MEDINFO 2015: eHealth-enabled Health
I.N. Sarkar et al. (Eds.)

doi:10.3233/978-1-61499-564-7-852

An Architecture for Continuous Data Quality Monitoring in Medical Centers

Gregor Endler[a], Peter K. Schwab[a], Andreas M. Wahl[a], Johannes Tenschert[a], Richard Lenz[a]

[a] *Computer Science 6 (Data Management), Friedrich-Alexander-Universität, Erlangen-Nürnberg, Germany*

Abstract

In the medical domain, data quality is very important. Since requirements and data change frequently, continuous and sustainable monitoring and improvement of data quality is necessary. Working together with managers of medical centers, we developed an architecture for a data quality monitoring system. The architecture enables domain experts to adapt the system during runtime to match their specifications using a built-in rule system. It also allows arbitrarily complex analyses to be integrated into the monitoring cycle. We evaluate our architecture by matching its components to the well-known data quality methodology TDQM.

Keywords

Medical Informatics; Data Quality; Quality Control; Quality Improvement; Data Collection; Databases, Factual; Group Practice.

Introduction

Data quality is an important concern in data integration scenarios. Integration and data quality influence each other bilaterally. Many data quality issues only become evident if several data sources are available, and integration can benefit from high data quality [1].

In the medical domain, one such integration scenario is created by medical practitioners affiliating into group practices or medical centers [2]. They do so for various reasons, among them financial benefits or better collaborative treatment of patients. To reap these benefits, organization wide planning is required. This in turn makes it necessary to create a general view over the data and processes of every participant's practice. This information is found in the respective local patient-data management systems and databases and may exhibit various deficiencies, like missing entries or wrong values. An integrated central database is fed new and possibly erroneous data from the center's locations continuously.

It is thus not enough to do a single pass of data cleaning over local databases. Indeed, it may even be impossible to validate data at the local databases since practitioners may be unwilling to relinquish control of their system or may be unaware of newly established quality considerations. Thus, central continuous monitoring and improvement of data quality is necessary to support the center. Frequent changes also induce significant software aging [3] in any system dealing with data quality. Data quality systems must be able to evolve with newly arising requirements, especially in the health sector. Medical or legal changes must be implementable by domain experts, since these changes may happen too short-term and too often to allow assigning a programmer for substantial code redesign [4].

Contribution

In this paper, we describe an architecture for continuous data quality monitoring. We start out by giving an overview over the project context, and describe our approach and the resulting architecture. To evaluate the architecture's usefulness, we show how it supports data quality methodology. In conclusion, we briefly discuss the ongoing implementation of a prototype.

MEDITALK

The research project MEDITALK is an example of an integration scenario in the medical domain [5]. A so called practice manager is responsible for controlling a medical center. This role necessitates an overview over the data of all the centers' locations. All local practitioners' data is integrated into a central database in a standard Extract-Transfrom-Load (ETL) process [6]. This solution is already in place at three medical centers, together encompassing 50,000 patients, 80 practitioners, and 3 practice managers. Since new data arises at the local practices continuously, ETL is repeated at set intervals. Copies of all local data arrive at daily intervals in the central database. The practice manager interacts with the system through a controlling application, which essentially provides a predefined dashboard of information about the centers' locations and their data.

Data Quality

Data quality is often generically defined as "fitness for use" [7]. This means that for data to be of high utility, it has to conform to requirements according to its application. We divide data quality considerations into two broad groups. Checking their fulfillment may be conceptionally easy, like verifying the length of a string. It may also be more involved, for instance requiring complex data mining or statistical models. Obviously, the boundary between these two difficulty groups is fuzzy - deciding which group a requirement belongs to is a domain expert's responsibility. Within both groups, requirements are further (sometimes implicitly) classified into the usual data quality dimensions [1,7]. We make the assumption that both groups are important to continuous data quality monitoring in medical centers.

Methods

In a first step, we conducted a literature review focusing on data quality both in general as well as specifically in healthcare. To familiarize ourselves with the domain, we afterwards conducted interviews with the three practice managers involved in MEDITALK. We asked questions about the data quality dimensions most mentioned in literature, their impact on the centers' work and about how detected quality problems had been addressed. We worked with the developers of the integration environment of MEDITALK, and designed an architecture to monitor data quality on top of the existing software.

Results

Literature Review

Many methodologies for data quality improvement have been proposed (see [8] for a survey). A commonality between these is that they offer a bird's eye view of necessary steps for data quality improvement and thus are highly generic. The arguably widest used one is Wang's Total Data Quality Management (TDQM) [9] (see Figure 1). TDQM is suited for continuous monitoring and is open to system evolution - both capabilities important for data quality monitoring in the medical domain. Therefore, we make the assumption that being able to support the TDQM steps is a minimum requirement for any data quality solution to be used by practice managers.

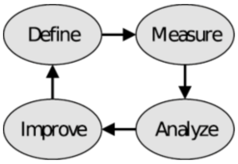


Figure 1 - TDQM cycle

There are many publications specifically considering data quality in healthcare applications, for example health simulations [10], free text analysis of diagnoses [11], or healthcare processes [12]. Concerning data completeness, Miller et al. report about two studies that found "pieces of information perceived as being needed for clinical decisions were missing 13.6% and 81% of the time" [13]. The ACM Journal of Data and Information Quality devoted a special issue to the topic of information quality in healthcare [14]. In 2013, Dixon et al. described the Health Data Stewardship framework and called for software tools to monitor and improve data quality [15]. It follows that supporting high data quality in the medical sector is an ongoing effort.

Interviews

All practice managers reported occurrences of low quality data, with incomplete data being identified as the biggest issue. To be more specific, *population completeness* [1] was deemed the most impactful [16]; a center's practitioners operate on a budget. Once they surpass a certain threshold, their activities are only reimbursed fractionally. This becomes evident at the latest at the end of each fiscal quarter, when practitioners report their activities to a central authority for reimbursement. At this point, however, it is too late for the practice manager to counter this effect. While countersteering would have been possible during the quarter, incomplete data at that time obfuscated arising budgeting problems, leading to loss of revenue.

All practice managers reported that at their centers there already were rules in place about data. Most of these were not checked automatically and had to be evaluated manually. One location reported presence of 119 standardized queries to review some quality constraints, again with the restriction that these queries had to be triggered by hand [17]. This produces significant effort since these queries have to be evaluated frequently. There are some exceptions to this, for example automatic sanity checks performed by patient-data management systems. Still, all practice managers expressed interest in being able to define their own rules for automatic evaluation.

Monitoring Architecture

While our architecture (see Figure 2) is built on top of the MEDITALK packages, it is designed to allow implementation as a standalone application. All user interaction happens through the Interaction package. The Data Service package is responsible for accessing the central database and for storing monitoring results and metadata. Actual data quality measurements are performed by the Monitoring package.

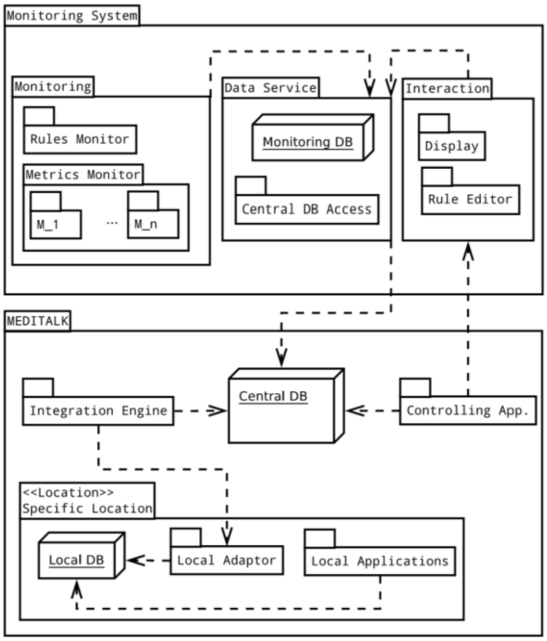


Figure 2 - MEDITALK Integration Environment and Monitoring System

Monitoring Package

Our proposed architecture splits assessment of data quality into two separate concepts: *rules* and *metrics*.

A *rule* is a formulation of a constraint on data, for example R1: ICDCode = 'J00' and Diagnosis != 'Acute nasopharyngitis'[1]. Any tuple in the database for which R1 holds (meaning that the ICD code has an attached diagnosis different from the standard) may be erroneous according to the formulated rule (see [18] for other examples). Another possible form of rules are association rules [19], and conditional functional and inclusion dependencies (see [20] for a survey). Expressing data quality constraints by rules has the advantage that they can be created without intervention by a programmer. Domain experts can formulate rules according to their own requirements.

The **Rules Monitor** package keeps track of the rule base stored in the Monitoring Database, executes rules according to their schedule and in turn stores the results. Rules fall into one of two categories: aforementioned examples are *data quality rules*, and define the fitness of data entries. The second category are *improvement rules*. These are used to trigger events to improve data quality, for example alerting the practice manager to violating data or performing automatic edits[2]. All rules are checked on the available data according to a set schedule. For the MEDITALK project, it is enough to check them once every 24 hours because data arrives in daily intervals. For other contexts, the intervals can be changed according to need. Every rule can also be triggered manually.

[1] This is not necessarily a practical example – we only use R1 to exemplify our system's usage.

[2] Automatic edits bear some challenges like conflicting edits [21]. We assume the practice managers to be aware of these perils.

The **Metrics Monitor** contains all functions that implement metrics, and triggers recalculation according to the needs of the respective metrics. A *metric*, for our purpose, is defined as any measure indicating the quality of a data item or set of data items. In distinction to the mathematical definition of a metric, we denote not only a distance between a data item and its theoretical perfect quality counterpart by the expression "metric", but any indicator which makes a statement about data or data quality. As an example, the number of NULL values within a database table may indicate the table's completeness (depending on NULL semantics [22]). We use the term metric interchangeably as the indicator delivered or the mechanism to calculate the indicator.

We distinguish between metrics and rules for the following reasons: some desirable functions, for instance statistical models (see [23] for an example), may be too complex for easy representation by rules. Also, some rules may prove to be of interest to more than one center. In the case of large sets of rules, creating these rules at all interested centers would be redundant. Expending onetime effort encoding or generating the set in a metric, then integrating the resulting module at each center would lower overall effort. So if for these reasons additional programming is warranted, an additional metric is created. This will usually be done by a programmer in response to a new requirement by a domain expert.

Data Service Package

All results of calculations within the Monitoring package are stored in the **Monitoring Database** to encapsulate all information necessary to assess gathered indicators. A monitoring result is a statement about data, valid at a certain point in time. Each result is identified by;

- the originating rule's / metric's identifier,
- the date and time the result was obtained, and
- all involved tuples and attributes.

Apart from this identifying data, each issue has a payload according to the specific rule / metric, for example an indicator calculated by a metric. The payload may be empty if all necessary information is already present in the result's key, but may also be arbitrarily complex, for example delivering an XML data structure. Usually, a rule's payload will be the percentage of applicable tuples it holds on, and the identifiers of violating tuples. The payloads of metrics may be as simple as a single indicator (e.g. "expected number of patients tomorrow: 32"), possibly including a confidence interval for the value. They may also deliver structures of any number of values and of any depth (e.g. a decision tree derived from patient data).

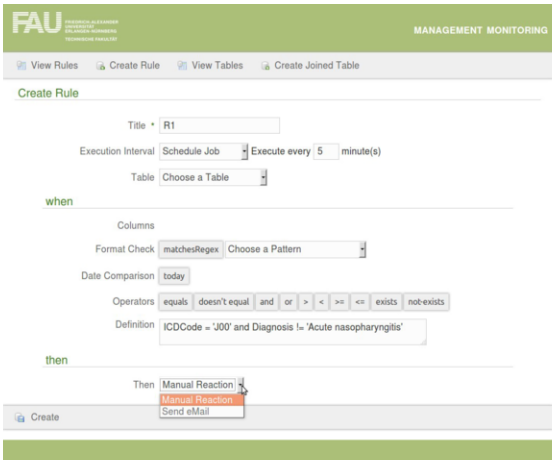
The **Central Database Access** package serves as a mapper between the Central Database and the data requirements of the Monitoring System. This decouples the Central Database and all other packages that need access to it, preventing changes at the Central Database from propagating to other parts of the system.

Interaction Package

Through the **Rule Editor**, practice managers can define their own data quality rules. This solution has several advantages in comparison to the standardized queries used to date:

- Rules can still be triggered manually, but can also be checked automatically and periodically (at set execution intervals).
- Rules can not only be defined on data, but also on metadata, meaning that rules can be used to alert the practice manager when a metric (or even another rule) changes.
- Drag-and-drop generation of rule conditions may lower initial training effort compared to explicitly spelling out the queries.

The prototypical rule editor (see Figure 3) developed in our group [24] can be used to create Boolean rules, for example to check whether a value matches a regular expression. Users can write their own regular expressions, or choose from standard checks like string length or number format. The editor is able to recreate the standardized queries, and can be extended to allow (conditional) functional dependencies and other kinds of rules not yet implemented.



Title * R1

Execution Interval Schedule Job Execute every 5 minute(s)

Table Choose a Table

when

Columns

Format Check matchesRegex Choose a Pattern

Date Comparison today

Operators equals doesn't equal and or > < >= <= exists not-exists

Definition ICDCode = 'J00' and Diagnosis != 'Acute nasopharyngitis'

then

Then Manual Reaction

Figure 3 - Rule creation

The **Display** package is responsible for visualizing the state of the Monitoring System. This includes all existing rules and metrics as well as their results on the currently available data. It also enables exploratory analysis of the central database to get an overview over schema and extension. All subpackages of Interaction have their own user interfaces, which can be used on their own or be implemented as plugins for the controlling application in the MEDITALK package.

Discussion

To evaluate whether our architecture is capable of continuous data quality monitoring, we show that each step of TDQM is supported by one or more packages of our architecture, as stated in Figure 4.

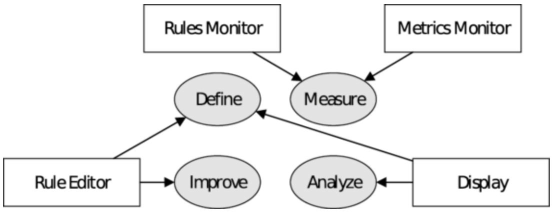


Figure 4 - Supporting TDQM

Define

"The definition component of TDQM cycle identifies important data quality dimensions (...) and the corresponding data quality requirements." [7, p.5, "The TDQM Cycle"]

Defining fitness of data is closely tied to an organization's business goals. As such, some of the effort expended in this step is outside of the scope of the Monitoring System. The

Display package assists in this effort by delivering an overview over available data and the schema of the central database. The formal definition of data quality requirements however is supported by the **Rule Editor** in full.

Example A practice manager (PM in the following) discovers that at some locations, ICD code and diagnosis text are both entered manually. The PM decides that mismatches between ICD code and diagnosis should be avoided, and formulates rules to check for these mismatches. One of these rules is R1: ICDCode = 'J00' and Diagnosis != 'Acute nasopharyngitis'. A catalog of like constraints to check the quality requirement "ICD code and attached diagnosis may not mismatch" is defined.

Measure

"The measurement component produces data quality metrics." [7, p.5, "The TDQM Cycle"]

The measure step is performed by the **Rules** and **Metrics Monitoring** packages. Both calculate indicators for data quality and store them in the Monitoring Database.

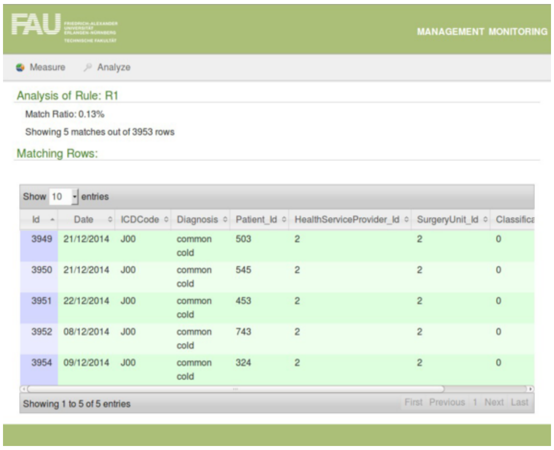
Example A new practice joins the center. Once its data becomes available in the central database, Rules Monitoring checks whether all applicable rules hold. Assuming the practice uses the diagnosis "common cold" instead of "Acute nasopharyngitis" in 5 cases with ICDCode = 'J00', all of these fulfill R1. A result is placed in the Monitoring Database stating that 5 tuples (the ones coming from the new practice) fulfill R1 and therefore are of low quality.

Analyze

"The analysis component identifies root causes for data quality problems and calculates the impacts of poor quality information." [7, p.5, "The TDQM Cycle"]

The Analyze step is supported by the interaction of the **Monitoring Database** and the **Display** package. The Display package shows all Monitoring results. It allows drilling down into their information, showing affected tuples and attributes, their provenance, and involved rules and metrics.

Example The result R1 match ratio: 0.13% is shown in the Display package (see Figure 5). This means that out of all diagnosis entries on the central database, 0.13% violate the data quality requirement defined by R1. The PM checks the definition text of the rule as well as which tuples are involved and their provenance. In this way, the new practice is identified as the origin of the problem.


FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG TECHNISCHE FAKULTÄT
MANAGEMENT MONITORING
Measure
Analyze

Analysis of Rule: R1

Match Ratio: 0.13%

Showing 5 matches out of 3953 rows

Matching Rows:

Show 10 entries

Id	Date	ICDCode	Diagnosis	Patient_Id	HealthServiceProvider_Id	SurgeryUnit_Id	Classifica
3949	21/12/2014	J00	common cold	503	2	2	0
3950	21/12/2014	J00	common cold	545	2	2	0
3951	22/12/2014	J00	common cold	453	2	2	0
3952	08/12/2014	J00	common cold	743	2	2	0
3954	09/12/2014	J00	common cold	324	2	2	0

Showing 1 to 5 of 5 entries

Figure 5 - Analyzing a data quality rule

Improve

"(...) the improvement component provides techniques for improving data quality." [7, p.5, "The TDQM Cycle"]

Again, some of the actions performed during this step are outside of the scope of the Monitoring System (e.g. organizational changes or in-service training of staff). However, the **Rule Editor** can be used to establish improvement rules. An example of such a rule is sending an email to a location that has not entered data by a certain deadline. Data cleaning rules such as automatic edits can also be formulated.

Example Judging the low data quality to be a one-time problem, the PM instructs the new practice to revise the offending tuples. Should the problem occur again, an improvement rule can be created that on detection of a violating tuple either notifies the person responsible or automatically changes the values to conform.

Conclusion

We described an architecture supporting continuous data quality monitoring for medical centers. Lenz identified several important factors in making software sustainable [25]. Our architecture satisfies two of these by design. *Separation of Concerns* is granted by grouping functionality into fitting packages, separating rules, metrics, data access, and user interaction into their own packages. This minimizes the probability that change in one package requires redesign of others. *Deferred Design* is satisfied by the presence of the rule editor and rule monitoring: The set of rules is not hardcoded into the system, but can be built, corrected, and extended at runtime by domain experts as demand requires. This ensures evolvability of the rule system with changing requirements. Two additional factors, *Loose Coupling* and *Service Oriented Architecture*, are implementation specific.

While our application example is an integration scenario, this is not imperative. Since the architecture only requires access to a single central database and not the sources, it can just as easily be applied to databases that are not involved with integration, for example hospital information systems. This design does not diminish the potential to make use of information that arises in the ETL process: Any data and metadata that is stored in the connected database is fair game for monitoring.

By enabling continuous monitoring, system evolution and deferred design by domain experts, we also support frameworks like Health Data Stewardship (HDS) [15]. The connection to health outcomes that HDS demands is implemented by the system's ability to monitor any kind of data. Since outcomes may also be stored in the central database, any information about those can be used by rules and metrics as well as in the display component.

Several parts of the architecture are already implemented. We are currently developing methods for measuring population completeness. Access to the central database is handled directly through the accessing components and not through a common mapper as of now. Monitoring DB and Display are not yet complete, but are already capable of storing respectively presenting the results of rules. Rule Editor and Rule Monitoring are fully functional. Once complete, we will evaluate the effectiveness of our solution through user studies.

Acknowledgements

Parts of this work are supported by the German Federal Ministry of Education and Research (BMBF), grant 13EX1013D.

References

[1] Batini C, Scannapieco M. Data Quality: Concepts, Methodologies and Techniques. Springer; 2006.

[2] Endler G. Data quality and integration in collaborative environments. In: Proceedings of the SIGMOD/PODS 2012 PhD Symposium. New York, NY, USA. p. 21–26.

[3] Parnas DL. Software aging. In: Proceedings of the ICSE '94. Los Alamitos, CA, USA: IEEE Computer Society Press; 1994. p. 279–287.

[4] Lenz R, Kuhn KA. Towards a continuous evolution and adaptation of information systems in healthcare. International Journal of Medical Informatics. 2004;73(1):75 – 89.

[5] Endler G, Langer M, Purucker J, Lenz R. An Evolutionary Approach to IT Support for Medical Supply Centers. In: Proceedings der 41. Jahrestagung der Gesellschaft für Informatik e.V. (GI); 2011.

[6] Chaudhuri S, Dayal U. An overview of data warehousing and OLAP technology. SIGMOD Rec. 1997 March;26:65–74.

[7] Wang RY, Ziad M, Lee YW. Data Quality. Springer US; 2002

[8] Batini C, Cappiello C, Francalanci C, Maurino A. Methodologies for data quality assessment and improvement. ACM Computing Surveys. 2009 July;41:16:1–16:52.

[9] Wang RY. A product perspective on total data quality management. Communications of the ACM. 1998 February;41:58– 65.

[10] Baumgärtel P, Lenz R. Towards Data and Data Quality Management for Large Scale Healthcare Simulations. In: Proceedings of the International Conference on Health Informatics. 2012. p. 275–280.

[11] Lauria EJM, March AD. Combining bayesian text classification and shrinkage to automate healthcare coding: a data quality analysis. J Data Inf Qual. 2011;2(3):13:1–13:22.

[12] Lenz R, Reichert M. IT support for healthcare processes - premises, challenges, perspectives. Data Knowl Eng. 2007;61(1):39-58.

[13] Miller DW, Yeast JD, Evans RL. Missing Prenatal Records at a Birth Center: A Communication Problem Quantified. In: AMIA Annu Symp Proc. American Medical Informatics Association; 2005.

[14] Madnick SE, Lee YW, editors. Special Issue on Information Quality: The Challenges and Opportunities in Healthcare Systems and Services. vol. 4 Issue 1 of ACM JDIQ. New York, NY, USA: ACM; 2012.

[15] Dixon BE, Rosenman M, Xia Y, Grannis SJ. A vision for the systematic monitoring and improvement of the quality of electronic health data. Proceedings of the 14th World Congress on Medical and Health Informatics. 2013;192:884–888.

[16] Endler G, Baumgärtel P, Lenz R. Pay-as-you-go data quality improvement for medical centers. In: Proceedings of the eHealth 2013; 2013.

[17] Gorupec M. Entwicklung eines Frontends für Regelsysteme zur Datenqualitätsverbesserung medizinischer Informationssysteme. Master Thesis, FAU; 2014.

[18] Rakovac I, Maharjan B, Stein C, Loyola E. Program for validation of aggregated hospital discharge data. Proceedings of the 14th World Congress on Medical and Health Informatics. 2013;192:1155.

[19] Agrawal R, Imielinski T, Swami A. Mining Association Rules Between Sets of Items in Large Databases. In: Proceedings of the 1993 ACM SIGMOD '93. New York, NY, USA: ACM; 1993. p. 207– 216.

[20] Fan W, Geerts F, Jia X, Kementsietsidis A. Conditional functional dependencies for capturing data inconsistencies. ACM Trans Database Syst. 2008 Jun;33(2):6:1–6:48.

[21] Fellegi IP, Holt D. A systematic approach to automatic edit and imputation. J Am Stat Assoc. 1976;71(353):17–35.

[22] Zaniolo C. Database relations with null values. In: Proceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of database systems. PODS '82. New York, NY, USA: ACM; 1982. p. 27–33.

[23] Huang Y, Hanauer DA. Patient No-Show Predictive Model Development using Multiple Data Sources for an Effective Overbooking Approach. Appl Clin Inform. 2014;5(3):836–860.

[24] Gorupec M, Endler G. ruleDQ: Ein Regelsystem zur Datenqualitätsverbesserung medizinischer Informationssysteme. In: GI Lecture Notes in Informatics (LNI) Seminars 13 / Informatiktage 2014; 2014. p. 37–40.

[25] Lenz R. Information Systems in Healthcare - State and Steps towards Sustainability. IMIA Yearbook 2009. 2009;1:63– 70.

Address for Correspondence

Gregor Endler
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Computer Science 6 (Data Management)
Martensstraße 3, 91058 Erlangen, Germany
gregor.endler@fau.de