

Adaptive Model of Data Exchange for Cross-System Interaction

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

Abstract—The article approaches the problem of adaptive data exchange model making for cross-system data interaction. The authors demonstrated the need in development of system integration methods that would facilitate the standardised data exchange across heterogeneous sources with an opportunity to adapt flexibly to ever changing conditions of system operation. The work gives a formal description of a model of cross-system interaction. It showcases some algorithms that were developed to generate a tree of an adaptive model and a storage scheme for objects of data exchange in XML format.

Keywords—cross-system interaction; data integration; adaptive model; data representation; XML; relational data base

I. INTRODUCTION

Cross-system interaction of data is one of the most essential steps in business process automation. Comprehensive one-size-fits-all corporate data systems now tend to be replaced by task-specific software solutions tailored to particular needs [1]. Nevertheless, to be of any good use such systems must be able to interact with one another and exchange data with external resources. Therefore it becomes necessary to work out and develop techniques and tools for cross-system integration.

One of the spheres where interaction and data exchange between heterogeneous resources are especially asked for is bidding for government and municipal procurement. On the one hand, Government encourages using open-source data in procurement procedures. On the other hand, the procurement law forces authorities participating in procurement activity and purchasing agencies funded with the money coming from the federal or municipal budget into integrating their data systems with the All-Russian Official website zakupki.gov.ru and facilitating bidirectional data exchange with the federal electronic trading platforms [2].

Cross-system integration of data can be approached in a few different ways, File Transfer serving as the most basic one. You just need to specify the data format that will be used for all integrated systems and leave out the rest. It will be sufficient for information exchange to take place. But since the data exchange here largely depends on manipulations with files, there is a great risk of data going out of sync at the moment of creation, transfer or processing [3]. Also, databases can be shared. It is a different way of integration. The systems work with the most relevant information which is perfectly synchronised. It is impossible to alter any one given entry in the database - any attempt to access it from a few various sources, will be stopped. However, the shared part tends to be

the weaker spot of this integration solution, which can lead to a system crash if more systems are integrated [4]. So far, we have been speaking about the data exchange. But sometimes we may have to carry out some other business operations, for instance, address validation. The system can call other systems' functions by way of Remote Procedure Invocation. It means each system controls its data integrity without damaging other systems [5]. The biggest disadvantage of this approach is that the integrated systems are too closely connected, it hampers sequential operations that depend on more than one system. To avoid most of the faults in solutions described earlier one can apply to Messaging. The information is updated through asynchronous exchange of small pieces of data. The integrated systems have no access to the information, because it is hidden in the message body. Also, they are less tied up to one another because of the option to convert messages [6]. Regardless of all its strong points, this approach needs to be taken to a higher implementation level with better development, testing and adjustment [1]. Web-oriented systems are quite often based on Service-oriented Architecture (SOA), with web-services providing a universal interface to combine disparate information systems [7]. Along with the above-mentioned solutions, there are some commercial ones, the most popular of them 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 entails large financial investment and requires a lot of time, EAI-resources - attraction of specifically qualified professionals [8].

All in all, 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

Informational cross-system interaction can be modelled as having three parts $M=\{I,O\}$, where I – represents a set of interacting subsystems, O – a set of objects taking part in the data exchange. In municipal procurement, the subsystems engaged in data exchange are comprised of the modules such as Regional automated system of municipal order placement, Electronic trading platforms, and Official All-Russia procurement website. The items of data interaction are

electronic documents that are produced as part of procurement procedure according to all laws and regulations regarding order placement. For instance, when the Regional system sends a Procurement notification to the Official procurement website, it at the same time forwards protocols for entry consideration, procurement results protocols etc. to the Electronic trading platform RTS-Tender. In its turn, the Regional system receives data about active contracts, procurement road maps, reference information (OKPD2, OKVED2), etc. from the Official website and data about the entries, supplier, or composition of the Auction Commission, etc. from the Electronic trading platform RTS-Tender.

Data exchange is facilitated between the two interacting subsystems, one of which plays a role of a sender and another one plays that of a receiver. The sender's subsystems can be given by I_a , and the receiver's subsystem by I_b , where $I_a \subseteq I$ and $I_b \subseteq I$. Each of the subsystems from the set I uses one of the formats of data exchange (XML – eXtensible Markup Language, RDB – Relational Data Base, DOCX – Microsoft Word Open XML Document). Data integration would facilitate data exchange no matter which format the interacting subsystems use. The structure of data representation is based on the subsystem sending or receiving data, and the object of data exchange, $(I, O) \Rightarrow S_i$, where S_i – scheme of data storage in the format that is used by the subsystem I . As a rule, data exchange problem is tackled by means of some extra programming units that convert data from one format to another directly for each object of data exchange [10, 11, 12]. This sort of approach leads to creation of new units (if new objects or subsystems have been added to the data interaction system) and constant reprogramming (if data storage schemes have been changed).

Taking into the account all mentioned above, we can posit that storage of meta-data about the parameters and conditions of data transfer in form of an adaptive model could create a universal adjustable programming tool for data interaction without any need in continuous reprogramming. Taken formally, transformation of the data storage scheme in the format used in the sender's subsystem into unified representation can be given by $T:S_{Ia} \rightarrow U_{Ia}$; transformation of the data storage scheme in the format used in the receiver's subsystem into unified representation is given then by $T:S_{Ib} \rightarrow U_{Ib}$, with T – the operator of data exchange format transformation in XML, RDB and DOCX, U_i – unified representation of the data exchange structure for the format used in the subsystem I_i ; matching of the unified representations of data storage is therefore given by $G(U_{Ia}, U_{Ib})$. So the adaptive model should be given by $Q = (U_{Ia}, U_{Ib}, G)$.

III. FORMATION OF ADAPTIVE MODEL

The suggested adaptive model of cross-system data interaction can be formed through development of tools for packet exchange [13, 14, 15].

The adaptive model of data exchange $Q = (U_{Ia}, U_{Ib}, G)$, in terms of storage is basically an xml-file. The fragment of xml-file for Procurement results protocol is presented in Figure I. The process of data transfer starts with the analysis of the adaptive model file and derivation of a tree for the adaptive model. Algorithmically tree derivation for the adaptive model

using a data storage scheme in XML format is given in Figure II.

Function Parse() takes a set of XML-nodes from the analysed file of the adaptive model as a parameter. If the node has child objects, it is added to the list of parent objects and initiates recursive invocation with the parameters being passed over to the node's descendant. If the current node does not have any descendants, it is added to the list of child objects. The first invocation of the Parse() function as a parameter is transferred to the XML-file root node. The algorithm procedure results in derivation of a tree for the adaptive model which contains data about the structure of communicated objects of data exchange and values of the key parameters.

```
<?xml version="1.0" encoding="utf-8"?>
<fcsNotificationEF table="PRC" condition="idproc = $param(idproc) and cldate is null" dsname=
xmlns="http://zakupki.gov.ru/oos/extension/1" xmlns:oos="http://zakupki.gov.ru/oos/
"http://www.w3.org/2001/XMLSchema-instance">
  <oos:id sql="select loadid from im_protocols where idproc = $param(idproc) and cldat-
  dsname="DS_PRTCLS" source="DS_PRTCLS.LOADID" notnull="DS_PRTCLS.LOADID"/>
  <oos:docPublishDate sql="select sysdate, round(sysdate+(2/24),'HH') startdate from
  source="CURDATE.sysdate" required="Не указаны время и дата публикации извещения"/>
  <oos:purchaseObjectInfo source="DS_PRC.PRCNAME" required="Не указано наименование"/>
  <oos:purchaseResponsible>
    <oos:responsibleOrg source="global(uo_name)" required="Нет наименования организа-
    <oos:regNum source="global(uo_codespz)" required="Нет кода СПЗ организатора"/>
    <oos:fullName source="global(uo_name)"/>
  </oos:responsibleOrg>
  <oos:responsibleRole const="RA"/> роль организации осуществляющей закупку
  <oos:responsibleInfo table="CR$USERS" condition="idcr$users = $DS_PRC.idcr$users
  dsname="DS_USERS">
    <oos:orgPostAddress source="global(uo_lctn)" required="Нет почтового адреса"/>
    <oos:orgFactAddress source="global(uo_lctn)" required="Нет адреса организатора"/>
    <oos:contactEmail source="DS_USERS.useremail" notnull="DS_USERS.useremail"/>
    <oos:contactPhone source="DS_USERS.userphone" notnull="DS_USERS.userphone"/>
    <oos:contactFax source="global(uo_fax)"/>
    <oos:addInfo const="Информация отсутствует"/>
  </oos:responsibleInfo>
</oos:purchaseResponsible>
```

FIGURE I. THE FRAGMENT OF XML-FILE OF PROCUREMENT RESULTS PROTOCOL

```
procedure Parse(nodes)
  for i from 0 to nodes.Count
    item = nodes.child[i];
    if item.type == EKEMENT_NODE
      tag = getName(item);
      if hasAttributes(item)
        addAttributes(tag);
      endif
      if hasChilds(item)
        addParent(tag);
        Parse(getChilds(item));
        closeParent();
      else
        addChild(tag)
      endif
    endif
  endfor
end
```

FIGURE II. THE ALGORITHM OF TREE DERIVATION FOR ADAPTIVE MODEL IN XML FORMAT FOR CROSS-SYSTEM DATA EXCHANGE

The process of data storage scheme production for an object of data exchange in XML format based on the adaptive model is comprised of two major stages: the first stage analyses the tree of the adaptive model, the second stage works out a data storage scheme for data exchange objects in XML format. Algorithmically the two stages of the process are shown in Figure III and Figure IV respectively.

At the first generating stage the execute() function receives the adaptive models' tree as a parameter. Then all nodes of the

tree are checked one after another. If the property of the node "name" is empty or carries an attribute such as not(isAdding), the node is used for connection with the data source and completion of the storage schemes for data exchange objects. The property of the node "mult" bearing indication "true" points out that in the final storage scheme this node corresponds to a set of elements that are added by means of the function addChildren().

The function of addChildren() at the second stage creates an element in the XML storage scheme of data exchange object for the current node and adds it to the set of parent nodes. If the node has some descendant nodes, while going round the cycle they get processed by the function of execute(). As soon as all descendant nodes of the current node have been processed, it goes on to the next node. Finally the algorithm ends up to generate a storage scheme for data exchange object in form of XML documents.

```

procedure execute(Tree t)
    tag = getRoot(t);
    currentAttributes = getAttributes(tag);
    executeCurrentAttributes();
    if tag.Name= ∅ or not (tag.adding()) return;
    if hasChildren(t)
        if tag.mult
            executeMult(tag);
            for i from 0 to tag.rowsCounter
                addChildren(t)
            endfor
        else
            addChildren(t);
        endif
    else
        if lastParent=∅
            addChildren(t);
        else
            addXMLElement(lastParent, tag);
        endif
    endif
end

```

FIGURE III. THE ALGORITHM GENERATING STORAGE SCHEMES FOR DATA EXCHANGE OBJECTS IN XML-FORMAT BASED ON THE ADAPTIVE MODEL (STAGE 1)

```

procedure addChildren(Tree t)
    tag = getRoot(t);
    currentAttributes = tag.getAttributes();
    Element = createElement(tag);
    lastParent = Element;
    parents.add(Element);
    childSet = getChildSet(t);
    foreach child in childSet
        execute(child);
    endfor
    closeparent();
end

```

FIGURE IV. THE ALGORITHM GENERATING STORAGE SCHEMES FOR DATA EXCHANGE OBJECTS IN XML-FORMAT BASED ON THE ADAPTIVE MODEL (STAGE 2)

If data exchange takes place between subsystems in format of RDB and XML, database tables represent data sources, and

fields of chosen tables represent values of elements. Figure V shows an example of data source tree to complete the XML storage scheme for data exchange objects (with regard to municipal procurement). Figure VI gives an example of the finalised XML document of Entry Consideration Protocol.

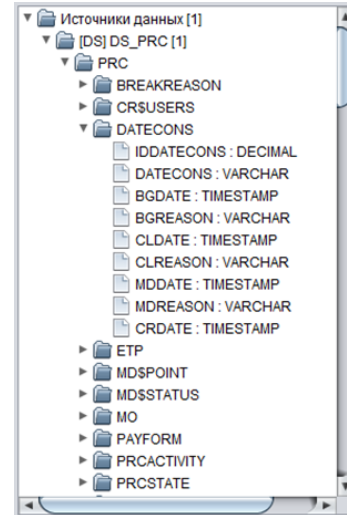


FIGURE V. EXAMPLE OF A DATA SOURCE TREE FOR COMPLETION OF XML STORAGE SCHEME FOR MUNICIPAL PROCUREMENT MANAGEMENT

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<fcsNotificationEF xmlns="http://zakupki.gov.ru/oos/extegration/1" xmlns:oos="
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" schemeVersion="5.2">
  <oos:docPublishDate>2016-11-11T14:05:03</oos:docPublishDate>
  <oos:purchaseResponsible>
    <oos:responsibleOrg>
      <oos:regNum>991111111116</oos:regNum>
      <oos:fullName>Департамент муниципального заказа Администрации г.К
    </oos:responsibleOrg>
    <oos:responsibleRole>РА</oos:responsibleRole>
    <oos:responsibleInfo>
      <oos:orgPostAddress> 660049, Красноярский край, Красноярск г, Кар
      <oos:orgFactAddress> 660049, Красноярский край, Красноярск г, Кар
      <oos:contactFax>7-391-2261984</oos:contactFax>
      <oos:addInfo>Информация отсутствует</oos:addInfo>
    </oos:responsibleInfo>
  </oos:purchaseResponsible>

```

FIGURE VI. EXAMPLE OF THE FINALISED XML DOCUMENT OF PROCUREMENT RESULTS PROTOCOL

IV. CONCLUSION

The article approaches the problem of adaptive data exchange model making for cross-system data interaction. The authors demonstrated the need in development of system integration methods that would facilitate the standardised data exchange across heterogeneous sources with an opportunity to adapt flexibly to ever changing conditions of system operation. The work gives a formal description of a model of cross-system interaction. It showcases some algorithms that were developed to generate a tree of an adaptive model and a storage scheme for objects of data exchange in XML format. The adaptive model stores meta data about parameters and conditions of data transfer, so it will contribute to creation of some universal customised software supporting data interaction skipping the stage of reprogramming in case of addition of new objects or

subsystems to the system of data interaction or alteration of data storage schemes.

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.
- [13] A. I. Belorusov, "Integration of information systems based on XML standards and WEB-services in the case of municipal procurement", Young scientist, N11, 2015, p. 9-15 (in Russian).
- [14] A. I. Belorusov, D. V. Zhuchkov, "Automating the interaction of municipal information systems with external web-services in the case of municipal procurement", Informatization and Communications, N3, 2014, p. 79-83 (in Russian).
- [15] R. V. Morozov, "Consolidation of procurement planning functions in a single system "Municipal customer", Proceedings of XIV Russian conference "Problems of Region informatization", Krasnoyarsk, 2015, p.161-165 (in Russian).