*Journal of Universal Computer Science, vol. 26, no. 1 (2020), 107-126*
*submitted: 30/12/18, accepted: 15/11/19, appeared: 28/1/20* *CC BY-ND 4.0*

# User-Oriented Approach to Data Quality Evaluation

**Anastasija Nikiforova**
(University of Latvia, Riga, Latvia
Anastasija.Nikiforova@lu.lv)

**Janis Bicevskis**
(University of Latvia, Riga, Latvia
Janis.Bicevskis@lu.lv)

**Zane Bicevska**
(University of Latvia, Riga, Latvia
Zane.Bicevska@lu.lv)

**Ivo Oditis**
(University of Latvia, Riga, Latvia
Ivo.Oditis@lu.lv)

**Abstract:** The paper proposes a new data object-driven approach to data quality evaluation. It consists of three main components: (1) a data object, (2) data quality requirements, and (3) data quality evaluation process. As data quality is of relative nature, the data object and quality requirements are (a) use-case dependent and (b) defined by the user in accordance with his needs. All three components of the presented data quality model are described using graphical Domain Specific Languages (DSLs). In accordance with Model-Driven Architecture (MDA), the data quality model is built in two steps: (1) creating a platform-independent model (PIM), and (2) converting the created PIM into a platform-specific model (PSM). The PIM comprises informal specifications of data quality. The PSM describes the implementation of a data quality model, thus making it executable, enabling data object scanning and detecting data quality defects and anomalies. The proposed approach was applied to open data sets, analysing their quality. At least 3 advantages were highlighted: (1) a graphical data quality model allows the definition of data quality by non-IT and non-data quality experts as the presented diagrams are easy to read, create and modify, (2) the data quality model allows an analysis of "third-party" data without deeper knowledge on how the data were accrued and processed, (3) the quality of the data can be described at least at two levels of abstraction - informally using natural language or formally by including executable artefacts such as SQL statements.

**Keywords:** Data quality, Data object, Domain-specific language, Platform-independent model, Platform-specific model, Executable model
**Categories:** H.0, H.1.0, H.2.0, E.0, I.6.5

## 1 Introduction

The issue of data quality has been topical for many decades, and its importance doesn't decrease. Currently, data are everywhere and are even considered the most valuable resource, even in comparison with oil [The Economist, 17]. In most cases, data are used in analysis and as a basis for decision-making. It means that in order to

make informed decisions, data must be of high quality, as low data quality can lead to huge losses. As an example, according to [ComputerWorld, 15], the US Postal Service (USPS) loses $3.4 billion per year due to only incorrect address data.

Most existing studies on data quality focus on the definition of data quality dimensions and their relation to the specific issues. The main disadvantage of such approaches is the demand to analyse quality dimensions in context of specific solutions, and it is a difficult task for non-IT and non-data quality experts (DQ-experts). This process is very important but at the same time very time-consuming, as the correctness of the users' choice impacts the results of the analysis. One of the main ideas of the proposed approach is to avoid data quality dimension concept, replacing it with a more general concept of "data quality requirement". This doesn't require researching dimensions, their meaning etc. The paper presents a data object-driven approach to data quality evaluation, which consists of three components that form a data quality model: (1) data object; (2) data quality requirements; (3) data quality evaluation process.

This paper is an extension of work originally presented at the "*Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS-2018)*" [Bicevskis, 18a]. The main idea presented in our conference paper was to build a data quality model in two steps: (1) creating a platform-independent model (PIM) and (2) converting the created PIM into a platform-specific model (PSM). This allows to describe data quality at two levels of abstraction: (a) informally, using natural language and (b) including executable artefacts, for instance, SQL statements, in the result deriving PSM from PIM. In the conference paper we discussed existing solutions for the data quality issue, briefly explaining the idea of a division of the data quality model into PIM and PSM, providing a very limited overview of its application to real data sets without explaining analyses' relationship with use-cases and a discussion of the results. In comparison with [Bicevskis, 18a], this paper presents a more detailed description of the proposed approach, including a description of every component of the proposed data quality model, their nature, and possible ways to create them. The main new points discussed in this paper are (a) the rationale for choosing every component and their comparison with possible alternatives, (b) a description of the implementation, (c) a detailed description of its application to real open data sets, summarizing the list of advantages of the proposed solution that arise as a the result of its application to numerous "third-party" data sets. As a result, a list of possible extensions of the presented idea arose, some of which were partially implemented.

The paper is structured as follows: an overview of related researches (Section 2), a description of the presented solution (Section 3), a description of the implementation of the presented solution (Section 4), an analysis of data quality of Companies House of UK, and comparative analysis of four European Company registers (Section 5), conclusions (Section 6) and future work (Section 7).

## 2 State of the art

More than 20 existing solutions on data quality issue were already discussed in detail in [Bicevskis, 18a] and [Nikiforova, 18b], where their pros and cons were highlighted, therefore, this discussion won't be repeated in this paper. However, the list of the

most influential groups of the existing researches should be provided with the list of examples for the further discussion: (a) general studies on data and information quality, mostly focusing on the definition of quality dimensions and their groupings - [Wang, 96], [Van den Berghe, 17], [Ferney, 17], [Redman, 01]; (b) assessments of specific industry data and information quality, analysing data from a specific industry with sector-specific methods, for instance, healthcare – [Van den Berghe, 17], [Schmidt, 15], [Tomic, 15], chemical hazard and risk assessments - [Bevan, 12]; (c) data quality assessment frameworks [Vetrò, 16], [Neumaier, 16], [Umbrich, 15]; (d) quality assessment of open data portals or Open Government Data [Vetrò, 16], [Neumaier, 16], [Sáez Martín, 16], [Sasse, 17]; (e) quality assessment of Linked Data [Acosta, 13], [Paulheim, 14]; (f) data quality guidelines for data publishers [Vetrò, 16], [Sasse, 17]. Another research takes a significantly different approach [Baker, 18] and discuses cloud-based tools for next-generation sequencing. It provides a review of 20 tools and is probably one of the most comprehensive studies on the topic.

The first group that focuses on the definition of data quality dimensions and their grouping has been probably the most frequently used in the last decades. However, this research method presents multiple disadvantages and limitations. The total number of dimensions used in these researches is very high. There are many different classifications of data quality dimensions where one of the most widely known is Wang's and Strong's 15 data quality dimensions [Wang, 96] divided into four categories: intrinsic, contextual, representational, accessibility. According to Batini and Scannapieco [Batini, 16], these dimensions were identified as a result of empirical research. Theoretical approach used by Wang and Wand highlighted such 5 most important dimensions as accuracy, reliability, timeliness, completeness and consistency as the most important aspects of data quality [Wand, 96]. Division of data quality dimensions into three categories - conceptual schema, data values and data format - is an example of the intuitive approach [Redman, 97]. One of the newest definitions of data quality dimensions is the set of 6 dimensions, namely, completeness, uniqueness, timeliness, validity, accuracy, consistency, defined by Data Management Association International UK Working Group [Askham, 13]. Furthermore, some researches suppose that every dimension has a list of criteria. However, these researches aren't the soles, as many researches provide other data quality dimensions as more "comprehensive". Moreover, the number of dimensions can vary from 3 [Ferney, 17] and [Adedugbe, 18] to 30 [Caro, 07]. Usually, there are two ways to determine the dimensions: (1) pre-defined by authors; (2) defined by users. Pre-defined data quality dimensions are usually defined: (a) as a result of surveys (involving developers, students, end-users etc.) [Vetrò, 16]; (b) as a result of the existing research overview; (c) by authors in accordance with their "vision". The main disadvantage of these approaches is the necessity to make deep analysis on dimensions in the context of a specific solution. It is time-consuming but at the same time very vital process as the results of analysis are highly dependent on the correctness of the choices made by users.

In addition, sometimes the number of data quality dimensions is not only too high, but also the difference between some of them is almost invisible. Moreover, although many solutions propose the same names for various data quality dimensions, their meanings are believed to be different, and vice versa – different names are used to describe the same semantics. In [Scannapieco, 02], six research papers providing

set of data quality dimensions were compared. The main result was that only 1 of 23 analysed data quality dimensions, namely, "accuracy", had the same semantical meaning in all 6 researches. This is in line with [Askham, 13] stating that "even amongst data quality professionals the key data quality dimensions are not universally agreed. This state of affairs has led to much confusion within the data quality community and is even more bewildering for those who are new to the discipline and more importantly to business stakeholders". In addition, according to [Batini, 16] and [Scannapieco, 02], dimensions can't be defined in a measurable and formal way, and it is also not known how they should be assessed. It means that not only a data quality concept is very complex and doesn't have a straightforward definition, but the same applies to the concept of a data quality dimension and every particular dimension.

As a result, these researches become useful and suitable mostly for DQ-experts. Moreover, even for them an analysis of data quality dimensions sometimes become inadequately time-consuming. As a result, we conclude that data quality dimension is too relative and at the same time complex concept. It requires deep knowledge and skills in data quality area, as well as an effort to understand what each dimension means for a particular solution. It creates the necessity to involve DQ-experts at each stage of the data quality analysis. However, as data quality depends on use-cases defined by particular users, at least the first steps of data quality analysis - defining quality requirements - should be done by end-users. Involving of DQ-experts in initial stages of data quality analysis means that data quality requirements and conformity to end-users needs significantly depend on the interpretation of user requirements by DQ-experts; though it is often hard to explain and understand a specific problem.

One of the main points of the presented solution is to let users analyse data quality as much as possible independently from DQ- and IT- experts. It means that the proposed solution should be clear and simple enough. The data quality dimensions concept is replaced with a more general and comprehensive concept of "data quality requirements", whose subset is "data quality dimension". It doesn't require definition or application of pre-defined dimensions since all requirements are fully defined by end-users and strongly depend on the use-case.

## 3 Basics of the proposed solution

The main idea of the presented data quality solution was initially proposed in [Bicevska, 17]. It was discussed in the context of Model-Driven Architecture (MDA) in [Bicevskis, 18a]. MDA was also discussed by other researchers at SNAMS-2018 conference - [Khider, 18] considered MDA for business process model recommender. It shows the continuous importance of MDA in the IT-related research.

This paper provides a more detailed description of the proposed data object-driven approach, PIM and PSM models of the proposed quality mode. It justifies the approach, as well as highlights its pros and new perspectives.

The proposed data object-driven data quality solution consists of three main components: (1) data object, (2) description of quality requirements, and (3) description of quality measuring process [Bicevska, 17]. These components form the data quality model: the data object description defines data whose quality should be analysed, the quality requirements description defines conditions that must be met to admit data as qualitative, and the description of quality measuring process defines a

procedure that must be performed to evaluate data quality. Each component is defined using graphical flowchart-based diagrams.

The proposed data quality model can be built in two steps: (1) firstly, a platform-independent model (PIM) of data quality is created; (2) then a platform-specific model (PSM) is derived from the PIM, by replacing informal descriptions with some executable artefacts such as programming code routines or SQL statements. The PIM consists of informal descriptions of data object, data quality specification and data quality measuring process. The PSM contains implementation of the data quality model thereby making it executable.

The division of data quality model into two models corresponds to Model-Driven Architecture (MDA). Possibly, it is not MDA in its traditional meaning, however, the principles are the same. According to [Kleppe, 03], MDA by itself "… is based on widely used industry standards for visualizing, storing and exchanging [software designs and] models". This is the core idea of the presented approach. Following Object Management Group (OMG) [Soley, 00], in the proposed solution "models become assets instead of expenses". Moreover, "modelling technology to pull the whole picture together" is one of the main aims of diagrams use. In comparison with OMG, a PIM is created based on the block diagrams (flowcharts) instead of Unified Modelling Languages (UML) diagrams that are widely used in MDA [Kleppe, 03]. However, UML diagrams often require specific knowledge and previous experience. According to [Kleppe, 03], UML is one of the most appropriate choices for engineers, as it allows exchanging and documenting their ideas. However, UML is not suitable for non-IT and non-DQ experts, therefore it can't be used for the proposed solution. At the same time, flowcharts are a simple and intuitive way to express ideas even for non-IT and non-DQ experts, and they are often included in educational programs for secondary schools (at least in Latvia). As a result, authors suppose flowcharts are easy to create, read and modify for the majority of users because they have all necessary components for data quality analysis. Therefore, flowchart-based diagrams were chosen as a more appropriate option for the proposed solution.

As programming languages and platforms may have significant differences in their semantics, PIM transformation into PSM takes place manually. Despite there are many options for automated and semi-automated transformation of PIM into PSM, it is almost impossible to ensure the correct translation of PIM defined by users into the PSM. Besides, as it was previously mentioned, the presented solution doesn't follow MDA in its traditional understanding. One of the main reasons to choose manual transformation is the fact, that the manual transformation of models task isn't effort- and time- consuming in this case – it is relatively simple task, especially for users with basic programming skills which will be required in the later stages of quality analysis only (agrees on [Miller, 03], [Pauker, 16], [Chungoora, 13]).

All concepts of the presented data object-driven approach are defined and described using graphical Domain Specific Languages (DSLs). DSL syntax and semantics are developed in such a way that they are (a) easily applicable to a new information system, (b) simple enough to let non-IT experts define data object and quality specification without IT-experts involving. Graphical models for data quality analysis were chosen for several reasons. Firstly, models are usually used as a communication tool [Mellor, 04], improving the readability of information since graphical representation in models is perceived better by readers than textual

representation. Visual information is easier and faster to read and to modify. Moreover, using models reduces the risk of misunderstandings between users. Secondly, according to [Mellor, 04], models are "cheaper to build than the real thing". Mellor emphasizes that the effectiveness of models depends on two aspects: abstraction and classification. By abstraction Mellor understands "ignoring information that is not of interest in a particular context". In the presented approach, it is achieved by using data object exclusively with the parameters representing real objects that are of interest for specific users in specific use-cases. Parameters that are not of interest for specific use-cases are ignored, hence they aren't included in the particular data objects. By classification Mellor means "grouping important information based on common properties". This principle is partially followed when grouping quality conditions for each parameter involved in data quality analysis. In [Kleppe, 03], the authors propose to create machine-readable models instead of the paper-based to reduce time- and effort- consuming activities. They offer to store machine-readable models in standardized repositories. In the presented approach, a graphical DSLs editor DIMOD [Bicevskis, 18a] is used to store created diagrams in repository. The basics of the proposed solution are close to those in [Kleppe, 03]: models can be easily updated, modified and reused. The diagrams can be transformed multiple times depending on user's needs as soon as use-case changes or new details appear. The changes can be introduced by multiple users if several people are involved in data quality analysis (in diagrams creation). Besides, involved people can represent different units, for instance, technical- and business- units or any of them.

Each diagram represents one of two data quality lifecycle control phases proposed by Total Data Quality Management [TDQM, 18]: (1) data quality definition that is divided into (a) data quality definition and (b) data quality measuring; (2) data quality analysis. Sometimes, data quality definition and data quality measuring represent two standalone phases. In general, TDQM introduces an additional stage – data quality improvement. The presented solution doesn't cover data quality improvements phase leaving it up to user as there are many useful and user-friendly solutions for this purpose. One of such solutions is Microsoft Data Quality Services that was previously analysed in [Bicevskis, 18a] and [Nikiforova, 18b]. In accordance with TDQM, data quality is assured systematically repeating phases of data quality cycle. The necessity of the repeating appears when data are continuously changing. New data or data modifications can bring new data quality problems and cause new requirements [Nikiforova, 18b]. The proposed approach ensures high quality of data because quality assessment criteria are changed, and new data quality requirements are defined in each cycle's iteration. As a result, the created diagrams represent the current stage of each data quality analysis step – not only the final result of the design phase, that also corresponds with [Kleppe, 03]. To sum up, the general idea of data quality models' definition using PIM and PSM corresponds with the MDA. However, there are differences - the proposed solution is adapted for non-IT and DQ experts as their involving in data quality analysis is one of the main aims of the proposed approach.

### 3.1 Data Object

The first step of the proposed data object-driven approach is a data object definition. According to [Bicevska, 17], a data object is "a set of values of the parameters that

characterize a real-life object". The data object requires to select only those parameters, in which quality specific user is interested in. This affects performance of data quality analysis significantly, since only the parameters whose quality matters are selected for further processing. The parameters of data objects depend on use case defined by data end-user. A similar idea is discussed in [Salesi, 18]: limiting the data features before submitting data to deep learning model improves performance.

When analysing data of Companies House of UK [Companies House, 18], it is obvious that the data object is Company ("Company_UK"). Every company is described using 55 parameters. Two very simple and intuitive use-cases were chosen: (a) identify company by its name, registration number and incorporation date; (b) contact company via mail post using its address and postal code. Therefore only 5 attributes for the data object "Company_UK" are necessary to cover the use-cases: "*CompanyNumber*" – company registration number, "*CompanyName*" – company *name*, "*IncorporationDate*" – company incorporation date, "*RegAddress_ AddressLine1*" – company address, "*RegAddress_PostCode*" – company postal code. Other 50 parameters are out of the scope and can be ignored.

A collection of data objects of the same structure forms a data object class. This concept is necessary as information systems usually deal with many data objects that should be processed in a unified way if they have the same structure. According to [Bicevskis, 18a], "a data object class has a name, and its elements have the same structure as they all are characterized by the same parameters". The data objects class consists of several specific data objects, called instances, which are described by fields of arbitrary number and other data objects' classes. Each particular data object can have one to all parameter's values. It means the data object's class has a tree structure. Data object class allows defining data quality requirements for the data collection. It also allows to specify when data quality is considered as high or low by introducing a threshold which can't be exceeded. For instance, if the total error rate of data quality problems of the data class "Company_UK" is lower than 5%, the data set is considered to be of high quality, however, otherwise data quality should be improved immediately. The total rank is calculated by relating the number of records having data quality problems to the total number of records. It also means that every user can introduce his own threshold thereby specifying tolerance intervals.

A document-oriented database is a commonly used example of data object class. It contains documents with the same structure, and the fields of each specific data object can be filled in partially or completely. In the above described particular example, Companies House collects not only current companies' names but also up to ten previous names. Two parameters describe previous name of company: the name itself "*Company_Name_Previous*" and the date when the name was changed "*Change_of_Name_Date*". Hence, the data object class "Company_UK" would have five parameters, and the data object "*PreviousName*" - two parameters.

### 3.1.1 Platform-independent model of data object

Firstly, a platform-independent model (PIM) of the data object should be created. According to [Kleppe, 03], PIM models "work independently of details and specific of the target platforms, there is a lot of technical detail that they do not need to bother with." Figure 1 shows the platform-independent model of the data object "Company_UK" with its 5 attributes. The description of company is informal as no

rules for attribute values' syntax are given. Only parameters' names and a description of stored data in natural language are depicted. The denotation is simple enough [see Fig. 1]. The description of the stored data can be retrieved in several ways: (a) from documentation accompanying data sets, if it is provided; (b) from parameters names; (c) by exploring data set. The first option is time-saving and user-friendly as it doesn't require any additional steps, however, documentation is provided very rarely. The second option - create data sets and publish the data - is widely spread, as it is a kind of "good practice" and nowadays often taken into account.

For the presented example, the data publishers (Companies House of UK) provide documentation containing additional information about the published data. In addition, the names of almost all parameters are self-explaining and don't require any additional analysis. However, this is the only such "user-friendly" data set among all four analysed Company registers (Latvia, Estonia, Norway and the United Kingdom).

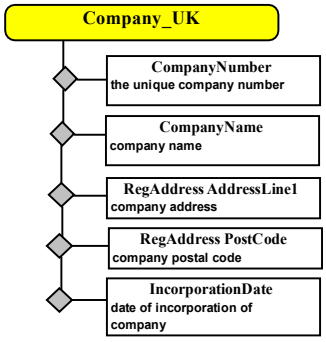


*Figure 1: PIM of data object "Company_UK"*

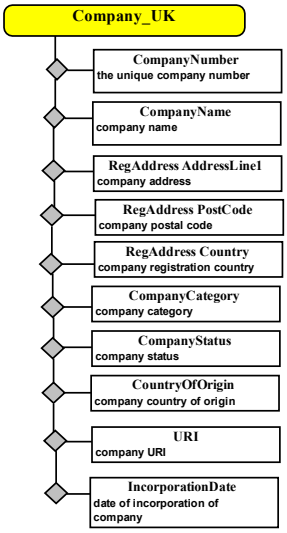


*Figure 2: PIM of the extended data object "Company_UK"*

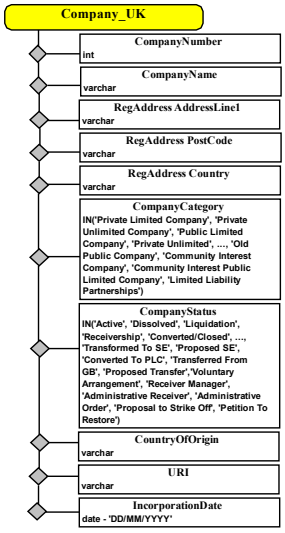


*Figure 3: PSM of the extended data object "Company_UK"*

Figure 1 demonstrates how simple is data quality analysis in the specific use-case is when only few parameters are required, whereas Figure 2 demonstrates an extended version of data object. It is also vital to note that the initial data object can be easily extended at any stage of analysis. It is a flexible object, and the number of data object parameters is determined by the user and the use-case.

As discussed in [Bicevskis, 18a], the data quality checking for one of the data object parameters' values can be reduced to an examination of individual values' properties. The most common controls are: (1) may a string serve as registration number? (2) is a postal code value correct? (3) are the provided dates valid and trustful? Checking of parameters' values is local and formal, i.e., this process does not

respect contextual interlinks with other data objects and does not check the compliance of data with the true characteristics of the real company.

### 3.1.2 Platform-specific model of data object

The second step is to create a Platform-Specific model (PSM) for a data object. It contains technical details that were not included in PIM.

Descriptions of data objects' parameters are semi-formal at this stage as rules for attribute values syntax are provided. The syntax rules for describing the allowable values for the data object's fields can be formulated at different abstraction levels - from formal language grammar to definitions of variables in programming languages. In the latter case, the data object model is closely related to its implementation environment. Figure 3 shows a PSM for the data object "Company_UK" that is derived from the PIM [see Fig. 2]. More interesting example appears - parameters "*CompanyCategory*", "*CompanyStatus*" get attribute "enumerator" (in MS SQL "IN") supplied with list of allowable values that further can be used as data quality requirements' input. The informal rules are replaced by formal rules at this stage, specifying more appropriate data type for each field depending on the values it stores. For instance, company name and address are strings, the format of an incorporation date is "*DD/MM/YYYY*". This information can be obtained in multiple ways: (a) from documentation about data sets provided by a data publisher; (b) from pre-processing, analysing data the most part of parameters contains. In this way, PIM is converted into the PSM.

In the presented example, the data publisher (Companies House of UK) provides the data set together with documentation containing additional information about data and maximal lengths of values. Moreover, another document provides lists of allowable values for parameters. The parameters' format depends on the technique used for replacing informal descriptions with executable artefacts at later stage – data quality evaluation. In the presented example, SQL is used, therefore parameters' types correspond to the SQL data types in most cases [see Fig. 2].

## 3.2 Data Quality Requirements Specification

According to [Bicevskis, 18a], a data quality specification for a specific data object consists of conditions that must be satisfied in order to consider the data object as of high quality. Data quality requirements can be defined at different level of abstraction: as for (a) specific data object, (b) data object in scope of its attributes, (c) data object in the scope of database, and (d) data object in scope of many databases [Nikiforova, 18b].

### 3.2.1 Platform-independent model of data quality requirements specification

If PIM is used for data quality specification, only informal descriptions of conditions are defined, for instance, in natural language. In other words, the nature of quality conditions is fixed in the PIM. The most commonly appeared conditions are checking of value existence and formats. However, other conditions can also appear as well.

The data quality specification is retrieved from the data stored in specific fields or from the description of data set, for instance, from an overview of fields with short summary on them, their length and their possible values provided by Companies

House of UK. This description can be used as the first step in creating of data quality specification, additional quality conditions can be added, if needed. However, some requirements can be specified during pre-processing of the dataset. For instance, in order to decide whether empty values are allowed for a parameter, it is possible to check the number of non-empty values, and, if the ratio of non-empty values is lower than 3%, there could be assumed that the specific parameter may not be empty. Such requirements can be formulated from users' viewpoint. For instance, in the example with Company register, it is obvious that "*CompanyName*" and "*RegistrationNumber*" must have values. Moreover, depending on the register, a registration number should conform to some pattern or correspond to some specific format. The defined conditions are collected and grouped by each parameter involved in data quality analysis. The PIM of data quality specification for the extended data object "Company_UK" is shown in the Figure 4. Successive checking of data object's "Company_UK" fields, describing each condition in a natural language is provided. The 1st – 4th and 10th boxes represent quality requirements for the initial short version of data object [see Fig. 1].

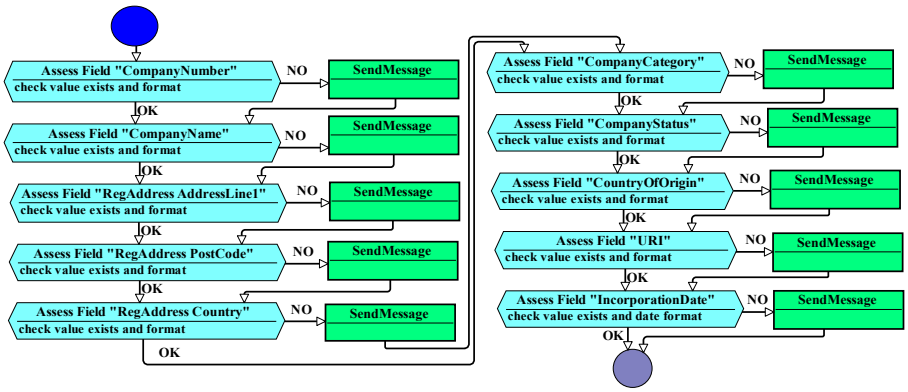


*Figure 4: PIM of data quality requirements specification*

The procedure of data object's class processing is as follows: (1) all instances of data object's class are selected from the data sources and written into collection; (2) all instances are cyclically processed, for each individual instance examining the fulfilment of the quality requirements, likewise in the case of processing an individual data object [Bicevskis, 18b]. In other words, when quality of value of particular parameter is checked, two alternatives are possible: (a) the value is correct and meets all defined quality requirements, or (b) the data have quality problems. In both cases, the quality of the check of the value of the next parameter follows the previous check, however, in the second case, if the previous value had at least one quality problem – didn't satisfy at least one quality condition, an appropriate message will be sent [see Fig. 4].

As described in [Bicevskis, 18b], such a data quality specification can be a programming task for development of data input forms. The next step after defining the PIM is to define the PSM, and this is described in the next Section.

### 3.2.2 Platform-specific model of data quality requirements specification

When creating a PSM, in accordance with PSM nature, data quality requirements for a data object are defined using logical expressions. The PSM is a detailed while a PIM is an abstract model of quality specification.

The structure of the diagram is the same as for the PIM, however, informal descriptions are substituted with logical expressions where names of data object's attributes serve as operands in the logical expressions (statements in programming languages may be used as operations). These logical expressions are mainly derived from the previously defined data object (mainly from the PSM of data object [see Fig. 3]) together with PIM of data object's quality specification [see Fig. 4]. For instance, (a) does the format of parameter "*IncorporationDate*" correspond to the defined? (b) is the value of the parameter "*CompanyCategory*" included into the list of allowable values? (c) does "*URI*" meet the pattern? etc. Data quality specification for the data object is described in a pseudocode written in the elements of the chart. Figure 5 demonstrates quality specification for the data object "Company_UK". Despite the fact that pseudocode sometimes is related to PIM (for instance, in [Ruiz, 18]), this time it can be considered as PSM since the pseudocode is closely related to the art of its implementation, for example, in programming language C#. This choice conforms to other researches ([Coutinho, 12], [Shi, 15], etc).

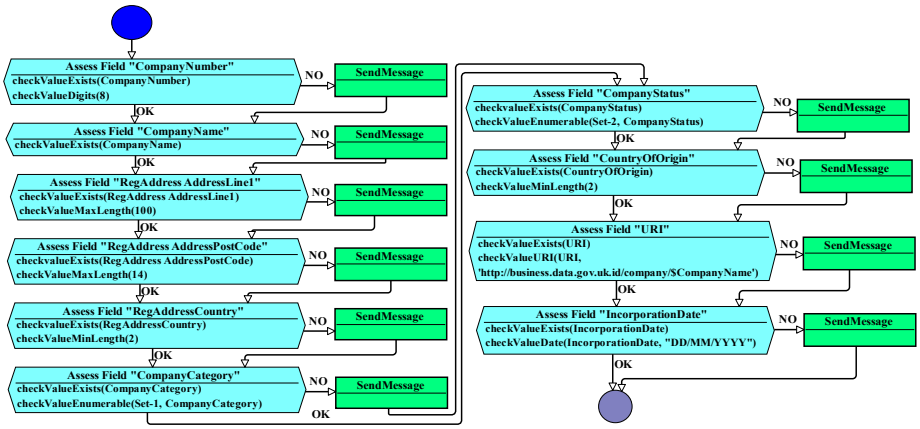


*Figure 5: PSM of data quality requirements specification*

An extended data object brings several specific checks, for instance, in case of the parameter "*URI*", correspondence to a defined pattern is checked. According to this quality requirement, every "*URI*" should start with a certain string while the second part should contain company's name taken from the first parameter ("*CompanyNumber*").

Likewise to previous researches ([Nikiforova, 18a, 18b, 19a], [Bicevskis, 18b]), the most commonly used data quality requirements are: (1) existence of values, (2) relevance to specified type of data, (3) format of stored values (for example, length of the stored value), (4) conformity to a specific pattern, (5) relevance to the list of enumerable values, (6) validity of value (for example, trustful date) and other

conditions that follow from the data set and type of data that can be stored in the specific field.

### 3.3 Data Quality Evaluation

Data quality evaluation process starts with description of activities that are necessary to be taken to select data object values from the data source. When data objects' values are read from the data source and written into database, one or more steps should be taken assessing data quality of the selected values. Each step describes one check for the compliance of the data object with specific quality specification. Therefore, if particular values don't meet defined data quality requirements, an appropriate message is sent [see Fig. 6]. Non-empty "SendMessage" values form data quality problems' protocol that is saved in database for further processing. It can also be used for improving data quality of particular data set automatically or manually by triggering changes in the data source.

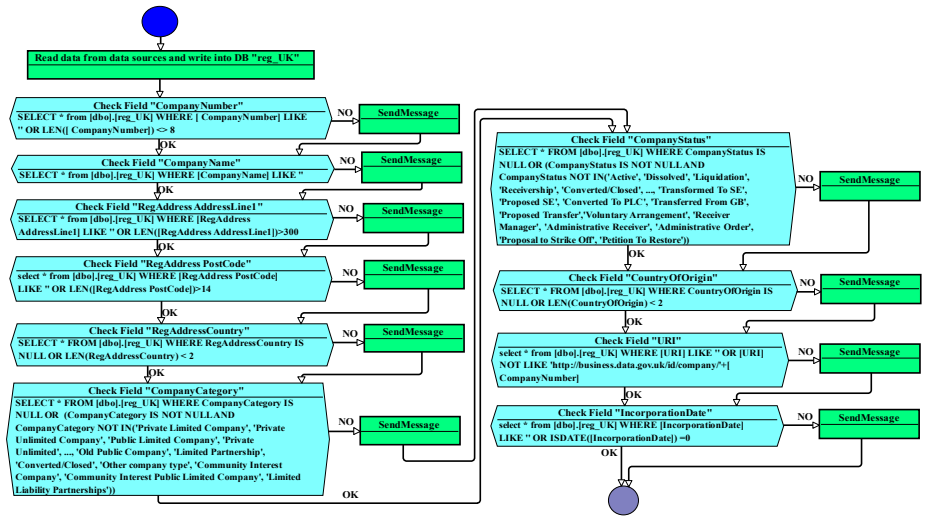


*Figure 6: PSM of data quality evaluation for data object "Company_UK"*

A PSM model of data quality evaluation process is obtained by modifying the PIM into an executable model. The data object class or the data object defined at the first stage (subsection 3.1) is used as an input for quality evaluation process. In the case of a specific data object, when data are entered into the data object fields, the data objects quality conditions are subsequently verified. In the case of data object classes, firstly, instance values of the data object class are selected from the database. Further, instances of the accumulated collection are inspected by verifying quality conditions (defined at the second stage (subsection 3.2)) of each instance. The quality verification process creates a test protocol that identifies the data objects nonconforming with quality requirements. Figure 6 demonstrates the PSM of data quality evaluation process for the data object "Company_UK".

As the first step, data are loaded into database (in the particular case, Microsoft SQL Server is used). Then, data quality of data objects parameters' values is analysed using executable artefacts. In the example, every individual operation evaluates the data quality of the data field using a SQL statement where a SELECT statement defines the data object and a WHERE clause – data quality requirements.

Nature of an executable artefact that replaces informal conditions depends on the person involved at this stage. As these conditions should be executable and thereby syntactically correct, appropriate knowledge and skills are required. This stage requires involvement of person with knowledge in IT area. In other words, executable texts can be represented not only using SQL statements, but also using some programming language although it may impact implementation. SQL is observed as the most appropriate and natural option since the basic construction of every SQL statement is simple and close to the construction of data quality requirements.

## 4 Implementation

As mentioned in Section 3, the presented data object-driven approach uses graphical DSLs for defining data quality models. Every component of data quality model is described using its own graphical DSL. As data quality requirements are different and the used DSLs can also be different, it is highly recommended to not use one specific graphical editor that supports only one DSL (that can become very complex) but to create your own editor for each used DSL [Bicevskis, 18a]. In other words, three DSLs are proposed for the presented approach: for (a) data object definition, (b) data quality specification definition; (c) data quality evaluation. All these DSLs are graphical and each of them has its own structure that was already demonstrated by examples in previous sections.

Currently, there are several platforms providing generation of graphical DSL editors. One of such platforms, called DIMOD, was used for this research. DIMOD lets users define a DSL using a meta-model that is stored into model's repository. Moreover, DIMOD is also based on MDA. It lets define DSL parameters and modification using a separate configuration component "Configurator" [Sprogis, 13]. It creates a meta-model of the DSL and stores it in a repository. Once it is saved in the repository, the corresponding modelling editor is created automatically, the meta-model of DSL is interpreted, and all necessary graphical features become available. Example of a meta-model for data object class is shown in Figure 7.

DIMOD can also be used to create and edit data quality diagrams by highly qualified modelling experts in collaboration with domain experts ("the clever users"). Furthermore, it is possible to check the data quality models' internal consistency involving both – IT and domain experts. This choice lets a wide range of users to use the created data quality model since the created models can be published in WEB. The creation of the DSL meta-model is probably the most complex step in the process as it requires deep knowledge in modelling. In this case, the possibilities provided by DIMOD allow to create the meta-model just once and to publish the created models in WEB. Once the model is shared, other users can start using it without any knowledge on how it is created. Moreover, the published data quality models let users explore previous models before creating the models by themselves.

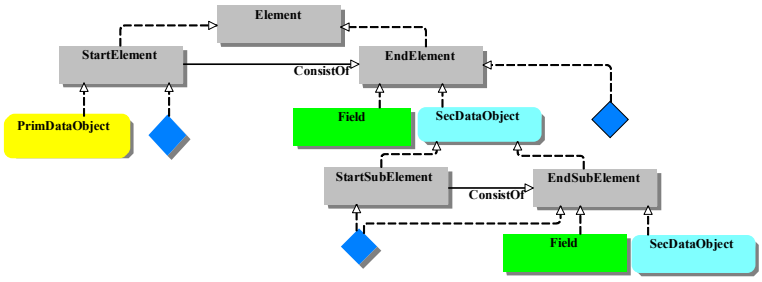


*Figure 7: Data object's class meta-model*

The PSM model is implemented by replacing the PIM's descriptions with executable artefact, for instance, with program code. Another option is to create a compiler that converts created data objects' and data quality specifications' flowcharts into executable code that can be used as PSM for data quality evaluation. However, as already discussed in Section 3, this approach may occur more time- and effort- consuming. Moreover, there is also no confidence that transformation in each particular case will be correct and accurate.

As the last step, a validator is involved to execute the obtained PSM. The defined requirements are used to execute the code. A protocol of data quality problems is generated for each particular data object stored in the new database, and it can be used for analysis and data quality improvement.

## 5 Appliance of the data object-driven approach to open data

The presented data object-driven approach was applied to multiple open data sets to (a) approbate the proposed methodology in real tasks, and (b) analyse the quality of open data sets. In total, more than 25 data sets were analysed ([Nikiforova, 18a, 18b]), including the analysis of a specific domain – Latvian open health[care] data [Nikiforova, 19a], with a description of the analysis and a summary of the results of the application of the proposed approach. The research showed that more than 84% of the analysed open data sets have at least several data quality issues, even in the scope of primary parameters.

Hereafter a short overview of results obtained by analysing four open data sets of company registers from four European countries - Latvia, Estonia, Norway and the United Kingdom - is given. In all cases, a company was selected as the data object class. The analysis was performed according to two use-cases: (a) identifying a company, and (b) contacting a company. The unified analysis of several company registers not only looked for data quality flaws in particular company registers but also tried to identify general data quality problems if any. The structure of data object for Companies House of UK is shown in Figure 1. However, there were additional data quality checks, not only those of defined use-cases, made for all data object parameters in the company registers to have in-depth data quality analysis. As data quality specification was made by users, it was an interpretation of data by users'

viewpoint. At this stage, data object analysis was done in scope of a single data object, mainly the syntactic accuracy was analysed. Figures 2 and 3 demonstrate data quality requirements for the data object class of Companies House of UK.

Applying of the proposed approach to company registers' data led to surprising results. Since (1) the defined use-cases are focused on the primary data characterizing companies and (2) the main aim of company registers is to collect data about companies, it was supposed that the primary data (a) should be complete; (b) can't contain any doubtful or (c) incorrect data. The results showed that the assumption was incorrect (see Table 1).

| **Country** | **Latvia** | **UK** | **Norway** | **Estonia** |
|---|---|---|---|---|
| | the 1st use-case: company identification | | | |
| **Name** | 10 (0.003%) | 1 (0.0001%) | 0 | 0 |
| **Registration number** | 0 | 0 | 0 | 0 |
| **Incorporation date** | 94 (0.02%) | 3 (0.0004%) | 9 (0.001%) | - |
| | the 2nd use-case: company contacting | | | |
| **Address** | 366 (0.09%) | 7 518 (1%) | 68128 (6.2%) | 29918 (11.24%) |
| **Postal code** | 20498 (5.16%) | 12155 (1.6%) | 14683 (1.3%) | 22621 (8.5%) |
| **Parameters with quality problems** | 11 (50%) | 17 (30.9%) | 8 (19%) | 7 (50%) |

*Table 1: Company Registers' analysis. Number of data quality defects*

Regarding the first use-case "company identification" only Company Register of Estonia didn't have any quality problems in the analysed data, but the register didn't provide incorporation dates for registered companies, thus the data didn't completely correspond to the defined use-case. Company Register of Norway had 9 invalid values of incorporation date; though, names and registration numbers didn't have any quality problems. The Register of Enterprises of Latvia and Companies House of UK had data quality problems not only in incorporation dates but also in companies' names. Nevertheless, the presence of these data quality problems does not mean that the data from these company registers could not be trusted when a company must be identified since the number of detected quality problems were low, and their correction would not require much resources.

In the second use-case, the number of detected quality defects was significantly higher. All analysed parameters for all four registers had at least some quality problems. Companies House of UK had 4 invalid addresses and postal codes, 7 514 empty addresses and 12 151 empty postal codes. In case of Company Registers of Estonia and Norway, all numbers provided in Table 1 are related to empty values. However, The Register of Enterprises of Latvia contained 3 invalid postal codes (incorrect length of values), 20 495 empty postal codes and 366 empty addresses. It means that none of the analysed registers may be used to contact every company.

Furthermore, apart missing values and dubious/ invalid dates, several common data quality problems for company registers were highlighted in the result of the extended analysis: (1) missing values for field containing (a) abbreviation while text

field with its explanation is provided and vice versa, (b) ID and textual value; (2) non-allowable values.

Alternatively, according to [Global Open Data Index, 18], Companies House of UK and Company Register of Norway took the 1st place among 94 countries in 2016, and this could let to users believe the data sets are of very high quality. Just as in the presented example, parameters that characterize company are company name, unique identifier or registration number and company address. However, in comparison with our research, mainly the correspondence to the "open data" term is evaluated by them. What matters here is that data quality isn't analysed at all. It means, that even very highly ranked data sets may have data quality problems as data publishers have not spent much effort on analysing the quality of their open data and such services as Global Open Data Index don't check quality of the data sets – this isn't their aim. The presented object-oriented approach could be helpful in such cases as it provides possibility to check quality of "third-party" data sets without any knowledge on how these data were collected and processed by data publishers.

During an extended data analysis of Companies House of UK, data quality problems were detected in 17 parameters. The most common problem observed in 11 parameters was empty values in mandatory fields. However, multiple interesting quality problems were also detected in fields storing countries' names (*"CountryOfOrigin"* and *"RegAddress_Country"*). For example, (1) different names denote one country: (a) 926 044 values are "*United Kingdom*" while 3 – "*UK*"; (b) 911 – "*United States*", while 1 – "*USA*" and 1 - "*United States of America*"; (c) 107 – "*Virgin Islands*", 1 – "*British Vigin Islands*" and 22 – "*Virgin Islands, British*" etc.; (2) companies from such non-existing countries as *Czechoslovakia*, *Yugoslavia* and *USSR* although the companies were registered after the collapse of these countries. On the one hand, in some cases such data quality defects may be insignificant, however, depending on the use-case, they may cause inaccurate results and, sometimes, lead to false decisions. But the point of these results was not only the fact of quality problems presence in the analysed data sets but also the nature of the detected problems. In case of problems in fields storing countries names, only the fact of the existence of given quality problems was detected. However, at this stage, it is difficult to point out which of the previously mentioned values are incorrect. It means that there is no confidence that all anomalies and defects in these fields were detected. The proposed approach should be extended to perform data quality analysis for multiple data objects.

## 6 Conclusions

Most of existing data quality researches are focused on data quality dimensions, more precisely, their definition, grouping and application to real data sets. Unfortunately, data quality dimension is very complex concept that requires deep knowledge not only in IT but also in data quality area. Exploration of data quality dimensions in case of every particular data quality research is unreasonably resource-consuming process. Moreover, the degree of understanding of what particular dimension means in scope of the particular research impacts the results of analysis. It means, that these approaches aren't suited for non-IT and DQ-experts.

One of the main ideas of the proposed data object-driven approach is to substitute the data quality dimension concept with more general and comprehensive concept of

"data quality requirement". As a result, users don't need making in-depth research on data quality dimensions, their meaning etc. Users may focus on the data they want to analyse and quality requirements that must be applied to the defined data object in order to evaluate the quality of the data set only. The obtained practical results prove that this substitution was appropriate.

The second point of the proposed approach is the use of graphical DSLs for defining of components. This also should positively impact non-IT and non-DQ user-experience since the diagrams are easy to create, read and modify, and this also support users' interaction. This paper explains why the graphical DSLs is the best option for this purpose.

The third point is that the core of the proposed implementation mechanism is the use of informal PIM and executable PSM models. PIM and PSM models can be used to define every component of the proposed data quality model. This approach allows describing data quality at two levels of abstraction: (1) informally, using natural language and (2) formally, including executable artefacts. This ensures successive specification/ detailing of data quality model.

As a result, the proposed approach is intuitive and therefore suitable for non-IT and non-data quality experts. Although the approach was mainly applied to open data sets, it can also be tailored for analysis of structured and semi-structured data. The analysis of open data sets is just am example to demonstrate how easily "third-party" data can be analysed in accordance with particular users' use-cases without an involvement of data holders. This was achieved as the proposed data quality model describes data quality independently from the IS that accumulates the data.

And the last but not the least significant point is that the results of application of the proposed approach demonstrate that (a) open data have numerous quality issues even in primary data, since data quality issues were detected in 84% of analysed data sets, (b) the proposed approach is suitable for identifying of data quality problems, and as a result for (c) overall data quality of open data could be improved when data is inspected from different perspectives in the context of numerous use cases.

## 7 Future Work

As follows from the results of the appliance of the proposed solution to open data, the research should be continued in order to provide easier and more effective way to analyse data quality in context of multiple data objects. This possibility will provide many different possibilities that currently aren't available or aren't user-friendly enough. Such an extension would ensure reusability of these data for further quality analyses of other data sets. First steps to solve this problem were already taken in [Nikiforova 19b], however, there is a lot of space for further research, including approbation of the proposed solution for complex data object's structures. As a result, detection of possible limitations of the proposed extended approach will be possible.

Moreover, the further research will be focused on proposal of data quality theory, the initial version of it is already outlined in [Bicevskis, 19].

**Acknowledgements**

This work was supported by University of Latvia Faculty of Computing project AAP2016/B032 “Innovative information technologies".

## References

[Acosta, 13] Acosta M., Zaveri A., Simperl E., Kontokostas D., Auer S., Lehmann J.: Crowdsourcing linked data quality assessment, In International Semantic Web Conference (pp. 260-276). Springer, Berlin, Heidelberg, 2013.

[Adedugbe, 18] Adedugbe, O., Benkhelifa, E., & Campion, R.: A Cloud-Driven Framework for a Holistic Approach to Semantic Annotation. In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS) (pp. 128-134). IEEE, 2018.

[Askham, 13] Askham N., Cook D., Doyle M., Fereday H., Gibson M., Landbeck U., Lee R., Maynard C., Palmer G., Schwarzenbach J.: The six primary dimensions for data quality assessment, DAMA UK Working Group, 432-435, 2013.

[Baker, 18] Baker, Q. B., Al-Rashdan, W., & Jararweh, Y.: Cloud-Based Tools for Next-Generation Sequencing Data Analysis. In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS) (pp. 99-105). IEEE, 2018.

[Batini, 16] Batini C., Scannapieco M., “Data and information quality” Cham, Switzerland: Springer International Publishing, Google Scholar, 2016.

[Bevan, 12] Bevan C., Strother D.: Best practices for evaluating method validity, data quality and study reliability of toxicity studies for chemical hazard risk assessments, Washington (DC): American Chemical Council, Centre for Advancing Risk Assessment Science and Policy, 2012.

[Bicevska, 17] Bicevska Z., Bicevskis J., Oditis I.: Models of Data Quality, 12th Conference, ISM 2017, Held as Part of FedCSIS, Prague, Czech Republic, Extended Selected Papers. Lecture Notes in Business Information Processing, Vol. 311, pp. 194-211, 2017.

[Bicevskis, 19] Bicevskis J., Nikiforova A., Bicevska Z., Oditis I.: A Step Towards a Data Quality Theory, In 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), IEEE, 2019 (in print).

[Bicevskis, 18a] Bicevskis J., Bicevska Z., Nikiforova A., Oditis I.: An approach to data quality evaluation, In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 196-201, IEEE, 2018.

[Bicevskis, 18b] Bicevskis J., Bicevska Z., Nikiforova A., Oditis I.: Data quality evaluation: a comparative analysis of company registers’ open data in four European countries, In FedCSIS Communication Papers (pp. 197-204), 2018, http://dx.doi.org/10.15439/2018F92.

[Caro, 07] Caro A., Calero C., Piattini M.: A Portal Data Quality Model for Users and Developers, In ICIQ (pp. 462-476), 2007.

[Chungoora, 13] Chungoora N., Young R. I., Gunendran G., Palmer C., Usman Z., Anjum N. A., Cutting-Decelle A. F., Harding J. A., Case K.: A model-driven ontology approach for manufacturing system interoperability and knowledge sharing, Computers in Industry, 64(4), 392-401, 2013.

[Companies House, 18] Companies House: Free Company Data Product, http://download.companieshouse.gov.uk/en_output.html

[ComputerWorld, 15] ComputerWorld: The "All In" Costs of Poor Data Quality. It goes beyond dollars and cents, 2015, https://www.computerworld.com/article/2949323/the-all-in-costs-of-poor-data-quality.html

[Coutinho, 12] Coutinho C., Cretan A., Jardim-Goncalves R.: Negotiations framework for monitoring the sustainability of interoperability solutions, In International IFIP Working Conference on Enterprise Interoperability (pp. 172-184). Springer, Berlin, Heidelberg, 2012.

[Ferney, 17] Ferney M., Estefan L., Alexander V.: Assessing data quality in open data: A case study, Congreso Internacional de Innovacion y Tendencias en Ingenieria (CONIITI) (pp. 1-5). IEEE, 2017

[Global Open Data Index, 18] Global Open Data Index, 2018, https://index.okfn.org/

[Khider, 18] Khider, H., Hammoudi, S., Benna, A., & Meziane, A. Social Business Process Model Recommender: An MDE Approach. In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS) (pp. 106-113). IEEE, 2018.

[Kleppe, 03] Kleppe A. G., Warmer J., Warmer J. B., Bast W.: MDA explained: the model driven architecture: practice and promise, Addison-Wesley Professional, 2003.

[Mellor, 04] Mellor S. J., Scott K., Uhl A., Weise D.: MDA distilled: principles of model-driven architecture, Addison-Wesley Professional, 2004.

[Miller, 03] Miller J., Mukerji J.: MDA Guide Version 1.0. 1, Object Management Group (OMG), Needham, MA, 2494, 2003.

[Neumaier, 16] Neumaier S., Umbrich J., Polleres A.: Automated quality assessment of metadata across open data portals, Journal of Data and Information Quality (JDIQ), 8(1), 2016.

[Nikiforova, 19a] Nikiforova, A.: Analysis of open health data quality using data object-driven approach to data quality evaluation: insights from a Latvian context. In IADIS International Conference e-Health 2019, MCCSIS 2019, (pp. 119-126). IADIS, 2019.

[Nikiforova, 19b] Nikiforova A., Bicevskis, J.: An Extended Data Object-driven Approach to Data Quality Evaluation: Contextual Data Quality Analysis, Proceedings of the 21st International Conference on Enterprise Information Systems. In ICEIS 2019.

[Nikiforova, 18a] Nikiforova A.: Open Data Quality, In Baltic DB&IS 2018 Joint Proceedings of the Conference Forum and Doctoral Consortium, Trakai, Lithuania (Vol. 2158), 2018.

[Nikiforova, 18b] Nikiforova A.: Open Data Quality Evaluation: A Comparative Analysis of Open Data in Latvia, Baltic Journal of Modern Computing, 6(4), 363-386, 2018.

[Pauker, 16] Pauker F., Frühwirth T., Kittl B., Kastner W.: A systematic approach to OPC UA information model design, Procedia CIRP, 57, 321-326, 2016.

[Paulheim, 14] Paulheim H., Bizer C.: Improving the quality of linked data using statistical distributions, International Journal on Semantic Web and Information Systems (IJSWIS), 10(2), 63-86, 2014.

[Redman, 01] Redman T. C.: Data quality: the field guide, Digital press, 2001

[Redman, 97] Redman T. C., Blanton A.: Data quality for the information age, Artech House, Inc., 1997.

[Ruiz, 18] Ruiz M.: TraceME: A Traceability-Based Method for Conceptual Model Evolution, Springer International Publishing, 2018.

[Salesi, 18] Salesi, S., Alani, A. A., Cosma, G.: A Hybrid Model for Classification of Biomedical Data Using Feature Filtering and a Convolutional Neural Network. In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS) (pp. 226-232). IEEE, 2018.

[Sasse, 17] Sasse T., Smith A., Broad E., Kennison J., Wells P., Atz U.: Recomendations for Open Data Portals: from Setup to sustainability, https://www.europeandataportal.eu/sites/default/files/edp_s3wp4_sustainability_recommendations.pdf, 2017

[Sáez Martín, 16] Sáez Martín A., Rosario A. H. D., Pérez M. D. C. C.: An international analysis of the quality of open government data portals, Social Science Computer Review, 34(3), 298-311, 2016.

[Scannapieco, 02] Scannapieco, M., Catarci, T.: Data quality under a computer science perspective, Archivi & Computer, 2, 1-15, 2002.

[Schmidt, 15] Schmidt M., Schmidt S. A. J., Sandegaard J. L., Ehrenstein V., Pedersen L., Sørensen H. T.: The Danish National Patient Registry: a review of content, data quality, and research potential, Clinical epidemiology, 7, 449, 2015.

[Shi, 15] Shi X., Han W., Huang Y., Li Y.: Service-oriented business solution development driven by process model, In the Fifth International Conference on Computer and Information Technology (CIT'05) (pp. 1086-1092). IEEE, 2015.

[Soley, 00] Soley R.: Model driven architecture, OMG white paper, 308(308), 5, 2000.

[Sprogis, 13] Sprogis A., Barzdins J.: Specification, Configuration and Implementation of DSL Tool, In Databases and Information Systems VII: Selected Papers from the Tenth International Baltic Conference, DB & IS 2012 (Vol. 249, p. 330). IOS Press, 2013.

[TDQM, 18] TDQM. The MIT Total Data Quality Management program. Available: http://web.mit.edu/tdqm/

[Tomic, 15] Tomic K., Sandin F., Wigertz A., Robinson D., Lambe M., Stattin P.: Evaluation of data quality in the National Prostate Cancer Register of Sweden, European journal of cancer, 51(1), 101-111, 2015.

[The Economist, 17] Economist, T.: The world's most valuable resource is no longer oil, but data. The Economist: New York, NY, USA, 2017.

[Umbrich, 15] Umbrich J., Neumaier S., Polleres A.: Quality assessment and evolution of open data portals, In 2015 3rd International Conference on Future Internet of Things and Cloud (pp. 404-411). IEEE, 2015.

[Van den Berghe, 17] Van den Berghe S., Van Gaeveren K.: Data quality assessment and improvement: a Vrije Universiteit Brussel case study, Procedia Computer Science, 32-38, 2017.

[Vetrò, 16] Vetrò A., Canova L., Torchiano M., Minotas C. O., Iemma R., Morando F.: Open data quality measurement framework: Definition and application to Open Government Data, Government Information Quarterly, 33(2), 325-337, 2016.

[Wand, 96] Wand Y., Wang R. Y.: Anchoring data quality dimensions in ontological foundations, Communications of the ACM, 39(11), 86-96, 1996.

[Wang, 96] Wang R. Y., Strong D. M.: Beyond accuracy: What data quality means to data consumers, Journal of management information systems, 12(4), 5-33, 1996.