IRSTI 20.23.29, 20.23.25, 28.23.29, 20.53.19
UDC 004.043;424

TECHNOLOGY OF BUILDING ADAPTIVE GRAPHICAL WEB INTERFACES FOR DATA EDITING

A.Mukhitova[1], A. Yerimbetova[2]
[1]al-Farabi Kazakh National University, Almaty, Kazakhstan
[2]Institute of Information and Computational Technologies, Almaty, Kazakhstan

[1]mukhitova.aigul@gmail.com
[2]aigerian@mail.ru
[1]ORCID ID: https://orcid.org/0000-0002-4081-7694
[2]ORCID ID: https://orcid.org/0000-0002-2013-1513

Abstract. The article discusses the technology of creating and modifying data in a heterogeneous information system using adaptive graphical web interfaces. The purpose of the work is to create a set of technical developments and techniques that implement a systematic and integrated approach to the construction of adaptive user interfaces. The research method is based on a systematic analysis of modern technologies for creating adaptive graphical interfaces, developing their own technology and testing it experimentally. A server-side Web application has been developed that provides on-screen forms for creating and editing XML documents in accordance with the selected XSD schema. On-screen forms are generated on the server side and are provided to the user as HTML pages. The field of application of the above-mentioned technology is the integration of the created adaptive graphical web interface in a distributed information system in order to increase the efficiency of user access to heterogeneous information and the ability to manipulate it.

Keywords: adaptive web interfaces, integration of heterogeneous information, xml document navigation, xslt technologies.

Introduction

The technological basis for building distributed information systems (DIS) is based on three technological "whales": WWW technologies, the LDAP directory model, and the Z39.50 protocol model.

By far the most convenient is the Z39.50 standard. No other international standard contains a complete set of necessary rules for searching for information, extracting this information from storage and providing it to the user in the necessary formats. At the same time, the standard regulates network access to databases and abstracts from specific schemas and data structures [1].

Technologies based on the LDAP protocol model are most suitable for implementing the functions of managing access to DIS and distributed information management. This is due to the fact that LDAP is the easiest way to organize the technology of a "single point of access" to information resources, including identification, authentication and authorization of users in distributed information systems. An essential circumstance in this case is the fact that LDAP is based on the ideology of distributed storage of information - directory trees [5]. Against the background of global identification of all directory elements, it contains definitions of mechanisms and procedures for data replication between different servers and is very well supported by application and system software developers. All this makes it quite easy to move from local to distributed management of information systems and control access to their resources.

To work with databases, remote document repositories, and repositories, select the Z39.50 protocol and its extensions: SRW/SRU protocols. If they are not available in the open access archive, you can choose the OAI-PMH protocol [1, 5]. This is extremely rare when the remote archive supports the OAI-PMH protocol. Here you have to use NTTP, without any metadata. To extract metadata, algorithms for parsing the title page, extracting keywords in their absence, and an algorithm for constructing an abstract based on a graph connected to the text are used [6].

Heterogeneous distributed information systems should include user and administrative interfaces (graphical WEB interfaces) that provide the ability to manage data from heterogeneous information sources. Interfaces that adapt to the structure and functionality of information resources are adaptive. Each specific information resource usually corresponds to only a fairly narrow range of parameters from their potentially possible values. Therefore, when selecting control parameters for various information resources, it is necessary to attract additional information about a specific information resource [11].

Over the past two decades, XML has become not only the standard format for data exchange between different web applications, but also the model for a family of some new databases or NoSQL databases, called XML databases. In addition to the efficiency of managing regular data, XML is also an excellent support for storing, processing, and querying temporary data due to its hierarchical structure. [2]

The technology of creating and modifying data in a heterogeneous information system using adaptive graphical WEB interfaces will be discussed below. It is assumed that the data can be extracted from the relevant information sources in XML format, since it is the most suitable for building WEB-based data editing interfaces.

Technology for building WEB interfaces

To simplify the perception of information by users, there are various methods of presenting it. One of the most common ways of visual representation that allows you to display the structure of information is to write it in the form of an HTML page in the browser. XML and HTML are subsets of SGML, and therefore inherit its basic principles [7]. The structure of an XML document is similar to a regular HTML page. By simple manipulations, information from an XML document can be presented as an HTML page. HTML forms and XML elements are logically equivalent in terms of the content of the information. This makes it possible to automatically convert them via a web browser using a single software tool for any XML data. The generated HTML form that acts as a data entry form is essentially a simple low-level web-based XML editor, and is an adaptive graphical interface [8,9].

The process of converting XML documents from one XML schema to another can be divided into three main parts [10]:

- Create XML code-Create code using regular text or specialized XML code editors. Many DBMSs are equipped with the function of direct generation of XML code.

- Create XSL style sheets for processing and displaying an XML document. XSL tables are created by the administrator-programmer.

- Overlay of XSL style sheets. The XML document is overlaid with XSL style tables according to certain formatting rules, and the result is a document that corresponds to the client's request in the following output formats: HTML, PDF, WML, and XML.

On the client-side

The conversion of XML documents by overlaying XSL style sheets is usually performed on the client side. The style sheet is loaded only once, reducing the load on the server. Based on the principle of overlaying a CSS file with an HTML file, browsers can attach styles to XML documents and form the final document.

Server-side

If browsers do not understand full-fledged HTML, and cannot perform xml\xsl transformations themselves, then the style application to an XML document can be performed on the server in two ways [7-10]:

- Manually - compiles xml files/style sheets and places the resulting files on the server;

-Automatically – content management framework (CMF) performs the required transformation of the xml document and transmits to the user: at the request of the Acrobat Reader program-the published page will come in PDF format, at the request of the WAP device - the client will receive a WML page, Web browsers will receive an HTML/XHTML page.

To represent structured information in XML format, the main thing is to have a description of the rules for constructing the structure of an XML record, i.e., a description of the data schema. Typically for XML, these rules are formulated in terms of XSD and represent an XML structure that can be processed by standard means, such as XSLT [11].

Thus, graphical web interfaces built on the basis of XML technologies allow you to display any structure of a file presented in XML format.

When using the XML format to represent structured information, it is essential to have a description of the rules for constructing the structure of an XML record, i.e., a description of the data schema. Usually for XML, these rules are formulated in terms of XSD [2, 3] and represent an XML structure that can be processed by standard means, for example, XSLT [10].

When extracting a record from a specific information source in a heterogeneous information system and presenting this record in XML format for editing, you need to get a complete description of the possible structure of the extracted record (XSD).

To get the XSD, follow these steps [11]:

- Get a reference to the original XSD data schema as a URL as the attribute schemaLocation when defining the namespace to use. It is usually contained in the root element of an XML record.

- Make a request to the information system to provide XSD by the ID database, if the XML record extracted for editing contains the Uniform Resource Identifier (URI), but does not contain a link to the XSD data schema used in the form of a URL. Such a request can be processed by the Explain service.

- Make a request to the information system for the provision of XSD by the name of the information resource (database) or the use of XSD, which corresponds to the scheme requested when forming the request for data extraction, in the case when the XML record extracted for editing does not contain namespace definitions.

In any of these cases, data modifications are required to initialize graphical interfaces [11]:

- Description of the data schema in the form of an XML structure in accordance with the XSD rules.

- XML structure containing extracted data for editing (not required to create a new record).

- Description of styles for forming GUI elements (optional).

- Description of the rules for forming the elements of the graphical interface in accordance with the rules of the XSD used and the value of the elements of the XML record being edited. However, these rules can be XSLT transformation rules applied to XSD.

The principle of operation of the XML record editor in the client-server architecture is as follows:

- For the client part, a ready-made HTML form is provided for entering and / or editing data. At the same time, the form already contains all the necessary tools (java scripts) for correct data entry;

-The editing form is generated on the server side by XSLT transformation of the modified XSD structure. First, an empty edit form is generated (without data), which, after the XSLT processor finishes working, is filled with the record data in XML format.

For the client part, a ready-made HTML form for entering and / or editing data is provided, which contains all the necessary tools for correct data entry, including:

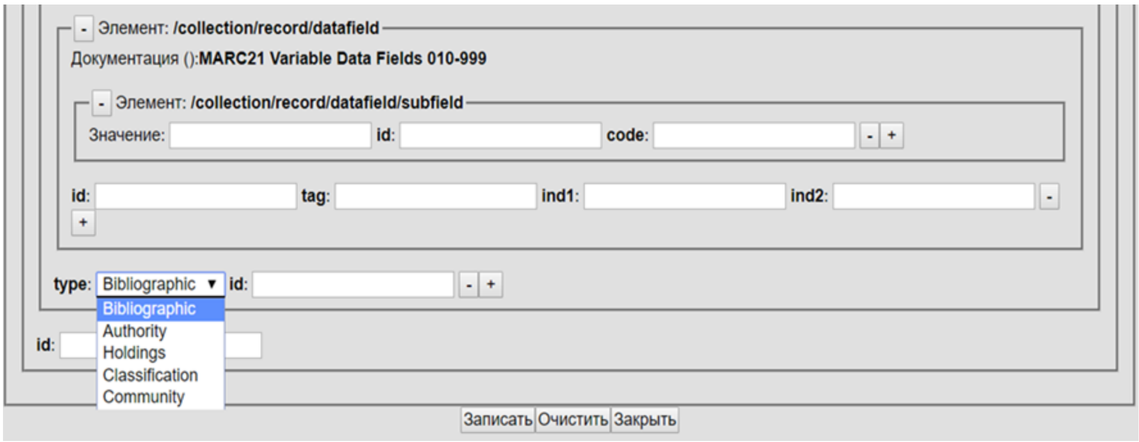
- Script for duplicating duplicate elements according to XSD.

- A script for deleting items that can be deleted according to XSD.

- A script for checking the correctness of data input, provided that the corresponding template is present in the form of a regular expression in XSD.

- The scenario of closing then opening any data element in the edit form.

Figure 1 shows a fragment of an HTML form generated by the XML record editor. Next to the field names are the information input fields.



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 1 - Graphical editor for XML records: data entry fields

When processing XSD with XSLT, various features of the XSD definitions are taken into account, including references and recursion. The entered data is automatically saved in the same database from which the record was extracted.

Conclusion

The developed technology for building adaptive graphical WEB interfaces allows you to effectively solve the problem of integrating heterogeneous information resources based on the database schemes of the XSD application using XSLT transformations. The described methodology has quite general ways of using it and can be used to build adaptive graphical WEB interfaces that allow you to generate sent HTML forms for entering and editing data.

The developed adaptive graphical editor of XML records allows you to import any XML data and transform its structure efficiently and simply, while the same processing process allows you to transform the source data of any structure without any modification of the program code.

Currently, the adaptive graphical editor of XML records is implemented in the ZooSPACE-W subsystem of the ZooSPACE platform of the ICT SB RAS (Institute of Computational Technologies, Novosibirsk). Work is underway to improve the functionality of the editor in terms of expanding the list of supported XSD and JSON elements.

This software product can be used as an independent module for creating and editing data presented in XML format, as well as an embedded module in various server software for heterogeneous information systems.

Acknowledgements

This research has been funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. AP09057872).

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] Brahmia, Z., Hamrouni, H., Bouaziz, R. XML data manipulation in conventional and temporal XML databases [electronic resource]: A survey (2020) Computer Science Review, 36, art. no. 100231. Cited 1 time.DOI: 10.1016/j.cosrev.2020.100231

[3] Mukhitova A., Zhizhimov O.L. “Application of XML-technologies in the Construction of Input and Editing Data Model in the Adaptive Administrative Graphical Web Interfaces for Heterogeneous Information Systems”. Data Analytics and Management in Data Intensive Domains: XX International Conference DAMDID/RCDL'2018 (October 9-12, 2018, Moscow,

Russia): Conference Proceedings / Edited by Leonid Kalinichenko, Yannis Manolopoulos, Sergey Stupnikov, Nikolay Skvortsov, Vladimir Sukhomlin / FRC CSC RAS. – Moscow, 2018. 157-160.

[4] Mukhitova Aigul, Zhizhimov Oleg L. Implementation of an adaptive model of input and editing information based on XSLT transformation for heterogeneus data. Open Semantic Technologies for Intelligent Systems / Minsk, 2019. 3. 173-178.

[5] Zhizhimov O. L., "Explain Services on ZooSPACE Platform and Adaptive User Interfaces // CEUR Workshop Proceedings, (2015). 1536. 30-36.

[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. 17. 2. ISSN 1562-5419.

[7] XML Schema Part 0: Primer Second Edition: W3C Recommendation. URL: http://www.w3.org/TR/xmlschema-0 (date of the application: 03.01.2021)

[8] XML Schema Part 1: Structures Second Edition: W3C Recommendation. URL: http://www.w3.org/TR/xmlschema-1 (date of the application: 03.01.2021)

[9] XML Schema Part 2: Datatypes Second Edition: W3C Recommendation. URL: http://www.w3.org/TR/xmlschema-2 (date of the application: 03.01.2021)

[10] XSL Transformations (XSLT) Version 2.0: W3C Recommendation. URL: http://www.w3.org/TR/xslt20 (date of the application: 03.01.2021)

[11] 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.- 42–149.ISBN:978-5-905569-10-4.
http://elib.ict.nsc.ru/jspui/bitstream/ICT/1467/20/paper16.pdf(2017)