

# Creation of the adaptive graphic Web interfaces for input and editing data for the heterogeneous information systems on the bases of XML technology

A. Mukhitova and O. Zhizhimov

Novosibirsk State University, Novosibirsk, Russia  
Institute of Computational Technologies SB RAS, Novosibirsk, Russia

**Abstract** The work is devoted to an appliance of constructing adaptive technology of graphic administrative WEB-interface for the resolution of integration problems of heterogeneous information resources, on the ground of the XSD application database schemes with the appliance of XSLT-transformations. An approach is described for elaboration, as well as an example of implementation, of adaptive entry model and data editing in the way of created editor prototype of the XML-records. The described methodology has quite general ways of usage and can be applied for constructing adaptive graphic WEB-interfaces, enabling to generate shippable HTML-forms for entering and editing data.

**Keywords:** :adapting graphic administrative and users web-interfaces, integration of heterogeneous data, XML editor, SRU, XML, XSD, XSLT-transformations.

The paper deals <sup>1</sup> with the development of technologies for creating adaptive graphical interfaces for heterogeneous information systems. Heterogeneous distributed applications should consist of user and administrative interfaces (graphic WEB-interfaces) supplying an opportunity to govern data from various informational sources. Interfaces, adjusting to structure and functionality of informational resources, are adaptable. Each of a particular informational resource, in general, has quite narrow range of features from the potentially existing meanings. Therefore, attraction of additional data on a particular informational recourse is required, while choosing the component governing features for various informational resources.

The inputting information about functional qualities of each data sources is essential for implementation of adaptable interfaces in mentioned point of client interfaces. Depending on the technology access, used to obtain informational assets, that sort of data can be received and processed. The constructing adaptive

---

<sup>1</sup> Work is performed within the Integration Project of the Siberian Branch of the Russian Academy of Science (AAAAA 18-118022190008-8), the Project on basic scientific research (AAAA-A17-117120670141-7) and RFBR 18-07-01457-a.

technology of client interfaces was considered in the work in order to manage the information search with the method of displaying extracted data using Z39.50 [1] and SRW/SRU [2],[3] on the basis of Explain services [3], [4], [5] in the various modifications. The implementation of these adaptable interfaces for the ZooSPACE [6],[7] platform was also illustrated. Constructed applications enable to automatically adjust graphic interfaces to possibilities of one or the other information resource:

- supported sets of searching attributes;
- supported combinations of searching attributes;
- supported schemes, patterns and sets of elements.

Despite the fact that there is a considerable number of web-resources and services, files of documents and libraries, which are applied as a data exchange format XML, the usage of JSON should be considered in constructing new distributed informational systems. At the present time, this format is developing rapidly and is easy for implementation in systems of data exchanging. For the reason of being simple and easy to operate data, JSON is better to use on the client's side, while the XML technologies are more convenient on the side of a server. In view of the specific character of the given problem the decision was made in favour of XML-technology.

The descriptions of the constructing rules of XML-records structure are essential in the use of XML format to present structured information, i.e. descriptions of data scheme. In general, the rules for XML are formulated in terms of XSD [8]-[10] and they present XML structure which can be processed with standard ways, for example XSLT [11].

The question about where the full description of potential structure of the derived record can be obtained arises in the process of extracting record from a particular informational source in heterogeneous informational system and presenting the record in the XML format for editing. The following options are possible [12]:

- The XML record, derived for editing, includes a reference on the applied XSD data scheme in the form of URL with schemaLocation as an attribute in determination of employed namespaces. It is usually contained in the XML record root element. In that case the issue of receiving XSD is solved in a trivial manner.
- The XML, record derived for editing, includes the namespace identification (URI), though it does not include a reference on the applied XSD data scheme in the form of URL. In that situation the informational system should be requested to provide the XSD in the use of namespace identification. For the ZooSPACE platform the similar request can be processed with Explain service.
- The XML record, derived for editing, does not include definitions of namespaces. In this case the informational system should be requested to provide the XSD (as a default) by the name of informational resource (database), or by using the XSD, which before corresponded to the scheme requested in an inquiry formation for extracting data.

In any of the listed cases the following steps are necessary for graphic interfaces initialisation of data modifications:

- The data scheme description in the way of the XML structure in accordance with the XSD rules.
- The XML structure consisting extracted data for editing (not required for creating a new record)
- The description of entity generation patterns of graphic interface (not necessary).
- The description of entity generation patterns of graphic interface in accordance with the XSD rules and with elements of editing the XML record.
- In these conditions, the XSLT conversion rules can be the rules applied to XSD.

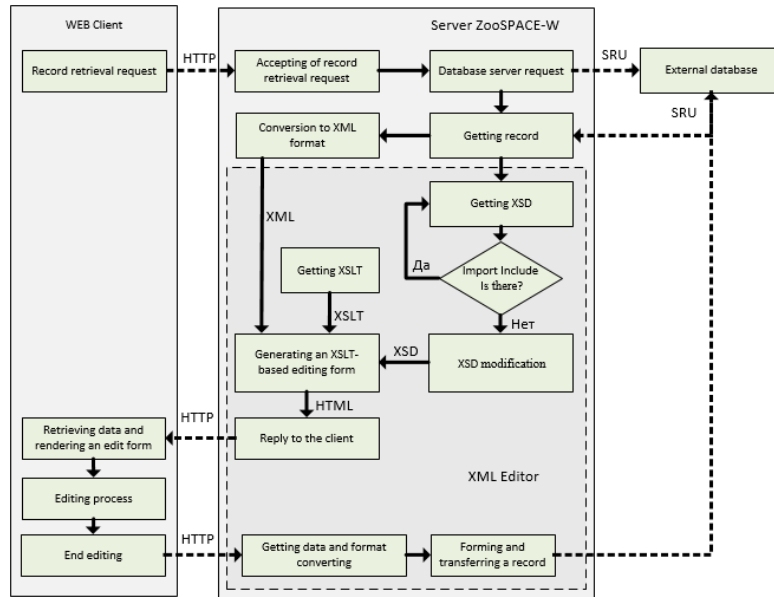
The algorithm is presented in the Figure 1, as an illustration of work of the XML records prototype adaptive editor in the format of client-server, built in WEB server of the ZooSPACE (ZooSPACE-W) platform. The XML editor is relevant to an area restricted by a dashed line for server side. As for the client part, the beforehand prepared HTML form to input and/or edit data is provided. In these conditions, the form already contains the all needed tools (java scripts) for correct data input, which includes:

- A script for duplicating elements, the repetition of which is possible according to the XSD.
- A script for removing elements, providing that the removing is possible according to the XSD.
- A script for checking the accuracy of data entry, if there is a relevant pattern in the way of regular expression in the XSD.
- A script for hiding- revealing any data elements in the form of editing.

It should be taken into account that the XSD data scheme definitions can contain references to other XSD data scheme definitions, which complement definitions both in the current namespace (element `xsd:include`), and in the other namespaces (element `xsd:import`). Therefore the initial XSD structure, before being processed by the XSLT processor requires modifying to register extra definitions.

The editor of the XML records operating principle, in format of client-server built in WEB server, can be described as follows:

- as for the client part, the beforehand prepared HTML form to input and/or edit data is provided. In these conditions the form already contains the all needed tools (java scripts) for correct data input;
- generation of editing forms occurs on the server side with the XSLT method of transformations of the modified XSD structure. At the beginning of the process an empty editing form is produced (without data). As soon as the XSLT processor has completed its action, the empty form is filled with record data in XML format.



**Figure 1.** The work algorithm of editor process BD record in the heterogeneous system with example of the ZooSPACE-W

For generation of empty form of editing (refer to Figure 2) the following rules are performed:

- The frame indicating the identification of data scheme is generated.
- The file of documents (annotation) for data scheme is generated.
- For each specified data element in XSD the following is generated:
  - the frame indicating the element name and its location (in the XPath pattern) in the XML record structure;
  - The key button of hiding-revealing element in a form of editing;
  - The file of documents (annotation), if any, with an indication of a language;
  - The nested elements ( for complex);
  - The field of entry element definition (for simple);
  - Names and data entry fields for each of potential attributes;
  - Key buttons for deleting (if allowed) or duplicating (if allowed) elements;
- The following key buttons are generated:
  - «Record»- for storage a result of editing;
  - «Clear»-for regeneration of empty editing form;
  - «Close»- for closing editing form without data storage.

The type of data and the placed restrictions are taken into account in the process of generation of data entry fields. In particular, the field of entry elements and attributes are presented with a list of dropdown definitions (refer to Figure 3) if there is XSD definitions such as:

Figure 2. Graphical interface of XML record editor

```
<xsd:simpleType name="recordTypeType">
  <xsd:restriction base="xsd:NMTOKEN">
    <xsd:enumeration value="Bibliographic"/>
    <xsd:enumeration value="Authority"/>
    <xsd:enumeration value="Holdings"/>
    <xsd:enumeration value="Classification"/>
    <xsd:enumeration value="Community"/>
  </xsd:restriction>
</xsd:simpleType>
```

If the XSD element contains indication for a pattern (RegEx), for example:

```
<xsd:simpleType name="indicatorDataType" id="ind.st">
  <xsd:restriction base="xsd:string">
    <xsd:whiteSpace value="preserve"/>
    <xsd:pattern value="[\da-z ]{1}"/>
  </xsd:restriction>
</xsd:simpleType>
```

In that case, the access to checking function of correspondence with a pattern of data entry in the form of editing is generated, that is XSLT code will be performed:

...

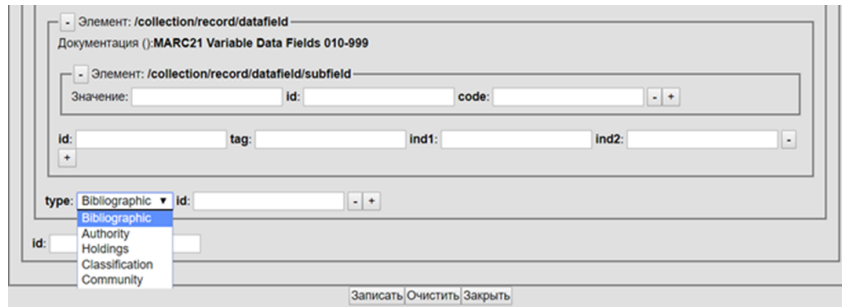


Figure 3. Graphical XML editor: data entry fields

```
<xsl:for-each select="xsd:simpleType/xsd:restriction/xsd:pattern">
  <xsl:attribute name="onChange">
    <xsl:text>e_change(this, /</xsl:text>
    <xsl:value-of select="@value"/>
    <xsl:text>/);</xsl:text>
  </xsl:attribute>
</xsl:for-each>
...

```

Which in turn generates the forms of elements

```
<input type="text"
  onChange="e_change(this, /[\da-z ]{1}/);" . . . />

```

A problem of recursive definitions arises from the described approach in XML formation on the ground of XSD. Recursivity may occur in the appliance of references to types and names. A fragment of a recursive determination is provided in the scheme with the help of the XSD.

```
\begin{flushleft}
<xsd:complexType name="organization">
  <xsd:sequence>
    <xsd:element name="id" type="int"/>
    <xsd:element name="name" type="string"/>
    <xsd:element name="sub-org" type="tns:organization"/>
  </xsd:sequence>
</xsd:complexType>
  <xsd:element name="region">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="id" type="int" />
    <xsd:element ref="tns:region" />
  </xsd:sequence>

```

```

</xsd:complexType>
</xsd:element>
  <xsd:element name="record">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="id" type="int" />
    <xsd:element name="organization" type="tns:organization" />
    <xsd:element ref="tns:region" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>

```

The XML elements with unrestricted length of Xpath are possible:

```
/record/organization/sub-org/sub-org/sub-org . . .
```

```
/record/region/region/region/region . . .
```

The attachment number control can be used for eliminating the endless number of item attachments in generation of graphic interfaces of editing records and for restricting them in accordance with the current demand. The list of processed elements, XSD (rules), is depicted by editor prototype in the table 1.

At the present time, the created prototype of mentioned the XML records adapting editor is being tested for various data charts. Furthermore, it is planned to boost its functional capabilities in the area of widening a list of supported elements of the XSD and JSON. At the end of the testing the editor will be built in the ZooSPACE-W subsystem of the ZooSPACE platform.

## References

1. ANSI/NISO Z39.50-2003. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. - NISO Press, Bethesda, Maryland, U.S.A. — Nov 2002. — ISSN: 1041-5653. — ISBN: 1-880124-55-6.
2. SRU- Search/Retrieve via URL / The Library of Congress, 26.12.2013, <http://www.loc.gov/standards/sru/>
3. SRU - Explain Operation / The Library of Congress, 6.09.2013, <http://www.loc.gov/standards/sru/explain/>
4. Zhizhimov, O.L. : Explain Services on ZooSPACE Platform and Adaptive User Interfaces. CEUR Workshop Proceedings. - Vol.1536.,- ISSN 1613-0073. pp.30–36 - <http://ceur-ws.org/Vol-1536/paper4.pdf>.(2015)
5. ZeeRex: The Explainable "Explain"Service, <http://zeerex.z3950.org>
6. Zhizhimov, O.L., Fedotov, A.M., Shokhin, Y.I. The ZooSPACE platform- access organization to various distributed resources. Digital libraries: The Russian scientific e-magazine. - Vol.17. - № 2. - ISSN 1562-5419.(2014)
7. Zhizhimov, O.L., Lobykin, A.A., Turchanovskiy, I.Y., Panshin, A.A., Chudinov, S.A. Computer -assisted acquisition system of statistics event information in distributed information system. Vestnik NSU. Ser.: The Information technology. - Vol.11. - № 1. - pp.42–52. - ISSN 1818-7900.(2013)

**Table 1.** Supported XSD elements

Element	Attribute
annotation	
appinfo	
attribute	name, ref, type, use
choice	
complexContent	
complexType	name
documentation	
element	name, ref, type, substitutionGroup, maxOccurs, minOccurs
extension	base
group	name, ref, maxOccurs, minOccurs
import	namespace, schemaLocation
include	schemaLocation
list	itemType
restriction	base
schema	attributeFormDefault, elementFormDefault, blockDefault, finalDefault, targetNamespace, version, xmlns
sequence	maxOccurs, minOccurs
simpleContent	
simpleType	name
union	memberTypes
unique	



8. XML Schema Part 0: Primer Second Edition: W3C Recommendation 28 October 2004. <http://www.w3.org/TR/xmlschema-0/>
9. XML Schema Part 1: Structures Second Edition: W3C Recommendation 28 October 2004. <http://www.w3.org/TR/xmlschema-1/>
10. XML Schema Part 2: Datatypes Second Edition: W3C Recommendation 28 October 2004. <http://www.w3.org/TR/xmlschema-2/>
11. XSL Transformations (XSLT) Version 2.0: W3C Recommendation 23 January 2007. <http://www.w3.org/TR/xslt20/>
12. Mukhitova, A.A., Zhizhimov, O.L., Adaptive technologies in the context of designing administrative graphic interfaces for heterogeneous information systems of inputting and editing data // XVI Russian conference of 'The distributed informational-computational resources. Science in digital economy' (DICR-2017): Proceedings of XVI All-Russia Conference (4-7 of December, 2017) Novosibirsk / under the editorship of Zhizhimov O.L., Fedotov A.M.- Novosibirsk: ICT SB RAS.- pp.142-149.- ISBN: 978-5-905569-10-4.- [http://elib.ict.nsc.ru/jspui/bitstream/ICT/1467/20/paper16.pdf\(2017\)](http://elib.ict.nsc.ru/jspui/bitstream/ICT/1467/20/paper16.pdf(2017))