# IMPLEMENTATION OF RESTRICTION AND VALIDATION RULES[1]

Olegas Vasilecas and Evaldas Lebedys
*Vilnius Gediminas technical univercity, Sauletekio av. 11, LT-10223 Vilnius, Lithuania*

Keywords: Data quality, Data validation, Restriction rules, Validation rules.

Abstract: The paper discusses implementation of restriction and validation rules. Restriction rules based and validation rules based data cleaning approaches are discussed. The differences between restriction rules and validation rules are distinguished. The paper presents a method for automated data validation rules implementation in data cleaning procedures.

## 1 INTRODUCTION

At the moment 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. 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. The application of business rules approach in data quality assurance is widely discussed in the publications of recent years. 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 it may not be changed or rejected at the entry in the database in certain domains.

Section 1 introduces the paper. Section 2 briefly presents restriction rules and validation rules. The quality of data in the context of the clinical trials is discussed in section 3. The purpose of restriction and validation rules is analysed in section 4. Section 5 presents automated validation rules based approach. Section 6 concludes the paper.

## 2 RELATED LITERATURE

The importance and criticality of data quality (DQ) is widely discussed in recent publications (Bertolazzi, 2001), (Motro, 1996). Many external and internal factors impact the quality of data. Constraints are implemented in most of information systems, but errors in data still occur. The whole of desirable data characteristics are analysed in the context of data quality. Although, “fitness for use” is the most widely adopted concept of data quality (Strong, 1997), DQ is defined using various terms in different application domains (Bertolazzi, 2001), (Ehlers, 2003), (Jilovsky, 2005), (Wang, 1996). Data quality can be defined as a lack of intolerable defects in data (McKnight, 2004). Although, “fitness for use” is the most widely adopted concept of data quality. Different levels of data quality assurance are implemented in most of systems, especially those with high data quality importance.

At least two most important procedures are implemented (Galhardas, 2001):

- Restriction of entry of ineligible data (Figure 1);
- Cleaning of collected data (Figure 2).

[1] The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "VeTIS" (Reg.No. B-07042)

Vasilecas O. and Lebedys E. (2009).
IMPLEMENTATION OF RESTRICTION AND VALIDATION RULES.
In *Proceedings of the International Conference on Health Informatics*, pages 230-236
DOI: 10.5220/0001542202300236

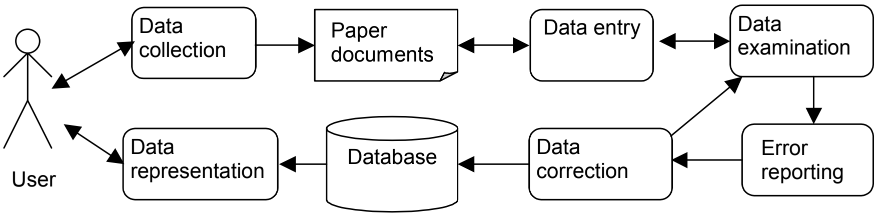

Figure 1: Data processing using restriction rules.

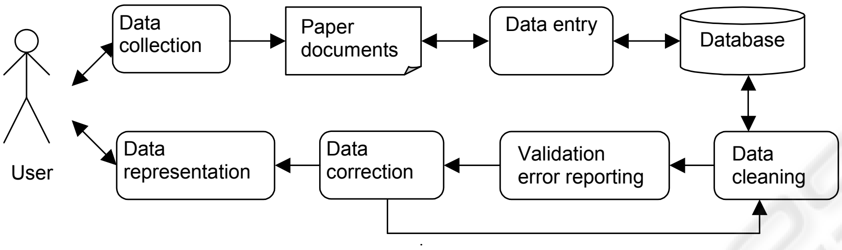

Figure 2: Data processing using validation rules.

Collected data may become invalid even if restriction procedures are implemented, thus constraints prohibiting the entry of erroneous data are not enough. Data cleaning consists of three steps: auditing of data to find violations, choosing transformations to fix violations and applying the transformations on datasets (Vijayshankar 2001). Depending on the application domain, all steps of data cleaning may vary: different methods of data auditing may be used, different methods for data transformation can be used. Rules for data auditing have to be defined, in most cases. These rules are used to validate date and identify discrepancies. Data transformation is the next step and it is based on automated correction of discrepancies or changes in data requiring user interaction.

Implementation of both procedures, restriction of entry of ineligible data and cleaning of collected data, are based on rules (Galhardas, 2001). These rules may be semantically identical, but the intent of use and the way of implementation is different. Depending on the intent of use and the way of implementation these rules are classified into:

- Restriction rules;
- Validation rules.

Restriction rules are also called constraints in software systems. Constraints in software systems were used to avoid errors in data for many years. Constraints are effective to avoid user errors and are widely used up until now. Although, constraints are implemented in most of information systems, errors in data still occur. Constraints are not suitable to eliminate errors that occur due to erroneous program code, security problems and unexpected access to the database, inaccurate mass data updates, inadequate data representation (Akiyama, 2005), (Ballou, 2006). 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. Validation rules are used to examine collected data with intent to identify data units non compliant with defined requirements.

The following are the active database management system (ADBMS) features used to implement restriction rules:

- Data types;
- Not null constraints;
- Lists and ranges of values;
- Multiplicity of relationships between tables;
- Primary and foreign keys;
- Referential integrity constraints;
- Stored procedures;
- Triggers.

Validation rules can be implemented in ADBMS using stored procedures. All the other database management system features are real-time based and are proceeded at the time of data manipulation – data entry, data update or data delete. As validation rules are executed manually and react to real-time data changes, these ADBMS features are not suitable for validation rules implementation. Rule repositories are also often used in parallel with stored procedures for data validation.

Table 1: The use of ADBMS features to assure data quality.

| Data quality category | ADBMS feature | Restriction rules | Validation rules |
|---|---|---|---|
| Integrity | Data types | X | |
| Completeness | Not null constraints | X | |
| Consistency, Integrity | Lists and ranges of values | X | |
| Completeness | Multiplicity of relationships between tables | X | |
| Consistency, Integrity | Primary and foreign keys | X | |
| Consistency, Integrity | Referential integrity constraints | X | |
| Completeness, Consistency, Integrity | Stored procedures | X | X |
| Completeness, Consistency, Integrity | Triggers | X | |

## 3 PURPOSE OF RESTRICTION RULES AND DATA VALIDATION RULES

We distinguish restriction rules based data processing and validation rules based data processing. Restriction rules are implemented in almost all systems and even in the systems based on data validation, some restriction rules are used. Restriction rules and validation rules are mostly used to ensure the quality of data in respect to the following data quality categories (Strong, 1997), (Wang, 1996):

- Completeness;
- Consistency;
- Integrity.

Missing values and data inconsistencies have to be identify to make data clean. Different types of inconsistencies have to be considered when designing both restriction and validation rules:

- Data inconsistency within one record;
- Data inconsistency within one table
- Data inconsistency within one database
- Data inconsistency within one different data sources

Different types of inconsistencies influence different data quality categories and different tools are used to clean these inconsistencies (Table 1).

As already mentioned before, restriction rules are triggering each time data are inserted, updated or deleted. Restriction rules are based on an automatic reaction to the changes in data state. Usually validation rules are executed manually or at the scheduled time points, but these rules never react to data changes at the time data are changed. Thus, the means suitable for implementation of restriction rules and validation rules differ.

This paper is focusing on validation rules based data processing and only implementation of validation rules is discussed further.

## 4 DATA QUALITY IN CLINICAL TRIALS

It is obvious that the means for data collection and analysis in clinical trials have to be precise, qualitative, verified and validated. These requirements also stand for the applications used in clinical trials for data inter-change, data entry, data clarification, data records tracking, etc. The lack of the system for gathering and managing all the requirements for particular trial complicates the control of quality. Requirements for particular clinical trial may be expressed as business rules. The use of business rules approach principles may facilitate the control of the quality. This is especially applicable to clinical trial applications. Trial related knowledge and know-how knowledge stored in business rules repository gives a broad view on a whole of all requirements. The question how to gather all the rules in to one repository arises here. We propose to use the model of a clinical trial to gather all the rules in to rules repository. The modelling of a clinical trial may slightly prolong clinical trial deign and may require additional resources, but it is definitely advantaged. First of all a graphical model of a clinical trials gives a broad view on the organisation and the procedures of a clinical trial. Besides, clinical trial model may be suitable to capture trial related rules.

There is no general specification of all requirements for clinical trial and it complicates quality control. The model of the clinical trial is not created during the design of the clinical trial mostly. As a result of the clinical trial design a clinical trial protocol is produced. The clinical trial protocol presents all the information needed for the conduct of the clinical trial, but the information is represented in natural language. The use of natural language for clinical trial description has both negative and positive aspects:

- the positive aspect of the use of natural language is clarity of the protocol for everyone interested in the clinical trial. In other words the protocol is understandable, easy readable and does not require any special knowledge;
- the negative aspect of the use of natural language is ambiguity of natural language. Natural language is informal and can be interpreted. As clinical trial protocol is the primary document for the conduct of clinical trial it is desirable to have unambiguous specification of all trial procedures.

The use of some formal or semi formal modelling language for clinical trial modelling may allow reduce the ambiguity of the protocol. But as there are special requirements for the clinical trial protocol and it has to be approved before the start of the trial, it is impossible to present a model of the trial instead of clinical trial protocol to the responsible authorities. Thus a model of a clinical trial cannot replace protocol. A model of a trial may be prepared in parallel with the construction of protocol instead of replacing the clinical trial protocol with the model of the trial. It would be even better to start the design of the trial from the model, but it may be impossible, because the design of the model may prolong the design of the study. Therefore the trial should be modelled using any formal or semi formal language to represent the procedures of the trial in a graphic way in parallel with the design of the protocol or just after the clinical trial protocol is created.

There are many modelling languages suitable to represent different aspects of systems – UML, IDEF, conceptual graphs, etc (Vasilecas, 2005). As UML became the most popular modelling language for any kind of systems in recent years, we analyse the use UML for clinical trial modelling in this paper. The Unified Modelling Language is a visual language for specifying, constructing and documenting the artefacts of systems. It is a general-purpose modelling language that can be used with all major object and component methods, and that can be applied to all application domains (e.g., health, finance, telecom, aerospace) (OMG, 2006). UML diagrams can be classified into three different classes (Shen, 2005):

- diagrams describing the roles and obligations of system users generally (Use Case diagrams). In the clinical trial models these diagrams should represent the roles and obligations of the clinical trial team members and participants. For example, the right to revoke the patients informed consent or the obligation of investigator to record medical history in the Case Report Form can be represented in the UML Use Case diagrams;
- diagrams describing structural system aspects (class and object diagrams). In the clinical trial model class diagrams should be used to represent the organisation of a trial in detail. For example, each examination, visit, laboratory assessment, etc., should be represented as classes with attributes and operations. Class model may be used to create the structure of the database for the clinical trial data;
- diagrams describing the internal and external behaviour of system (state transition diagrams, sequence and collaboration diagrams). In the clinical trial models these diagrams should be used to represent the sequence of actions in each step of a clinical trial. For example, the proceeding of screening visit can be described in sequence or collaboration diagrams and the states of the patient diary can be represented in state transition diagrams.

UML models are not fully formal. Some information represented in UML diagrams can be interpreted, but generally UML models are suitable for automation of systems development. We assume that the use of UML would greatly improve clinical trial design and data validation processes. We highlight the following main advantages of UML usage for clinical trial research:

- UML model would give a broad graphical view on the whole trial. This would improve quality control and documentation of clinical trial procedures;
- Duties and responsibilities of clinical trail team members represented in UML Use Case models would simplify preparation of operational manuals for investigators and other team members;

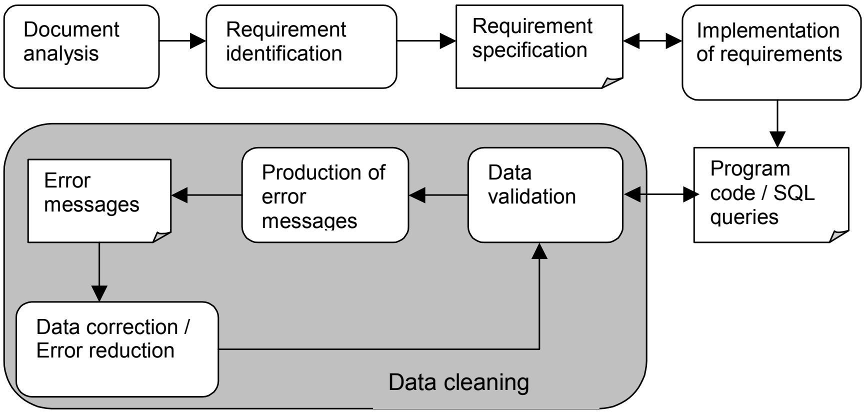
Document analysis
Requirement identification
Requirement specification
Implementation of requirements
Program code / SQL queries
Data validation
Production of error messages
Error messages
Data correction / Error reduction
Data cleaning

Figure 3: Common data cleaning process.

- The organisation of clinical trial structural components represented in UML Class diagrams, can be used for clinical trial database design;
- Representation of all requirements for valid clinical trial data in one model would give a broad view on all rules for data validation;
- The rules described in UML models for valid data, may be retrieved from UML model and placed into rules repository (Vasilecas, 2005). It would simplify the extraction of rules from source documents.

Because of the limited space of the paper clinical trial modelling using UML is not discussed in detail. The novelty of this paper is not the use of UML for modelling the specific domain – clinical trials. The aim is to present a way for automation of clinical trial data validation and improve data clarification.

## 5 IMPLEMENTATION OF VALIDATION RULES

Common data cleaning approach is based on manual identification of system requirements and manual implementation of data validation rules. All related documentation is analysed to identify system requirements. Requirements specification is the basis for manual implementation of validation rules expressed in requirements specification. Data validation is executed then to identify data errors and produce error messages (Figure 3).

It is always desirable to reduce the amount of manual job, because automated activities are more reliable. Previous analysis showed that rules may be derived from systems models represented by UML (Vasilecas, 2006). Derived rules can be used for further analysis and automated implementation. The Unified Modelling Language was chosen for analysis, because it is a general-purpose modelling language that can be used with all major object and component methods, and that can be applied to all application domains (OMG, 2006). UML diagrams can be classified into three different classes (Shen, 2002):

- diagrams describing the roles and obligations of system users generally (Use Case diagrams);
- diagrams describing structural system aspects (class and object diagrams);
- diagrams describing the internal and external behaviour of system (state transition diagrams, activity diagrams, sequence and collaboration diagrams.

Business rules in Use Case diagrams mostly appear as statements describing system actors competence boundaries and obligations. Business rules in Use Case diagrams are represented relating system functions with domain actors. Each Use Case can be depicted in details using sequence and collaboration diagrams. Sequence and collaboration diagrams represent how system actors act and exchange information to execute the tasks they are assigned. Sequence and collaboration diagrams include business rules describing the exact order of actions to be executed to perform a task. State transition diagrams are used to specify the sequences of changes of states of business objects. Event-Condition-Action (ECA) rules are mostly represented in state transition diagrams. UML activity diagrams can be used to model the logic of the operations captured by a use case or a few use cases. Activity diagrams represent both the basic sequence of actions s well as the alternate sequence of actions. ECA rules are mostly represented in activity diagrams. Class diagrams include rules that express constraints of business objects, properties of

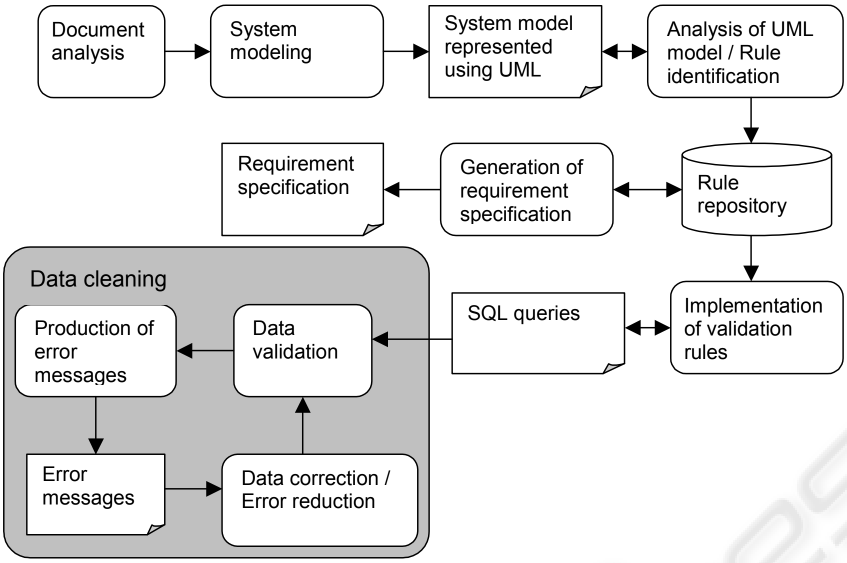
Document analysis
System modeling
System model represented using UML
Analysis of UML model / Rule identification
Requirement specification
Generation of requirement specification
Rule repository
Data cleaning
Production of error messages
Data validation
SQL queries
Implementation of validation rules
Error messages
Data correction / Error reduction

Figure 4: Automated data cleaning process.

relationships between business objects. The rest of UML diagrams. The rest of UML diagrams are used to represent aspects of the development and implementation of software systems and are not analysed further.

An automated data cleaning approach based on derivation of rules from system models represented by UML is presented further (Figure 4). Commercial case tools are storing UML model data in XML files usually. In the current stage of research, only Sybase PowerDesigner was chosen for analysis. The file PowerDesigner produces is used for analysis and identification of validation rules. Only the rules that meet the rule representation templates are derived from UML models. First of all the system parses XML file that stores UML model specification. The system looks for the following components in the XML file:

- actors, use cases and associations that were represented in Use Case diagrams;
- classes, attributes, operations and relations that were represented in Class diagrams;
- objects and messages that were represented in sequence and collaboration diagrams;
- start points, activities, transitions, decisions and end points that were represented in activity diagrams;
- start points, states, transitions, and end points that were represented in state chart diagrams.

Each model component is copied to rule repository. Repository data are indexed after all model components are copied to rule repository. All UML diagram components are merged using the available rule templates. The rules in the repository can then be used to generate natural language requirement specification. SQL queries for data validation are also generated using the rules in the repository. Further steps of data cleaning a similar to restriction rules based data processing. The main difference between these two approaches is automated generation of data validation queries using the rules derived from system models.

Further analysis is focused on manual definition of data validation rules in addition to the rules derived from system models.

# 6 CONCLUSIONS

Analysis of recent researchers in data quality area showed that data quality is relevant for each organisation and due to its complexity is a problematic research area. Implementation of restriction rules is now always sufficient and additional data cleaning procedures have to be implemented to have data of a high quality. On the basis of the previous research we decided that data quality requirements might be derived from system models represented by UML. Thus we proposed an automated data validation rules based approach that focuses on automatic derivation of data quality requirements from system models. The proposed method was implemented in software prototype and was briefly presented in the paper.

# REFERENCES

Akiyama, I., Propheter, S. K. (2005) Methods of Data Quality Control: For Uniform Crime Reporting Programs. Federal Bureau of Investigation. URL: http://www.fbi.gov/hq/cjisd/data_quality_control.pdf.

Ballou, D. P., Chengalur-Smith, I.N., Wang, R.Y. (2006) Sample-Based Quality Estimation of Query Results in Relational Database Environments. IEEE Transactions on Knowledge and Data Engineering, Vol.18, No.5, 639-650.

Bertolazzi, P., Scannapieco, M. (2001) Introducing data quality in a cooperative context. In the proceedings of the Sixth International Conference on Information Quality (IQ2001), Boston, MA, USA, 431-444

Ehlers, U.D., Goertz, L., Hildebrandt, B., Pawlowski, J. M. (2005) Quality in e-learning: Use and dissemination of quality approaches in European e-learning. A study by the European Quality Observatory, Luxembourg: Office for Official Publications of the European Communities, URL: http://www2.trainingvillage.gr/etv/publication/ download/panorama/5162_en.pdf.

Galhardas, H., Florescu, D., Shasha, D., Simon, E., Saita, Ch. (2001) Declarative data cleaning: Language, model, and algorithms. In the International Journal on Very Large Data Bases(VLDB), 371–380.

Jilovsky, C. (2005) Data quality – what is it and does it matter? Presented at Information Online 2005 Exhibition & Conference, Sydney, URL: http://conferences.alia.org.au/ online2005/papers/c12.pdf

McKnight, W. (2004) Overall Approach to Data Quality ROI. White Paper, Firstlogic Inc. URL: http://www.oracle.com/technology/products/warehouse/pdf/Overall%20 Approach%20to%20Data%20Quality%20ROI.pdf.

Motro, A. and Rakov, I. (1996) Estimating the Quality of Data in Relational Databases. In Proceedings of the 1996 Conference on Information Quality, 94-106.

Object Management Group (OMG). (2006) Unified Modeling Language (UML) Specification: Infrastructure version 2.0. URL: http://www.omg.org/docs/ formal/05-07-05.pdf.

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

Strong, D. M., Lee, Y. W., Wang, R. Y. (1997) Data Quality in Context. Communications of the ACM, Vol. 40, No.5, 103-110.

Vasilecas, O., Lebedys, E., Laucius, J. (2005) Repository for Business Rules Represented in UML Diagrams. Izvestia of the Belarusian Engineering Academy, V1 No. (19)/2, 187-192.

Vasilecas, O., Lebedys, E. (2006) Moving business rules from system models to business rules repository. INFOCOMP, V5, No 2, 11-17.

Vijayshankar, R. and Hellerstein, J.M. (2001) Potter'sWheel: An Interactive Data Cleaning System. In the International Journal on Very Large Data Bases(VLDB), 381-390.

Wang, R. Y., Strong, M. D. (1996) Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems, Vol. 12, Issue 4, 5-33.

Yin, T.C.T. and Chan, J.C.K. (1988) Neural mechanisms underlie interaural time sensitivity to tones and noise. In: W.E. Gall, G.M. Edelman and W.M. Cowans (Eds.), Auditory Function: Neurobiological Bases of Hearing. John Wiley, New York, pp. 385-430.

Smith, J., 1998. The book, The publishing company. London, 2nd edition.