ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2007, Vol. 36, No. 3

APPLICATION OF BUSINESS RULES FOR DATA VALIDATION

Olegas Vasilecas, Evaldas Lebedys

Vilnius Gediminas Technical University
Sauletekio al. 11, LT-10223 Vilnius, Lithuania

Abstract. There are many tools suitable to model systems and to generate software code from system models, but these tools do not support data validation. Available data validation tools are domain specific and require manual definition of data validation rules. Thus, the lack of the tool supporting both system modelling and automated generation of data validation rules from system models is obvious. The paper discusses the use of system models represented by UML in automation of data validation. The method to derive rules from UML models is presented. A fragment of UML model and derivation of rules represented in the model is demonstrated. The tool supporting proposed method is introduced and the process of automated data validation rules generation from UML models is presented.

1. Introduction

The importance of business system modelling for information system development is widely discussed [2], [12]. At the moment, a variety of methods and commercial tools are available that can be used to model business systems and implement data integrity constraints through the functionality of active database management systems (ADBMS) [2], [5]. Unfortunately, these tools do not support data validation – the implementation of business rules as integrity constraints, triggers stored procedures is used only to avoid entry of erroneous data into the database. Regardless the use of data quality checks at the entry of data into the database, errors in data exist [3]. The application of business rules approach in data quality assurance is widely discussed in the publications of recent years [14], [11]. This shows the importance and relevance of approach. Currently only domain specific data management tools support data validation, but these tools do not support system modelling at all or are suitable to model only some aspects of system. Therefore, there are no tools that support both system modelling and data validation. Errors in software, unintended access to data and other considerations may be the sources of errors in data. These circumstances may be crucial for the quality of data in certain domains, such as statistical data processing, clinical trials or telecommunications. Besides, even if the data are erroneous they may not be changed or rejected at the entry in the database in certain domains [3]. For example, data have to be entered into the database exactly as it was collected in clinical trails. In this case, only after data are in the database, data validation can start to list the discrepancies and get confirmation from the origin of data. The need of the tool for validation of data is obvious. There are commercial and non-commercial tools supporting validation of data, but the functionality of these tools is limited to the manual entry of data validation rules by the user [9], [10]. As system models can be used to retrieve domain knowledge, we state that a part of business rules can be derived from system models and implemented as data validation scripts. To check this assumption, we have developed an experimental tool. The results of the experiment are presented in the paper.

The paper is organised as follows. Section 1 introduces the paper. Section 2 briefly presents the principles of data quality assurance. Section 3 demonstrates the ways business rules are represented in UML diagrams. Section 4 presents the experimental tool and the course of experiment performed to evaluate the tool. Section 5 concludes the paper.

2. The Related Works

Although constraints are implemented in most of information systems to increase the quality of data, errors in data still occur [13]. There are many reasons for this:

- Errors in program code;
- Security problems;
- Unexpected access to the database;
- Mass updates to data.

Data quality means have to be applied to reduce the amount of errors in data. It may be impossible to make the data absolutely clear as data processing is continuous and it is hard to identify all potential

sources of errors [15]. Data quality can be defined as a lack of intolerable defects. There is a finite set of possibilities for data errors and these possibilities can be listed as data quality rules [8]. Data validation is the first step in assessing data quality. Data Validation is an analytic and domain-specific process that extends the evaluation of data beyond method, procedural, or contractual compliance to determine the analytical quality of a specific data set [16]. Data validation is also referred as process that consists of an examination of all the data collected, in order to identify and single out all the elements that could be the results of errors or malfunctioning [1].

Previous analysis showed that business rules could be derived from systems models [17], [18]. Though a plenty of rules can be derived from system models, in most cases these rules do not compose the whole set of rules applicable in particular sys-tem. Additional logical rules have to be defined to get a full set of rules. An example from clinical trail may be the difference between the dispensed medication and returned medication. It is obvious that if the patient received medication A for treatment of some disease, at the next visit unused medication A has to be returned. If by mistake medication B is recorded as the returned medication, discrepancy occurs. The checks to catch this and similar discrepancies are not represented in a model, as a plenty of such checks would overweight the model.

Mostly the following checks are represented in system models [6], [8]:

- the checks of mandatory values, so called not null constraints;
- the checks of the ranges of values;
- if … then … else rules;
- the cross checks between values of different items.

Most of available tools allow implementing these rules to restrict the entry of erroneous data that do not comply with the defined rules [7]. Commercially available products such as Clintrial® or Oracle Clinical® support data validation, but the rules for data validation have to be defined by the user [9], [10]. There are also researches and non commercial tools supporting data management and assurance of data quality, but system models can not be used to automate the process of data validation [4].

3. Data quality rules in UML models

The software prototype we developed is an experimental tool to process UML models and derive business rules represented in different UML diagrams. UML was chosen for analysis as it is likely the most popular modelling language and there are many modelling tools supporting UML [11]. The second reason UML was chosen is the ability to model different aspects of system of interest using UML [5].

A simple experiment is presented in this paper to demonstrate the functionality of created software prototype. Further we briefly describe application domain. Each patient is examined and evaluated to meet requirements, before he or she receives investigational drug. If the patient fits the requirements, the patient gets randomisation number and receives investigational drug. Mostly, in every clinical trial, demographics of each participant in clinical trial is collected to be included in data analysis. Depending on the purpose of the trial, specific restrictions may be defined on demographics for inclusion of the person into the trial. We assume that only woman of age from 18 up to 45 years can take part in the trial. It means that each person willing to participate in the trial is examined to meet the following requirements:

- Patient sex at birth has to be female;
- Person age has to be within 18 and 45 years.

The patient itself may take part in the study, but if he or she does not fulfil inclusion requirements, he or she may not get the investigational drug. The person is assigned a randomisation number and receives investigational drug, if he or she fulfils all the trial requirements, otherwise the patient is excluded from the study.

Vitals and medical history data are also collected at pre-randomisation visits in clinical trials. There are acceptable ranges for each vital sign. For example, acceptable adult person weight should be within 40 and 150 kilograms. This means that the results out of these ranges may be erroneous and should be checked. In such a narrow model as chosen here, similar rules may be defined, but in a full model of particular system it may be redundant. This is why we decided not to represent these constraints in the model below. It is assumed that these data quality checks should be defined additionally outside the model.

It is important to note that different stages of data processing are discussed here and this is why overlapping rules appear:

- original data are recorded in paper forms and at this step no restrictions may be applied. The clinical trial proceeding instructions define the rules for data collection, but there are no means to assure that the instructions are followed;
- original data from paper forms are entered in the database as it were recorded in original. No corrections are allowed even if the errors are obvious;
- the automated processing of rules can only be applied in data validation. Only at this stage the erroneous, illegible data can be identified. As no automated data corrections are allowed, only error messages can be generated.

Figure 1 presents a fragment of the class diagram of the collection of patient data on pre-randomisation visit. As the demographics, vitals and medical history data of each patient have to be collected, the

corresponding classes are represented. The above described constraints are represented in the diagrams.

It is important to note that the following restriction in class diagram exists for demographics of each person "Patient sex at birth has to be male or female", but we assumed that only female patients can take part in the trial we are analysing. This is not a discrepancy as the patient before the randomisation can be either male or female and only female patients can be randomised. As mentioned above, the reason for overlapping rules are three stages of data processing. The same applies to the restriction of patient age.

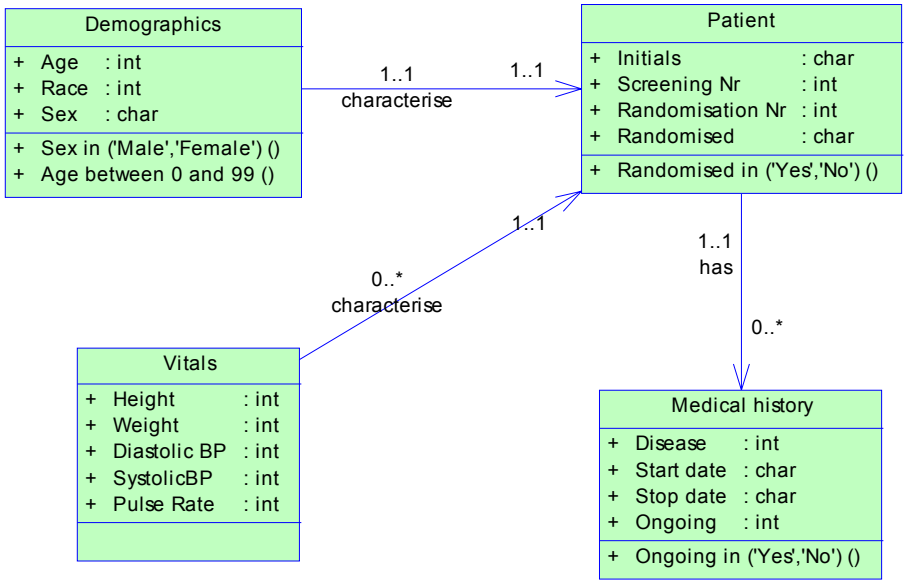

Figure 1. A fragment of the class diagram

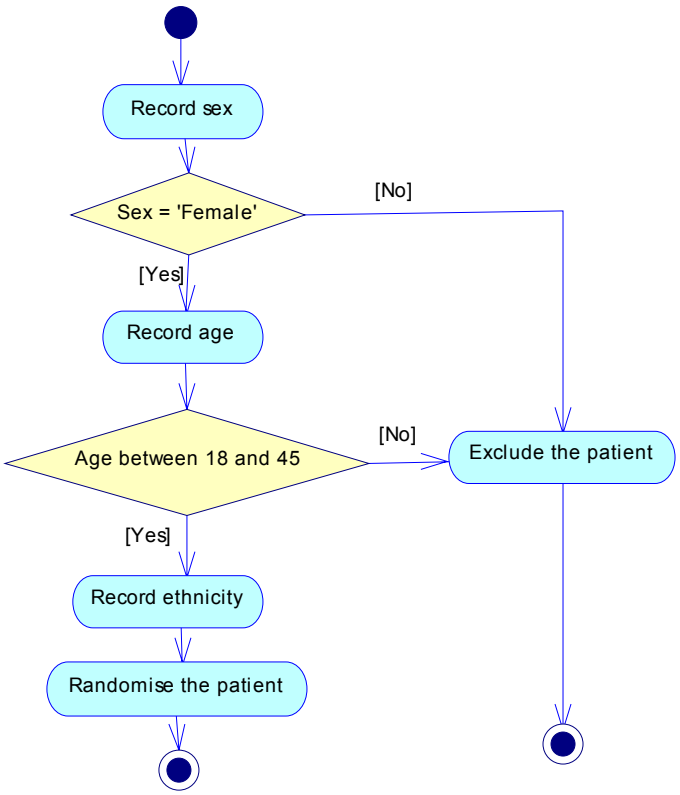

Figure 2. Activity diagram represents the course of pre-randomisation visit

Figure 2 represents the actions physician performs to collect data and to estimate patient eligibility for randomisation. The above discussed rule "Only female patients can enter the trial" is represented in the diagram. As already mentioned above, we have got two different rules concerning the same data item – patient sex:

- the rule "Patient sex at birth has to be male or female" defined in class diagram is used to check if the recorded sex is legible. For example, this rule will trigger an error message if the physician by mistake recorded the age of patient in the paper form where the sex had to be recorded. In this case an error message may be generated "Patient sex is illegible. Patient sex should be either male or female";
- the rule "Only female patients can enter the trial" represented in the activity diagram is used to check if no male patients were randomised. The error message will be generated in case the male patient was not excluded from the study and was randomised. The error message similar to this will be generated "A male patient was included in the trial. Only female patients can be randomised".

4. Deriving the rules from UML models

Further we briefly present the software prototype that may be used to generate the scripts for data validation. Data validation scripts are generated on the basis of rules defined in the UML model. The tool does not support generation of error messages at the current stage of software prototype development. We plan to implement it in the further versions of the software prototype.

The internal behaviour of the software was demonstrated in [17], [18] and is not discussed in detail in this paper. After the UML model is analysed and data are imported into the repository, business rules are identified. For each business rule a script is generated that may be used to implement data quality checks. The scripts are supposed to be run to check for erroneous data in the database of particular system. The domain of clinical trials and a fragment of model presented above is analysed further.

Figure 3 represents the rules that were derived from the class diagram (Figure 1) and activity diagram (Figure 2). Scripts were generated for each rule derived from the class diagram. Each script executed against the collected data in the database would return the erroneous data records. The scripts were also generated for the rules in the activity diagram that had an action to be performed. We have defined the action "Randomised='No'" for the activity "Exclude the patient" in activity diagram. No actions were defined for other activities and, although rules were derived, the scripts were not generated.

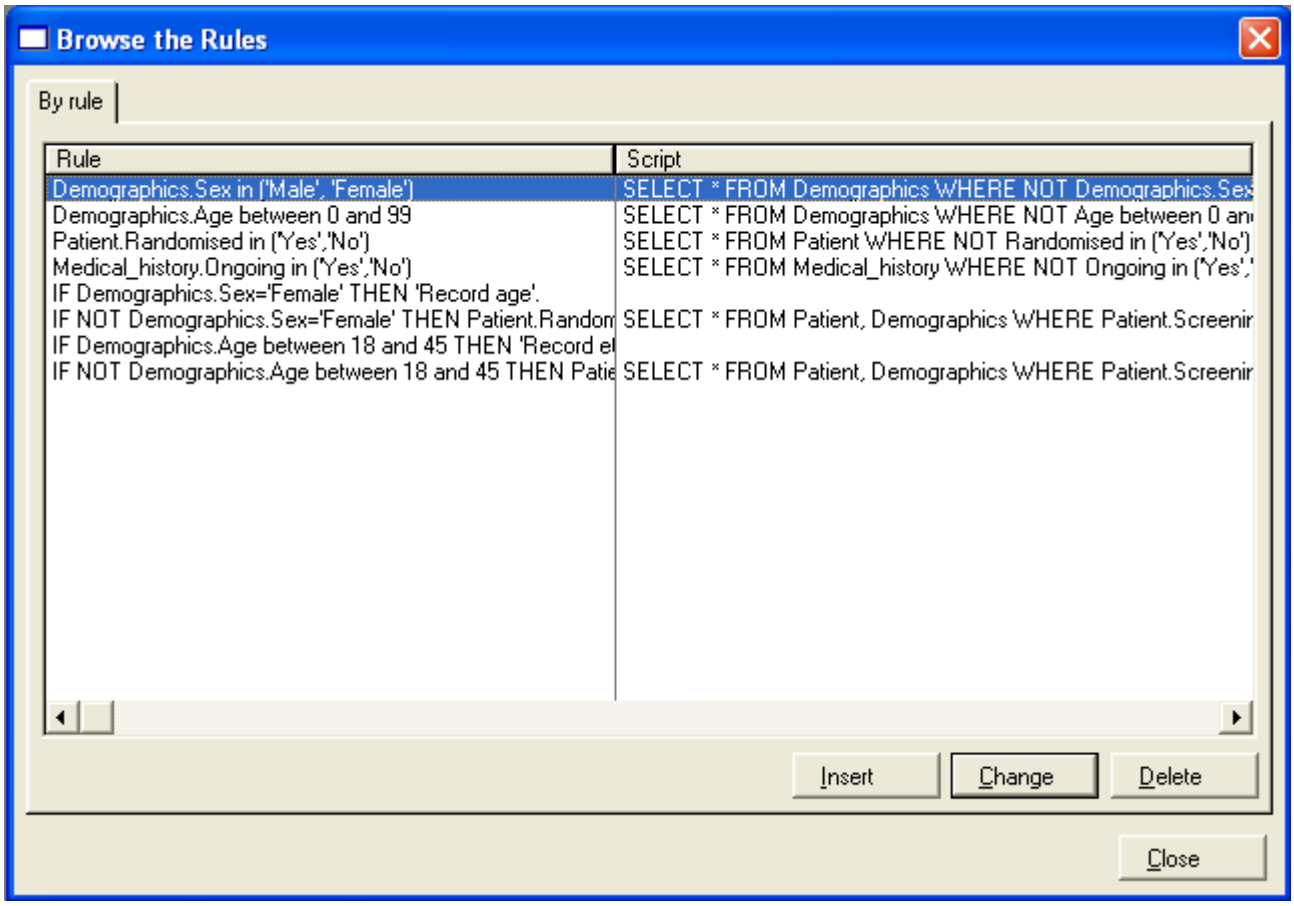

Figure 3. The list of rules derived from the sample model

The created software prototype does not support the automated execution of generated scripts in any database management system. At the current stage of development the scripts can only by copied and executed by the user to list data discrepancies. We plan to implement automated execution of scripts in further versions of the software prototype.

5. Conclusions

The analysis showed that data quality is widely discussed in the literature and is a common topic of today's researches. There are commercial and non commercial tools supporting the assurance of data quality, but available tools may only be used to de-fine quality rules manually. Our previous research showed that system models contain the rules that define the ways to collect and process data. Rules are defined when different aspects of systems are modelled. We presented the method to derive business rules from UML models in this paper. The prototype application was developed to implement the proposed method and an experiment was performed to evaluate the software prototype. The results showed that the UML models can be used to derive the rules from system models and to generate data validation scripts to automate data quality assurance. The functionality of the tool is planned to be expanded in the future.

References

[1] **S. Bandini, D. Bogni, S. Manzoni**. Knowledge Based Environmental Data Validation. *Proceedings of the Biennial Meeting of the International Environmental Modelling and Software Society*, (*iEMSs* 2002), *Lugano, Vol*. 3, 2002, 330-335.

[2] **A. Caetano, A. Vasconcelos and others.** A Framework for Modeling Strategy, Business Processes and Information Systems. *In proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing, EDOC* 2001, *IEEE Press. Seattle, USA*, 2001, 69-81.

[3] **J.R. Davis, V.P. Nolan, J. Woodcock, R.W. Estabrook.** Assuring Data Quality and Validity in Clinical Trials for Regulatory Decision Making. *Workshop Report, Roundtable on Research and Development of Drugs, Biologics, and Medical Devices, Division of Health Sciences Policy, Washington,* 1999.

[4] **G. Duftschmid, W. Gall, E. Eigenbauer, W. Dorda.** Management of data from clinical trials using the ArchiMed system. *Medical Informatics and the Internet in Medicine, Vol*. 27 (2), 2002, 85-98.

[5] **H-E. Eriksson, M. Penker.** Business Modelling with UML: Business Patterns at Work. *J. Wiley & Sons*, ISBN 0-471-29551-5, 2000.

[6] **H. Herbst, G. Knolmayer.** The Specification Of Business Rules: A Comparison Of Selected Methodologies. *Methods and Associated Tools for the Information Systems Life Cycle, Maastricht, The Netherlands,* 1994, 29-46.

[7] **J. Laucius, E. Lebedys, O. Vasilecas.** Realisation of ECA rules by ADBVS triggers. *Information sciences, Vilnius University*, 2003, 129-133.

[8] **W. McKnight.** Overall Approach to Data Quality ROI. *White Paper, Firstlogic Inc*., 2004. *URL: http://www.oracle.com/technology/products/warehouse/pdf/Overall%20, Approach%20to%20Data%20Quality%20ROI.pdf, retrieved on 2007.06.02*.

[9] **Oracle Corporation.** Oracle Clinical. *An Oracle white paper*, 2002. *URL: http://www.oracle.com/industries/life_sciences/oc_data_sheet_81202.pdf, retrieved on 2007.05.20.*

[10] **Phase Forward Inc.** Clintrial. *Phase Forward white paper, 2006. URL http://www.phaseforward.com/products/Clintrial.pdf, retrieved on 2007.05.27.*

[11] **W. Shen, K. Compton, J. K. Huggins.** A Toolset for Supporting UML Static and Dynamic Model Checking. *In proceedings of the* 26*th International Computer Software and Applications Conference* (*COMPSAC* 2002), *Prolonging Software Life: Development and Redevelopment, Oxford, England, IEEE Computer Society*, 2002, 147-152.

[12] **P. Sinogas, A. Vasconcelos, A. Caetano.** Business Processes Extensions to UML Profile for Business Modeling. *In proceedings of the 3rd International Conference on Enterprise Information Systems* (*ICEIS* 2001), *Setubal, Portugal*, 2001, 673-678.

[13] **Society for Clinical Data Management.** Good Clinical Data Management Practices, Version 3. *September*, 2003.

[14] **E. Ugboma.** Assuring Information Systems' Effectiveness Through Data Integrity: Essential Guidelines For Information Systems Databases. *In The Proceedings of ISECON* 2004, *Vol*.21 (*Newport*): §3252, 2004.

[15] **United States Department of Health and Human Services,** Food and Drug Administra-tion. Guidance for Industry. *E6 Good Clinical Practice: Consolidated Guidance* (1996). *URL: http://www.va.gov/chrr/regulatory/vrb/files_forms/GoodClinicalPractices.pdf, retrieved on 2007.05.17*.

[16] **United States Environmental Protection Agency (USEPA).** Guidance on Environmental Data Verification and Data Validation EPA QA/G-8, Washington (2002). *URL: http://www.epa.gov/QUALITY/qs-docs/g8-final.pdf, retrieved on 2007.06.05*.

[17] **O. Vasilecas, E. Lebedys.** Business rules repository for business rules represented using UML. *In R. Rachev, A. Smrikarov (Eds.). Proc. of the International Conference on Computer Systems and Technologies* (*CompSysTech*, 05), *Varna, Bulgaria*, 16-17 *June*, 2004, II.5-1 – II.5-6.

[18] **O. Vasilecas, E. Lebedys.** Moving business rules from system models to business rules repository. *INFOCOMP, Vol*.5, *No*.2, *June* 2006 (ISSN 1807-4545), 2006, 11-17.

Received June 2007.