Application of XML-technologies in the Construction of Input and Editing Data Model in the Adaptive Administrative Graphical Web Interfaces for Heterogeneous Information Systems

© Aigul Mukhitova
Novosibirsk State University,
Novosibirsk, Russia
mukhitova.aigul@gmail.com

© Oleg Zhizhimov
Institute of Computational Technologies
SB RAS, Novosibirsk Russia
zhizhim@mail.ru

Abstract. Graphical user and administrative interfaces have become an integral part of the development of heterogeneous information systems. Interfaces, which are capable to adapt to the structure and functionality of information resources, will be adaptive. Today, most of data exchanged between information systems is done with the help of the XML document. It became necessary to develop a universal model of data input and editing in order to adapt graphical web interface to changes in requirements and implementation technologies.

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: adaptive graphical user and administrative web-interfaces, integration of heterogeneous data, cross-infrastructure interoperability, data representation, new data analysis methods development, XML editor, SRU, SRW, XML, XSD, XSLT-transformations.

1 Introduction

To manage heterogeneous information in a heterogeneous information system, it became necessary to develop a universal model for constructing an adaptive graphical web interface for data input and editing. Such interfaces, which are capable of adapting to the structure and functionality of information resources, will be adaptive.

For the construction of adaptive administrative WEB-interfaces of data input and editing, the most suitable format of all available structured formats extracted from the corresponding information sources is the XML format. It is this format that allows the end user to work with a large amount of data from more heterogeneous sources and get the best result from working with this data.

To represent structured information, the XML format, it is essential to have a description of schema the XSD data schema. Typically, these rules are formulated in terms of XSD and represent an XML structure that can be handled by standard tools, such as XSLT.

Thus, graphical web interfaces based on XML technologies allow you to display any structure of a file represented in XML format.

The paper deals 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.

2 The methodology for constructing adaptive graphic WEB-interface

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 technology of uses 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 [3-5] in the various modifications. The implementation of these adaptable interfaces for the ZooSPACE [6, 7] platform was also illustrated. Constructed applications

Proceedings of the XX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2018), Moscow, Russia, October 9-12, 2018

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, less 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 Schema Location 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 initialization 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 applied to XSD.

The functional diagram is presented (see 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.

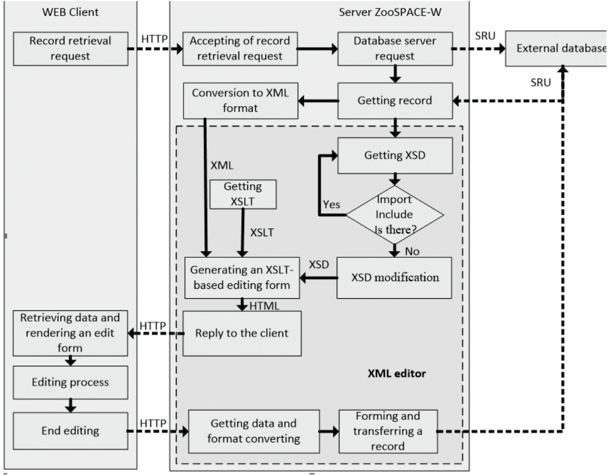


Figure 1 The functional diagram of XML editor for the heterogeneous system ZooSPACE-W

3 Implementation and validation

The XML editor is relevant to an area restricted by a dashed line for server side (see Figure 1). 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.

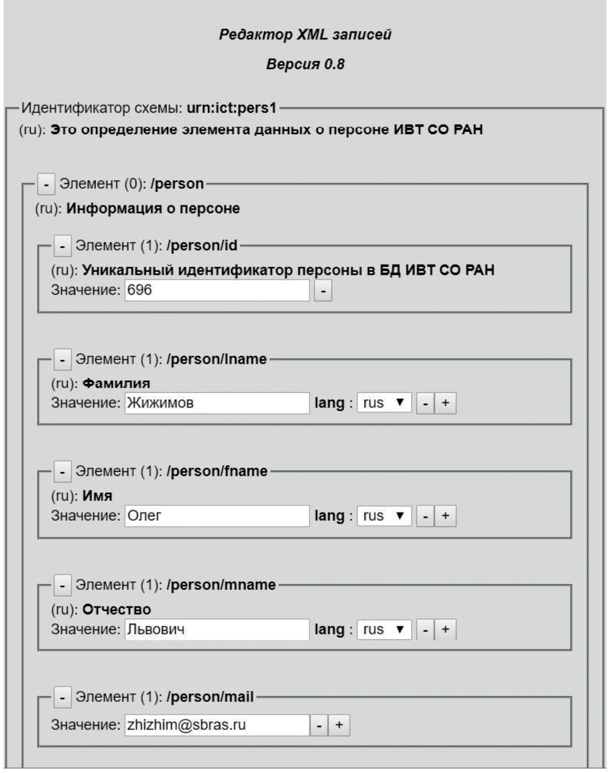
The editor of the XML records operating principle, in format of client-server built in WEB server, can be described as follows:

- 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.

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



Редактор XML записей

Версия 0.8

Идентификатор схемы: urn:ict:pers1

(ru): Это определение элемента данных о персоне ИВТ СО РАН

- Элемент (0): /person

(ru): Информация о персоне

- Элемент (1): /person/id

(ru): Уникальный идентификатор персоны в БД ИВТ СО РАН

Значение: 696 -

- Элемент (1): /person/lname

(ru): Фамилия

Значение: Жижимов lang : rus ▼ - +

- Элемент (1): /person/fname

(ru): Имя

Значение: Олег lang : rus ▼ - +

- Элемент (1): /person/mname

(ru): Отчество

Значение: Львович lang : rus ▼ - +

- Элемент (1): /person/mail

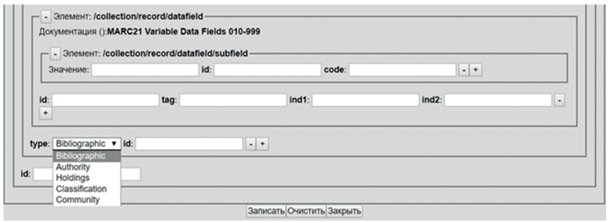
Значение: zhizhim@sbras.ru - +

Figure 1 Graphical interface of XML editor

- 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 (see Figure 3) if there is XSD definitions such as:

```
<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>
```



- Элемент: /collection/record/datafield

Документация ():MARC21 Variable Data Fields 010-999

- Элемент: /collection/record/datafield/subfield

Значение: id: code: - +

id: tag: ind1: ind2: - +

type: Bibliographic ▼ id: - +

id:

Figure 2 Graphical XML editor: data entry fields

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:

```
…
<xsl:for-each
select="xsd:simpleType/xsd:restriction/xsd:patt
ern">
    <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. Recursiveness 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.

```
<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.

4 Conclusion

An approach for elaboration and implementation of adaptive entry model and data editing has been presented. The created prototype of the described adaptive XML editor enable to import any XML-data and transform their structure efficiently and simply, while the same processing process will allow transforming the initial data of any structure without any modification of the program code.

At the present time, the created prototype of the described adaptive XML editor is being tested for various data schemes. It is also planned to increase its functionality capabilities in terms of expanding the list of supported elements of the XSD and JSON.

Upon completion of testing, the editor will be integrated into the ZooSPACE-W subsystem of the ZooSPACE platform. Moreover, it can be used by users as an independent functional system as an editor for working with XML-format files.

Acknowledgments. Work is performed within the Integration Project of SB RAS (AAAA-A18-118022190008-8), Project for basic scientific research (AAAA-A17-117120670141-7) and RFBR project № 18-07-01457-a.

References

[1] ANSI/NISO Z39.50-2003. Information Retrieval (Z39.50): Application Service Definition and Protocol Speciation. 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. http:// www.loc.gov/standards/sru. Accessed 29 July 2016.

[3] SRU-Explain Operation. The Library of Congress. http://www.loc.gov/standards/sru/explain. Accessed 6 September 2013

[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 scien-tic 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)

[8] XML Schema Part 0: Primer Second Edition: W3C Recommendation. http://www.w3.org/TR/xmlschema-0. Accessed 28 October 2004

[9] XML Schema Part 1: Structures Second Edition: W3C Recommendation. http://www.w3.org/TR/xmlschema-1. Accessed 28 October 2004.

[10] XML Schema Part 2: Datatypes Second Edition: W3C Recommendation. http://www.w3.org/TR/xmlschema-2. Accessed 28 October 2004.

[11] XSL Transformations (XSLT) Version 2.0: W3C Recommendation http://www.w3.org/TR/xslt20. Accessed 23 January 2007.

[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 information-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)