Towards Measuring Test Data Quality

Johannes Held
Advisor: Prof. Dr. Richard Lenz
University of Erlangen and Nuremberg
Chair for Computer Science 6 (Data Management)
Martensstraße 3
91058 Erlangen
{johannes.held,richard.lenz}@cs.fau.de

ABSTRACT

In order to enable proper system and integration testing, it is often necessary to have huge test data inventories, reflecting the heterogeneous live system. Although the maintenance of large data stores can be guided by advice obtained from data quality evaluations, this technique can be only partly applied to test data inventories. Assessing test data quality is difficult, as the well-known data quality dimensions are not applicable in an easy fashion. For example, an otherwise good value of 100% for *correctness* would not allow to store erroneous test data items. The need for data quality dimensions dedicated to assessing test data quality can't be satisfied by well-known data quality dimensions. In this paper, we present our thesis approach to identify and validate new quality dimensions applicable for test data quality and develop quantification methods. We propose **proximity to reality** and **degree of coverage** as two new test data quality dimension and sketch quantification approach to measures, specifically suited for test data.

Categories and Subject Descriptors

D.2 [**Software Engineering**]: Metrics; H.1 [**Models and Principles**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous

General Terms

Management, Measurement

Keywords

Test Data Quality, Data Quality Dimensions, Testing

1. INTRODUCTION

For some domains, proper system and integration testing requires huge test data inventories, which mirrors the heterogeneous live system. This is especially evident in the field of medical engineering, as there are many competing companies and different compositions of medical devices deployed in clinics. Moreover, a lot of standards for the exchange of data exist in the medical domain, like *Digital Imaging and Communication in Medicine* (DICOM), or *Health Level 7* (HL7) [1]. Organizations like *Integrating the Healthcare Enterprise* (IHE) offer standardized protocols for communication and interoperability of medical devices [7]. Unfortunately, nearly all these standards evolved over time. Thus, they are not clearly specified in every detail, which leads to slightly different implementations. Every year, companies meet at *Connectathons*[1] held by the IHE, to test their system's interoperability and to track down errors before the systems are deployed in the clinic. Despite existing standards, these meetings are necessary.

Proper test date must therefore mirror the heterogeneous environment that is to be found in the realm of medical engineering. There is a lot of complex test data items for system and integration tests to choose from: Data exported from hospitals, data gathered through in-house studies and test data created to test functional requirements are part of a test data inventory, which evolves and grows over the years. Maintaining such huge test data inventories is a daunting task regarding the detection of out-of-date, redundant, or missing test data items.

Having huge test data inventories brings the issue of **data quality** into question: How is data quality measured for test data items? What test data items must be stored to improve the test data inventory and the testing process? Most well-known data quality dimensions like *correctness* or *timeliness* [3, 8] – to name just a few – are only partly applicable. It is obvious that e.g. *correctness* can't be used as a reasonable data quality dimension for test data: Achieving correctness of 100% for test data inventory would lead to error-free test data items, not being suitable to test systems coping with faulty input. *Timeliness* is just as difficult to quantify, as some test data items might be intended to be out-of-date to test the system's compatibility with old data. Therefore, traditional data quality dimensions will play a subordinate role concerning test data quality.

To improve the maintenance of test data inventories, we propose the identification of new data quality dimensions to assess test data quality. We want to set up a course of actions leading to new data quality dimensions, their quantification methods and validations at the testing sites.

We introduce our thesis approach in section 2, and present our preliminary results and sketch our quantification method

[1]http://www.ihe.net/Connectathon/index.cfm

EDBT-ICDT '12 Proceedings of the 2012 Joint EDBT/ICDT Workshops
Pages 233-238
http://doi.acm.org/10.1145/2320765.2320830

in section 3. Section 4 addresses other research related to our work. The contribution and acknowledgements close this paper in sections 5 and 6.

2. THESIS PROPOSAL

During this thesis, we want to identify and validate new data quality dimensions which are suitable to assess and prove test data quality, limiting ourselves to individual test data and system and integration testing. With our approach we want to permit test engineers to take a new angle of view on their test data. Using data quality metrics, the maintenance of large test data inventories should be facilitated. This includes the identification of missing or possibly redundant test data. Test engineers – whether having to maintain the test data repository or designing and running test cases – should able to profit from our proposed methods. To achieve this goal, we interview test engineers of companies in the field of medical engineering and ask about their day-to-day experience with test data and how they cope with data quality issues. During these interviews, we derive use cases which describe the test engineer's needs in a more formal way. Together, we phrase possible data quality dimensions and measurements which comply with the use cases. We formalize the dimensions, develop and implement measurement methods in close contact with the test engineers. In order to assess the test data inventories' quality, a comparison to the live system's data can be necessary. Therefore, we cooperate with clinical divisions to get knowledge about the live system's data. To validate the results, the developed methods will be deployed and additional interviews and questionnaires are carried out.
Our active cooperation with a company in the testing domain (exchanging a lot of knowledge) helps us to understand the process of testing for various applications and a tester's needs.

3. PRELIMINARY RESULTS

In this paper, based on the interviews held with test engineers of a company in the medical domain, we propose two preliminary results from our ongoing research: Two new data quality dimensions **proximity to reality** and **degree of coverage** along with concepts for an associated method to quantify them. Using our method, the tester will be provided with valuable advice for maintaining the test data inventory, discover redundant test data items and query mechanisms to select appropriate test data items matching the requirements of given test cases. For this thesis we are interested only in the test data items and not on the combination of test data item, test case and expected result, as we want to improve the maintenance of test data inventories.

3.1 Methods

We asked test engineers in the field of medical engineering to explain their notion of data quality regarding test data items and the test data inventory.

During the interviews, we experienced that the test engineers want to know whether their test data items mirror the live system's data, if their test data repository contains all needed test data items and if they use the best matching test data item for a given test case. Having test data resembling the live system's data increases the chance to catch everyday problems.

Additionally we got an introduction to a live system – a clinic radiotherapy division – to see the productive use of a system under test in conjunction with heterogeneous software and the regulations which producers of medical equipment (hard- and software) have to adhere to.

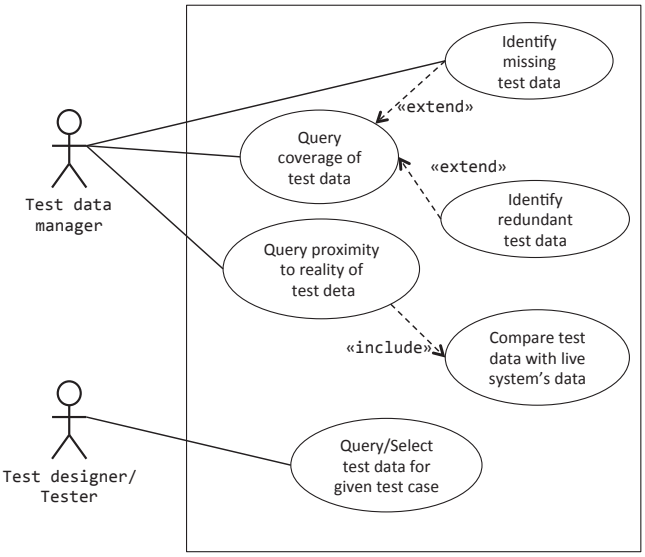

Figure 1: High-level use cases gathered during interviews

Figure 1 shows an overview of the developed use cases, which we explain in a thematically grouped order:

Assist test data management: In order to test the system under test comprehensively, theoretically every possible combination of inputs has to be available as a test data item. Nevertheless, a test data manager should be able to measure the coverage of available test data. Since a complete coverage would lead to an infinite number of test data items, it is necessary to restrict the number of input variables. Therefore this quantification will be based on test relevant properties and combinations of them, all predefined by domain experts. Next, a test data manager needs to be able to find gaps in the test data inventory and to reveal possibly unneeded, redundant test data items.

Using any kind of distribution analysis, a test data manager wants know how realistic his test data are. A comparison of the set of available test data and the live system's data using cluster analysis can be triggered.

Assist test data selection: The test designers or tester needs support for selecting appropriate test data items which match test case requirements. Instead of using a test data item, because it is subjectively 'good' and has been used for ages for some kinds of test cases, this selection support ensures that a test data item meets the requirements of a given test case.
If no matching test data items are found, similar test data items should be displayed with a measure of similarity showing their difference to the exact match.

3.2 Definition of new Quality Dimensions

Based on the gained knowledge, we identified two new quality dimensions: Test data's **proximity to reality** and **degree of coverage** in terms of chosen criteria.

Proximity to reality describes how a test data inventory

– with a given set of test data items – covers the distribution of data in the live system regarding test relevant properties. The quantification is based on a comparison of the live system's data with the available test data. Therefore, real data has to be collected and processed. It's obvious that the live system's data can't be collected at once and that the collection must be designed with an incremental approach in mind.

The **degree of coverage** for a test data inventory (regarding a test data type) describes the coverage by means of test relevant combinations of properties of the test data type. It measures the amount of available test data for a given set of combinations. Selecting the right set of properties and the right set of combinations of these properties is a vital part of measuring this data quality dimension and must therefore be carried out by experts in the application domain. Strong et al. supports this approach stating that the data customer needs to be deeply involved in the process of data quality measurement [9]. This dimension can be seen as a realisation of the fitness-for-use principle [11, 2], as it quantifies how well the available test data matches the given requirements.

3.3 Method Definition

Our proposed method is based on a multidimensional classification of test data items based on a subset of the test data type's attributes. The chosen attributes are named 'criteria' and may create non-orthogonal dimensions. Our method is based on two repositories, as shown in figure 2. Both repositories are incrementally populated by some kind of *extract/transform/load* (ETL) process, which is able to extract data from test data inventories or use exported live system's data, transform the data to match the repositories' data structures, and finally load them into the repository. The test repository stores the values per criterion for all loaded test data items and a pointer to the respective test data item in the test data inventory. It is important to note that some test data types may require multiple values per criterion. In contrast to the test repository, the reference repository, populated with exported live system's data, stores only some pointers and is restricted to counting the quantity of exported data, matching combinations of criteria. As both repositories shall not reimplement the duties and features of specialized data stores (e.g. a *Picture Archiving and Communication System* (PACS) for clinical data), only pointers, depicted with dotted arrows in figure 2, are stored. These pointers link to their originating test data items and are used to access them while testing, describing where the needed test data item can be found. Additionally, the test repository contains information about test data types, selected criteria and their mapping. These information are needed to enable a generic ETL or query process.

All examples are based on a simplified DICOM-RT data type as test data type. DICOM-RT are DICOM files with special extensions for radiotherapy. As shown in figure 3, our simplified test data type has the five attributes `PatientPos`, `NumberOfBeams`, `RadiationType`, `GantryAngle`, and `PlanningSystem`. A test data item – in this case a treatment plan – is therefore defined as a treatment given through a number of beams of some radiation type, e.g. electron, coming from different angles[2]. The patient has a position relative to the treatment system, e.g. *Head First Supine* (HFS), and

[2]We constrain for our examples that it is sufficient to know

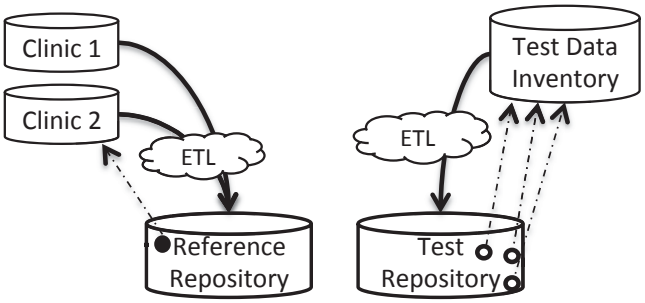

Figure 2: Schematic repository overview

the treatment plan was created with a specific treatment planning software.

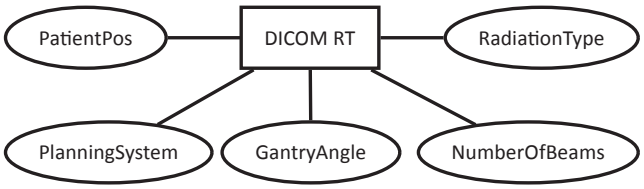

Figure 3: Simplified *DICOM-RT* test data type

The definition of a test data type's attribute can be described as tuple $a = (n, d, \delta, g, b)$. An attribute has a unique name n, a data type d (alphanumeric, numeric, boolean, ...), and a distance function δ which computes a normalized distance between two values and defaults to 1. δ should be defined by domain experts as it is used to search for similar test data items. Finally, an attribute has a valid range g and an invalid range b[3]. To describe incorrect test data items, the definition of an invalid range is necessary. Ranges can be defined as a set of values, a regular expression, or by an interval from v_s to v_e with increments of v_i. In contrary to g, b has not to be complete, as every value $v \notin g$ is classified as invalid. b is used to enable a more explicit distinguishing feature for valid and invalid test data items.

Example: The attribute `RadiationType` of our simplified test data type DICOM-RT is written as an alphanumeric value representing nominal values: $b = \{Helium\}$ and $g = \{Electron, Proton, Photon, Neutron\}$. The distance function δ for this attribute is defined as:

$$\delta(v_i, v_j) = \begin{cases} 1 & \text{if} \quad v_i \neq v_j \\ 0 & \text{else} \end{cases} \qquad \forall v_i, v_j \in g \cup b$$

A lot of test data types T – regardless of their structure and complexity – can be described as set of attributes $T = \{a_1, \ldots, a_n\}$. In reality, the test data type DICOM-RT is hierarchical structured. According to the interviewed test engineers, it is sufficient to have a flattened view per plan.

As showed in figure 1, domain experts may identify a set P of additional, artificial properties for the test data type. These properties are described in the same manner as the test data type's attributes. Possible candidates are peculiarities introduced by technical design decisions or abstract

values for e.g. `GantryAngle` per plan and not per beam. For more information about the DICOM standard the reader is referred to `http://www.nema.org/stds/dicom.cfm`.

[3]g for 'good' and b for 'bad'

classification for the test data items. Most of these additional properties can be computed by rules based on the test data type's attributes, whilst other are based on the testers' knowledge and need to be set externally.
For those properties, which rely on human input, a lazy and pay-as-you-go manner can be applied: The property is set to a special value, e.g. `not_set`. If this property is used by the system, all test data items with this value are highlighted and the user is prompted to enter the correct value.

Based on all describing attributes and properties, a subset $C \subseteq T \cup P$, the criteria, are chosen by experts to determine a standard set of relevant attributes for which values are gathered to describe a test data item in the repositories. Although it is possible to add more criteria later on, it is recommended that all necessary criteria are initially chosen by a group of domain experts who know which attributes will be relevant for various test cases. Due to the fact that the reference repository – the repository which stores information about exported live system's data – uses the standard set of criteria, there are no values available for these later added criteria. Therefore, these additional criteria can only be used to measure the *completeness* of a test data inventory and not the **proximity to reality**.

The power set of C is used as basis for selecting a set of useful combinations of criteria which are sufficient to make statements about a test data item: $H \subseteq \mathcal{P}(C)$. For each combination $h \in H_t$ a hypercube is These hypercubes are views onto the multi dimensional classification space and are utilized to measure the new test data quality dimensions. The hypercubes are constructed using each $c \in h$ as dimension. Each dimension has a combined range of $g_c \cup b_c$. For each dimension it is possible to adjust the range, e.g. into equivalence classes, to minimize the combinatorial explosion of possible cells in the hypercube.

Example: Assume the selected criteria C are `RadiationType` (RT), `NumberOfBeams` (NB), `PatientPosition` (PP) and `GantryAngle` (GA). The multidimensional classification in this example is defined by the chosen criteria as $(RT \times NB \times PP \times GA)$. $H_t = \{\{RT, NB, PP\}, \{PP, GA\}\}$ is chosen by a group of experts. Based on H two hypercubes $h_1 = \{(g_{RT} \cup b_{RT}) \times (g_{NB} \cup b_{NB}) \times (g_{PP} \cup b_{PP})\}$ and $h_2 = \{(g_{PP} \cup b_{PP}) \times (g_{GA} \cup b_{GA})\}$ are constructed as views on the multidimensional classification, helping to measure the **degree of coverage**. In order to limit the number of cells in the hypercube, the range of GA can be coarsened e.g. into steps of 20° instead of steps of 1° .

Hypercubes are the basis of the measurement of **degree of coverage**. The distribution of stored items in the test repository mirrors the distribution of test data items in the test data inventory. Evenly filled cells are a sign of an uniform distribution of test data items. For each hypercube a fill level indicator can be measured. Combining these indicators leads to an evaluation of the overall **degree of coverage**. Aided by software, the test engineer is able to find gaps and determine requirements (combinations of distinct values for criteria) for the acquisition of new test data items to best fill the gaps. A visually enhanced tracking of each test data item in each hypercube can reveal overlapping and therefore redundant test data items, which can possibly be removed from the test data inventory.

To sketch a possible measurement, we need to introduce some functions. Let `cells(`h`)` be a function returning the set of cells of a hypercube $h \in H$ and `val(`x`)` a function returning the number of matching test data items for the cell $x \in$ `cells(`h`)`. Let `eval(`a, x`)` be a function which returns 1 if `val(`x`)` $= a$, 0 else. The **degree of coverage** for a test repository can then be described as $q = \prod_{h \in H} q_h$ with q_h being the measurement for hypercube h:

$$q_h = \frac{\sum_{x \in \texttt{cells}(h)} \texttt{eval}(a_h, x)}{|\texttt{cells}(h)|}$$

A large amount of overfull cells contradicts the desired amount of test data items per combination – viewed from the test data maintainer's standpoint. As q_h penalizes overfull cells – which are no indication for bad test data – we introduce a second variant for `eval` to cope with this conflict: Let $\texttt{eval}_{\geq}(a, x)$ be a function which returns 1 if `val(`x`)` $\geq a$, 0 else. $q_{\geq}$ uses $\texttt{eval}_{\geq}$ instead of `eval`. q will be always less than or equal to $q_{\geq}$.

Example: Figure 4 shows a simplified visualisation of the hypercube h_2 with randomly chosen values. `PatientPos` (reduced to four values) is put on the first column and `GantryAngle` (reduced to the range 0° - 60° in steps of 20°) on the first row. Each cell contains the amount of matching test data items. For example, there exists only one test data item with HFS as position and a gantry angle between 0° and 20° . Assume the desired amount of matching test data items is set to 2 and the threshold for overfull cells is set to 5. In the example, cells are colored dependent on their value: Empty cells have a red border and overfull cells are displayed on red background and white text color. In our example, with $a_{h2} = 2$, q_{h2} evaluates to $\frac{5}{12} \approx 41.6\%$ and $q_{\geq,h2} = 75\%$. These results show, that the test designer can choose between at least 2 test data items for $\frac{3}{4}$ of the possible combinations (cells) described by hypercube h_2. The gap between q_{h2} and $q_{\geq,h2}$ illustrate that there are many cells with more matching test data items than required.

	0° - 20°	- 40°	- 60°
HFS	1	2	2
HFP	0	0	2
FFS	3	3	2
FFP	2	8	6

indicate empty cells
indicate overfull cells

Figure 4: Simplified visualisation of hypercube h_2 for `PatientPos`×`GantryAngle`, filled with sample data.

Obviously, overfull cells indicate a agglomeration of test data items for a combination of criteria. This agglomeration may be unintended, created through unrestrictedly collecting of possible test data.

On the other hand, this may represent the distribution of the live system's data. To verify if these combinations are common, the test repository and the reference repository are compared. A cluster analysis is performed for each data set, highlighting often used combinations of criteria. These clusters are then compared between both sets in order to determine whether the test data inventory shows the same distribution as the live system's data or if there's a need for new test data items covering combinations of criteria which seem to be common in the live system. Matching test data items may be acquired by querying the reference

repository for the stored item's pointer into the exported live system's data, if data security of the exported data allows this usage. Using this pointer, the data can be copied into the test data repository. This shows a high quality in means of **proximity to reality**. Obviously, this comparison yields better results the more data from live systems is collected. The incremental population of the reference repository is guided by the repository itself. When the distribution of data is known, missing combinations can be detected and live system's data which fills these gaps can be requested.

Figure 5 sketches such a comparison for the simple case of a two dimensional classification. It shows that the test repositories distribution diverges from the reference repositories distribution. Thus the test data repository does not fully mirror the live system.

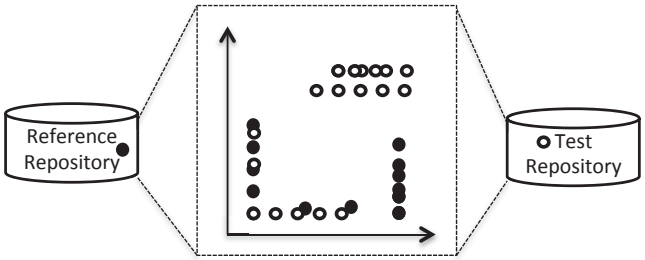

Figure 5: Simplified comparison of the test repository and the reference repository

Additionally, the sketched method provides easy selection of suitable test data items for a given test case. Via queries based on the chosen criteria – often preconditions of the test case – it is possible to filter all available items. A list of matching items and their pointers into the test data inventory is returned. This query based selection assures that adequate test data items are used for testing. If no matching test data items are found, the repository can be searched for similar test data items by solving the optimization problem of minimizing all distances using standard algorithms. The similarity search is not used as standard search, as the optimization problem can be hard to solve.

3.4 Discussion

As preliminary result, we proposed **proximity to reality** and **degree of coverage** as two new data quality dimensions suitable to describe aspects of data quality for test data inventories. Our sketched quantification method will be based on multidimensional classification and its detailed specification is part of our ongoing work. Additionally, we are working on a metric to quantify the measure for **proximity to reality**. Also, the introduced query mechanism will be specified in more detail.

The interviews on the topic of test data quality were performed at a company working in the field of medical engineering. The method's validation is in progress within the company's testing division. During the project's progression, more domain experts will be interviewed and given the opportunity to validate the method. Also, additional metrics will be developed based on the test engineers' demands.

Data types are often standardized or defined by requirements. The special properties are mostly already part of the testers' workflow, e.g. if they are used to distinguish test data. Thus, the definitions of attributes and properties are likely to be correct. A faulty definition of a distance function can be corrected without hampering the system on a grand scale. The cluster analysis to quantify the **proximity to reality** had to be rerun, which is needed also if new data is fed to the system. If new attributes or properties are added, the values have to be obtained for each repository. If this cannot be covered through the application of rules, this introduces – in the worst case – a need for human inputs, which might be handled in a lazy, pay-as-you-go fashion. Although the system depends on the domain experts' knowledge, the costs of wrong decisions are acceptable. The retrieval of this knowledge may be difficult, as the experts may have problems to bring their knowledge into a formalized notion, e.g. the distance function.

The accessibility of a repository with exported live system's data to measure **proximity to reality** is no conceptual barrier but may be a practical one. A joint cooperation of companies would be needed to run the reference repository, request clinical data and populate the repository. Without a reference repository, our method will only be able to make statements about one company's individual test data quality regarding **degree of coverage**.

A prototype is currently under construction and will be deployed at our validation partner to validate our preliminary results. It is expected that the new data quality measurement facilitates the test data inventories maintenance and provides answers concerning the usage of the most adequate test data items. This will lead to positive effects on the test process, like shorter test cycles or reduced test costs. Although this thesis focuses on test data quality for medical devices, we expect that our proposed methods for measuring test data quality are also applicable to other domains, as the fundamental approach is not restricted to the domain of medical engineering.

4. RELATED WORK

Test data selection has been deeply investigated [4, 5, 6]. All developed approaches use some flavor of path or data flow analysis of a program under test as criteria for test data selection. That is a common approach when doing white-box testing, as the structure of the system or program under test is known. Being scheduled in later testing phase and assuming a functionally tested system, our approach can't take advantage of white-box testing, as the system under test is treated as a black-box.

Wang et al. describe the annotation of attributes to make statements about *believability* and *interpretability* [10]. They propose an extension to the entity relationship model, attaching data quality attributes to every regular attribute. Therefore each attribute has defined values for data quality dimensions and data customers can act accordingly. However, we use the distribution of values from selected attributes as a whole to assess test data quality.

Strong et al. propose a new point of view to measure data quality. Instead of relying on intrinsic data quality dimensions, they claim to incorporate the data customer's view to an overall understanding of actual data quality problems beyond intrinsic data quality issues [9]. We support this finding, as our method recommends a lot of communication and cooperation with domain experts.

5. CONTRIBUTION

In this ongoing Ph.D. thesis, we are going evaluate established data quality dimension regarding test data. We will identify and describe new data quality dimensions suitable for assessing test data quality. As we show with our preliminary results, there is a demand for test data quality which can't be addressed using only standard data quality dimensions. We propose a method to measure and validate the new data quality dimension and illustrate their benefit in the testing domain.

6. ACKNOWLEDGMENTS

This project is supported by the German Federal Ministry of Education and Research (BMBF),
project grant No. 01EX1013G.

7. REFERENCES

[1] T. Aden, J. Riesmeier, and A. Dogac. A survey and analysis of Electronic Healthcare Record standards. *ACM Computing Surveys (CSUR)*, 37(4):277–315, Dec. 2005.

[2] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino. Methodologies for data quality assessment and improvement. *ACM Computing Surveys*, 41(3):1–52, July 2009.

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

[4] T. Chusho. Test data selection and quality estimation based on the concept of essential branches for path testing. *Software Engineering, IEEE Transactions on*, (5):509–517, 1987.

[5] U. Linnenkugel and M. Müllerberg. Test data selection criteria for (software) integration testing. In *Proceedings of the First International Conference on System Integration*, pages 709–717, 1990.

[6] S. Rapps and E. Weyuker. Selecting software test data using data flow information. *Software Engineering, IEEE Transactions on*, (4):367–375, 1985.

[7] J. G. Rhoads, T. Cooper, K. Fuchs, P. Schluter, and R. P. Zambuto. Medical device interoperability and the Integrating the Healthcare Enterprise (IHE) initiative. *Biomedical instrumentation & technology / Association for the Advancement of Medical Instrumentation*, Suppl:21–7, Jan. 2010.

[8] M. Scannapieco, P. Missier, and C. Batini. Data quality at a glance. *Datenbank-Spektrum*, 14:6–14, 2005.

[9] D. M. Strong, Y. W. Lee, and R. Y. Wang. Data quality in context. *Communications of the ACM*, 40(5):103–110, 1997.

[10] R. Y. Wang, M. Reddy, and H. B. Kon. Toward quality data: An attribute-based approach. *Decision Support Systems*, 13(3-4):349–372, Mar. 1995.

[11] R. Y. Wang and D. M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–34, 1996.