J. Mantas et al. (Eds.)

doi:10.3233/SHTI210866

A Framework for User-Configurable Data Quality Assurance of Electronic Patient Records

Marina ROMANCHIKOVA[a,1] and Jean-Laurent HIPPOLYTE[a]
[a] *National Physical Laboratory, UK*

Abstract. Electronic Patient Records (EPRs) are valuable data resources for clinical and operational research. The heterogeneity of medical software coupled with the changing data formats and long lifespan of the patient datasets stored in EPRs results in data inconsistencies that hinder operational activities and increase personnel efforts for data lookup and cleaning. This study presents an approach for automated data quality reporting that was developed and tested within a real-world hospital setting at Royal Surrey County Hospital NHS Foundation Trust in 2020. 81 data quality tests configurable via spreadsheets were defined and executed to yield standardised human-readable reports in comma-separated value format. The data evaluation and reporting routines provided manyfold improvement over existing data quality reporting mechanisms.

Keywords. Data quality, electronic patient records, automation

1. Introduction

Positioned at the heart of clinical information exchange, Electronic Patient Records (EPRs) are a valuable resource of data for clinical and operational research. However, heterogeneous nature of clinical tools that feed data to EPR leads to variable data quality, making data reuse or exchange a challenging exercise. The demand for efficient data quality assurance mechanisms has been exacerbated by the additional pressure on EPR data management posed by the onset of Covid-19 pandemic that required rapid access and exchange of patient records. This study presents an approach for automated data quality evaluation that was developed and tested within a real-world hospital setting at Royal Surrey County Hospital (RSCH) NHS Foundation Trust in 2020 to aid the migration of patient data into a new EPR system.

Migration-ready EPRs must fulfil the data quality criteria defined by the hospital Data Quality department and the target EPR provider. All patient records that do not meet these criteria need to be corrected pre-migration. Numerous data quality checks (DQCs) must be run to detect the non-compliances. These checks must be executed at least once per day to capture the changes resulting from normal hospital activity and from the ongoing EPR correction efforts. The existing DQC process demanded involvement of several NHS Trusts departments and needed to be adjusted to handle the high number of EPR database reports. This work aimed to create a user-friendly low-

[1] Corresponding Author, Marina Romanchikova, National Physical Laboratory, Hampton Road, Teddington, TW11 0LW, UK; E-mail: marina.romanchikova@npl.co.uk

technology solution for evaluation of EPR compliance with respect to each of the defined DQCs. Techniques, tools, and data flows used to create the solution are described in the following sections.

2. Materials and Methods

2.1. User requirements

The Data Quality team users needed to be able to run individual DQCs (i.e., find all patient records with missing home address) or groups of DQCs (i.e., locate all patient records that do not comply with any of the “red” category DQCs). The application output had to produce human-readable listings of patient records in need of correction, as well as snapshot statistics of compliant and non-compliant patient records.

2.2. Data Quality Checks

A total of 148 DQCs were defined by the RSCH Trust’s Data Quality department and refined within this collaboration. The DQCs specified individual EPR data items (e.g., NHS numbers) or sets of items (e.g., whether any data elements in the patient record contain certain characters) to be tested against a set of rules (i.e., an NHS number should be 10 digits long). Every DQC was assigned a Red/Amber/Green risk category described in Table 1.

Table 1. Types of data quality checks and their implications for data readiness for migration to new EPR.

DQC category	# DQC items	Example	Impact of non-compliance
Red	9	Hospital number is used in more than one patient record	Cannot be migrated as is or will be result in incorrect data.
Amber	126	Missing current home address	It is unclear whether target EPR can accept the record.
Green	13	Missing gender specification	Potentially can be migrated as is, but is likely to cause issues in target EPR.

2.3. Software framework

The application used the data quality rules definitions from the Data Quality Vocabulary [1] that describes concepts relevant to data quality such as data, reference data, quality rules and quality metrics. The implemented quality rules included legal/illegal/unique value, property completeness, functional dependency reference and custom rules.
The application presented in Figure 1 was implemented in Python 3.7 using *pypika*[2] library for SQL query generation and *pyodbc*[3] library to connect to the EPR Microsoft SQL Server database.

[2] https://github.com/kayak/pypika
[3] https://github.com/mkleehammer/pyodbc

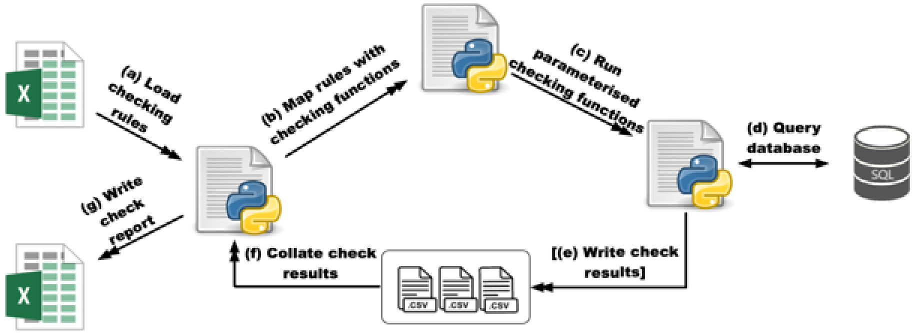
(a) Load checking rules
(b) Map rules with checking functions
(c) Run parameterised checking functions
(d) Query database
SQL
[(e) Write check results]
.csv
.csv
.csv
(f) Collate check results
(g) Write check report

Figure 1. Data quality checking application. (a) The DQCs to be executed are loaded from the spreadsheet provided by the user. (b) Data quality rules associated with each DQC are combined in checking functions. (c) Parametrised SQL queries are generated according to rule types and database mappings. (d) The generated queries are run on the EPR database. (e, f) The query results are collated into human-readable reports and returned to the user.

From the total of 148 DQCs, 81 were mapped to the EPR (i.e., it was known which tables and columns contain the data) and implemented in the software. The input spreadsheet structure for several data quality rule types is illustrated in the Table 2.

Table 2. Configurable spreadsheet with data quality rules definitions. Table names and DQC IDs have been altered. Queries are generated by the software based on the table & column names, rule type and rule constraint.

DQC ID	Table.Column	Rule Type	Rule Constraint	Generated SQL Query
1	Patient.NHS_ID	Valid Value	<pattern>	SELECT Hospital_ID FROM Patient WHERE NHS_ID NOT LIKE <pattern>
2	Patient.DateOfDeath Patient.Status	Conditional Property Completeness	<condition>	SELECT Hospital_ID from Patient WHERE <condition>
3	Patient.Gender	Missing Value	IS NULL	SELECT Hospital_ID from Patient WHERE Gender IS NULL

2.4. Application outputs

The software produced 81 individual DQC reports listing patient identifiers and the elements of the EPR that failed the DQC. The reports enabled location and subsequent correction of the identified records in the EPR. Additionally, the software generated a summary report with overview of failed record statistics against each completed DQC (Table 3).

Table 3. Fragment of the overview report produced by the software. The "#failed" column shows the total number of non-compliant patient records for a given DQC.

DQC ID	RAG	DB tables	Description	#failed
1	Red	Patient.NHS_ID	Invalid NHS number	270
2	Amber	Patient.DateOfDeath; Patient.Status	Missing date of death in deceased patients	334
3	Green	Patient.Gender	Missing gender value	6173

2.5. Performance

Development and tests were performed on a remotely accessed Windows 7 office workstation with 16 GB RAM and Intel Core i7 CPU. From the 81 implemented DQCs, 79 DQCs were completed in less than 7 minutes. The runtime for two less frequently used DQCs that scanned for invalid characters in all columns of all database tables took approximately 17 minutes. The runtime for 81 implemented DQCs was 24 minutes.

3. Discussion

The presented framework was developed to provide the hospital Data Quality personnel with a configurable tool to obtain up-to-date information on completeness and correctness of EPRs. The framework made use of a) user-configurable spreadsheets to instruct the software which reports are required; b) user- and machine-readable data quality checking rules and c) auto-generation of SQL queries from data quality rules and EPR database mappings. While the solution proved useful and fast to implement for simple DQCs such as verifying that a data attribute is present or follows a pattern, the method showed limitations for a small number of complex queries that involved multi-stage evaluation of several tables, for example, counting of NHS numbers by trace status[4]. Such queries were pre-defined in the input spreadsheet and would need additional development time to be fitted into the rule-based query generation mechanism.

Acknowledgements

This study was funded by the department of Business, Energy, and Industrial Strategy through the Covid-19 response programme[5]. The authors would like to thank the Royal Surrey NHS Hospital Foundation Trust's Data Quality and Data Warehousing teams for setting the clinical objectives and for providing the access to EPR database.

References

[1] Fürber C, Hepp M. Towards a vocabulary for data quality management in semantic web architectures. InProceedings of the 1st International Workshop on Linked Web Data Management 2011 Mar 25 (pp. 1-8).

[4] https://www.datadictionary.nhs.uk/attributes/nhs_number_status_indicator_code.html

[5] https://www.npl.co.uk/covid-response