# TaDaQua: Tangible Data Quality with Object Signatures

**PI: Simon Razniewski**
**UniBZ-Team: Mouzhi Ge, Werner Nutt, Ognjen Savkovic**
**External Partners:**
**Stefan Hellweger, Department of Hydraulic Engineering, Province of Bozen-Bolzano**
**Felix Naumann, Hasso-Plattner-Institut Potsdam**
**Divesh Srivastava, AT&T Labs-Research**
**Flip Korn, Google Research**
**Florian Daniel, University of Trento**

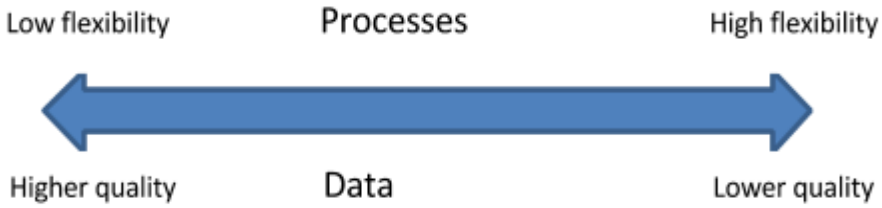
**Abstract**

In many applications, databases suffer from incompleteness e.g. due to flexible or varying processes for content creation. While missing values are relatively easy to notice, missing records can be harder to detect. For instance, this university requires that research proposals with a budget larger than 50k€ should list members from at least three different disciplines. If such a proposal is without links to members from at least three different disciplines, it will be considered as incomplete.
Knowledge about incompleteness is crucial both for data producers, who may not know which fields and links are important to fill, and for consumers, who may have a hard time assessing whether a data object is complete or not. Furthermore, data incompleteness can cause additional time to acquire information outside of the database.
The common approach to verify whether all important data is present is using a validation logic. However, the creation and maintenance of a validation logic are often a cumbersome process that requires considerable domain knowledge about edge cases and exceptions. To tackle this challenge, in this project we intend to investigate whether machine learning is a viable alternative to the use of a validation logic. For the features used in the machine learning, we rely on object signatures, which are linear representations of the occurrences of objects throughout a database. We also aim to provide explanations for the classifications done by the classifiers.
The goal of this project is the explorative analysis of an aspect of data quality for complex objects, which should create interesting links between data quality, database and business process research, and open up further options for collaboration.

## 1. Background

Data quality has become a critical concern to the success of projects and organisations. Numerous initiatives or projects have been delayed or even cancelled, citing data quality problems as the main reason. Among the data quality problems, missing data is a prevalent problem in many applications. Especially in scenarios where workflows are only loosely coupled with IT-systems, the documentation of real-world activities in databases is often incomplete or incorrect [31]. For example, in hospitals, since patients with different symptoms undergo very different analyses and treatments, there then exist a large number of processes. Consequently, without knowledge about the details of the processes it may be very hard to identify whether e.g. some test results should be there or not. Two key factors that may lead to data quality issues are flexible processes and user experience concerns. When the execution of processes is flexible, it may be hard to properly configure data input systems to ask for all the important data, without requiring any data that does not exist in reality. This can in turn lead to a bad user experience, for example if users have to submit data that does not exist in the real world, they will either submit dummy entries (e.g. “123-45-6789” for a US social security number when entering information

for a person without a US social security number), or they will avoid entering records into the IT system at all. In a nutshell, there is a tradeoff between process flexibility and data quality as shown below:

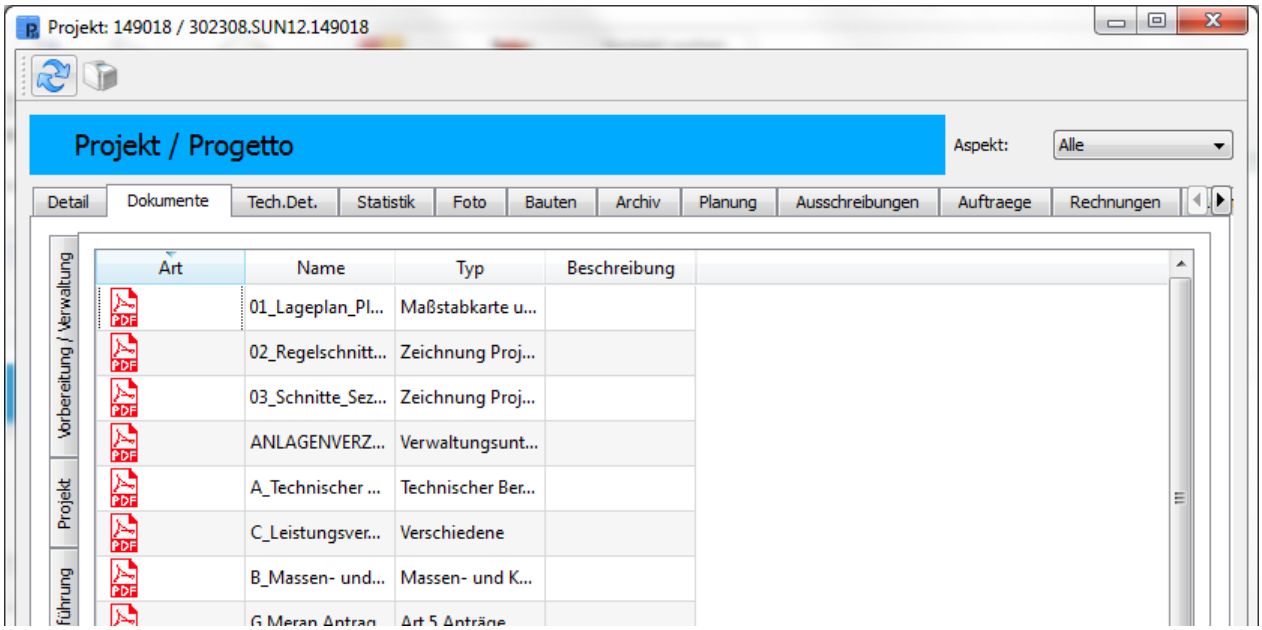



The organization of IT systems thus has to be balanced between flexibility and data quality. One extreme would be for instance a system that always requires a non-null US social security number, without anticipating the need for entering information about Europeans not having a US social security number. On the other hand, an example would be databases where data can be entered without any validation. Then, there is no guarantee about data quality, but it has the total freedom regarding the process of how the data will be stored in the database.

This proposal is based on the hypothesis that a trade-off between flexible processes and reasonable levels of data quality can be achieved. When the users can obtain the information about the quality of their data, a higher data quality can be reached while the process flexibility is kept. For example, one of the frequently used tools for this purpose is a validation logic. In this context, validation logic refers to a formalized collection of domain knowledge that allows to assess the quality of data. Such a logic can be both useful for data creators, to remind them, which data they still need to insert, and for data consumers, to make them be aware of the quality of the consumed data. Subsequently, processes can still run flexibly, but users have indicative tools that allow them to assess how well data describes a real-world process.

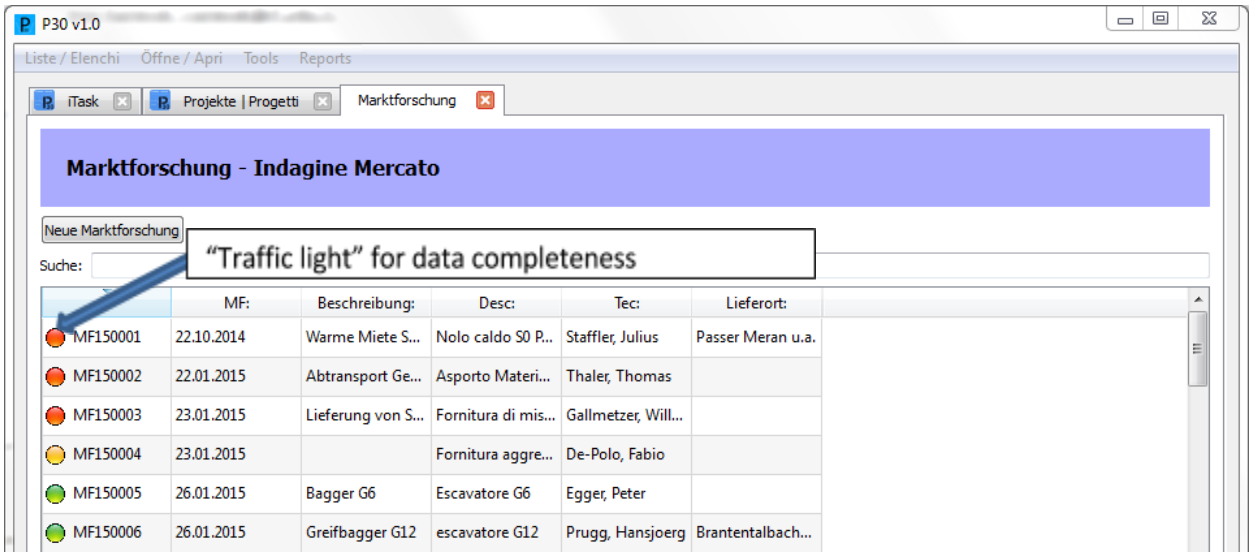
## Use Case in the Province of Bozen-Bolzano

As a real-world use case, we have investigated the project database in the department of hydraulic engineering of the Province of Bozen-Bolzano. Since there is a large amount of mountains in the Province of Bozen-Bolzano in South Tyrol, it is the local department's responsibility to construct and maintain hydraulic defense constructions like check dams, dams and avalanche nets etc. Subsequently, the department's project database covers a variety of topics concerning planning, execution and administrative documentations of such projects, as well as the acquirement procedures of raw materials and specialized works. The screenshot below shows the current system that is used in the local department.




There are different factors to determine how projects are created and operated, for example, whether a project is about construction or maintenance, whether it is locally or EU-funded, and whether it is small or large. Likewise, the required documentations can vary, for instance, small projects do not require a call for proposals, maintenance projects may not require a geological study, EU-projects may require a much more extensive documentation, or emergency measures can be executed with simplified and more result-oriented procedures.

The department of hydrology has realized that completeness of data is very important to them. Poor completeness results in studies or analyses being unnecessarily done several times, in wrong conclusions wrt. the state of projects (e.g. that no call for proposals was issued yet), or in wrong conclusions about aggregates (e.g. believing that half the budget of a project is still available, when in fact 90% were spent already). Therefore, in order to make data producers and consumers be aware of the quality of project data, the department of hydrology annotates each project with a “traffic-light-style” data quality indicator, which is either a green, yellow or red circle. Green means that all information that a domain expert believes should be there is there, yellow means that the core information is there, however some details are missing, and red means that core information is missing. From interviews with the local department, we found that those indicators are helpful in ensuring quality, because they are publicly visible, and no data creator wants to be seen as creating many bad entries. Below we show a sample interfaces with these quality indicators:




Currently, the color of the traffic lights is computed using a manually-created validation logic. Below we show some sample code as it is currently used for the computation of the traffic-light colors within the project database:

```
public void validate() {
    validate = new Vector<Validierung>();
    boolean hasAngebot = false;
    boolean hasAngebotsanfrage = false;
    boolean hasAuftrag = false;
    anz_warn = 0;
    anz_fehler = 0;
    for (DDocument d : doc) {
        if (d.getTypDocument() != null) {
            if (d.getTypDocument().getKey().equals("TD085")
                    || d.getTypDocument().getKey().equals("TD060")) {
                hasAngebot = true;
            }
            if (d.getTypDocument().getKey().equals("TD084")) {
                hasAngebotsanfrage = true;
            }
    }
    if (!hasAngebot && anz_antworten != null && anz_antworten > 0) {
        validate.add(new Validierung(Validierung.Typ.WARNING,
                "Kein Angebot!", "Nessuna Offerta!"));
        anz_warn++;
    }
    if (!hasAuftrag && s == null) {
        validate.add(new Validierung(Validierung.Typ.WARNING,
                "Kein Auftrag!", "Nessun Incarico!"));
        anz_warn++;
```

The challenge with the use of a validation logic is that writing this logic is labor-intensive and requires a lot of domain knowledge, because for some real-world objects, relevant information can be split over various tables.[1] Therefore, in this proposal, we focus on researching the alternatives to the manual creation of a validation logic.

[1] *In the following, we name such objects as Complex Data Objects.*

## Research Questions

Based on previous work in the literature, we hypothesize that the quality of data can be validated in a data-driven way from labelled samples. Thus instead of explicitly writing the validation code, we plan to use a set of objects, labelled with completeness indicators, to predict the completeness of other unlabelled objects. Therefore our main research question is:

**How can supervised learning be used for the completeness assessment of complex data objects?**

This main research question can be further divided into 3 sub-questions:

*Q1: Are there learning algorithms by which this can be done?*

*Q2: Which input should we give to these learning algorithms?*

*Q3: How can we explain the results of such algorithms to users?*

Regarding sub-question 1, there exist a variety of techniques that could be utilized. Nearest-neighbour techniques are interesting for their simplicity. Decision trees are favourable because humans can interpret them easily. In turn, on many tasks Support Vector Machines perform best, but are generally impossible to interpret.

Regarding sub-question 2, note that machine-learning techniques require a vector of attributes as input, and that in many machine-learning tasks, attribute engineering is the actual main challenge. We expect the same for our task. A complex object may be linked to many other objects, which have attributes and can be linked to other objects again, and so on. How to best aggregate linked objects, identify possibly relevant attributes, and where to prune the links, seems an open problem (for illustration, consider the related problem of recommendations over a friendship graph. It is well known that the preferences of your friends allow very well to predict your preferences. To a lesser degree, this holds also for the friends of your friends, and so on. But clearly not all transitive friends can be supplied as attributes).

Regarding sub-question 3, observe that classifiers learned by machine learning algorithms are often not human interpretable. However, for the usability of our approach it is crucial that user do not only receive information on whether an object is complete or not, but also, why it is not complete, so they actually know what to do in order to complete the object.

Together with process mining, learning important data fields might help in identifying common data patterns and subsequently missing links. We believe that databases about research projects face analogous challenges, when requirements regarding the documentation varies.

# 2. State of the Art

Data quality problems have been investigated in a substantial body of literature. For example, More than 60% of 500 medium-size firms were found to suffer from data quality problems [13,14]; it is estimated that 1% to 10% of data in organisational databases is inaccurate [14]. Those data quality problems are often associated with high costs [12]. For example, poor data quality management costs more than $1.4 billion annually in 599 surveyed companies [16]; a telecommunications company lost $3 million because of the poor data quality in customer bills [15]. It is estimated that poor data quality results in 8% to 12%

loss of revenue in a typical enterprise [31]. Therefore data cleansing, the process of detecting and correcting errors, inconsistencies, duplicates and other problems related to data correctness, has received considerable attention in the past [18,19,20].

Data completeness can be defined as “the extent to which data are of sufficient breadth, depth, and scope for the task at hand” [16]. This definition is task-centred and derived from the intended use of the data consumer. According to this objective and data-centred view, completeness is defined as all values for a certain variable are recorded[31]. From these two major definitions, we can observe two components that are vital to completeness: content and structure. Therefore high-completeness data is achieved when data content and structure are both at a high-quality level. That means high-completeness data must contain no NULL values and carry the full meanings for its intended task. In order to achieve high data completeness, extensive work has been done on statistical approaches to missing value imputation [21,22]. The difference to our problem is missing value imputation only deals with NULL values in record. In our proposal, the main and novel concern is focused on missing links.

There has been a number of work on detecting links in database research. Within the data profiling community, [23] and [25] have tackled the problem of foreign key discovery, which is also referred as inter-table analysis. In the semantic web community, tools such as R2O [26] and DB2OWL [27] are developed to convert relational databases to ontologies, in order to exploit explicit and implicit links in the database. Furthermore, the idea of computing similarities of objects based on links has also been used on the semantic web [28]. This idea is also similar to the vector space model used to describe the semantics of words in natural language processing. Graph and tree similarity are also well-studied problems with applications in XML document retrieval [29] and in graph databases [30].

In recent work [3] in which some authors of this proposal were involved, it was investigated how to obtain completeness information from process descriptions. The work formalized processes manipulating objects in the real-world and recording modifications in the database, and then presented algorithms to verify whether the processes guarantee that data at a certain point will be completely recorded in the database. The assumption in this work is that processes follow strict flows. Regarding the assessment of the completeness of complex objects, to the best of our knowledge, we are not aware of any previous work.

Once we have found the incomplete data, information about why an object is not complete is then needed. beside labelling the data as as incomplete, this is to provide further reason about why and how the data got incomplete. It allows data providers to append the required information, and also enables data consumers to know which aspects of the data are possibly with problems. Therefore, the explanation is important and relevant to this project. An explanation can be considered as a piece of information that is presented in a communication process to serve different goals, such as exposing the reasoning behind an algorithm [17]. Explaining results of machine learning algorithms is a known problem in various domains, for instance in recommender system research [24, 17]. Up to now, there are limited research on explanation about the potential problems and reasonings behind the data quality assessment. In this proposal, we intend to tackle this challenge and conduct research in explaining data incompleteness.

## 3. Objectives and Expected Outcome

We plan to verify the hypothesis that machine learning provides a viable alternative to the manual creation of a validation logic. Our objectives are therefore threefold:

1. To find machine learning techniques for the assessment of the completeness of complex data objects.
2. To identify features of complex objects that are useful for predicting the completeness of complex data objects.
3. To develop techniques to explain why objects have been classified as incomplete.

Regarding the outcomes of our project, we differentiate between the minimal outcome, the realistic outcome and the optimal outcome.

*Minimally*, we expect to achieve the following:

1. An evaluation of how well simple algorithms, e.g. k-nearest neighbour based on Hamming distance and weighted decreasing Hamming distance work for predicting incompleteness
2. Feature extraction methods that take into account direct links in the database
3. Simple explanation techniques based on the neighbour information used in k-nearest neighbour techniques
4. Two java libraries implementing these techniques
5. An online demo for DBpedia showing attribute extraction and completeness prediction

*Realistically*, we expect the following additional outcome:

- The evaluation of advanced algorithms such as SVMs
- Advanced features e.g. based on 2nd order links in the database, attribute entropy or string similarity
- Advanced explanation techniques
- An evaluation of our techniques using the project database of the department of hydraulic engineering

*Optimally*, we would also obtain machine learning techniques that have both a good accuracy, and a good explainability of their results. Furthermore, optimally, we would expect to design feature extraction techniques that scale well for big databases.

## 4. Methodology

Our approach is based on signatures, which are feature vectors containing as core features the occurrences of the IDs of objects in other positions in the database.

To illustrate what signatures are, consider an example motivated by the project database of the department of hydraulic engineering. This database contains three tables, *Project, Document* and *Section*. The *Project* table lists projects together with their managers. The *Document* table contains documents, together with their type, and the project to which they belong. The *Section* table contains sections of projects, and links them to their corresponding project:

| *Project* | |
|---|---|
| *Name* | *Manager* |
| *P1* | *John* |
| *P2* | *NULL* |
| *P3* | *Mary* |

| *Document* | | |
|---|---|---|
| *Name* | *Type* | *Project* |
| *P1_bill.pdf* | *bill* | *P1* |
| *P1_contract.odp* | *contract* | *P1* |
| *P2_vertrag.pdf* | *contract* | *P2* |
| *P3_contractV1.docx* | *contract* | *P3* |

| *Section* | |
|---|---|
| *S_Code* | *Project* |
| *375* | *P1* |
| *863* | *P2* |
| *864* | *P2* |

Suppose we are interested in the completeness of the documentation of projects. Note that projects are complex objects, since information about them is split over all three tables. Using classical profiling techniques, it would be easy to find out that project P2 has no manager, since a NULL value appears for that attribute. But what can be said about the documents related to the projects, and about the sections?

**Approach Taken.** We propose a data-centric approach to tackle determine the completeness of complex objects. Our approach uses so-called signatures as attribute vectors that summarize object links, which in turn give a quick overview and allow the application of machine learning techniques.

For the toy database above, the object signatures could look as follows:

| *Proj.Name* | *Proj.Manager* | *Doc.Type=bill* | *Doc.Type=contract* | *Sec.Project* |
|---|---|---|---|---|
| *P1* | *1* | *1* | *1* | *1* |
| *P2* | *0* | *0* | *1* | *2* |
| *P3* | *1* | *0* | *1* | *0* |

These signatures are interpreted as follows: Each number indicates how often the project in that row occurs in the attribute mentioned in the column header. For instance, the value “2” in P2’s row for the column *Sec.Project* indicates that P2 appears twice as value in the *Project* column of the *Section* table.

In this toy example, one can manually quickly identify potential anomalies: P2 has no manager, P2 and P3 have no billing document, and P3 has no section. Not having a billing document may be ok if a project has just started, whereas not having a manager or a section may indicate a data quality issue. Subsequently, P1 might get labelled as complete, while P2 and P3 might get labelled as incomplete.

Regarding the explainability of results, we see two main approaches. Either we restrict ourselves to interpretable techniques such as nearest neighbour or decision trees, or whenever an object is found to be incomplete, we can try some more complete variants of it and see whether they get classified as complete. In that case, the user could then be supplied with the information that the current object is not complete, but which other states of the objects would be considered as complete.

## 5. Working Plan

We plan to validate our work using two use cases: The project database of the department of hydraulic engineering, and DBpedia. The project database is an excellent use case from our collaboration partner. DBpedia we plan to use for two reasons. First, because the project database contains sensitive data, we cannot make it publicly available. Second, since DBpedia is freely available, this means we can work with

it independently without requiring any input from collaboration partners. We intend to select a class of objects (e.g. countries or Hollywood actors), and use some data source such as the CIA factbook or IMDB to assess the completeness of the DBpedia information.

We split our work plan into five work packages. Packages 2, 3 and 4 are core research packages, while package 1 is needed for enabling work on package 3, and 1 and 5 are also important for dissemination.

### WP1: Creating labelled data sets

To perform any experiments on object modelling and machine learning, we need data. This package shall provide the required data for the subsequent work. We want to create two data sets: First, a dataset from the project database of the department of hydraulic engineering. Second, a dataset about the online database DBpedia. The latter we also intend to make public to the community. The labelling of the first dataset will be done by the department of hydraulic engineering, for the second one, we will use a crowd-sourcing platform such as Amazon's Mechanical Turk. To this package, Stefan Hellweger will contribute the data from the department of hydraulic engineering, and Florian Daniel his expertise on the design of crowd-sourcing tasks.

Task 1.1: Create a labelled dataset for DBpedia

Task 1.2: Create a labelled dataset for the project database of the hydraulic engineering department

### WP2: Extracting features about complex objects

To apply machine learning techniques for predicting the completeness of complex objects, features describing the objects are needed (remember that complex objects are objects, for which information is split across various tables). In this package we create the feature vectors, here called signatures, that describe features of complex objects. We will try to reuse existing solutions for inter-table analysis to find overlapping fields, but then extend these with information theoretic measures such as entropy in order to identify fields with relevant information. In the example above, finding the documents related to projects is easy (as the column *Doc.Project* contains project IDs). However, determining the attribute that best describes which role the document plays is not straightforward. In this case, the Type attribute is important, while the *Name* attribute is not. But in general, there could exist many more attributes, which could require aggregation or prefiltering of attributes. Also note that the links between objects generally may imply a graph structure, which cannot straightforwardly be mapped onto a linear attribute vector (signature). Thus, we will investigate how to best reduce graph structures to signatures. To this package, Felix Naumann and Divesh Srivastava will contribute their experience on foreign key discovery and data profiling.

Task 2.1: Create an algorithm for computing simple binary signatures over databases

Task 2.2: Develop a technique to mine the structure of complex objects

Task 2.3: Create a technique to map graph-structured complex objects into signatures

### WP3: Development and evaluation of machine learning techniques

In this package we apply and evaluate machine learning techniques to our labelled datasets. We first test common existing techniques such as regression and nearest neighbor. Then we test custom techniques that take into account diminishing weights for attributes based on entropy measures found in WP1, and possible other application-specific customizations. To this package, Flip Korn will contribute his experience with machine learning algorithms.

Task 3.1: Evaluate simple machine learning algorithms

Task 3.2: Evaluate advanced machine learning algorithms

### WP4: Explaining classifications

With the techniques from packages 2 and 3 one can find out whether an object is complete or not. To add value, however, also information about why an object is not complete, is needed.
An issue with advanced machine learning techniques such as support vector machines is that even if their performance is good, it is basically impossible to explain in a human understandable way why a certain record has received a certain classification. In our use cases however, it is crucial not only to label certain projects as incomplete, but also to provide explanations why, so that data providers can provide the required information, or consumers know about which aspects of the data to be skeptical. The goal in this work package is therefore to translate the results from the machine learning techniques back into simplified labels ("red/yellow/green") together with explanations. We will first try naive guessing techniques (see if by creating a more complete version of an object, the object becomes classified as complete), exploit techniques from the recommender systems domain [24], and experiment with rule mining systems in order to create explanations.

Task 4: Translate classifications into a validation logic

### WP5: Java Framework + demo

In order to raise interest in our work, and to have a greater impact, we aim to make implementations of our work public. This package consists of three components. First, an implementation of our signature computation technique, and second, an implementation of our machine learning framework, both to be done in Java. The third component will be an online demo exemplifying our approach on an online database such as DBpedia.

Task 5.1: Develop a java library for signature computation

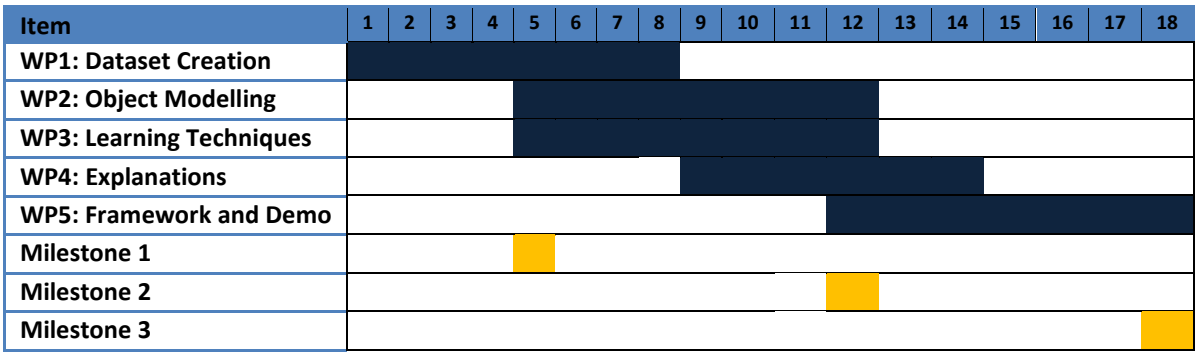
Task 5.2: Develop a java library for signature classification

Task 5.3: Develop an online demo for signature computation and classification

## Timeline and Milestones

There are three milestones. After 4 months, we plan to have the DBpedia dataset ready, as this lies entirely in our responsibility. After 12 months, we plan to have produced feature extraction techniques and classification techniques that allow to predict completeness. After 18 months, we plan to be able to generate explanations for classifications and to have produced a library for signature mining and classification.

Below is a time plan for the work packages and the milestones:




## 6. Dissemination and Continuation

Our dissemination strategy is twofold: On the conceptual side, we plan to present our results regarding packages 1, 3 and 4 at a major database conferences such as VLDB, SIGMOD or ICDE. On the implementational side, we plan to publish our demo implementation (Task 5.3) at a similar venue. Besides, we also want to make the labelled DBpedia dataset (Task 1.1) and the Java frameworks for signature mining and classification (Task 5.2 and Task 5.3) available online.

Strategically, the project has the aim to create links with the data profiling community at the European level to form a consortium that can apply for EU funding. In addition, we will use the outcomes of the project to apply for industry funding such as the Google Faculty Award.

We expect that an interesting line to continue this work would be to explore how process mining techniques can be used further refine the completeness required for an object based on its state. Process mining allows to determine the various stages that data objects may pass during their life. Depending on the state of an object, the required data might vary: For a project that has just started, it might be okay to not have any reports attached, while for a finished project, it might be crucial to have a final report attached. Thus, taking into account the state of an object might yield more precise classifications.

## 7. Budget

We request  a total budget of 50.000€, of which 40.000€ are personnel costs (items 1 and 2) and 10.000€ are travel costs (items 3 and 4):

1. A research assistant (AR) with previous experience in developing database-related software, to be hired for 15 months (38000€), who should work primarily on the implementation of feature extraction and explanation techniques.
2. Budget for crowd-sourced data labelling (2000€), to be used to label signatures of objects on DBpedia, e.g. using Amazon's Mechanical Turk.
3. Collaboration expenses (4000€), to be used to visit one of our collaboration partners and to invite one
4. Dissemination costs (6000€), to be used for the presentation of the work at two conferences such as VLDB, SIGMOD and VLDB, and at one workshops at such a conference (2000€ each)

## 8. Team

There will be four researchers from the Free University of Bolzano (FUB) working on the project. In addition, a research associate (AR) will be hired from the 4th month on.
Stefan Hellweger from the department of hydraulic engineering will lead the creation of a labelled dataset for the project database, and will provide feedback on the research results.
Prof. Felix Naumann from the Hasso-Plattner-Institut Potsdam, Florian Daniel from the University of Trento, Divesh Srivastava from AT&T Labs-Research and Flip Korn from Google Research will participate in the research as external academic partners.
All the team members contribute expertise and know-how to the project. In the past, they have achieved research and practical results on different aspects of the questions addressed in TaDaQua, which will now be brought together.

**Simon Razniewski** is an assistant professor at the Free University of Bozen-Bolzano. In his PhD research, for which he has received the best PhD student award in 2013, he has investigated the reasoning about completeness information in various context such as relational databases, business processes, spatial databases and on the semantic web. His work has been published at VLDB, CIKM, BPM, SIGSPATIAL and SIGMOD. He has industrial experience with machine learning techniques from internships at AT&T Labs and Globalfoundries Inc.

**Mouzhi Ge** is an assistant professor at the Free University of Bozen-Bolzano. His current research is focused on Recommender System and Data Quality Management. In the past, he has published more than 20 papers on Data Quality Profiling, Assessment and Management in a variety of international conferences and journals. Besides, he has hands-on experience on developing data quality tools such as plugins for Dataflux and Informatica Powercenter.

**Werner Nutt** is a full professor at the Free University of Bozen-Bolzano, working on formal aspects of data quality and process governance. In the past, he has researched probabilistic databases and data completeness.

**Ognjen Savkovic** is a fourth-year PhD student at the Free University of Bozen-Bolzano, working on data quality discovery over business processes. He has extensive experience with the development of systems for data quality assessment [7, 8].

**Stefan Hellweger** is the IT system developer and administrator of the department of hydraulic engineering of the province of Bolzano. He creates and maintains the database, the user interfaces and the validation logic, which is used to notify data creators and consumers of possible missing fields and links. He implemented a traffic-light-style visualization on top of this validation. He has also research experience with user experience issues [9, 10].

**Felix Naumann** is a full professor at the Hasso-Plattner-Institute in Potsdam, Germany. He has worked on a range of topics concerning data profiling in the past, and already visited the FUB in 2014.

**Divesh Srivastava** is the head of Database Research at AT&T Labs-Research. He has extensive experience with Data Quality research.

**Flip Korn** is a researcher at Google Research. He is specialized in data management and engineering.

**Florian Daniel** is a research associate at the University of Trento specialized in web services and with experience in the management of crowdsourcing platforms.

## 9. Selected Relevant Publications from the Team


[1] Identifying the Extent of Completeness of Query Answers over Partially Complete Databases, Simon Razniewski, Flip Korn, Werner Nutt and Divesh Srivastava, SIGMOD, Melbourne, Australia, 2015

[2]: Adding Completeness Information to Query Answers over Spatial Data, Simon Razniewski and Werner Nutt, SIGSPATIAL, Dallas, USA, 2014

[3]: Verification of Query Completeness over Processes, Simon Razniewski, Marco Montali and Werner Nutt, International Conference on Business Process Management (BPM), Beijing, China, 2013

[4]: Completeness Statements about RDF Data Sources and Their Use for Query Answering, Fariz Darari, Werner Nutt, Giuseppe Pirro and Simon Razniewski, Int. Semantic Web Conference (ISWC), Australia, 2013

[5]: Completeness of Queries over SQL Databases, Werner Nutt and Simon Razniewski, Conference on Information and Knowledge Management (CIKM), Maui, USA, 2012

[6]: Completeness of Queries over Incomplete Databases, Simon Razniewski and Werner Nutt, Int. Conference on Very Large Databases (VLDB), Seattle, USA, 2011

[7]: Ognjen Savkovic, Paramita Mirza, Alex Tomasi, Werner Nutt: Complete Approximations of Incomplete Queries. VLDB 2013 (Demo paper)

[8]: Ognjen Savkovic, Paramita Mirza, Sergey Paramonov, Werner Nutt: MAGIK: managing completeness of data, CIKM 2012 (Demo paper)

[9]: Hellweger, Stefan, Xiaofeng Wang, and Pekka Abrahamsson. "The Contemporary Understanding of User Experience in Practice." http://dx.doi.org/10.6084/m9.figshare.1319577 (2015).

[10]: Hellweger, Stefan, and Xiaofeng Wang. "What is User Experience Really: towards a UX Conceptual Framework." http://dx.doi.org/10.6084/m9.figshare.1319576 (2015).

[11] Ge M., Helfert M., Impact of Information Quality on Supply Chain Decisions, Journal of Computer Information Systems, Vol. 53, No. 4, 2013.

[12] Ge M., Helfert M., Cost and Value Management for Data Quality, Handbook on Data Quality - Research and Practice, Sadiq, Shazia (Ed.) pp 75-92. 2013.

[13] Ge M., Helfert M., Jannach D., Information Quality Assessment: Validating Measurement Dimensions and Process, 19th European Conference on Information Systems, Helsinki, Finland, 2011.

[14] Ge M., Helfert M., Effects of Information Quality on Inventory Management. International Journal of Information Quality, Vol. 2, No. 2, pp 176-191, 2008.

[15] Ge M., Helfert M, Data and Information Quality Assessment in Information Manufacturing System, 11th International Conference on Business Information Systems, Innsbruck, Austria, 5-7 May 2008.

[16] Ge M. and Helfert M., Develop a Research Agenda: A Review of Information Quality Research, 12th International Conference on Information Quality, MIT, USA. November 9-11, 2007.

[17] Gedikli F., Jannach D., Ge M., How should I explain? A comparison of different explanation types for recommender systems, International Journal of Human-Computer Studies, Volume 72, Issue 4, pp 367–382, 2014

## 10. References

[18]: L. Berti-Equille, T. Dasu, and D. Srivastava. Discovery of complex glitch patterns: A novel approach to quantitative data cleaning. In ICDE, pages 733–744. IEEE, 2011.

[19]: J. M. Hellerstein. Quantitative data cleaning for large databases. United Nations Economic Commission for Europe (UNECE), 2008

[20]: M. Yakout, L. Berti-Équille, and A. K. Elmagarmid. Don't be scared: use scalable automatic repairing with maximal likelihood and bounded changes. In SIGMOD, pages 553–564. ACM, 2013

[21]: P. Royston. Multiple imputation of missing values. Stata Journal, 4:227–241, 2004

[22]: A. C. Acock. Working with missing values. Journal of Marriage and Family, 67(4):1012–1028, 2005.

[23]: M. Zhang, M. Hadjieleftheriou, B. C. Ooi, C. M. Procopiuc, and D. Srivastava. 2010. On multi-column foreign key discovery. *VLDB Endow*

[24]: M. Zanker. The influence of knowledgeable explanations on users' perception of a recommender system. RecSys 2012

[25]: J. Bauckmann, U. Leser, F. Naumann, and V. Tietz. Efficiently detecting inclusion dependencies. In ICDE, pages 1448--1450, 2007.

[26]: Barrasa Rodríguez, J., Corcho, Ó., & Gómez-Pérez, A. (2004). R2O, an extensible and semantically based database-to-ontology mapping language.

[27]: Cullot, N., Ghawi, R., & Yétongnon, K. (2007, June). DB2OWL: A Tool for Automatic Database-to-Ontology Mapping. In SEBD (Vol. 7, pp. 491-494).

[28]: Giuseppe Pirrò: REWOrD: Semantic Relatedness in the Web of Data. AAAI 2012

[29]: Pawlik, M., & Augsten, N. (2011). RTED: a robust algorithm for the tree edit distance. Proceedings of the VLDB Endowment, 5(4), 334-345.

[30]: Yan, X., Yu, P. S., & Han, J. (2005, June). Substructure similarity search in graph databases. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data (pp. 766-777). ACM.

[31]: Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. 2009. Methodologies for data quality assessment and improvement. *ACM Comput. Surv.* 41, 3, 2009