Framework for preparing subject data in testing modules of scientific applications

E S Fereferov, A G Feoktistov and I V Bychkov

Matrosov Institute for System Dynamics and Control Theory of SB RAS,
Lermontov St. 134, Irkutsk, Russia, 664033

fereferov@icc.ru

Abstract. The paper addresses the relevant problem of data preparation for testing modules of scientific applications. Such testing requires the multiple executions of modules with different parameters for various scenarios of solving problems in applications. Often, data sources for parameters used for problem-solving are subject data (experimental results, reports, statistical forms and other information resources) created earlier as a result of functioning various objects of a subject domain. Usually, such data are heterogeneous and weakly structured. The developer of scientific applications has to make additional efforts in extracting, cleaning, integrating, and formatting data in order to achieve the correctness and efficiency of their use in applications. The aim of the study is the development of a framework for automating the description of semi-structured data and their transformation into target structures used by scientific applications. We proposed a conceptual model that allows us to represent knowledge about the structure of the source data, determine their relations with the target structures and set the rules for data transformation. Additionally, we developed a framework prototype. It is integrated into the technological scheme of continuous integration for modules of scientific applications (distributed applied software packages) that are developed with the help of Orlando Tools. The effectiveness of this prototype functioning is confirmed by the results of experimental analysis.

1. Introduction

The paper addresses relevant problems related to preparing data for scientific applications oriented to high-performance computing. Such applications are nowadays one of the main components in the process of carrying out large-scale experiments associated with solving complicated scientific and applied problems. These problems can arise in various spheres of human activity.

The need to obtain the values of the subject-oriented data contained in weakly structured sources often arises in the process of developing and using scientific applications.

Databases or files in different formats are such sources. Often, they do not conform to the strict structure of tables and relationships in relational database models.

Usually, the subject-oriented data are created earlier by subject domain specialists and contained experimental results, reports, or statistical information. Such data is required to set the initial parameters of problems in applications. In this regard, its developers are forced to carry out the non-trivial elicitation, refining, transformation, integration, and aggregation of the subject-oriented data to a specific form of their representation in applications.

We propose a new framework for marking weakly structured data with reference to a given target structure that can be an object structure.

The rest of the paper is structured as follows. In Section 2, we give a brief overview of the known tools for extracting and transforming data from weakly structured information sources. Section 3 represents a conceptual model of transformation tables. A framework prototype of preparing data for executing scripts of testing modules in scientific applications is proposed in Section 4. Section 5 shows the results of experimental analysis. Section 6 concludes the paper.

2. Related work

Nowadays, the development of new approaches to the integration of heterogeneous data sources with information and computation systems is a challenge [1]. Often, this problem is related to processing big data. In different cases, various approaches are used to solve it. Among them are the following approaches:

- Creation of tools for extracting data from semi-structured data sources given in the specific format or a large spectrum of documents,
- Development of tools for converting free-form spreadsheets into a relational data model,
- Ensuring the synthesis of the required integrated data structure based on a set of heterogeneous source data structures, etc.

As a rule, the extraction and transformation of data are partially automated.

It is known that the TextRunner [2] and WebTables [3] systems are focused on extracting data from web-pages and transforming it into the relational form.

Such tools as FlashRelate [4], Foofah [5], and Senbazuru [6] support the extraction of data and the relations between them from spreadsheets.

In addition, the TabbyXL system [7, 8] implements the transformation of arbitrary tables into the relational form based on a set of rules for their analysis and interpretation. MIPS [9], which is close to TabbyXL, performs a similar transformation based on the search for critical table cells.

The FlashExtract system [10] is more universal tool. It allows us to extract data from a wider range of documents, including text files, web pages, and spreadsheets. Data extraction is executed on the basis of examples provided by users of this system.

It should be noted that the source table model should be known for the effective operation of the Senbazuru and MIPS systems.

The feature of data preparation for the execution of testing scripts of modules in scientific applications is specific requirements of each module to the format of input parameters. Therefore, the data in the relational form provided by the above-listed systems is not always convenient and may require additional transformations.

Currently, there is a wide range of commercial tools for extracting and transforming data from weakly structured information sources. These include IBM InfoSphere DataStage [11], Talend Open Studio [12], Pentaho Data Integration [13], Informatica PowerCenter [14], Open Refine [15], etc. Usually, these tools are focused on solving the Extract Transformation Load (ETL) problems.

They provide forming the structured data storages for business intelligence systems (OLAP, OLTR systems, etc.). However, the use of commercial software products is extremely expensive and not always available for developers of scientific applications.

Unlike the aforementioned tools, the proposed framework supports a large spectrum of structures that can be both the relational and object ones. Knowledge about the markup of weakly structured data is stored in the structural specifications that can be used by various transformational procedures many times. Data transformation is based on applying the special templates reflecting the relations between semi-structured and target data. Conditions for the use of transformation operations are determined by a set of productions that can take into account the features of data sources.

3. Conceptual model of transformation tables

We assume that the weakly structured data is any intermediate data between the structured and unstructured. As a rule, their structure has uncertainties of various kinds.

In data processing, the degree of its correctness is not known in advance. The data scheme may not exist or does not fully correspond to the processed data. Some data attributes may be absent or not fully satisfy the correctness conditions defined for these attributes.

We propose the new model M for describing weakly structured data and the rules for their transformation into target structured formats. This model has the following structure:

$$M = \langle File, WS, TS, Rule, H: WS \to TS \rangle$$

where the parameters $File, WS, TS, Rule, H: WS \to TS$ are interpreted as follows.

The $File$ parameter sets the format of a source (file) of weakly structured data: $File = \{XLS, CSV\}$, where XLS and CSV are the valid file formats.

The WS scheme of weakly structured data is determined by the set $Tbls$ of tables and the set $Refs$ of links between them: $WS = \langle Tbls, Refs \rangle$, where $Tbls = \{t_1, \dots, t_n\}$ and $Refs = \{r_1, \dots, r_m\}$. The construction of such a scheme allows us to mitigate the aforementioned uncertainty of the initial weakly structured data.

The set Atr of attributes and the set Val of their values are assigned to each i-th table: $t_i = \langle Atr, Val \rangle$. The set Atr is determined by the range and data type: $Atr = \langle Cell_s, Cell_f, Type \rangle$, where $Cell_s$ and $Cell_f$ are the start and end bounds of the range correspondingly.

We use the following set of data types: $Type = \{I, F, S, D, B, R\}$, where I is the set of integers, F is the set of real numbers, S is the set of string values, D is the set of date and time values, $B = \{true, false\}$ is the set of Boolean values, and R is the reference type indicating to the relation $r_j \in Refs$.

Table attribute values are characterized by the data range $Val = \langle Cell_s, Cell_f \rangle$. The reference $r_j \in Refs$ is set by the name and attribute of the table in the markup file: $r_j = \langle t_i, a_k \rangle$, where $t_i \in Tbls$ and $a_k \in Atr$.

The structure TS describes the target structured data. $TS = \langle Objs, Obj\ Refs \rangle$, where $Objs$ is the set of target structure objects and $ObjRefs$ is the set of references between it. Objects from $Objs$ are characterized by a name and a set of fields: $Objs = \langle Name, Fields \rangle$. The references from $ObjRefs$ define relations between objects from $Objs$.

The $Rule$ parameter represents a set of data transformation rules: $Rule = \{rl_1, \dots, rl_n\}$, where $rl_i = \langle Atr, Fields, Trans, Q: Atr \to Fields \rangle$, Atr is a set of attributes from the weakly structured data schema, $Fields$ is a set of fields of target structure objects, $Trans = \langle Val, Op, StrOp \rangle$ is a structure that defines the applied value transformation operations, $Val = val \vee key$ determines the transfer of a value or key value from a related table (if reference is given $r_j \in Refs$), Op is an operation above attribute values (for example, combining attribute values into one field or splitting attribute values into several fields), $StrOp$ is a string operation above values.

The operation $Q: Atr \to Fields$ matches attributes to fields of objects. Conditions for the use of such operations can be determined by means of productions.

The operation $H: WS \to TS$ maps the weakly structured data to the target data scheme using rules from $Rule$.

The structural metadata-based approach is used to establish the correspondence between the scheme WS of weakly structured data and the structure TS that describes a scheme of the target data. Metadata include data types, table names, attributes, objects, their fields, and the relations between them.

The rules for transformation data from the source scheme to the target structure are based on the established matches.

4. Framework prototype

Based on the proposed model, we developed a framework prototype for marking and transforming data from weakly structured sources into target formats of scientific applications. The framework

prototype includes a markup tool and transformation modules. The markup tool provides the application developer with the ability to visually customize and specify the process of transforming data needed to solve a particular class of problems. Transformation modules provide the creation and translation of data into target structures of specific types. The general scheme of marking and transforming is shown in figure 1.

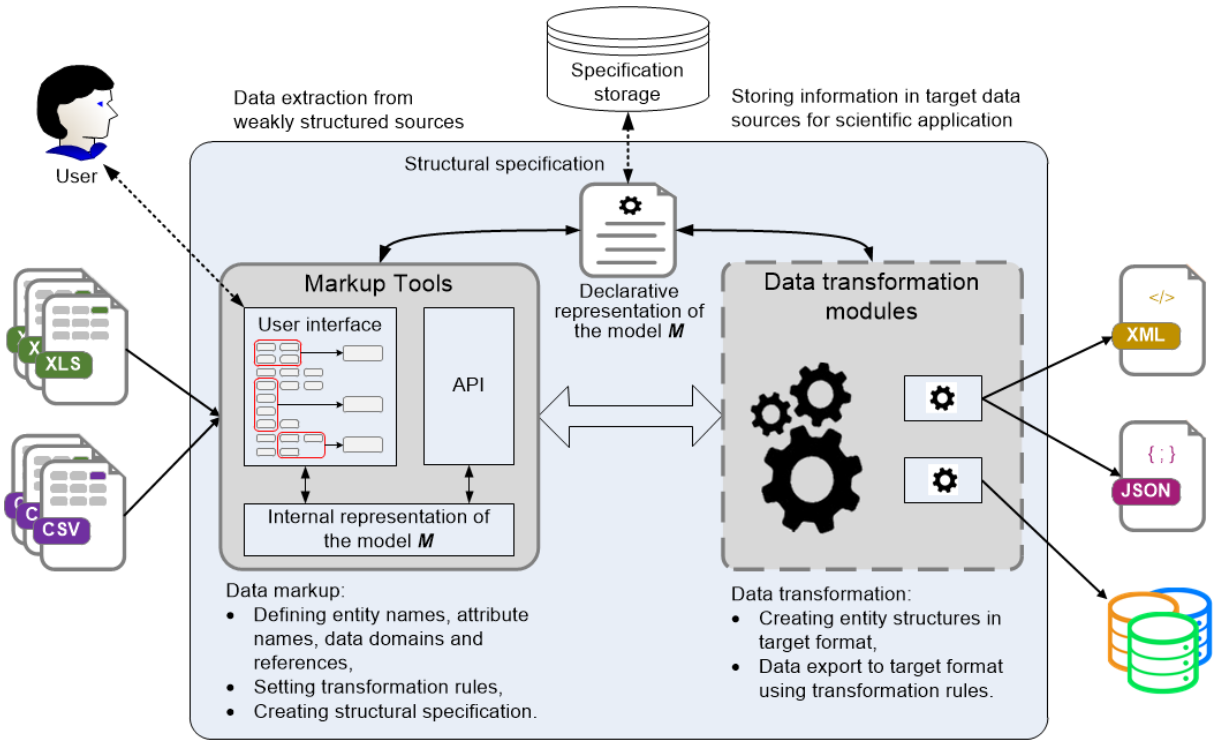

Figure 1. Marking and data transformation scheme.

The framework prototype implements loading and marking of spreadsheet files in the CSV and MS Excel formats. At the markup stage, the package developer forms visually the model *M* of the source data, specifying table ranges, their attributes, data types, values, and relationships in weakly structured documents. Next, transformation rules are configured. For example, we can specify a rule for splitting attributes into several attributes or assign conditional processing of values using the construction «if then else». The special package for working with regular expressions, which is a part of Embarcadero RAD Studio, is used to handle string values [16].

The description of the initial data scheme can be extended with additional structures to provide the formation of complex target structures. For example, such an extension can maintain a hierarchical structure. Developer's knowledge about the markup and transformation rules are saved to a structural specification file. Next, the specification can be processed by data transformation modules into specific target formats or loaded into a markup tool system for a correction. The application programing interface for accessing external subsystems is implemented in the markup tool. API methods provide access to weakly structured data through the structures of the created model. Such a software interface allows us to support interaction with different transformation modules without reworking the markup tool.

Today, the translation module in XML and JSON formats is implemented as a part of the framework prototype. This module allows us to create target structures taking access to the generated model and data through the API. Schemes of target structures for XML and JSON are set using

templates in the same file formats. In addition to other constructions, such a template contains specific tags to which values from table fields are passed. These tags are as follows:

$$\langle \#[Type\ of\ data]\ N = Name\ (Format) \rangle,$$

where

- $[Type\ of\ data] = \{FV, RN\}$, where FV is a value from table field, RN is a record counter,
- $N = Name$ is a name of the corresponding table field from which the data will be inserted,
- $Format$ is an optional parameter set. There is a set of parameters for each type of data (for example, we can specify the start value of RN or declination for text values from the table fields).

The specification created using the markup tool is enough to generate the database. We implemented a module that provides the generation of relational database schemas based on specifications and filling it with data through the API. Created structural specifications can be applied many times in solving typical problems of data extraction and transformation (for example, when statistical information for different periods are used). In addition, such structural specifications can be applied to automate the creation of application software systems for working with data of target structures.

The framework prototype is integrated into the technological scheme of continuous integration for modules of scientific applications (distributed applied software packages) that are developed with the help of Orlando Tools [17]. The main tasks of such continuous integration in Orlando Tools are receiving, storing, and testing versions of package modules, including the preparation and processing of test data.

5. Experimental analysis

The developed framework prototype was used to test the computing modules of the distributed applied software package (scientific application). This package is used to solve important practical problems of determining the critical objects of the gas supply system of Russia from the standpoint of energy security [18].

In the process of calculations, the package uses the database managed by the Firebird Database Management Software (DBMS) [19]. The database scheme and the structure of files containing the values of module parameters are determined by the computational model of the package [20].

In the experiment, the input data for solving such a problem are files in the MS Excel format with statistical parameters of objects of the gas industry in Russia for several time periods. Each file contains 83 parameters of the gas industry objects of Russia in the tables located on one sheet. One parameter matches to one attribute of the table. The structure of files with information for various periods differs by the location of single attributes. Package modules are tested with each data file.

10 tables with the total number of records more than 2000 were identified in the process of marking the file with object parameters for the initial period. The subject specialist spent about 1 man-hour on the markup of the initial data.

The number of constructions in specifications shown in figure 2. Additionally, it was necessary to correct specifications for files with object parameters for other periods. The correction was to add new table attributes and change the position of some previously created table attributes. In addition, figure 2 shows the number of specification corrections for a file with parameters of the initial period when preparing specifications for files with parameters for subsequent periods.

The created specification is applied to automatically translate the source data into the target structure (XML-files) of package modules parameters according to the given template. The development of macros in MS Excel for translating such data into the XML-file would require about 10 man-hours of programming.

The created specification has been applied to automatically generate the data structure and fill the database that is managed by the Firebird DBMS. For comparison, importing tables from MS Excel using MS SQL Server tools would require calling and setting up translation procedures for each table [21]. At the same time, information about the relations between the tables is not stored anywhere.

Comparison of working costs for data transformation is represented in figure 3.

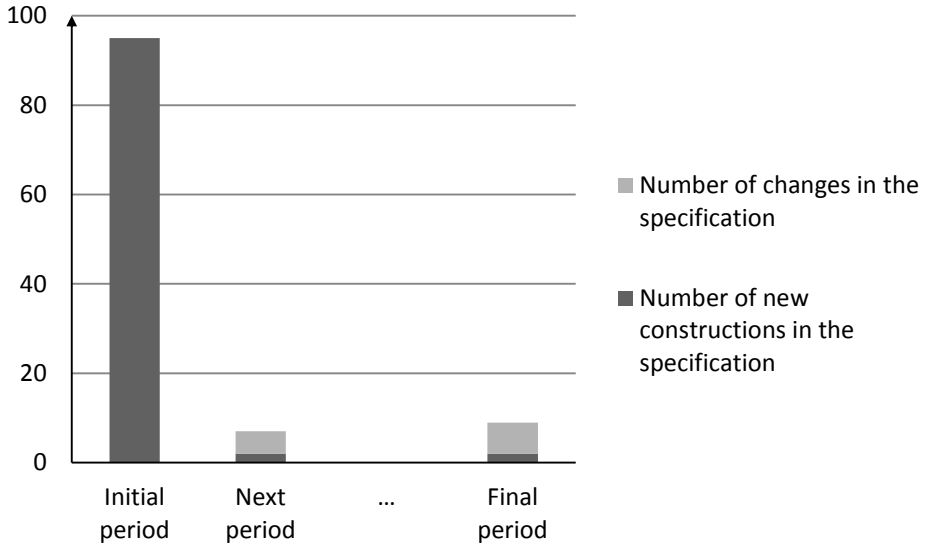

Figure 2. The number of specification constructions and its corrections.

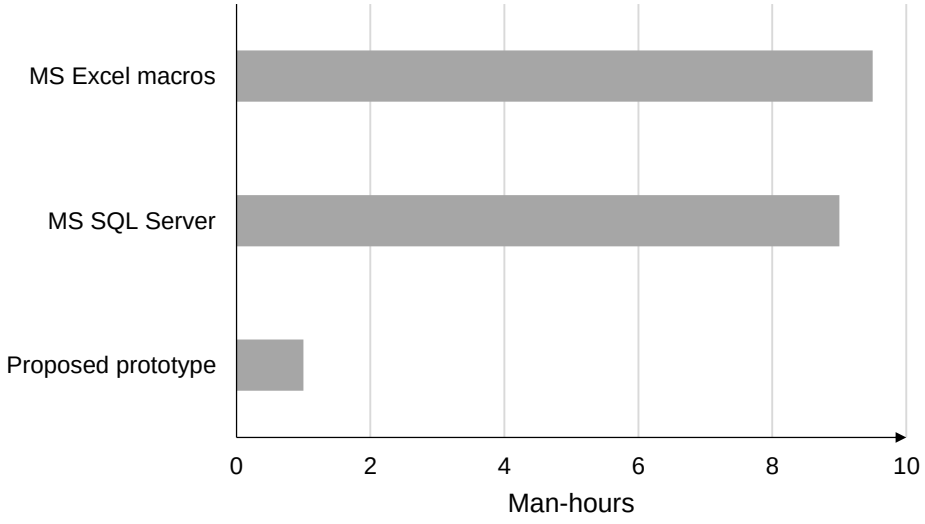

Figure 3. Comparison of working costs for data transformation.

The results shown in figure 3 show a significant reduction of data transformation in time using the proposed framework prototype in comparison with the development of the MS Excel macros and the use of the MS SQL Server tools. This is very important since such a data transformation procedure must be executed repeatedly when developing scientific applications in which the intensity of module modifications can reach several times a day.

6. Conclusions

In the paper, we consider the relevant problem of preparing data for testing modules of scientific applications. In such applications, testing modules requires the multiple executions of them with different parameters for various scenarios of solving problems. Often, the module parameters are

weakly structured data, which require additional efforts from the application developer to transform them into formats used by scientific applications.

We have developed the framework prototype that supports marking, extracting, and transforming subject data from semi-structured sources. The developed framework prototype allows us to describe knowledge about the structure of subject data sources and save it in the form of declarative specifications. This approach is based on applying the special templates reflecting the relations between semi-structured and target data. It is sufficiently flexible due to the specifications generated using these templates contain all necessary information to solve various problems of transforming the source data to target formats used by applications.

The experimental results showed a significant reduction in data transformation time through the applying proposed framework prototype in comparison with developing MS Excel macros and using MS SQL Server tools to this end.

Further study is directly related to extending a spectrum of target structures in transforming the semi-structured source data. In addition, we plan to develop effective algorithms for extracting and cleaning semi-structured data for new target structures.

Acknowledgment. The study is supported by the Russian Foundation of Basic Research, project no. 19-07-00097-a (reg. no. AAAA-A19-119062590002-7). This work was also supported in part by the Basic Research Program of SB RAS, projects no. IV.38.1.1 (reg. no. AAAA-A17-117032210078-4) and no. IV.38.1.2 (reg. no. AAAA-A17-117032210079-1).

References

[1] Cafarella M J, Halevy A, Wang D Z, Wu E and Zhang Y 2008 Webtables: exploring the power of tables on the web *Proc. of the VLDB Endowment* **1(1)** 538–549

[2] Banko M, Cafarella M J, Soderland S, Broadhead M and Etzioni O. 2007 Open information extraction for the web *Proc. of the 20th int. joint conf. on Artifical intelligence* pp 2670-2676

[3] Halevy A, Rajaraman A and Ordille J. 2006 Data integration: the teenage years *Proc. of the 32nd Int. Conf. on Very large data bases* pp 9–16

[4] Barowy D W, Gulwani S, Hart T and Zorn B 2015 FlashRelate: Extracting relational data from semi-structured spreadsheets using examples *ACM SIGPLAN Notices* **50(6)** 218–228

[5] Jin Z, Anderson M R, Cafarella M and Jagadish H V 2017 Foofah: Transforming data by example *Proc. of the ACM Int. Conf. Management of Data* pp 683–698

[6] Chen Z, Cafarella M, Chen J, Prevo D and Zhuang J 2013 Senbazuru: a prototype spreadsheet database management system *Proc. of the VLDB Endowment* **6(12)** pp 1202–1205

[7] Bychkov I V, Mikhalov A A, Paramonov V V, Rugnikov G M and Shigarov A O 2017 TABBYXL: The system for transforming data from arbitrary spreadsheets into a relational form *Proc. of the 16th all-Russian conf. on Distributed information and computing resources (DICR-2017)* pp 150–156

[8] Shigarov A, Khristyuk V and Mikhailov A 2019 TabbyXL: Software platform for rule-based spreadsheet data extraction and transformation *SoftwareX* **10** 100270

[9] Embley D W, Krishnamoorthy M S, Nagy G and Seth S 2016 Converting heterogeneous statistical tables on the web to searchable databases *Int. J. Document Analysis and Recognition* **19** 119–138

[10] Le V and Gulwani S 2014 FlashExtract: A Framework for Data Extraction by Examples *ACM SIGPLAN Notices* **49(6)** 542–553

[11] Blokdyk G 2017 *IBM InfoSphere DataStage: The Definitive Guide* (CreateSpace Independent Publishing Platform) p 120

[12] Rick Daniel Barton 2013 Talend Open Studio Cookbook (Packt Publishing) p 270

[13] Casters M., Bouman R and Dongen J 2010 *Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration* (Wiley) p 720

[14] Malewar R 2017 *Learning Informatica PowerCenter* 10.x. (Packt Publishing) p 426

[15] Kusumasari T F and Fitria 2016 Data profiling for data quality improvement with OpenRefine *2016 Int. conf. on Information Technology Systems and Innovation (ICITSI)* pp 1–6
[16] Regular Expressions – RAD Studio. Available at: http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Regular_Expressions (accessed: 20.06.2019)
[17] Feoktistov A, Gorsky S, Sidorov I and Tchernykh A 2019 Continuous Integration in Distributed Applied Software Packages *Proc. of the 42st Int. Conv. on information and communication technology, electronics and microelectronics (MIPRO-2019)* pp 1775–1780
[18] Feoktistov A, Gorsky S, Sidorov I, Kostromin R, Edelev A and Massel L 2019 Orlando Tools: Energy Research Application Development through Convergence of Grid and Cloud Computing *Communications in Computer and Information Science* **965** 289–300
[19] Firebird: The true open source database for Windows, Linux, Mac OS X and more. Available at: https://firebirdsql.org/ (accessed: 21.06.2019)
[20] Bychkov I, Oparin G, Tchernykh A, Feoktistov A, Bogdanova V and Gorsky S 2017 Conceptual Model of Problem-Oriented Heterogeneous Distributed Computing Environment with Multi-Agent Management *Procedia Computer Science* **103** 162–167
[21] Import and export data using SQL Server Import and Export Wizard - SQL Server Integration Services (SSIS). Available at: https://docs.microsoft.com/ru-ru/sql/integration-services/import-export-data/import-and-export-data-with-the-sql-server-import-and-export-wizard?view=sql-server-2017 (accessed: 18.06.2019)