Implementation of an adaptive model of input and editing information based on XSLT transformations for heterogeneous data

Aigul Mukhitova
Novosibirsk State University
Novosibirsk, Russian Federation
mukhitova.aigul@gmail.com

Oleg L. Zhizhimov
Institute of Computational Technologies
Siberian Branch of the Russian Academy of Sciences (ICT SB RAS)
Novosibirsk, Russian Federation
zhizhim@mail.ru

***Abstract*—Adaptability, the ability of interfaces to adapt to the structure and functionality of information sources is one of the main features of the information system's intelligence. Development of adaptive graphic web interfaces based on XML-technologies allows to visualize any structure of the XML-format file for further manipulation of data input and editing. The paper deals with the technology of constructing an adaptive graphical administrative WEB-interface for data input and editing in a heterogeneous information environment based on the use of XSD data schema definitions with the use of XSLT transformations. An example of implementation of the adaptive model of input and editing of information in the form of the created prototype of the XML-records editor is given. This editor, in the client-server architecture, on the server side generates an empty editing form by converting the modified XSD structure by XSLT method and then provides the client-side ready HTML-form with all the necessary tools (java scripts) for correct input and/or editing of heterogeneous data.**

***Keywords*—adaptive graphical user and administrative web-interfaces, integration of heterogeneous data, data representation, new data analysis methods, XML, XSD, XSLT-transformations, XML-editor.**

I. Introduction

To create responsive web-based administrative interfaces for data input and editing, the XML format is the most appropriate format from all available structured formats extracted from the relevant information sources. Namely this format allows the final user to work with a large number of heterogeneous data from many heterogeneous sources, as well as it has a good data type and allows the definition of its structure.

XML is more dynamic and allows to easily generate new data schemas and rules of transition between them, which are formulated in the same language (XSLT-transformation), and is supported by a large number of software manufacturers. The XML standard includes support for Unicode encodings, which makes it possible to use several different languages in the same application at the same time. XML provides for the transfer of data, such as graphics or audio/video, without which the information environment is unthinkable. A particularly important advantage of XML is its integration with the web environment, as well as platform independence. All this makes XML a de facto standard for data exchange.

It became necessary to create fundamentally new high-level applications based on the integration of information technologies and ensuring the integration of heterogeneous information resources. This direction is actively developing in many scientific centers of different countries and is associated with the creation of information systems for XML-messages exchange, functioning in the Web environment.

There are a number of works that have been analyzing the problem of adaptive web application interfaces, giving possibility to input and output information in xml format. In work [1] is represented model and algorithm for constructing a syntax-directed editor of xml descriptions. The main purpose of intelligent XML editors is to provide a high-level user interface. Such interface must ensure full information on possible user actions at each point of dialogue and on appropriate XML tags. The editor can be built around a logical processor — realization of a special abstract automaton. The control table of this automaton is built using the contents of the DTD file and based on calculation of the modified Wirth—Weber ratios. This approach allows the user to create syntactically correct XML files without knowledge of DTD [1]. The author of another paper [2] proposes the concept of XML documents with a built-in dynamic model. A general structural diagram of this model is presented, and a method for its interpretation is described. The architecture of the developed software tool for creating and maintaining dynamic XML documents is discussed, and a brief overview of its modules is given. One of the last works in this direction is the work [3], where deals with modern and previously proven in practice means to verify the structure of the documents to the appropriate document description scheme.The problem of using schemes for validating (validating) XML documents in

the model of a situationally-oriented database (SODB) with the help of an XML-based document structure. As a result, the ability to work with third-party SODB extends, XML documents stored on thirdparty services are downloaded and used in a web application with a preliminary validation check. Thus, the dynamic model is expanded with specifications for connecting circuits to monitor the data being downloaded. As a result, the SODB allows not only to process data, but also to control the data downloaded by users or from third-party RESTful-services. The obtained results are discussed on the example of the XML file of the information system about dissertational councils [3]. We can state the absence in the open scientific information space of domestic and foreign analogues of our proposed adaptive model of input and editing information based on XSLT transformations for heterogeneous data.

A variety of distributed information systems should provide the ability to manage data from heterogeneous information sources, i.e. generate administrative and user interfaces that provide the ability to manage heterogeneous database (input and editing).

For homogeneous information sources with a fixed record structure, the task of co-building and editing data is simple. While for heterogeneous sources of information (with an arbitrary recording structure) there is a need to use adaptive technologies for building graphical interfaces with the required functionality [4]. XML is a useful tool for structurally describing data, but it is not intended to represent data visually. For this the XML data must be converted into another form that is easy for the user to view and edit through a browser, such as an HTML document.

Such conversions are performed using constructs defined by the XSLT language. XSLT transformations are used to present information to the user, and as a means to convert XML documents to other formats.

XSLT describes the rules for converting the source structure of an XML document into a destination document (XML, HTML, Text). The final structure can be modified in contrast to the structure of the original tree during the construction process, when the elements of the original tree can be reordered and filtered, as well as by adding new elements. Each of the specific information resources, as a rule, has a rather limited range of possible formats and schemes of providing data and their possible values. However, due to the heterogeneity of data sources, it is necessary to attract additional information about a particular information resource when selecting components that regulate data processing of various information resources.

The development of adaptive web-based graphical interfaces based on XML technologies allows to visualize any structure of the XML-format file for the possibility of further manipulation of data input and editing.

The technology of creation of the adaptive graphic WEB-interface realizing model of input and editing information for heterogeneous data, built-in heterogeneous information system will be considered below.

This assumes that the data can be extracted from the appropriate information sources in XML format. As a matter of principle, other structured formats can be used. The system needs to be activated modules, which convert data into XML format and back, because the XML format is described in the most appropriate technology for building WEB interfaces to edit data.

II. Description of technology

In paper [4] described technology of constructing adaptive user interfaces for controlling the search of information and method of displaying retrieved information by using the Z39.50 [5] and SRW/SRU [6] on the basis of services Explain [7-8] in its various modifications. The implementation of these adaptive interfaces for the ZooSPACE [9-10] platform is also demonstrated. The ZooSPACE complex is based on several loosely coupled distributed subsystems providing configuration (ZooSPACE-L), access to resources (ZooSPACE-Z), user and administrative web-interfaces (ZooSPACE-W), statistics collection (ZooSPACE-S) and monitoring (ZooSPACE-M) of the whole system [9]. The implementation of adaptive user and administrative interfaces in the ZooSPACE-W subsystem should minimize user actions for searching, viewing and editing information from heterogeneous sources.

Depending on the technologies used to access information resources, input information about the functional properties of each data source should be obtained.

To present structured XML information, it is important to have an XSD (XML Schema) data schema description. XML Schema technology allows you to check the correctness of the XML document according to the described rules and apply code generation tools for various web programming tools, which speeds up the application development process. In general, the rules for XML are formulated in terms of XSD [11-13] and they present XML structure which can be processed with standard ways, for example XSLT [14]. 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 [15]:

- 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.

For all of the above methods for initializing GUI data conversion are needed:

1) a description of the data schema in the form of an XML structure in accordance with the rules of XSD;
2) an XML structure containing the extracted data for editing (not required to create a new record);
3) a description of the styles of formation of elements of the graphical interface (optional);
4) description of templates for generating graphical interface objects in accordance with XSD rules and XML record editing elements. Under these conditions, XSLT transformation rules may apply to XSD.

The functional diagram is presented (see Figure 1), as an illustration of work algorithm of the XML records prototype adaptive editor in the format of client-server, built in WEB server of the ZooSPACE (ZooSPACE-W) platform.

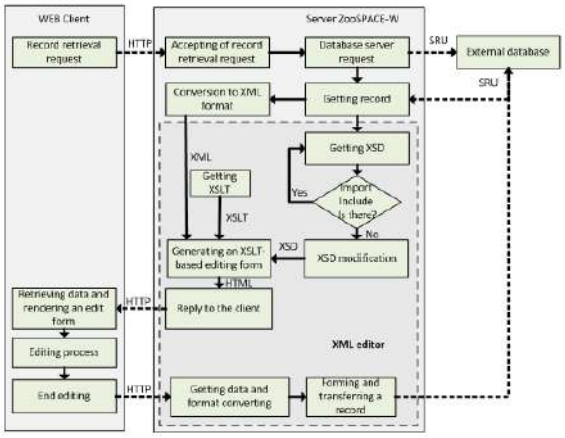

Figure 1. The functional diagram of XML-editor in the heterogeneous system ZooSPACE-W

III. 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 checking the accuracy of data entry, if there is a relevant pattern in the way of regular expression in the XSD;
- a script for removing elements, providing that the removing is possible according to the XSD;
- a script for duplicating elements, the repetition of which is possible according to 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 [15]:

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;
2) 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:

- 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);

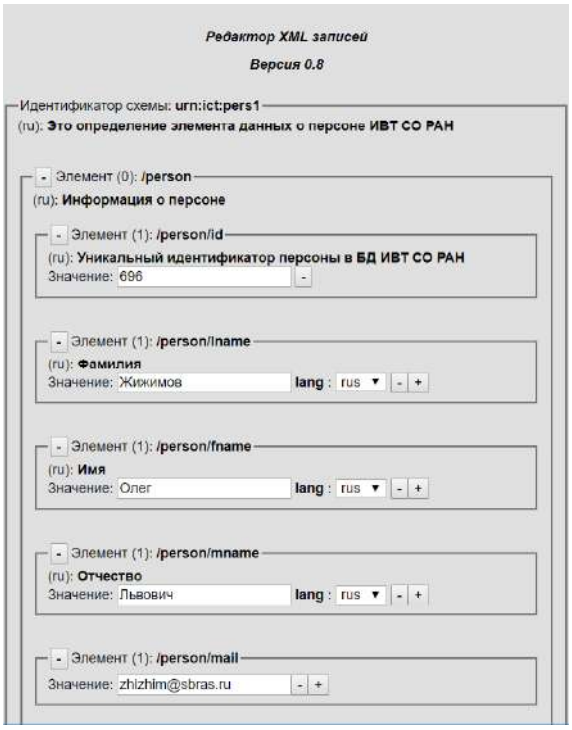

Figure 2. Graphical interface of XML-editor

  - 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:

Figure 3. Graphical XML-editor: data entry fields

```
<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:

```
  ...
<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 (see Figure 4). 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="org">
  <xsd:sequence>
  <xsd:element name="id"
     type="int"/>
  <xsd:element name="name"
     type="string"/>
```

```
    <xsd:element name="sub-org"
     type="tns:org"/>
  </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="org"
     type="tns:org"/>
    <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...
```

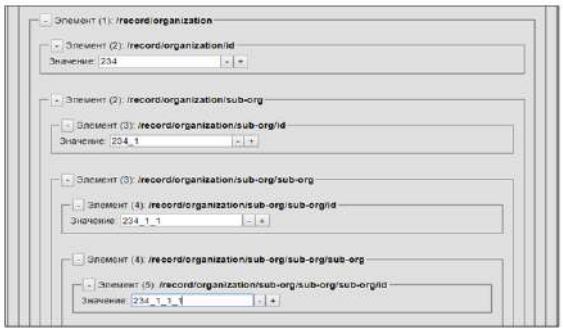

Figure 4. Recursion fragment

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 I.

Table I
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	

IV. CONCLUSION

The above described technology for creating adaptive graphical Web interfaces for data editing is implemented in a prototype editor, which is a server application. The presented approach to the formation of information editing interfaces for heterogeneous data allows the developed graphic web interfaces to automatically tune into the structure of one or another information resource. The created prototype of the described adaptive XML editor allows you to process any XML data by converting the source data of any structure without any modification of the program code. In the future, it is planned to increase the functionality of the editor in terms of expanding the list of supported XSD elements and supporting JSON format, due to its popularity. Upon completion of testing, the editor will be integrated into the ZooSPACE-W subsystem of the technological platform for the mass integration of distributed heterogeneous data sources ZooSPACE. Moreover, it can be used by users as an independent functional system, in the form of an editor for working with files in XML format.

ACKNOWLEDGMENT

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 No 18-07-01457-a.

REFERENCES

[1] Yu. M. Sherstyuk, " Model and algorithm for constructing a syntax-directed editor of xml descriptions", "Nauchnoe Priborostroenie", 2000, vol. 10, no. 4, pp.72-78

[2] V. V. Mironov, G. R. Shakirova, "Programmno-instrumentalnoe sredstvo dlya sozdaniya i vedeniya dinamicheskikh xml-dokumentov", Vestnik UGATU, 2007, vol. 9, no. 5, pp.54-63.
[3] V. V. Mironov, N. I. Yusupova, A. S. Gusarenko, "Validation of the xml documents in web-applications based on situation-oriented databases (DTD, XML SCHEMA, RELAX NG)", Proceedings of the 5th All-Russian Conference «Information Technologies for Intelligent Decision Making Support», 2017, May 16-19, Ufa, Russia, Volume 1, pp.10-14.
[4] O. L. Zhizhimov, "Explain Services on ZooSPACE Platform and Adaptive User Interfaces", CEUR Workshop Proceedings, 2015, Vol.1536, pp.30-36. Available at: http://ceur-ws.org/Vol-1536/paper4.pdf.
[5] 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.
[6] SRU-Search/Retrieve via URL. The Library of Congress. Available at: http:// www.loc.gov/standards/sru (accessed 2016, July 9).
[7] ZeeRex: The Explainable "Explain" Service. Available at: http://zeerex.z3950.org
[8] SRU-Explain Operation. The Library of Congress. Available at: http://www.loc.gov/standards/sru/explain (accessed 2013, September 6)
[9] O. L. Zhizhimov, A. M. Fedotov, Y. I. Shokhin, "The ZooSPACE platform-access organization to various distributed resources. Digital libraries: The Russian scientific e-magazine. Vol.17. No 2. ISSN 1562-5419.
[10] O.L Zhizhimov, A.A. Lobykin, I.Y. Turchanovskiy, A.A. Panshin, S.A. Chudinov, "Computer -assisted acquisition system of statistics event information in distributed information system". Vestnik NSU. Ser.: The Information technology, 2013, Vol.11. ISSN 1818-7900. pp.42-52.
[11] XML Schema Part 0: Primer Second Edition: W3C Recommendation. Available at: http://www.w3.org/TR/xmlschema-0 (accessed 2004, October 28)
[12] XML Schema Part 1: Structures Second Edition: W3C Recommendation. Available at: http://www.w3.org/TR/xmlschema-1 (accessed 2004, October 28)
[13] XML Schema Part 2: Datatypes Second Edition: W3C Recommendation. Available at: http://www.w3.org/TR/xmlschema-2 (accessed 2004, October 28)
[14] XSL Transformations (XSLT) Version 2.0: W3C Recommendation. Available at: http://www.w3.org/TR/xslt20 (accessed 2007, January 23)
[15] A. A. Mukhitova, O. L. Zhizhimov, "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. pp.142-149. Available at: http://elib.ict.nsc.ru/jspui/bitstream/ICT/1467/20 /paper16.pdf

РЕАЛИЗАЦИЯ АДАПТИВНОЙ МОДЕЛИ ВВОДА И РЕДАКТИРОВАНИЯ ИНФОРМАЦИИ НА ОСНОВЕ XSLT-ПРЕОБРАЗОВАНИЙ ДЛЯ РАЗНОРОДНЫХ ДАННЫХ

Мухитова А., Жижимов О. Л.

Адаптивность, способность интерфейсов подстраиваться под структуру и функциональность информационных источников, является одним из основных признаков интеллектуальности информационной системы. Разработка адаптивных графических веб-интерфейсов на основе XML-технологий позволяет визуализировать любую структуру файла XML-формата для возможности дальнейших манипуляций по вводу и редактированию данных. В работе рассмотрена технология построения адаптивного графического административного WEB-интерфейса для ввода и редактирования данных в разнородной информационной среде на основе использования определений схем данных XSD с применением XSLT преобразований. Приводится пример реализации адаптивной модели ввода и редактирования информации в виде созданного прототипа редактора XML-записей. Данный редактор, в архитектуре клиент-сервер, на стороне сервера генерирует пустую форму редактирования путем преобразования модифицированной структуры XSD методом XSLT и затем предоставляет на клиентской части готовую HTML-форму с полностью необходимым инструментарием (java скрипты) для корректного ввода и/или редактирования неоднородных данных.

Received 10.01.19