InDaQu: Enabling user-centered definition and exchange of consistency constraints for data cleaning *

Stefan Brüggemann\textsuperscript{1} and Yvette Teiken\textsuperscript{1} and Hans-Jürgen Appelrath\textsuperscript{2}

\textsuperscript{1} OFFIS - Institute for Information Technology, Oldenburg, Germany, \{brueggemann,teiken\}@offis.de
\textsuperscript{2} University of Oldenburg, Department for Computer Science, Oldenburg, Germany, appelrath@informatik.uni-oldenburg.de

Abstract. Severe data quality problems exist in most public health care systems and inconsistent data sets often occur. Consistency constraints can be used to define valid and invalid data. Existing solutions of such constraints like rule systems are often difficult to maintain, not human-readable, and of a bad quality like containing contradictory rules. With InDaQu we present an approach that allows domain experts to easily create and maintain consistency constraints using an introduced domain-specific language. These constraints are being stored in an ontology, which allows for an automated inconsistency detection in the defined rules themselves. We identified several scenarios in which consistency constraints can be interchanged and exchanged between different participants. The approach has been successfully evaluated in the cancer registry of Lower Saxony.

1 Problem Statement

Data quality is a critical aspect in most public health care systems. Inconsistent data can be found in hospitals, cancer registries, and other data producers [12]. These data quality problems often result in high costs and low medical quality [2].

A variety of problems in health care can be identified:

\begin{itemize}
  \item Medical errors may kill or cause long-term damage to patients [14]. It is stated that between 44,000 and 98,000 lives are lost every year due to medical errors in U.S. hospitals. Not all of these errors occur due to inconsistent data, but a number of studies have shown that poor data quality is responsible for medical errors and subsequent poor quality of care [16, 20, 7].
  \item The accuracy of insurance bills is often reduced due to poor data quality. In a study [6] medicare data has been analyzed and 2.7 percent of about 12 million
\end{itemize}

\* This work has been presented at ACM SAC 2010 in the track ”The semantic web and its applications” with the title ”Interchangeable data quality definitions for public health care systems”
records, approximately 321,300 insurance bills, could be identified containing coding errors. These errors often lead to problems in the patient’s insurance reimbursement and to time-consuming tasks of identifying and correcting these errors.

– Cancer registries analyze collected data and create reports based on these data. The evaluation of the Cancer Registry of Norway has shown that about 7 percent of the tumor cases could not be verified and contained inconsistencies [15]. Following the principle of ‘garbage in, garbage out’, it is obvious that recommendations based on analyses with these data might lead to misleading decisions.

– Inconsistent data is critical in networked environments like integrated care networks. When several organizations and medics are involved in a patient’s medical attendance and data is being exchanged between them, data quality problems may arise. Different organizations often use different classification schemata, quality definitions, and consistency constraints. This may result in wrong medications, operations, or therapies when data is not interpreted correctly.

The reasons for the existence of inconsistent data can be clearly identified: Existing classification schemata are often complex and difficult to learn. Doctors have to be familiar with diverse regulations and coding systems. A study [12] has shown that one of three doctors has not retrieved systematic training in medical coding. Furthermore doctors and hospitals do not always use equal classification schemata and consistency constraints. This results in inconsistencies occurring at different locations, e.g. in data exchange or data integration. The need for technical solutions for medical coding has been stated in [19], because inconsistencies can be detected during data production. For instance, when a pathologist classifies a tumor, consistency constraints can detect quality problems in real-time and inconsistencies can be avoided.

Since standardized coding systems exist, the authors of [1] request for standardized and exchangeable consistency constraints. Uniform coding standards must be consistently required, promoted, and uniformly applied across sites of service.

In this paper we propose an approach for interchangeable consistency constraints. Several environments in health care systems exist, in which data have to be integrated from different data sources. We focus on data containing attributes which belong to existing coding systems. These coding systems define closed world data in our scenarios. As an example, a tumor registry collects tumor describing data. These data contain attributes describing the classification of a tumor, or the number of nodes or metastases. These data may contain inconsistencies like coding prostate disease for women. In our ontology based scenario, we give the ability to manually refine existing consistency constraints, which can then be applied in data quality environments. The approach enables to deliver rules bundled with a technique for consistency checking. Consistency constraints are often complex, non-trivial, and might contain contradictory rules. These contradictions can be avoided in our approach. It provides an excellent
basis for interchanging these rules between organizations. When data is being delivered to data consumers, it can be bundled with according consistency constraints. These definitions are an ideal basis for discussions about consistency constraints.

This paper is structured as follows. In Section 2 we introduce our approach InDaQU. The following section 3 introduces different application scenarios. In section 4 we show how we successfully introduce InDaQu in the cancer registry of Lower Saxony. The paper ends with section 6 giving an overview of further work and conclusions.

2 Our approach InDaQu

Data is often being exchanged between acteurs of public health care systems. In our approach InDaQu consistency constraints are being defined for these data. We do not focus on personal patient data like name, age, or sex, but on closed world data like ICD\(^3\)-codes, tumor classifications, or operation and procedure codes. Some of these are already defined in taxonomies or ontology-based systems like UMLS\(^4\) or SNOMED\(^5\). Further organizations like NAACR\(^6\) or IACR\(^7\) define specialized versions of these classifications. These standards can easily be integrated in InDaQu. Figure 1 shows the metamodel of InDaQu. It consists of dimensions and rule definitions. Dimensions can have a hierarchical structure and can be used to create rules. A rule definition has a left and a right side. This left and right side describes which codes are used in this rule.

InDaQu serves a set of known dimensions like \(A_1, \ldots, A_n\) with instances \(a_1, \ldots, a_n\) in a specific domain or scenario. Each of these dimensions may consist of distinct attributes \(A_{ij}\). InDaQu allows for defining consistency constraints with

\(^3\) International Classification of Diseases: Defining a set of codes that classify malignant neoplasms and other diseases [http://www.who.int/classifications/icd/en/](http://www.who.int/classifications/icd/en/)


these dimensions and attributes by domains experts. In such a domain several rule descriptions can be defined using the introduced dimensions: Rule $r : \{A_{Left_1}, ..., A_{Left_n}\} \rightarrow \{A_{Right_1}, ..., A_{Right_m}\}$ with \(\{A_{Left_1}, ..., A_{Left_n}\} \cap \{A_{Right_1}, ..., A_{Right_m}\} = \emptyset\). An instance of this rule can then restrict the combinations of the instances of the used dimensions: $r_i : \{a_{left_1}, ..., a_{left_n}\} \rightarrow \{a_{right_1}, ..., a_{right_m}\}$.

We identified two types of rules, unique and non unique rules. In InDaQu rules can be seen as mathematic relations. A unique rule has for every left side one and only one right side. So it is possible to determine one side when the other side is given. In a non unique rule it is impossible to determine the left side when the right side is given. The reason is that there exists more than one right side for a left side.

This metamodel can be instantiated in a concrete scenario. The metamodel is not limited to a specific domain. It allows the integration of a number of classification systems that may occur in medical environments and other domains like product categories or e-business. These classification systems only have to be conform to the structure of the metamodel. These classification systems act as dimensions in InDaQu. Based on the dimensions ICD and TNM[11] we can define the following rule $ICD \rightarrow \{T, M, N\}$ An example for this rule is $C11.0 \rightarrow \{T = 0, M = 1, N = 1\}$. ICD defines specific malignant neoplasms, and T, N, and M define a set of tumour describing attributes. For each specific neoplasm only some T, N, and M values are valid. Medical experts agree that such rules exist for the description of tumours. These rules are defined by the previously described IARC, UICC, or an other institution.

In InDaQu we use ontologies for modeling and checking the described rules. Ontologies are well known concepts in computer science and provide a formal representation of a set of concepts of a specific domain and define relationships between them [9]. They can be used for knowledge representation and can be applied in data interchange. They are also used for semantic modeling. Ontologies are also used in the area of health care.

A lot of classifications are already available in OWL [18], the web ontology language that can be used to describe ontologies. Other can easily be realized in OWL. In listing 1.1 a part of an ICD definition in OWL, as we modeled them for InDaQu, is shown. Based on these classifications we can model rules in an ontology. Ontologies are also a known concept in data quality management. Several utilizations of ontologies providing domain knowledge for data quality management are introduced in [5].

With conditional functional dependencies[3] and edit/imputation-systems[8] several approaches for the definition of consistency constraints already exist. In these approaches the defined consistency constraints are restricted to a concrete database. Our approach provides logically independent consistency constraints that can be applied to any data set like a database, XML-file, or else. The mentioned approaches further do not allow the integration and use of existing
dimensions. Our approach enables the integration of dimensions that exist in any namespace.

In InDaQu we use OWL-DL. This enables to automatically validate a set of defined rule instances. These rule instances, which in fact are consistency constraints, can contain contradictory rule instances like 'C20 is only valid with T-value 1' and 'C20 is only valid with T-value 2'. A reasoner automatically identifies this as an inconsistency [22], since ‘C20’ is only valid with T value 1 or value 2. The domain expert then has to decide which value is correct.

Another advantage of the use of ontologies is that is possible to use well known standards like Dublin Core\(^8\). Dublin Core is a standard for describing meta data of documents. We use this standard to describe rules using the concepts title, description, date, creator, and literature source.

As imaginable after seeing listing 1.1 native OWL might not be practical in use for domain experts. Existing tools like Protégé\(^9\) work on ontologies, not on rules. The domain expert who works with InDaQu is interested in modeling rules and mappings for data quality, not in OWL modeling. So we had to rise the level of abstraction for domain experts. We created a graphical domain specific language (DSL) for InDaQu. For DSL a meta model is needed [13]. This metamodel is shown in figure 1 as a MOF-model (Meta object facility \(^{10}\)). We use this meta model as a fundament for our graphical domain experts interface. Now we have the advantage that domain experts can model rules in an intuitive way and are able to store them correctly in an ontology. This is possible because of our meta model and the defined transformations within the model. Figure 2 shows how the abstraction between the meta-model, DSL, and ontology works. The end user DSL stores its data in an instance of the meta model. This instances are then stored in the ontology. This is done by a transformation that transforms instances of the meta model into the ontologies. This approach has some advantages. Sometimes it is necessary to change the internal of an ontology. In InDaQu we can change the ontology without changing the meta model or the end user DSL. All we have to change is the transformation. Another advantage is that the end user DSL can be changed more easily. Because we don’t interact with the ontology but with the meta model. So it’s possible to implement/deploy

---

\(^8\) [http://dublincore.org/](http://dublincore.org/)

\(^9\) [http://protege.stanford.edu/](http://protege.stanford.edu/)

other DSLs for other kinds of end user who are not familiar with the given representation[21]. Other possibilities would be a textual language or a language that is based on graphs.

This is possible because we accomplish with this approach a separation of concerns. We can differ between the ontology and the end user interface.

![Fig. 2. InDaQu meta-model in use](image)

### 3 Application scenarios of our approach

We now provide two distinct scenarios in which these quality constraints can be used. In the first one, illustrated in figure 4, a centralized institution exists which defines quality constraints. This may be an institution like UICC\(^{11}\), IACR, or the BQS\(^{12}\). In such an organization data quality commissioners have to define quality constraints. These constraints can act as a system-wide data quality standard. Data that is being exchanged in such a public health care system has to fulfil these requirements. These constraints can act as an ideal basis for discussions in a feedback loop again. When quality constraints are being changed frequently, these versions can be identified and referenced using URIs and namespaces.

Another application scenario is shown in figure 3. Here, smaller institutions like doctors in private practice can choose one of these definitions and do not have to define quality constraints on their own. Hospitals deliver data to different recipients. These recipients may be other hospitals, doctors in private practice, health insurance companies, or institutions that perform analyses on these data, like cancer registries. As we have seen in section 1, data quality is a critical aspect in public health care systems. Therefore, InDaQu can be used in hospitals to define quality constraints. These constraints may be frequently modified, for instance when new versions of the ICD are being introduced. Without our approach recipients might disagree with the quality of the delivered data or might not be able to interpret them correctly. With InDaQu data can be delivered together with corresponding consistency constraints. This allows data recipients to understand the data. When they disagree with the defined and delivered consistency constraints, these ontologies serve as an ideal basis for discussions in a feedback loop.

In this scenario smaller institutions like doctors in private practice can choose one of these definitions, as they do not want to define quality constraints on their own.

---

\(^{11}\) International Union against Cancer [http://www.uicc.org](http://www.uicc.org)

\(^{12}\) Bundesgeschäftsstelle Qualitätssicherung gGmbH [http://www.bqs-online.de](http://www.bqs-online.de)
Fig. 3. Hospitals are able to define consistency constraints on their own. Data then can be delivered to consumers bundled with these constraints.

4 Evaluation

We have successfully introduced InDaQu in the cancer registry of Lower Saxony (EKN)\(^\text{13}\). Cancer registries integrate data from several distinct information sources like pathologists or hospitals into a data warehouse. Experts perform complex analyses using this data warehouse. These analyses are being used for decision support and to research several cancer specific questions. Data quality is a critical aspect in this environment. There was a need for a rule system that is able to integrate existing classifications like the ICD or the TNM-System. Based on these classifications rules should have been described. In this concrete scenario the following rules were defined:

- ICD2TNM \(\{ICD\} \rightarrow \{T, N, M\}\)
- ICD2Histology \(\{ICD\} \rightarrow \{Histology\}\)
- ICD-T-N-M-TNMVersion2UICC \(\{ICD, T, N, M, TNM\text{Version}\} \rightarrow \{UICC\}\)

The latter rule definition contains a validation of a UICC-stadium that assigns a specific stadium to concrete tumours. These rules show the benefits of using OWL in our approach because the rule ICD2TNM is embedded in ICD-T-N-M-Version2UICC and can be internally represented using transitivity.

\(^{13}\text{http://www.krebsregister-niedersachsen.de}\)
A centralized institution defines consistency constraints that can be referenced by doctors or hospitals.

Listing 1.2 shows the concrete rule definition ICD2TNM and an instance of this rule. The Dublin Core elements `dc:title` and `dc:description` are used to describe the rule ICD2TNM, the concrete instance ICD2TNM4 identifies (C54.1, is, 1, 0) as a valid combination. The instance is labeled with other Dublin Core elements containing the last modification date (`dc:date`), the source it is based on (`dc:source`), and the responsible user that has created this rule (`dc:creator`).

Listing 1.2. Rule Definition

Domain experts have a need for an intuitive visualization that enables them to define these rules. Therefore the previously defined DSL has been used to
build a user-friendly graphical user interface, that is shown in figure 5 and figure 6.

![Diagram](image)

**Fig. 5.** An interface that allows the creation of new rule definitions and to load existing ones

The figures show how new definitions can be created and how existing rule definitions can be loaded. In the first case existing dimensions can be chosen either for the left or the right side of a rule definition. A rule definition can be classified as a unique or a cartesian product rule. The rule in figure 6 shows a rule that is not unique. In a unique rule exactly one value for T, one for N, and one for M would be possible. The rule is an example of an cartesian product rule because rules are defined for every combination of the selected values for T, N, and M.

In the latter case the screen shown in figure 6 will appear. It shows an example of the rule ICD2TNM. In this perspective ICD-codes can be selected on the left side. The user then can choose the valid values for T,N, and M for the selected ICD-code.

On the left side the ICD-code 'C15.4' is selected (highlighted in green). Then on the right side the T-,N-, and M-values are shown that are valid for this ICD-code. The highlighted T-,N-, and M-values are currently valid for the given ICD-code, but the user can simply select other values and modify the selection.

Classifications like the ICD often have a hierarchical structure. This structure can be found in figure 6, too. On the left side groups can be chosen from this structure from a drop-down list. In this example the group 'GruppeC15' contains the subgroups 'C15', ..., 'C20'. When a user selects one of these subgroups only the concrete codes from this group are being shown in the panel labeled 'Nodes'. In this example no subgroup is selected, but 'GruppeC15'. Therefore all codes that are more specific than 'GruppeC15' are being shown. These include the codes 'C15.0', 'C15.1', 'C15.3', 'C15.4', for instance. When a user defines rules, he can choose whether to select a node from the 'Nodes' panel or a group or subgroup from the 'Layers' panel. This allows for creating rules not only for each concrete
code, but simplifies this so that rules can be created for a group and the rules can then be inherited to the codes.

Fig. 6. A specific rule for the rule definition ICD2TNM

The rule definition ICD-T-N-M-TNMVersion2UICC determines exactly one UICC-value for combinations of ICD, T, N, M, and TNM-Version. With InDaQu it is possible to identify inconsistent rules. An inconsistent rule set would contain more than one UICC-value for a given combination. An example for such an inconsistent rule set is shown in listing 1.3.
The following rules are inconsistent:

17423: 'C10.3', 'T_3', 'N_1', 'M_0', 'TNM_Version_5' -> 'III' and
23223: 'C10.3', 'T_3', 'N_1', 'M_0', 'TNM_Version_5' -> 'IV'.

Listing 1.3. Detection of inconsistent rules

The defined rules have been applied in the registries' data integration processes, where data from different pathologists is being integrated. Listing 1.4 shows some identified inconsistencies. These identified inconsistencies then are sent back to the corresponding pathology together with the rule set in a feedback loop.

Listing 1.4. Results of a consistency check

5 Related work

In the surrounding of InDaQu exists related work. In [17] an W3C initiative for semantic web and health care is introduced. The goal of this initiative is to identify use cases and methods for extracting Semantic Web representations from existing medical record terminologies. These representations will be based on OWL and they started with some with a focus on a sub-domain of SNOMED-CT.

Related work also exists in the field of coding. These more or less handle with rules or edits for coding. These edits are not based on ontologies but on other rule engines. One implementation of these systems is given by IARC[14]. This implementation is named DEPedits and it is used for data quality issues by reporting to IARC [10]. It is possible to convert between different ICD versions. The conversion rules defined by IARC are within the program so it is not possible to edit and not even to see these rules. And it is not even possible to use these conversion rules in a own process.

A common approach in public health care is the use of edit and imputation systems.[8] In these systems errors in data is automatically replace by another valid combination. The replacement is based on a statistical function. One problem in this approach is that the data is consistent but not valid. Based on a statistical function it is not determined a if tupel is correct because correction is only frequency-based.

6 Conclusions and Future Work

We have presented an approach for the definition of consistency constraints and shown that public health care systems often have problems regarding data

\[14\] http://www.iarc.fr/
quality and that there is a need for standardized and interchangeable consistency constraints. Starting with existing medical classifications, we integrated them in an ontology and provided formalisms to create rules with these classifications. We developed a domain specific language and used it to create a user interface that can easily be used by domain experts to create rules. The approach presented in this paper has been evaluated in the cancer registry of Lower Saxony. The evaluation has shown that a set of existing rules has been successfully defined and that user can easily maintain these rules. The rules successfully identified inconsistencies in data integration processes and inconsistent rule definitions could be avoided with the presented approach.

More work has to be done in the field of ontology mapping. In a scenario where several rule definitions from different vendors or hospitals exist in ontologies, these rule definitions have to be mapped against each other. This mapping could identify the corresponding rules for a given ontology in the other ontologies and allows for an automated rule mapping and could provide a feedback basis for domain experts.

Future research will focus on semi-automatic data cleaning. The described approach provides a basis for consistency checking. Identified inconsistencies have to be repaired by domain experts. InDaQu has to be extended to provide correction suggestions that can be presented to domain experts. User interaction has to be minimized in this task and existing work for an automation of this task has to be used as described in [4].

References