxe module</body>
</html>
```

`/db/apps/xrxeditor/examples/example1/example.xql`



<http://localhost:8080/exist/apps/xrxeditor/examples/example1/example.xql>

⁸ EXIST_HOME bezeichnet den Pfad der jeweiligen lokalen Installation von eXist

Dieses und die folgenden Beispiele befinden sich durch die Installation con XRXEditor in der Datenbank und können bei einer Standardinstallation und laufender Datenbank über die jeweils angegebenen URLs aufgerufen werden. Die Skripte und Dokumente können unter den angegebenen Datenbankpfaden gefunden, angesehen sowie zum Testen modifiziert werden. Dazu kann beispielsweise die in eXist integrierte Entwicklungsumgebung eXide verwendet werden (<http://localhost:8080/exist/apps/eXide/index.html>)

2.3.2 Die Funktion xrxe:editor()

Ist das XQuery Modul xrxe.xqm richtig in einer XQuery-Seite eingebunden, so kann der Editor über einen Funktionsaufruf generiert und an einer beliebigen Stelle in die Seite eingebettet werden. Die Funktion heißt xrxe:editor(\$param). Sie erwartet ein Element als Argument. Der Name dieses Elements ist beliebig bspw. „div“. Dieses Element erhält als Attribute weitere Informationen für XRXEditor. Es gibt zwei obligatorische Angaben. Das Attribut xsd gibt den absoluten Datenbankpfad oder die URI des zu verwendenden XSDs an. Das Attribut doc gibt den absoluten Datenbankpfad oder die URI des zu editierenden XML-Dokumentes an.

```
...
<body>
{
    xrxe:editor(<div
        doc="/db/apps/xrxeditor/examples/example2/example.xml"
        xsd="/db/apps/xrxeditor/examples/example2/example.xsd"
    />
)
}
</body>
...

```

/db/apps/xrxeditor/examples/example2/example.xql

Beim Aufruf der Webseite wird der Editor dynamisch generiert und eingebettet:

collection

entry

+ id

header

did

title

date

12/12/2012

+ number

body

abstract

I lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt

text

published

p

Select Annotation

Lorem ipsum dolor person sit amet person sit amet, consetetur sadipscing place elitr, sed diam nonumy place eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

p

Select Annotation

Lorem ipsum dolor name firstname sit amet firstname lastname consetetur sadipscing lastname name elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

+ p

footer

link

+ location

+ description

+ url

comment

+ comment

+ entry

<http://localhost:8080/exist/apps/xrxeditor/examples/example2/example.xql>

Dieses Beispiel bezieht sich auf das folgende XML Schema. Weitere Beispiele beziehen sich auf dasselbe oder darauf aufbauende Schemata.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://vdu.uni-koeln.de/NS/example" xmlns:xrxe="http://www.monasterium.net/NS/xrxe"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://vdu.uni-koeln.de/NS/example"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:group name="headerGroup">
    <xs:sequence>
      <xs:element name="title"/>
      <xs:element name="date" type="xs:date"/>
      <xs:element ref="number" minOccurs="0"/>
    </xs:sequence>
  </xs:group>
  <xs:attributeGroup name="linkAttributes">
    <xs:attribute name="location">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="ext"/>
          <xs:enumeration value="int"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="description"/>
    <xs:attribute name="url"/>
  </xs:attributeGroup>
  <xs:complexType name="name">
    <xs:sequence>
      <xs:element name="firstname"/>
      <xs:element name="lastname"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="number" type="xs:int"/>
  <xs:element name="collection">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="entry" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="header">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="did">
                      <xs:complexType>
                        <xs:group ref="headerGroup"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="body">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="abstract" minOccurs="0"/>
              <xs:element name="text">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="p" maxOccurs="5" minOccurs="3">
                      <xs:complexType mixed="true">
                        <xs:choice minOccurs="0" maxOccurs="unbounded">
                          <xs:element name="person" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
```

```

<xs:simpleContent>
    <xs:extension base="xs:string">
        <xs:attribute name="alsoKnowAs"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="place" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="name" type="name" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="important" minOccurs="0"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="published" type="xs:boolean"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="footer">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="link">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:string">
                            <xs:attributeGroup ref="linkAttributes"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element minOccurs="1" maxOccurs="10" name="comment" />
        <xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="id"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

/db/apps/xrxeditor/examples/example2/example.xsd

Das zu bearbeitende XML-Dokument aus diesem Beispiel sieht folgendermaßen aus:

```
<?xml version="1.0" encoding="UTF-8"?>
<ex:collection xmlns:ex="http://vdu.uni-koeln.de/NS/example">
  <ex:entry>
    <ex:header>
      <ex:did>
        <ex:title/>
        <ex:date>2012-12-12</ex:date>
      </ex:did>
    </ex:header>
    <ex:body>
      <ex:abstract>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor
invidunt</ex:abstract>
      <ex:text published="true">
        <ex:p>Lorem ipsum dolor <ex:person alsoKnowAs="Sitament">site amet</ex:person>sit amet,
consetetur sadipscing <ex:place>elitr, sed diam nonumy</ex:place> eirmod tempor invidunt ut labore et dolore
magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita
kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet,
consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea
takimata sanctus est Lorem ipsum dolor sit amet.</ex:p>
        <ex:p>Lorem ipsum dolor <ex:name>
          <ex:firstname>sit amet</ex:firstname>
          <ex:lastname>consetetur sadipscing</ex:lastname>
        </ex:name>,elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam
erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no
sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero
eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet.</ex:p>
        <ex:p>Lorem ipsum dolor <ex:important>sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt</ex:important> ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero
eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor
invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo
dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</ex:p>
      </ex:text>
    </ex:body>
    <ex:footer>
      <ex:link/>
      <ex:comment/>
    </ex:footer>
  </ex:entry>
</ex:collection>
```

/db/apps/xrxeditor/examples/example2/example.xml

Alternativ zur Referenz über das Attribut doc kann das XML-Dokument auch direkt als Knoten an die Funktion übergeben werden. Dazu muss der XML-Knoten in ein xrxe:doc-Element in das Argumentenelement geschrieben werden.

```
...
<body>
{
  let $xml-doc := doc('/db/apps/xrxeditor/examples/example3/example.xml')/element()
  return
  xrxe:editor(<div xsd="/db/apps/xrxeditor/examples/example3/example.xsd">
    <xrxe:doc>
      {$xml-doc}
    </xrxe:doc>
  </div>
)
</body>
...

```

/db/apps/xrxeditor/examples/example3/example.xql

2.3.3 Das verwendete Schema

XRXEditor verwendet die Regeln und Informationen des XSDs um den Editor zu generieren. Das verwendete Schema unterliegt folgenden Beschränkungen: das Schema muss einen targetNamespace besitzen, sowie den Prefix „xs“ für alle XSD-Elemente verwenden. Der default-Namespace muss dem targetNamespace entsprechen.

2.3.4 Das verwendete XML-Dokument

Das zu editierende XML-Dokument, das in den generierten Editor geladen wird muss ein wohlgeformtes XML-Dokument sein und valide gegen das verwendete Schema sein. Alle Elemente müssen zu einem Namensraum gehören und über einen Präfix verfügen.

3 Konfigurationsmöglichkeiten

Es gibt zwei verschiedene Arten den generierten Editor zu konfigurieren. Das Dokument xrxo-conf.xql bietet mehrere Variablen, die verändert werden können um den Editor zu konfigurieren. Diese Konfiguration wirkt sich auf alle mit XRXEditor erzeugten Editoren der Anwendung aus. Soll eine Konfiguration nur für einen Editor gelten, so ist es möglich dem Argumenten-Element im Funktionsaufruf weitere Attribute mitzugeben, die die Default-Einstellungen in xrxo-conf.xql für den jeweiligen Editor überschreiben. Die Attribute heißen jeweils gleich wie die Variablen in xrxo-conf.xql ohne den Präfix „\$xrxo-conf:default-“. Außerdem gibt es weitere Konfigurationsmöglichkeiten ausschließlich über Attribute

3.1 Konfigurationsattribute

In diesem Abschnitt werden die Konfigurationsparameter beschrieben, die nur als Attribute des Argumentenelements angegeben werden können. Diese überschreiben keine Variable in xrxo-conf.xql da diese nur für spezifische Editoren Sinn ergeben.

save

Soll ein XML-Dokument nach der Bearbeitung speicherbar sein, so kann über das Attribut *save* die URI angegeben werden, zu der das zu speichernde XML-Dokument über HTTP geschickt wird. Im einfachsten Fall ist das die URI des XML-Dokuments. Besitzt das Funktionsargument kein save-Attribut, bekommt der Editor keinen Speicher-Button.

```
...
xrxo:editor(<div
    doc="/db/apps/xrxeditor/examples/example4/example.xml"
    xsd="/db/apps/xrxeditor/examples/example4/example.xsd"
    save="http://localhost:8080/exist/rest/db/apps/xrxeditor/examples/example4/example.xml"
)
...
/db/apps/xrxeditor/examples/example4/example.xql
```



<http://localhost:8080/exist/apps/xrxeditor/examples/example4/example.xql>

save-close

Mit dem Attribut save-and-close kann dem Editor eine weitere URL angegeben werden. Ist das Attribut gesetzt, bekommt der Editor einen weiteren Speichern-Button („Speichern und Weiter“). Wird dieser betätigt wird das Dokument gespeichert und der Benutzer direkt zur angegebenen URL weitergeleitet.

```
...
<xrx:editor><div
    doc="/db/apps/xrxeditor/examples/example5/example.xml"
    xsd="/db/apps/xrxeditor/examples/example5/example.xsd"
    save="http://localhost:8080/exist/rest/db/apps/xrxeditor/examples/example5/example.xml"
    save-close="http://localhost:8080/exist/apps/dashboard/index.html"
/>
...
/db/apps/xrxeditor/examples/example5/example.xql
```



<http://localhost:8080/exist/apps/xrxeditor/examples/example5/example.xql>

node-path

Soll nicht das gesamte XML-Dokument (das gesamte Wurzel-Element) bearbeitbar sein, sondern nur ein tiefer liegendes Element, so kann dieses Element über einen XPath-Ausdruck (ausgehend vom Dokumentknoten) durch das Attribut node-path angegeben werden. Die Steuerungselemente des Editors werden nur für dieses Element generiert und nur dieser Teil des Dokuments ist bearbeitbar.

```
...
<xrx:editor><div
    doc="/db/apps/xrxeditor/examples/example6/example.xml"
    xsd="/db/apps/xrxeditor/examples/example6/example.xsd"
    save="http://localhost:8080/exist/rest/db/apps/xrxeditor/examples/example6/example.xml"
    save-close="http://localhost:8080/exist/apps/dashboard/index.html"
    node-path="/ex:collection/ex:entry"
/>
...
/db/apps/xrxeditor/examples/example6/example.xql
```

entry

+ id

header

did

title

date

12/12/2012

+ number

body

abstract

text

published

p

Select Annotation

Lorem ipsum dolor person sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

p

Select Annotation

Lorem ipsum dolor name fname sit amet fname lname conseetur sadipscing lname name elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

+ p

footer

link

+ location

+ description

+ url

comment

+ comment

<http://localhost:8080/exist/apps/xrxeditor/examples/example6/example.xml>

3.2 Konfigurationsparameter

In diesem Abschnitt werden die Konfigurationsparameter beschrieben, die eine Variable in xrx-conf.xql besitzen. Sie können entweder direkt dort für alle Editoren geändert werden oder für einzelne Editoren durch ein Attribut im Argumentenelement überschrieben werden.

(\$xrxe-conf:default-)services

Ein mit rxeditor erzeugter Editor benutzt Ajax-Anfragen um Teile des Editors nachzuladen oder das Schema während des Bearbeitens abzufragen. Alle Ajax-Anfragen werden an die XQuery Datei xrxe-services.xql gesendet, die Teil des rxeditor-Pakets ist. „services“ gibt die URL an unter der diese Datei zu erreichen ist. Weicht die eXist- und rxeditor-Installation von der hier beschriebenen ab, so muss die URL angepasst werden.

(\$xrxe-conf:default-)save-method

Dieser Konfigurationsparameter gibt die Methode an mit der die Submission beim Aufruf zum Speichern eines XML-Dokumentes ausgeführt wird. Die beiden möglichen Werte sind „POST“ oder „PUT“.

(\$xrxe-conf:default-)place-triggers

Die Schaltflächen zum Speichern des XML-Dokumentes können entweder unterhalb („bottom“), oberhalb („top“), oder unterhalb und oberhalb („top-and-button“) des Editors, über diesen Parameter platziert werden.

```
...
  xrxe:editor(<div
    doc="/db/apps/rxeditor/examples/example7/example.xml"
    xsd="/db/apps/rxeditor/examples/example7/example.xsd"
    save="http://localhost:8080/exist/rest/db/apps/rxeditor/examples/example7/example.xml"
    save-close="http://localhost:8080/exist/apps/dashboard/index.html"
    node-path="/ex:collection/ex:entry"
    place-triggers="top-and-bottom"
  />
...
/db/apps/rxeditor/examples/example7/example.xql
```

[Save](#) | [Save and continue](#)

entry	
+ id	
header	
did	
title	
date	12/12/2012
+ number	
body	
abstract	<p>Lore ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt</p>
text	<p>published</p> <p><input checked="" type="checkbox"/></p> <p>p</p> <p>Select Annotation</p> <p>Lore ipsum dolor person site amet person sit amet, consetetur sadipscing place elitr, sed diam nonumy place eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lore ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p> <p>p</p> <p>Select Annotation</p> <p>Lore ipsum dolor important sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt important ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lore ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.</p> <p>+ p</p>
footer	
link	<p>+ location</p> <p>+ description</p> <p>+ url</p>
comment	<p>+ comment</p>
	Save Save and continue

<http://localhost:8080/exist/apps/xrxeditor/examples/example7/example.xml>

(*\$xrxe-conf:default-*)wrapper

In Fällen in denen das zu bearbeitende XML-Dokument nicht wirklich einem gegen das Schema validen Dokuments entspricht, da das eigentliche Dokument in andere Elemente eingebunden ist, beschreibt wrapper einen XPath-Ausdruck, der zum gegen das Schema validen Element führt.

```

...
xrx:editor(<div
    doc="/db/apps/xrxeditor/examples/example8/example.xml"
    xsd="/db/apps/xrxeditor/examples/example8/example.xsd"
    save="http://localhost:8080/exist/rest/db/apps/xrxeditor/examples/example8/example.xml"
    save-close="http://localhost:8080/exist/apps/dashboard/index.html"
    node-path="/ex:collection/ex:entry"
    place-triggers="top-and-bottom"
    wrapper="/atom:entry/atom:content"
)
...

```

/db/apps/xrxeditor/examples/example8/example.xql

```

<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom">
    <atom:id>http://vdu.uni-koeln.de/ID/123456</atom:id>
    <atom:title>Title</atom:title>
    <atom:published>2010-10-06T12:55:55+02:00</atom:published>
    <atom:updated>2010-10-06T12:55:55+02:00</atom:updated>
    <atom:author>
        <atom:email>daniel.ebner@uni-koeln.de</atom:email>
    </atom:author>
    <app:control xmlns:app="http://www.w3.org/2007/app">
        <app:draft>no</app:draft>
    </app:control>
    <atom:content type="application/xml">
        <ex:collection xmlns:ex="http://vdu.uni-koeln.de/NS/example">
            <ex:entry>
                ...
            </ex:entry>
        </ex:collection>
    </atom:content>
</atom:entry>

```

/db/apps/xrxeditor/examples/example8/example.xml

3.3 Weitere Konfigurationsvariablen

Es gibt Konfigurationsparameter, die ausschließlich durch eine Variable in xrx-conf.xql geändert werden können. Sie können nicht durch ein Attribut im Argumentenelement überschrieben werden. xrx-conf.xql beinhaltet eine Reihe an Variablen, die Texte festlegen, der auf der Benutzeroberfläche angezeigt wird. Diese Texte können über die Variablen geändert werden. Weitere hier nicht beschriebene Konfigurationsvariablen und -funktionen dienen zum Debuggen, Testen oder für spezielle Anforderungen für VDU und Mom-CA.

4 XSD mit Informationen anreichern

XML-Schema Definitionen sind ein Standard zur Beschreibung von XML-Anwendungen. Sie beschreiben das Vokabular und die Struktur der Anwendung. Dieser Regeln sind auch für den erzeugten Editor relevant und er kann anhand eines Schemas generiert werden. Jedoch ist XSD keine Beschreibungssprache für Editoren und so fehlt es an XSD an Möglichkeiten die Erscheinung und den Aufbau des Editors genauer mit XSD zu beschreiben. XSD bietet jedoch über das Element „appinfo“ die Möglichkeit an jeder beliebigen Stelle des Schemas zusätzliche Informationen für eine Anwendung zu beschreiben.

```
<xs:annotation>
  <xs:appinfo>
    ...Hier können Anweisungen für Software stehen...
  </xs:appinfo>
</xs:annotation>
```

XRXEditor bietet einige Konstrukte, die die Software beim Generieren des Editors zusätzlich aus den angegebenen Appinfos herauslesen kann. Diese Infos können in entsprechenden Elementen stehen, die den xrxe-Namespace besitzen.

4.1 Zusätzliche Informationen für den Editor

xrxe:label

Jedes Steuerungselement des Editors besitzt eine Benennung. Diese Benennung ist standardmäßig der Knotename des referenzierten XML-Knotens. Da diese Knotennamen meist nicht sehr menschenlesbar sind (sie bestehen oft aus Abkürzungen oder einzelnen Buchstaben bspw. „p“ für einen Abschnitt) bietet XRXEditor die Möglichkeit XML-Knoten mit zusätzlichen Labeln für den Editor auszustatten.

```
...
<xs:element name="p" maxOccurs="5" minOccurs="3">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Abschnitt</xrxe:label>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexType mixed="true">
    ...
  </xs:complexType>
</xs:element>
...
```

/db/apps/xrxeditor/examples/example9/example.xsd

Save | Save and continue

Eintrag

+ II)

Beschreibende Informationen

Titel

Datum
12/12/2012 ▾

+ Zahl

Inhalt

Zusammenfassung

Lore ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Text

veröffentlicht

Abschnitt

Select Annotation

Lore ipsum dolor Person sit amet Person sit amet, consetetur sadipscing Ortsangabe elitr, sed diam nonumy Ortsangabe eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abschnitt

Select Annotation

Lore ipsum dolor Name > Vorname > sit amet < Vorname Nachname > consetetur sadipscing Nachname Name elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

+ Abschnitt

Zusätzliche Informationen

Link

Or
extern ▾

+ Beschreibung

+ Uniform Resource Locator

Kommentar

+ Kommentar

Save | Save and continue

<http://localhost:8080/exist/apps/xrxeditor/examples/example9/example.xql>

Es gibt drei Stellen an denen eine solche Appinfo für ein Element oder ein Attribut stehen kann. In einer Referenz auf ein Element/Attribut, in der Definition eines Elements/Attributs oder in Beschreibung des Typens eines Elements/Attributs. Im ersten Fall gilt die Appinfo nur für den jeweiligen Kontext der Referenz. Gleiche Elemente in einem anderen Kontext besitzen diese zusätzliche Auszeichnung nicht. Im zweiten Fall besitzen alle Elemente der Definition die zusätzliche Auszeichnung. Im dritten Fall besitzen alle Elemente die den

beschriebenen Typ besitzen die zusätzliche Information, unabhängig vom Namen des Elements. Für RXEditor gilt folgende Hierarchie: Appinfo-Auszeichnungen im Typ werden durch Appinfo-Auszeichnungen in der Elementen-Definition überschrieben. Diese werden wiederum durch Auszeichnungen in der Referenz überschrieben.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xrxe="http://www.monasterium.net/NS/xrxe">
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="firstname">
          <xs:annotation>
            <xs:appinfo>
              <xrxe:label>First Name</xrxe:label>
            </xs:appinfo>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="firstname" type="firstname-type">
    <xs:annotation>
      <xs:appinfo>
        <xrxe:label>Vorname der Person</xrxe:label>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
  <xs:simpleType name="firstname-type">
    <xs:annotation>
      <xs:appinfo>
        <xrxe:label>Vorname</xrxe:label>
      </xs:appinfo>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:maxLength value="10"></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

In diesem Beispiel besäße das firstname-Element im XML-Knoten das Label „First Name“ aus der Referenz.

```
<person>
  <firstname></firstname>
</person>
```

xrxe:hint

Dem Benutzer wird dieser zusätzliche Hinweis bei der Bearbeitung des jeweiligen Knotens angezeigt.

```
<xs:element name="title">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Titel</xrxe:label>
      <xrxe:hint>Titel einfügen</xrxe:hint>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

/db/apps/xrxeditor/examples/example10/example.xsd



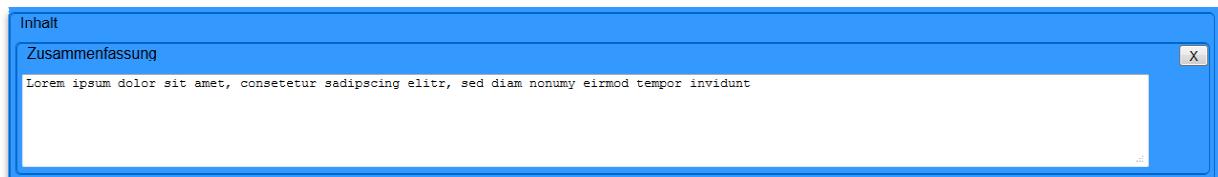
<http://localhost:8080/exist/apps/xrxeditor/examples/example10/example.xql>

xrxe:content-control

Angabe welches Steuerungselement für den Knoten verwendet werden soll. Hauptsächlich um ein mehrzeiliges Eingabefeld (textarea) gegenüber dem Standard-Steuerungselement einzeiliges Eingabefeld (Input) für die freie Eingabe von Text zu erzwingen.

```
<xs:element name="abstract" minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Zusammenfassung</xrxe:label>
      <xrxe:content-control>textarea</xrxe:content-control>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

/db/apps/xrxeditor/examples/example11/example.xsd



<http://localhost:8080/exist/apps/xrxeditor/examples/example11/example.xql>

xrxe:constraint

Eine zusätzliche Einschränkung des Knotenwertes mittels eines XPath-Ausdrucks

```
<xs:element name="abstract" minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Zusammenfassung</xrxe:label>
      <xrxe:content-control>textarea</xrxe:content-control>
      <xrxe:constraint>string-length(.) le 130</xrxe:constraint>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

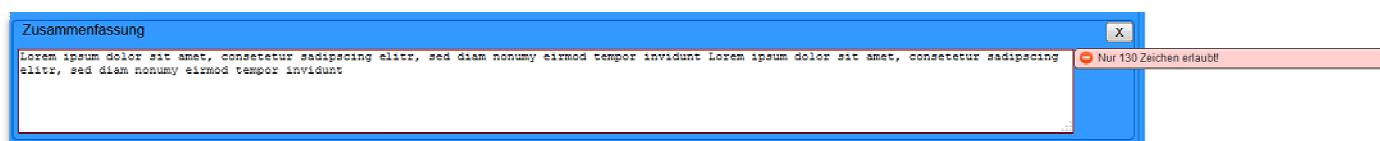
/db/apps/xrxeditor/examples/example12/example.xsd

xrxe:alert

Diese Warnung erscheint für den Benutzer falls seine Eingabe für diesen Knoten aufgrund des Knotentyps bzw. anderer Einschränkungen nicht valide ist.

```
...
<xs:element ref="number" minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Zahl</xrxe:label>
      <xrxe:alert>Nur Ziffern erlaubt</xrxe:alert>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
...
<xs:element name="abstract" minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Zusammenfassung</xrxe:label>
      <xrxe:content-control>textarea</xrxe:content-control>
      <xrxe:constraint>string-length(.) le 130</xrxe:constraint>
      <xrxe:alert>Nur 130 Zeichen erlaubt!</xrxe:alert>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
...
```

/db/apps/xrxeditor/examples/example13/example.xsd



<http://localhost:8080/exist/apps/xrxeditor/examples/example13/example.xql>

xrxe:default-value

Mit XSD kann zwar ein Default-Wert für Attribute angegeben werden, jedoch nicht für Elemente. Mit xrxe:default-value kann der Standardwert, den ein Knoten beim Einfügen bekommt für den Editor auch für Elemente angegeben werden.

```
<xs:element ref="number" minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Zahl</xrxe:label>
      <xrxe:alert>Nur Ziffern erlaubt</xrxe:alert>
      <xrxe:default-value>0</xrxe:default-value>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

/db/apps/xrxeditor/examples/example14/example.xsd

A screenshot of the XRX Editor interface. It shows a single input field with a blue header bar labeled "Zahl". The input field contains the number "0". Below the input field is a horizontal scroll bar.

<http://localhost:8080/exist/apps/xrxeditor/examples/example14/example.xql>

xrxe:children-control

Sollen die Kindelemente eines Elements auf verschiedene Reiterkarten verteilt werden kann dies durch den Wert „switch“ erreicht werden.

```
<xs:element name="entry" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:appinfo>
      <xrxe:label>Eintrag</xrxe:label>
      <xrxe:children-control>switch</xrxe:children-control>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="header">
        ...
      </xs:element>
      <xs:element name="body">
        ...
      </xs:element>
      <xs:element name="footer">
        ...
      </xs:element>
    </xs:sequence>
  ...
</xs:complexType>
</xs:element>
```

/db/apps/xrxeditor/examples/example15/example.xsd



<http://localhost:8080/exist/apps/xrxeditor/examples/example15/example.xql>

xrxe:mixed

Der Wert „false“ dient dazu eine strukturierte Eingabemaske für Knoten die gemischten Inhalt haben dürfen zu erzwingen, anstatt ein Textannotationsfeld anzubieten

xrxe:min-occur

Das Minimum, wie oft ein Element vorkommen muss, kann gegenüber dem offiziellen XSD-Minimum erhöht werden, ohne die eigentlichen Regeln des XSDs zu ändern. Ist das Minimum erreicht, werden die Lösch-Buttons ausgeblendet, so dass der Benutzer diese Anzahl nicht unterschreiten kann.

xrxe:max-occur

Das Maximum, wie oft ein Element vorkommen kann, kann gegenüber dem offiziellen XSD-Maximum verringert werden, ohne die eigentlichen Regeln des XSDs zu ändern. Ist das Maximum erreicht, werden die Einfüge-Buttons ausgeblendet, so dass der Benutzer diese Anzahl nicht überschreiten kann.

```

<xs:element minOccurs="1" maxOccurs="10" name="comment">
  <xs:annotation>
    <xs:appinfo>
      <xrx:label>Kommentar</xrx:label>
      <xrx:min-occur>3</xrx:min-occur>
      <xrx:max-occur>5</xrx:max-occur>
    </xs:appinfo>
  </xs:annotation>
  <xs:complexType mixed="true"/>
</xs:element>

```

/db/apps/xrxeditor/examples/example16/example.xsd

xrx:relevant

Kann für Elemente, Attribute, Gruppen und Attributgruppen angegeben werden. Der Wert „false“ lässt den Editor den Knoten oder die komplette Gruppe ignorieren. In xrx-conf.xql können auch alle Attribute durch xrx:conf-default-attribute-relevant="false" als nicht relevant gesetzt werden. Durch den Wert xrx:relevant können dann einzelne Attribute wieder als, für den Editor relevant, markiert werden. Analog gibt es Variablen für Attributgruppen, Elementgruppen und Elemente.

```

<xs:attribute name="id">
  <xs:annotation>
    <xs:appinfo>
      <xrx:relevant>false</xrx:relevant>
    </xs:appinfo>
  </xs:annotation>
</xs:attribute>

```

/db/apps/xrxeditor/examples/example17/example.xsd



<http://localhost:8080/exist/apps/xrxeditor/examples/example17/example.xql>

xrx:menu-item

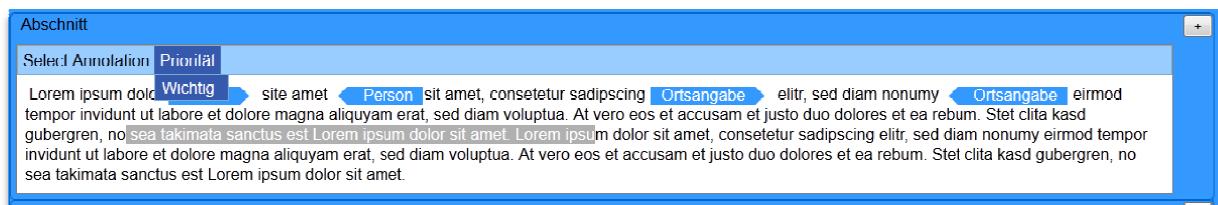
Kann ein Element als Annotation in einem Textannotationsfeld vorkommen (Das Element ist Kindelement eines Elements, das gemischten Inhalt besitzen darf), so kann über xrx:menu-item das Untermenü angegeben werden, in dem die Annotationsmöglichkeit im Textannotationsfeld angezeigt wird. Ist dies nicht angegeben erscheint ein Element im Standard-Untermenü „Select Annotation“. Dieses kann in xrx-conf.xqm geändert werden.

```

...
<xs:element name="important" minOccurs="0">
  <xs:annotation>
    <xs:appinfo>
      <xrx:label>Wichtig</xrx:label>
      <xrx:menu-item>Priorität</xrx:menu-item>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
...

```

/db/apps/xrxeditor/examples/example18/example.xsd



<http://localhost:8080/exist/apps/xrxeditor/examples/example18/example.xql>