

# Development of a Unified Model of Data Representation for Cross-System Interaction

Tatiana Penkova\*, Anna Korobko and Artem Belorusov  
Institute of Computational modelling SB RAS, Krasnoyarsk, Russia  
\*Corresponding author

**Abstract**—The paper presents an unconventional approach to formation of unified models for representation of cross-system interaction. The authors highlight the necessity of system integration method development that would bring in a standardised way to data exchange between heterogeneous resources with an opportunity of flexible adaptation to the actual conditions in which these systems operate. The paper gives a formal description of the model of cross-system data interaction. Some conversion algorithms were developed to turn the information storage schemes into the unified XML-format and RDB representation.

**Keywords**—cross-system interaction; data integration; unified model; data representation; XML; Relational data base

## I. INTRODUCTION

The sphere of corporate information system development has recently been undergoing a number of significant changes: complex universal automation systems that encompass all business processes are giving way to ad-hoc customised software products aimed at tackling specific tasks. [1]. Development of all-embracing information systems proved to be very time-consuming and quite inefficient in terms of their functional complexity. The efforts required for their creation and implementation make them a less attractive option when compared to the evolving market of shrink-wrapped applications that offer a great variety of effective software. However it is impossible to build a fully functioning corporate information system without enabling it to interact with its own components and exchange information with external resources. According to a number of studies, businesses end up spending 40% of their IT budget on cross-system interaction [2]. Compatibility of systems is becoming a prerequisite for automation, which substantiates the topicality of development and elaboration of methods and technologies in cross-system interaction of information.

Submission of bids for government and municipal procurement is one of the sectors where interaction across disparate systems and data exchange between heterogeneous resources are in greater demand than ever. The state policy orientated towards open source data, as well as legislation in the procurement sector predispose the authorities involved in the procurement process and organisations making their purchases using budgetary or municipal funds to integrate their information systems with the Official All-Russia website of procurement ([zakupki.gov.ru](http://zakupki.gov.ru)) and bidirectional data communication with the digital markets [3].

Currently there are a few approaches to cross-system integration that can be singled out. File Transfer, for example, is among the simplest integration tools. In case when files are used to exchange information, developers do not have to go into details about the processes that underlie the systems to integrate, it is enough to determine common data format. The main drawback, though, is that in the course of creation, transfer or processing the files there is a good chance of asynchronous behaviour in the data [4]. Another tool is to use Shared Database. Unlike the previous approach, the systems target current synchronised information, which means that any attempt to make changes to one and the same given database item through several different sources gets blocked by transactions. In this respect, the shared database can result to be a bottleneck on the way to integrity and even trigger the whole system to shut down if the number of integrated systems gets too high [5]. The above-mentioned approaches operate well at the level of information exchange, but sometimes may prove inefficient when there is a need to perform a set of business procedures, such as validation of the altered address. It can be made using Remote Procedure Invocation, which allows one system to call functions of other systems. As a result, every system sustains the integrity of its data independently, without interfering with any other systems [6]. The downside of this approach is that the integrated systems are too closely-knit, it becomes especially evident during execution of sequential function chart which involves a few systems at once. One option that can help to avoid most of the problems highlighted above is Messaging: the information is kept up-to-date by way of frequent asynchronous exchange of small portions of data; moreover, the message body does not contain any details about the information that could be seen by integrated systems; the possibility to convert the messages result in weaker bounds between the systems [7]. Despite the apparent advantages, this approach implies a rather more complex process of development, testing and debugging [1]. Web-oriented systems are quite often based on Service-oriented Architecture (SOA), with web-services providing a universal interface for disparate information systems to come into interplay [8]. Also, there are some commercial solutions, best-known of which are IBM WebSphere MQ and Microsoft BizTalk, based on the technology of Enterprise Application Integration (EAI). The applications provide a wide range of functional opportunities, but their adoption means significant financial investment and requires a lot of time, EAI-resources - attraction of specifically qualified professionals [9].

Consequently, the advancement in modern technologies, advantages of the existing approaches, and restrictions on existing solutions, all together substantiate the relevance of originally developed techniques and tools for cross-system interaction. They provide standardised information exchange between heterogeneous resources with a possibility of flexible adaptation to changes in the systems' functioning conditions.

## II. MODEL OF CROSS-SYSTEM INTERACTION

The model of cross-system interaction can be thought of as a set of three  $M=\langle I,O \rangle$ , where  $I$  – stands for multiple subsystems engaged in informational interaction,  $O$  – multiple objects of informational exchange. The subsystems engaged in information exchange in the sector of municipal procurement are comprised of the modules Regional automated system of municipal bidding, Official All-Russia procurement website and Digital markets. The objects of informational exchange are electronic documents that are compiled in the course of procurement in accordance with all laws and regulations concerning order placement. For example, the Official All-Russia website gets a Request of proposal handed over from the Regional system, meanwhile what the Digital market RTS-Tender gets is Entry Protocols for consideration, Procurement Results Protocol, etc. At the same time, the Official All-Russia website sends the Regional system data on signed contracts, procurement road maps, reference materials (e.g. OKPD2, All-Russian Classifier of Products; OKVED, All-Russian Classifier of Economic Activities) etc., meanwhile, Digital market RTS-Tender takes care of the Entries, Auction Commission Composition, Data about Supplier, etc.

Information exchange takes place between two interacting subsystems one of which acts like a "sender" and the other one - like a "receiver". Let us take the "sender's" subsystem as  $I_a$ , and the "receiver's" subsystem as  $I_b$ , where  $I_a \subseteq I$  and  $I_b \subseteq I$ . Each subsystem from the set  $I$  uses of the information exchange formats (e.g. XML – eXtensible Markup Language, RDB – Relational Data Base, DOCX – Microsoft Word Open XML Document). The task of data integration is to provide information exchange regardless of the format used by interacting subsystems. The structure of data representation is determined by the subsystem either sending or receiving information, and the object of informational exchange,  $(I,O) \Rightarrow S_i$ , where  $S_i$  – represents a scheme of data storage in the format of the subsystem  $I$ . The task of information exchange is normally fulfilled through development of separate program units that convert one format into another for each of the objects involved in the information exchange [10, 11, 12]. Using this approach makes it necessary to create new units (in case more objects or subsystems are added to the system of information interaction) and do the reprogramming (in case the schemes of data storage are changed).

Unified representation of information exchange structures of various formats makes it possible to automatise the matching of information storage schemes of items in different subsystems. Conversion of information storage schemes in the sender's subsystem format into a unified representation is formally given by  $T:S_{I_a} \rightarrow U_1$ , whereas conversion of information storage schemes in the receiver's subsystem format into a unified representation is given by  $T:S_{I_b} \rightarrow U_1$ , where  $T$  –

conversion operator of information exchange formats XML, RDB and DOCX,  $U_1$  – unified representation of information exchange structures in the subsystem's format  $I$ . Thus, the unified representations of storage schemes will be given by  $G(U_{I_a}, U_{I_b})$ . Let us look closer at the formation of a unified model of data representation in formats XML and RDB.

## III. FORMATION OF A UNIFIED MODEL OF DATA REPRESENTATION IN FORMAT XML

The structure and requirements for XML-documents content are described in data schemes in XSD format. Generally, one XSD file contains a structure description of some items of informational exchange. The purpose of a unified model of XML-documents representation boils down to analysis of an XSD-file, declaration of elements describing particular exchange items, and their representation in a hierarchical model (tree). The algorithm of how the operator converts the XML format information storage schemes into the unified representation is shown in Figure I.

```

function getSchema(source)
  schemaSet = getSchemas(source);
  map = new HashMap();
  foreach a in schemaSet
    elMap = getMap(a);
    elName = getName(a);
    map = mapU(elName, elMap);
  endfor
  return map;
end

function getMap (schema)
  map = new HashMap();
  foreach a in schema.elements
    if a.elements ≠ ∅
      elMap=getMap(a);
    else
      elMap = elMapU("minOccurs", a.minOccurs);
      elMap = elMapU("maxOccurs", a.maxOccurs);
      elMap = elMapU("type", a.type);
      elMap = elMapU("full name", a.annotation.documentation);
    endif
    elName = getName(a);
    map = mapU(elName, elMap);
  return map;
end

```

FIGURE I. THE ALGORITHM OF HOW THE OPERATOR CONVERTS THE XML FORMAT INFORMATION STORAGE SCHEMES INTO THE UNIFIED REPRESENTATION

To visualise the unified model in form of an n-ary tree, the tree nodes have to be stored as elements of associative arrays in HashMap. HashMap allows to store Key-Value pairs, where Key is the name of an element in the data storage structure and Value is defined by dynamically recursive function getMap(). As for intermediate tree nodes, the Value contains a HashMap containing a full list of child nodes. The tree nodes of type "leaves" are defined by a HashMap containing a list of available properties and their values. The main properties of the tree nodes are: minOccurs – minimum permissible number of elements in a scheme, maxOccurs – maximum permissible number of elements in a scheme, type – type of the scheme element, annotation.documentation – word description of a scheme element. Various combinations of property values make it possible to set the following parameters for the elements:

- optional nodes [minOccurs = 0, maxOccurs = 1]
- imperative nodes [minOccurs = 1, maxOccurs = 1]
- infinite number of nodes [minOccurs = 0, maxOccurs = unbound]
- set number [minOccurs = 3, maxOccurs = 5].

The example of a unified representation in form of a tree calculated for the item of information exchange "Protocol of entry processing for on-line auction" (document fcsProtocolEF1.xml) is shown in Figure II. Highlighted in red are the nodes that are mandatory to fill (value of property minOccurs = 1).

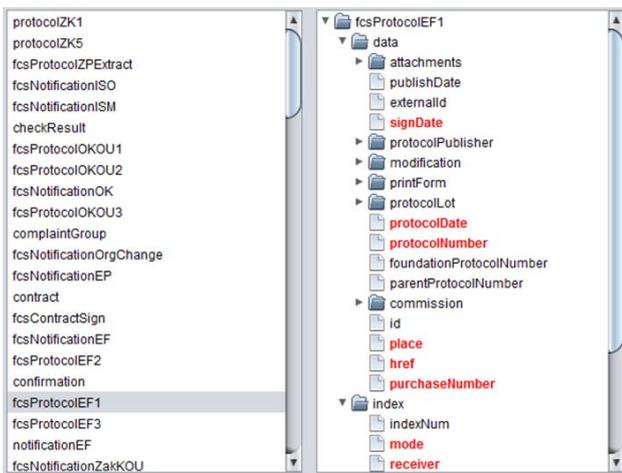


FIGURE II. VISUAL IMAGE OF THE UNIFIED MODEL OF INFORMATION REPRESENTATION FOR THE XML-DOCUMENT DESCRIBING "PROTOCOL OF ENTRY PROCESSING FOR ON-LINE AUCTION"

#### IV. FORMATION OF A UNIFIED MODEL OF DATA REPRESENTATION IN FORMAT OF RELATIONAL DATABASE

A data storage scheme in the format of relational database (RDB) is converted into a unified representation through analysis of the database meta-data and construction of an n-ary tree the nodes of which are charts, margins and their properties. The algorithm of how the operator converts RDB format information storage schemes into the unified representation is given in Figure III.

Unified representation of RDB format information storage schemes, just like in the case of XML-format, takes form of an n-ary tree the node of which are at the same time elements in associative arrays in HashMap. Each name of the element in the information storage system is used as a key to HashMesh, and the value is determined dynamically via function getColumns() for each chart. The tree nodes of type "leaves" are defined by a HashMap containing a list of available properties and their values. The main properties of the nodes are: key – primary key, fk – foreign key, type – a kind of element in the scheme. The values of the properties show the character of the elements contained in the scheme in the course of informational interaction.

```
function getTableList(source)
map = new HashMap();
metaData = connection.getMetaData();
tables = metaData.getTables();
foreach a in tables
elName = a.getString("TABLE_NAME");
elMap = getColumns(metadata, elName);
map = mapU(elName, elMap);
endfor
return map;
end

function getColumns(metadata, tableName)
map = new HashMap();
columns = metaData.getColumns(tableName);
foreach a in columns
elName = a.getString("COLUMN_NAME");
elMap = new HashMap();
elMap = elMapU("type", a.getInt("DATA_TYPE"));
elMap = elMapU("key", a.getBoolean("IS_KEY"));
elMap = elMapU("fk", a.getBoolean("IS_FK"));
map = mapU(elName, elMap);
endfor
return map;
end
```

FIGURE III. THE ALGORITHM OF HOW THE OPERATOR CONVERTS THE RDB FORMAT INFORMATION STORAGE SCHEMES INTO THE UNIFIED REPRESENTATION

An example of such a representation, which was made for the item in the information exchange tagged "Procurement" (Table PRC) is given in Figure IV.

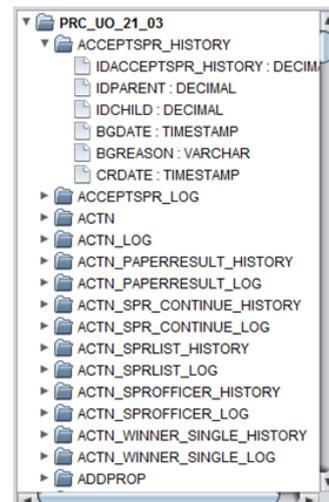


FIGURE IV. VISUALISATION OF THE UNIFIED MODEL OF RDB DATA STORAGE SCHEME FOR THE TABLE "PROCUREMENT"

#### V. CONCLUSION

The paper presents an unconventional approach to formation of unified models for representation of cross-system interaction. The authors highlight the necessity of system integration method development that would bring in a standardised way to data exchange between heterogeneous resources with an opportunity of flexible adaptation to the actual conditions in which these systems operate. The paper gives a formal description of the model of cross-system data interaction. Some conversion algorithms were developed to turn the information storage schemes into the unified XML-format and RDB representation, also, description was given to

algorithmic implementation of the conversion operator. Unified representation of information exchange structures of various formats makes it possible to automatise the matching of information storage schemes of items in different subsystems.

#### ACKNOWLEDGMENT

The reported study was funded by RFBR and Government of Krasnoyarsk Territory according to the research project N 16-47-243020.

#### REFERENCES

- [1] G. Hohpe, B. Wolf, "Enterprise integration patterns: designing, building, and deploying messaging solutions", Addison-Wesley Professional: Computers, 2004, 683 p.
- [2] D. Braue, "The Integration Imperative", 2002, // <http://www.cio.com>.
- [3] "Contract system in the procurement of goods, works and services for state and municipal needs": The Federal Law N 44-FZ, 05.04.2013 (ed. from 03.07.2016).
- [4] G. Fedak, H. He, F. Cappello, "BitDew: a programmable environment for large-scale data management and distribution", International conference for high performance computing, networking, storage and analysis, 2008, pp. 1-12.
- [5] L. M. Haas, E. T. Lin, M. A. Roth, "Data integration through database federation", IBM Systems Journal, vol. 41, Iss. 4, 2002, pp. 578-596.
- [6] A. Wellings, R. Clark, D. Jensen, D. Wells, "A framework for integrating the real-time specification for Java and Java's remote method invocation", 5th International symposium on object-oriented real-time distributed computing, Washington, April 29-May 1, 2002.
- [7] S. Damodaran, "B2B integration over the internet with XML – RosettaNet successes and challenges", WWW2004, May 17-22, 2004, New York. ACM 1-58113-912-8/04/0005.
- [8] B. Hochgurtle, "C# and Java: cross-platform Web-services", Svjas, 2004, 213 p. (in Russian)
- [9] J. Dědič, "Advanced Topics on System Integration", Jiří Dědič, Masaryk University, Brno, 2005, 108 p.
- [10] M. F. Fernandez, A. Morishima, D. Suci, W.-Ch. Tan, "Method for converting relational data into XML", Patent US 6785673 B1, 2004.
- [11] B. Vrdoljak, M. Banek S. Rizzi, "Designing Web Warehouses from XML Schemas", LNCS, vol. 2737, 2003, pp. 89-98.
- [12] E. Rahn and P. A. Bernstein, "A survey of approaches to automatic schema matching", Very Large Database J., 10(4), 2001, pp. 334-350.