DISS. ETH NO. 20770

A Requirement-Oriented Data Quality Model and Framework of a Food Composition Database System

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

KARL PHILIPPE PRESSER

Dipl. Informatik-Ing. ETH

December 3, 1976

citizen of Zurich, ZH, Switzerland

accepted on the recommendation of

Prof. Dr. M. Norrie, examiner
Prof. Dr. H. Hinterberger, co-examiner
Prof. Dr. C. Chan, co-examiner

2012

Abstract

Data quality is an important issue for data acquisition, maintenance and retrieval in an information system because data users demand to obtain best quality data. Data quality research is a relatively young field and is dominated by management scientists who propose solutions for classification and organisational process improvement. Concepts concerning the implementation in information systems are hardly available. For this reason, every information system architect and programmer has to design and implement their own solutions to deal with data quality problems.

To address these problems, we propose a definition of data quality, a data quality model and a data quality framework for IT professionals and users as a contribution to the interdisciplinary field of data quality. As data quality is influenced by multiple user requirements, we put data quality requirements centre stage and named our model and framework Requirement-Oriented Data Quality model (RODQ model) and Requirement-Oriented Data Quality framework (RODQ framework).

The objective of the RODQ model is to help information system architects and users to specify data quality requirements, measure and aggregate them to a total value of data quality. Total quality values can be used to gain an overview of data's quality and to make statements about the quality of considered data. The RODQ model has an RODQ schema with which data quality requirements can graphically be modelled. The RODQ model also includes a classification of data quality requirements according to characteristics that influence their implementation in an information system and includes a simplified assessment schema that can be applied to every data quality requirement.

The RODQ framework is a conceptual framework that proposes implementation concepts to get a comprehensive data quality framework in an information system. To the RODQ framework belongs the concept of data quality prevention with the main idea to evaluate data against RODQ schemas before they are stored in the database and hence to prevent data of low quality entering the information system. This idea is not only designed for graphical user interfaces but also for automated machine-to-machine interfaces. For this reason, the concept of data quality controller is defined as the single quality management place where data quality requirements are defined and evaluated. A further concept is the data quality analysis, which has the objective to support data quality sustainability and which is similar to data quality prevention. The difference is that data quality analysis operates on existing data in the database and in addition is able to analyse sets of data records. The idea behind this concept is also to present quality information to users while they are working with the system. Consequently, not only the data maintainer can use data quality information but also every data user. An additional aim of data quality prevention and data quality analysis is to make the data maintainer and data user quality conscious for the data by providing support at different places in an information system. Furthermore, the long-term analysis of data and their quality is a specific requirement for

scientific data. In addition to the existing concept of data versioning, we present the concept of data quality versioning that performs data quality measurements periodically. The resulting amount of quality values can become problematic over time and we also propose a solution to manage this amount of values.

The showcase application is our implementation of a food composition database management system, called FoodCASE, in which food items and nutrient values are managed by food scientists. Our main quality focus is therefore on empirical and scientific data collections. We finally present the implementation of all concepts in FoodCASE and provide a retrospective and critical discussion about our concepts.

Zusammenfassung

Datenqualität ist ein wichtiger Aspekt bei der Erfassung, Wartung und Abfrage von Daten in einem Informationssystem, weil Datennutzer die beste Qualität erwarten. Das Forschungsgebiet der Datenqualität ist ein relativ junges Gebiet und wird von Wirtschaftswissenschaftlern dominiert, die Lösungen für die Klassifizierungen von Qualitätsaspekten und organisatorische Prozessverbesserungen vorschlagen. Konzepte zur Umsetzung dieser Lösungen in Informationssysteme sind kaum vorhanden. Aus diesem Grund muss jeder Architekt und Programmierer von einem Informationssystem das Datenqualitätsmanagement selber konzipieren und realisieren.

Um diese Probleme anzugehen, definieren wir den Begriff der Datenqualität auf eine neue Weise. Wir schlagen auch ein Datenqualitätsmodell und ein Datenqualitätsframework vor, das sowohl den Entwickler als auch den Benutzern unterstützen soll. Weil Datenqualität durch mehrere Benutzeranforderungen beeinflusst wird, stellen wir diese Anforderungen in den Mittelpunkt und benennen unser Modell und Framework Requirement-Oriented Data Quality-Modell (RODQ Model) und Requirement-Oriented Data Quality Framework (RODQ Framework).

Das Ziel des RODQ Model ist es, Architekten und Anwendern von Informationssystem zu helfen, Anforderungen an die Datenqualität zu formulieren, diese zu messen und zu einem Gesamtwert an Datenqualität zusammenzufassen. Die zusammengefassten Qualitätswerte können verwendet werden, um einen Überblick über die Qualität von Daten zu gewinnen und um Qualitätsaussagen über die betrachteten Daten machen zu können. Das RODQ Model verfügt über ein RODQ Schema mit denen Anforderungen an die Datenqualität grafisch modelliert werden. Das RODQ Model enthält auch eine Klassifizierung von Qualitätsanforderungen nach Merkmalen, die die Umsetzung in einem Informationssystem beeinflussen. Das RODQ Schema beinhaltet auch ein vereinfachtes Bewertungsschema, das auf jede Datenqualitätsanforderung angewendet werden kann.

Das RODQ Framework ist ein konzeptionelles Framework, das Implementierungskonzepte enthält um ein möglichst umfassendes Qualitätshandling in einem Informationssystem zu erhalten. Zum RODQ Framework gehört das Konzept der Prävention mit der Hauptidee neue Daten gegen das RODQ Schema zu prüfen, und zwar bevor sie in der Datenbank gespeichert werden umso die Eingabe von Daten minderer Qualität zu vermeiden. Diese Idee ist nicht nur für grafische Benutzeroberflächen, sondern auch für automatisierte Maschine-zu-Maschine-Schnittstellen zweckmässig. Aus diesem Grund wird der Begriff des Datenqualität-Controllers eingeführt der einen zentrale Einheit beschreibt, wo Qualitätsanforderungen an Daten definiert und ausgewertet werden. Ein weiteres Konzept ist die Datenqualitätsanalyse, die das Ziel hat die Nachhaltigkeit der Datenqualität zu unterstützen und Ähnlichkeiten zur Prävention aufweist. Der Unterschied ist, dass die Datenqualitätsanalyse auf vorhandenen Daten in der Datenbank arbeitet und zusätzlich in der Lage ist, mehrere Datensätzen zusammen zu analysieren. Eine weitere Idee hinter diesem Konzept ist die Mitteilung von Qualitätsinformation für den Benutzer

während er im System arbeitet. Folglich können nicht nur die Personen Qualitätsinformationen sehen, die die Daten pflegen sondern auch Personen, die die Daten nutzen. Ein weiteres Ziel der Prävention und der Datenqualitätsanalyse ist es, das Qualitätsbewusstsein von Benutzern zu stärken und zu unterstützen indem an verschiedenen Orten in einem Informationssystem Qualitätsinformationen angezeigt werden. Darüber hinaus ist die Langzeitanalyse von Daten und deren Qualität eine spezifische Anforderung für wissenschaftliche Daten. Zusätzlich zum bestehenden Konzept der Versionierung von Daten, präsentieren wir das Konzept der Versionierung von Datenqualität, die Datenqualitätsmessungen in regelmäßigen Abständen durchführt und abspeichert. Die hieraus resultierende Menge an Qualitätswerten kann in seinem Umfang im Laufe der Zeit problematisch werden und wir schlagen eine Lösung vor wie diese Menge an Werten verwaltet werden können.

Die Implementierung von den oben erwähnten Konzepten haben wir im Informationssystem FoodCASE durchgeführt, mit dem Lebensmittel und deren Nährstoffzusammensetzungen verwaltet werden können. Der Schwerpunkt unserer Untersuchungen lag daher auf empirischen und wissenschaftlichen Datensammlungen. Am Ende dieser Arbeit präsentieren wir die Umsetzungsdetails aller Konzepte in FoodCASE und diskutieren unsere Konzepte retrospektiv und kritisch.

Acknowledgments

First of all, I would like to thank Hans Hinterberger, who gave me the chance to make this thesis in his group. The freedom he gave me for the research was as generous as his support for the Swiss food composition database. Also his input and feedback from numerous discussions were as advancing my work as his support in different projects around FoodCASE. The unlimited time he spend helping me to improve the thesis was more support than I could expect. The cooperation and communication was inspiring and open, which was the best premise to motivate me to go ahead with the work.

The reason why I start to thank my co-examiner is because he is the supervisor of this thesis. His retirement and my suboptimal time management forced us to change the examiner and co-examiner. Hans was therefore the supervisor with most scientific contributions.

After Hans' retirement Moira Norrie "inherited" me. I would specially thank Moira not only for the offered working place but also for her willing support she gave me for my work and her matter of course how she was and still is interested in the data quality topic. I would like to thank her providing me new views on scientific work and their publications and also for the support and uncomplicated procedure to acquire and perform new projects.

A special thank also goes to Paolo Colombani for endless hours of providing contextual input for food composition databases. I also learnt many scientific issues from him, not only in empirical sciences but also in computer science. And last but not least the discussion about data quality from a food compilers point of view brought fruitful new input for this thesis. It was also Paolo's merit that I was able to perform some research work in the EuroFIR projects and so I thank him also to open the doors to all these opportunities.

I also would like to thank the BAG, in particular Urs Stalder and Michael Bär, for their support in the joint project to introduce FoodCASE in Switzerland as the national food composition database. Their trust and openness gave us enough free space for own ideas to improve FoodCASE.

Also the team of Hans was always a backing during my work. A big thank goes to Kajetan Abt, who implemented the most parts of FoodCASE version 1.0 and hence prepared a research object, on which data quality investigations could be started. A thank goes also to Markus Dahinden, who was always open-minded for data quality or technical discussions. I also thank Barbara Scheuner for having an ear for my Ph.D. ups and downs and, being in the same situation, for being an excellent sparring partner for the final stage of the thesis. Lukas Fässler with his thesis experience, was a save guide for asking advise and to find a good way at different branches.

A very important backing during the time of the thesis was my wife Manuela, who organised our children and many other things to give me time and space. She always listened insightfully and was standing by my side. For all this, priceless thanks go to her.

I thank my parents for their support and for their guidance into the academic world. They basically laid the foundation that this thesis could be written.
I also thank all food compilers and persons participating in the FoodCASE project and other persons involved in EuroFIR, EuroFIR NEXUS and EuroFIR AISBL as well as around them. Through their help I was able to conduct the user study and got findings of this thesis. All these projects were the best setup to investigate data quality on food composition database systems.
I also would like to thank the whole GLOBIS team that because of the relatively closeness to my topic provide a lot of inputs for information systems and HCI. The discussion and feedbacks were always constructive, inspiring and broaden my horizon.
I also want to thank the BA&PM team and its head Christian Gassner of the University of Applied Sciences ZHAW to give me the opportunity to test some of my data quality concepts in their school administration software.
As it is impossible to thank all persons that accompanied me through the last 5 years of my thesis, I would like to thank all other persons that made contributions and gave support in any form during that time.

Table of Contents

1
Introduction

"Quality is free. It's not a gift, but it is free. What costs money are the unquality things - all actions that involve not doing jobs right the first time"

Crosby in [1]

The objective of the concepts “open data” and “open science data” is to make data publicly available. The idea goes back to the late fifties when during the International Geophysical Year (1957-1958) different data centres were installed and standards for descriptive metadata were established to regulate the exchange and use of this data. During the following years, many other scientific data collections have been made publicly available, which subsequently became valuable sources for many information systems. In 2007, the Organisation for Economic Co-operation and Development (OECD) published a recommendation that publicly funded research data should be made publicly available.

As the adoption rate for these publicly available data collections grows, the pressure to manage and publish information about the data’s quality increases correspondingly. Data quality information helps data producers to transparently show how data were generated and to increase the producer’s reputation. Data quality information also helps data consumers to find appropriate data for their intentioned use. The benefit for data consumers is not only valid for scientific publicly available data, but also for all kinds of data available on the Internet, for data collections in companies and for private data collections.

One of these publicly available scientific data collections is food composition data. A food composition database contains data about food items such as an apple or cheese and information about the nutrient values of these food items such as “Emmentaler cheese contains 33.8 grams of fat per 100 grams edible portion”. The data in a food composition database is typically collected over many years and from different sources such as laboratories and scientific publications. A person responsible for this task is called a food compiler. Science and also the public demand that data quality should be as high as possible. To identify data’s quality, requirements on the data must be evaluated. Such requirements can range from basic constraint checks such as “a nutrient value must be of type real” over data provenance based requirements such as “the laboratory must be accredited for the measurement method” to requirements of contextual nature such as “a food item needs to have at least the 4 nutrient values; energy, fat, protein and carbohydrate”. Therefore, data quality covers everything from basic constraint checks to sophisticated contextual requirements.

However, our experience of implementing the food composition database system FoodCASE for Switzerland showed that concepts for data quality assurance, except basic constraint checks, were absent. Although publications in the area of data quality research exist, as will be discussed in Chapter 2, implementation concepts or guidelines on how to support data producers, data maintainers and data consumers in the handling and management of data quality requirements in an information system are limited. Some implementation concepts exist but they only cover partial aspects, and a comprehensive design guideline such as a data quality framework is completely missing. As a consequence, existing software systems have implemented their own data quality functionalities. A frequently used functionality is to implement data checks on graphical user interfaces (GUIs) and to mark quality deficiencies of data by colouring the background of a mandatory field in red or annotating the field with a star. Once the data is in the database, a frequently used strategy is to generate reports to reveal data that have a certain quality deficiency. Nevertheless, as these solutions are usually tailored to particular problems, they are seldom applicable in other information systems and hence are not sufficient for general and comprehensive data quality handling in information systems. The only exception to the above statement can be found in postal address validation. Some data quality companies offer an advanced solution in that fuzzy logic is used to identify duplicate entries in an address list.

Hence, every software architect and every software programmer has to re-design and re-implement all necessary data quality functionalities for their information system. How much functionality was developed in parallel can only be estimated. The simple checks “does a mandatory field have a value” or “is a value of type Integer” are two small data checks that must be performed on the application level such as on a GUI or website but are probably re-implemented in every information system.

One reason for the situation above is certainly that many data checks are context dependent and impossible to apply in another information system. As an example, consider the data check

“a food item needs to have at least nutrient values for energy, fat, protein and carbohydrate”. Data checks are therefore implemented for a particular purpose and software architects and programmers do not bother about how such a data check can be slightly modified and applied in other information systems for similar forms of conditions.

A second reason is that data quality is a relatively young research area although data quality issues have been around since the first database systems. The first International Conference on Information and Data Quality was in 1996 and the first ACM Journal of Data and Information Quality (Volume 1, Issue 1) was published in 2009. Hence, the number of publications about data quality is relatively low compared to other established data management areas.

Although there are reasons for the lack of concepts and design guidelines, we are convinced that the objective of computer science in the area of data quality must be to provide concepts of automated support for data producers and consumers. An information system should automatically support users in every data manipulation step: from data input over data maintenance to data usage, the so-called data life cycle. A set of concepts including the entire data life cycle is a data quality framework that we propose in this work. Our goal is to show that the management of data checks can be made independent of the contextual meaning of a data check and hence we are able to get a general applicable data quality framework that could be used in many different contexts. The reason why such a data quality framework is necessary is to simplify the work of software architects and programmers when designing and implementing data quality functionalities in an information system. To understand why data quality is an important aspect, we have to think about the consequences of low data quality.

1.1 Consequences of Low Data Quality

Low data quality can cause serious problems, not only in science but also for governments, companies, and subsequently for the public. Particularly scientific data has to achieve an important goal: In science, data is used to develop and prove models and concepts that solve certain problems. If scientific results are based on low quality data, future research that is based on these results can produce incorrect insights. We have found several examples in the Swiss food composition database where information such as the origin, analysis method or age of data was missing. If Swiss food composition data comes from other countries or even continents, was not measured using accepted methods or is more than 10 years old, then this data has reduced reliability. If these results are used in science, governments and industry, the underlying low data quality can cause problems such as incorrect scientific findings, negative effects on governmental policies/recommendations and additional expenses. In that sense, science has a responsibility to produce high quality data.

The following points show six possible consequences of low data quality:

- *Negative effects of government policies and recommendations*

Government institutions make policies and recommendations that are based on data and want to be sure that the used data is reliable. If the public health department, for instance, conducts a food consumption survey to monitor consumers' behaviour and wants to formulate recommendations depending on the outcome, then low quality of data can lead, in extreme cases, to malnutrition of persons. As a consequence, new costs arise for health care, for the correction of the policy and their implementation. A famous example in the area of food composition science is the misplacement of the decimal point in the iron content value of spinach. Spinach was said to have ten times more iron than it actually has. The mistake was not harmful, but showed that such an error can survive over many years.

- *Additional expenses*

  A company that delivers faulty products and has to re-deliver, has to take additional actions that increase their costs. The reason for such a costly delivery can be low data quality in their ordering system such as incorrect product items, number of items or shipping address. English in [2] summarises: "The bottom line is that information quality problems hurt the bottom line". Some data quality problems such as the re-delivery of products can be measured in terms of costs, but other problems such as the loss of unsatisfied customers or missed opportunities can seldom be measured. Hence, the total costs for low data quality in general are hard to determine. Redman in [3] estimates 10 to 20% of a company's revenue is lost due to data quality problems and the Data Warehouse Institute estimates the costs for data quality problems of US companies to be 600 billion US dollars a year [4].
- *Additional data management effort*

  A number of steps must be taken to eliminate low quality data. First, data deficiencies must be identified. Second, the correct values for these data must be determined and, finally, the values need to be corrected. This additional effort causes not only additional expenses but also takes time that could be used for other tasks.
- *Missed opportunities*

  If low quality data is used to analyse a specific issue, it can happen that new findings and facts are not discovered and new opportunities of science and business development are missed. The progress of knowledge and industry can be decelerated or even prevented.
- Dissatisfied data users and stakeholders

  If data users encounter low quality data, their trust in the data decreases. This can lead to dissatisfaction of data users and, in the worst case, to disuse of the data. If a data producer loses customers and maybe also income, stakeholders will soon become dissatisfied too. The obvious consequences are reduced earnings and reduced funding which ultimately can put the existence of an institute at risk.
- Competitive disadvantage

In a highly competitive environment, low data quality can become a disadvantage. If data producers have to acquire new data users, high quality data and even indications about the quality bring additional benefit to users and is therefore an advantage. Also, the competition on market share can be influenced by data quality. In the course of our research it became apparent that quality can be seen as the second most important aspect of products after the price.

Although the consequences of low data quality are obvious and data workers attempt to counter low data quality, there are some challenges for this undertaking. In the following section, these challenges are listed as basis for the research questions addressed in this thesis.

1.2 Research Questions of this Thesis

Data quality is often considered to be equal to accuracy in the sense that the difference between the real value and the stored value determines the degree of data quality. This holds true up to a certain point. If we consider empirical sciences such as food science, medicine, physics, biology, environmental science, chemistry, psychology and education, real values are often not known and are only approximately measurable. Consider the Vitamin C value of an apple: Food scientists can measure this nutrient using the most current analysis method. But how close the measured value is to the real value is unknown. Whether the apple that we are eating have the same Vitamin C content as the measured value is also unknown. In this case, accuracy is less important because the real value is not known and other data quality issues such as reliability, timeliness and absence of missing values become more important. Identifying these issues for a specific context is a first challenge when dealing with data quality.

Another problem is how users perceive and rate data quality. During their work, users of an information system can encounter incorrect and inconsistent data. In the course of our research it became apparent that if users encounter several different data deficiencies, they soon suspect the data to be of low quality. The problem is that single examples of low quality data exist, but a comprehensive analysis of data and its quality is not performed. To overcome this situation a systematic measurement and analysis of the quality of existing data in the information system is necessary which leads to the challenge of finding appropriate measurement and analysis methods for data quality.

Another challenge is that data quality is often not considered as a long-term activity: If an institution or company decides to carry out a data quality activity, it leads mostly to a one-time correction project. This means that data quality deficiencies are searched for and users have to correct faulty data records. Once the quality improvement project is finished, the data quality decreases to the same level as before if the source problems are not solved and other prevention activities are not taken. The challenge is to find concepts that can help manage the quality of data as a continuous activity.

A further challenge that was mentioned earlier is that software architects and programmers have to implement their own data quality functionality because some of them are context specific. The challenge is to find a context-independent data quality framework that enables data quality to be automatically measured, managed and customised for individual users.
From these challenges the following research questions for this thesis are derived:

1. What do users mean by data quality?
2. How can data quality be modelled?
3. How can data quality be measured in an information system?
4. How can data quality be managed in an information system?
5. How can data quality assessment be automatically supported in an information system?
6. How can data quality improvement be automatically supported in an information system?

In the research literature, data quality is often categorised by so-called dimensions. Dimensions are subjective descriptions of data quality aspects such as accuracy or believability. These descriptions bring some problems for database and application designers in that they are defined on an abstract and not distinct level and sometimes overlap, as we will see in Chapter 2. If we want to model data quality and provide a data quality framework, we need a precise definition of data quality and data quality requirements. The first question covers these definitions.
To be able to manage and measure data quality requirements, one needs a formalisation that expresses these requirements in a manner that it becomes possible to apply mathematics, statistics, and automated heuristics to them. Some requirements are generally valid, for example that the postal code and address must match, but others are strongly context dependent, as for instance, a food item should contain at least four nutrient values. Our objectives have been to get away from these context-dependencies and to find a generally valid formalism of data quality requirements that are abstracted to such a level that data quality requirements can be managed and measured. Research questions 2 to 4 cover these objectives.
The last two research questions concern the maintenance of data quality. If we understand the nature of data quality requirements, we might be able to quantify them and find a way to assess and even to improve data quality. So a database application could be enabled to automatically assess data quality requirements and communicate the results to users. The maintenance of data quality focuses on continuous data quality assurance.

1.3 Project FoodCASE and Research Approach

We have seen that data quality is an important issue for all types of data. The development of a database at a research university places the emphasis on scientific data such as food composition data. As a test bed for our research work we work at a practical level with the Swiss

food composition database. The Swiss food composition database is the foundation of the information system FoodCASE. It is a key factor in our research work for the following reasons:
The first Swiss food composition database application at ETH Zurich was built in 1992. Since then, members of the Department of Computer Science have had a number of projects with scientists from the Department of Agricultural and Food Science to implement an information system for food composition data or to add functionalities to the existing information system. The Swiss Federal Office of Public Health has funded various projects to maintain and augment food composition data as well as to implement information systems. One of the projects was SwissFIR where we had the opportunity to implement a new food composition information system, called FoodCASE, with a strong research focus on data quality. FoodCASE turned out to be an excellent research object and the cooperation in the FoodCASE project with different food scientists across Europe helped to identify data quality requirements for food composition data from a broad user base.
We therefore used an application-driven approach for our research work in that we first implemented FoodCASE and then investigated concepts to manage data quality. The different data quality requirements as well as FoodCASE lead to a holistic approach with two different peculiarities: The approach is holistic because the requirements from different fields such as data provenance, data integration and context must be abstracted to a general form of requirement so that all requirements can be handled in the same way. The approach is also holistic in that the management of the requirements must cover all steps of data processing, from data input over further processing and maintenance until the dissemination of data. Throughout this thesis, food composition data is taken as examples of scientific data and the proofs for our concepts will be presented using examples from FoodCASE.

1.4 Contribution of this Thesis

Contrary to most publications in the field of data quality, which come from management science, this thesis examines data quality from a computer science perspective, in particular, information system design and implementation. The contributions of this work follow directly from the research questions:

- *A definition of data quality for data collections*
  We performed two user studies to understand what food composition scientists mean when talking about data quality. We present definitions for different terms that belong to data quality. We also present a set of standard requirements for the reliability of empirical scientific data.
- *Requirement-Oriented Data Quality model (RODQ model)*
  The RODQ model defines inherent properties of data quality requirements that are essential for their integration in an information system. It contains also a basic data quality assessment approach that is generally applicable. We also present an RODQ

model schema that is used to define how a total data quality score can be achieved from multiple data quality requirements.

- *Requirement-Oriented Data Quality Framework (RODQ framework)*
  The RODQ framework is a set of concepts for a software framework that can be implemented in an information system. The framework contains concepts for data quality management, data quality analysis and data quality sustainability and is based on the RODQ model.
- *A reference implementation with a scientific database application*
  We present the implementation results of the RODQ framework in the information system FoodCASE. We also present a field study where we investigated missing values in a school administration software.

1.5 Structure of this Thesis

Chapter 2 describes existing concepts that can be used to model and implement data quality. We also identify data quality issues that are related to our data quality model and framework but that, to our current knowledge of the state of the art, have not been addressed by the research community.

In Chapter 3, two user studies and their evaluations are presented where the users' notions of data quality have been investigated. The first user study investigates what data quality issues users mention when they have to feedback a software requirement specification. In the second user study, we used a questionnaire to investigate the importance of concrete data quality requirements and functionalities for food composition data with different user groups. Based on the results, we summarise issues related to data quality, based on which we provide definitions for the terms data quality, data quality requirement and data quality indicator.

Chapter 4 shows that traditional data models such as the ER model are not sufficient to satisfy all data quality requirements. We therefore present a conceptual model, the RODQ model, to describe data quality requirements and to model how data quality of an entity is determined. The model differentiates two assessment approaches: One approach is a heuristic with which database attributes and relations can be assessed while the second approach is a heuristic to calculate a summary of multiple data quality requirements. Both assessment heuristics are context independent and therefore generally valid. Finally, we present the schema definition of our data quality model. We also show that our schema definition can be combined with existing schema definitions such as ER schemas or UML schemas, or can be used as a stand-alone schema.

In Chapter 5, we present a data quality framework, the RODQ framework, which contains multiple data quality concepts for an information system. The concepts serve as guidelines to implement a software framework for the management of data quality in an information system. One concept concentrates on input validation that is aimed at improving the quality entry level

for new or modified data. Two other concepts are data quality evaluation and data quality analysis. Data quality evaluation is the concept that enables a context-independent evaluation of data quality requirements. The data quality analysis concept shows how data quality can be measured on an individual basis, how data quality deficiencies can be identified and how long-term behaviour can be studied.

In the first part of Chapter 6, the design and implementation of the RODQ model schema and the RODQ framework in the software FoodCASE are presented. In the second part, we present the results of a field study on missing values as well as a user study that we performed in cooperation with the University of Applied Science ZHAW in Winterthur.

We conclude the thesis in Chapter 7 with a critical discussion of advantages and disadvantages of the presented model and framework. In addition, an outlook on topics for further research is presented.

2

Background and Related Research

In this chapter, topics of the data quality research area related to our data quality model and data quality framework are presented. We start in Section 2.1 by discussing the relationship of our work to other research areas. Section 2.2 continues with the introduction of food composition and the Swiss food composition database to understand the context of our application-driven research. We then present different data quality problems that we were faced with in the Swiss food composition database. The problems help to identify different challenges concerning data quality and help to identify important data quality research topics, to which definitions of data quality terms, data quality modelling, data quality assessment, and other data quality issues for food composition data belong. Existing contributions to these topics are then summarised and discussed. Existing definitions of data quality terms are presented in Section 2.3. Section 2.4 presents existing work for data quality modelling, Section 2.5 presents existing work for data quality assessment, and Section 2.6 summarises related data quality work for food composition data. Finally, topics without existing publications are listed in Section 2.7 where the necessity for a data quality assessment model and a data quality framework is argued.

2.1 Positioning of this Thesis

The focus of this thesis is on data quality requirements, their characteristics and their implementation in an information system for food composition data. This work belongs mainly to the research area of data quality. As data quality has some contact points to other research topics, we will first describe where our work is situated and how it is distinguishable from quality management and from other information systems' topics such as data provenance, data integration, constraint management, data modelling as well as software engineering.

Quality management is a management research area which investigates how the production of a product is properly planned, performed, assured and controlled. The goal is to keep the quality of a product as high as possible measured by the requirements of customers and stakeholders. The research area of data quality started from quality management as data was regarded as a product and it was realised that concepts from quality management can also be applied to data. Some examples of these concepts are the clarification of user requirements, involvement of participating persons and continual processing improvement. Our work is also dealing with user requirements and proposes how these requirements to data can be formalised and measured.

Our work has a contact point to the research area of data provenance in that information about the provenance of data is used to determine data's quality. If a requirement such as "the institution that measured the nutrient value must be known" or "the laboratory must be accredited for the measurement method" can be evaluated with a positive answer, then the quality of a data record increases. If no information for these requirements exists, then the quality is decreased. For our work, it is therefore not important how data provenance information can be provided along with data records. Our RODQ model "only" provides a special form to define data quality requirements and to evaluate data quality regarding multiple requirements. Consider the following example with food composition data: researchers collect original publications of nutrient values and manage all meta-information such as laboratory or measurement method in the food composition database. If this information is available in the database, it is the domain experts' task to define requirements and to define how data quality can be determined. Using the RODQ model, the researcher has a tool to support these tasks.

Data quality does not only consist of data provenance requirements but also of data integration problems such as "no duplicate food items should exist" or contextual requirements such as "nutrient values should not be older than 10 years". The contact point of our work to the research area of data integration is that we again consider only the requirements for and the determination of data quality. How the detection, record linkage or removal of duplicates are solved, is a topic of data integration and comes into play when a programmer has to implement the data quality requirements. However, to specify and model how data quality is evaluated, the implementation details are not relevant. The same argument holds true for basic constraint management.

The contact point to data modelling is larger. As we will see in Chapter 4, the RODQ model includes attributes and entities in order to define data quality requirements. An established data model can simplify the work when data quality must be modelled because attributes and entities are known and can be used in the RODQ model. The RODQ model also provides concepts for attributes and entities and can be used as a standalone model.

Another contact point of our work is to the wide area of information systems and software engineering. Our work requires that a database exists together with at least one application through which data comes into the database. Data quality is evaluated on new and existing data and hence our proposed RODQ model and RODQ framework are designed to be used in an information system. An information system can have multiple entry points for data. This can be GUIs for user inputs but also machine-to-machine interfaces to exchange data with other information systems. Data quality must be evaluated on all these interfaces. Hence the RODQ framework proposes a classical software engineering concept, namely a design pattern, for a central data quality requirement check unit, see Section 5.3, which is connected to all these different data entry points.

2.2 Background and Open Challenges of our Work

A food composition database contains food items and nutrient values where the food items can be naturally grown such as apples and bananas or be processed food items such as brands or recipes. Ideally, food items and values are documented, which means that other meta-information values for the foods and their nutrients are provided. Classification of food items, measurement method information, food processing information such as preparation or cooking method or laboratory information are just a few examples. It can happen for example that there are three Vitamin C values for an apple from three different laboratory analyses. As these data are publicly available, users would get confused if they find three different nutrient values on the website. Hence, the food compiler has the task to aggregate the three values to get a single value. The food compiler has the possibility to choose one of the values or build a weighted sum of these three values. Important from a scientific point of view is that all data processing is digitally documented and can be retraced. Recipes are handled by a second step of data processing. Recipes are seldom analysed in laboratories and are instead typically calculated from their ingredients. In the calculation, the preparation method or the cooking method must be included as some of the nutrients gain or lose on values when they are heated or boiled in water. Again, the documentation of the calculation procedure is important.

In 2006, the Swiss food composition data was made publicly available on the Internet and in 2007, our project started to build a new food composition information system (FCDBMS) for Switzerland. The new system was implemented according to the standards defined by EuroFIR [5] [6], see Section 2.5, that define entities and attributes for a food composition database, different food composition thesauri, an XML template to exchange food composition data

between two systems and a reliability assessment for food composition data. Food composition data was then migrated from the old FCDBMS to the new FCDBMS.

During the project, we were faced with the challenge to manage data quality in order to deal with different data quality problems such as reported data quality deficiencies, unsatisfied EuroFIR constraints and suspected low data quality, which were reported by Internet users of the Swiss food composition database and food compilers. For instance, the sums of all fatty acids were bigger than the indicated values for total fat, the sums of all indicated nutrient values per 100 grams was not in an acceptable range of 98 grams to 102 grams and some indicated energy values did not correspond to the energy values calculated from the nutrient values.

The EuroFIR standard defines some constraints that the old data did not satisfy such as mandatory fields for English food names and for the acquisition type of food composition data. In addition, Swiss food compilers suspected the existing data to be of low data quality according to the EuroFIR standards. It was only suspected because the evaluation of about 38'000 data records each with 34 EuroFIR criteria would take much time and was not done until the end of our project.

The obvious approach to use SQL scripts to find data quality deficiencies turned out to be not the most efficient solution because

a) The data is already in the database so it is rather a data cleansing approach than a prevention approach.
b) SQL scripts are beyond the typical users' capabilities making them dependent on IT persons and their availability.
c) Users and food compilers are not only interested in the quality rating of a single food composition value but also in gaining an overview of quality over all data records. Visualisation tools to simplify such overviews are not common features of current SQL querying tools.

A more appropriate approach is to use a data quality framework, implemented in the information system to support food compilers in the tasks of entering and maintaining high quality data. An appropriate framework meets the following goals:

a) Provide a simplified data quality rating for laypersons and provide a detailed data quality rating for professionals.
b) Provide food compilers with functionality to analyse data during and after data input so that deficiencies can be identified.
c) Provide an extensible framework so that future requirements such as the EuroFIR reliability rating, which was not finalised during our project, can be adjusted and unsatisfied constraints such as a mandatory field for the English name, are temporarily allowed.

The basis for such a data quality framework consists of the following aspects:

- A suitable definition of data quality terms.
  How can we judge data's quality based on these definitions?
- Appropriate data quality assessment schemas.
  How do we measure data quality?
- Timely analysis and correction of data quality deficiencies.
  How can we find and correct data quality deficiencies quickly and efficiently?
- Preventing data quality problems.
  How can we design our database application to avoid the same data quality problems from occurring again?
- Sustainable data quality management.
  How can we manage data quality assurance and improvement in the project and in the long-term?
- Applying a data management that includes the modelling of data quality.
  How can we model and document data quality in the context of an information system?

Each of these aspects has challenges that must be overcome in order to define a data quality model and framework. The remainder of this section identifies the challenges in more detail.

A suitable definition of data quality terms

Which data quality issues are applicable and how they are related is context dependent as the following two examples show: As mentioned in Chapter 1, empirical sciences, such as food composition, cannot determine the accuracy for a particular value because the real value is simply unknown. Therefore, other data quality issues, more precise data quality requirements or data quality dimensions, such as reliability, reputation, believability, missing values, timeliness and consistency are needed to determine the quality of data. In contrast, accuracy is the most important dimension for the postal address table of a food composition database where contacts of industry companies and different laboratories are collected. The accuracy of data providers' addresses is relevant for correspondence and for further enquiries. In this case, completeness, correctness and timeliness can be seen as sub-dimensions of accuracy.

Both examples have similar but also different data quality dimensions to their data in their specific context. The challenges are to find an appropriate set of data quality dimensions and to define the exact meaning of every single dimension so that users have a common understanding of the terms such as reliability, accuracy, completeness and correctness. These definitions will also help food compilers to communicate what data quality means.

Appropriate data quality assessment schemas

In order to publish a simplified quality rating for publicly available data, the different data quality dimensions must be assessed and summarised to provide a total data quality rating

according to the importance of each data quality dimension. The importance of a data quality dimension determines its influence on the total data quality rating of a data record. The importance can also be used to define the order in which data quality deficiencies must be improved. For instance, choosing the most important dimensions first, brings the biggest data quality increase because of their influence on the rating. This approach is often used when limiting factors such as time or funding exist.

As an example, consider again the food composition database. Food scientists want to publish that an apple has 5 mg of Vitamin C and want to provide the total data quality rating for that nutrient value. As an example, we assume that a food compiler defines that the total data quality rating value is composed of the data quality dimensions reliability, completeness and timeliness. The example reveals that for the three dimensions different assessment schemas must be defined. Finally, the outcomes of the three assessment schemas are summarised to the total data quality rating value using weighting factors, which are defined by a food compiler and which are based on the importance of the dimensions.

The challenges of the assessment step are to find appropriate assessments for every data quality dimension, to determine the importance of every data quality dimension in relation to the others and to find a summarising assessment schema.

Timely analysis and correction of data quality deficiencies

Finding all data quality deficiencies in order to correct them requires a data search and analysis that is based on the data quality requirements defined above.

As an example, reconsider food composition data. To implement a search of a data quality requirement, the requirement formulated in natural language must be translated into a programming language. Data will then be analysed to find data quality problems that can be corrected, either automatically or manually. The former is preferable as it is faster and less defective. In the Swiss food composition database, the nutrient value table has about 38’000 entries and 66 data quality requirements. That makes 2.5 million evaluations that must be performed.

The first challenge is the translation of data quality requirements from natural language to a programming language in a way that all possible attribute and entities values are covered. The second challenge is the analysis and correction of data quality deficiencies according to importance and costs. For instance, if the budget in a data quality improvement project is limited so that not all data quality requirements can be investigated and, if necessary, corrected, a subset with the most important data quality requirements must be chosen. The last challenge is data quality management and performance that arises when real-time analysis is requested in the application with several million data records and several hundred data quality requirements.

Prevention of data quality problems

We reconsider the example of the address table for industry contacts and laboratories in the food composition database. Food compilers have the possibility to go through every entry and check the accuracy of the address information. Doing this once is a valuable opportunity to bring data quality to a preferred level. However, the level of the address quality will decrease after the activity is completed and as soon as new data with low quality is added or information about contact persons is no longer valid.

The challenge is to design a functionality that prevents data with low quality from entering the database without preventing efficient data entry. The second challenge is to design a data quality monitoring functionality for existing data that alerts data compilers either as soon as data quality decreases under a certain threshold or on a continuous basis.

Sustainable data quality management

The management of persons and processes is also involved in a data quality assurance or improvement project. Although the focus of this thesis is not on management science, it is important to define, manage and control how data comes into an information system. Managers are interested in improving business processes and their organisation. They look at data as results or outputs of process chains in which persons or automatic functions generate data at a certain quality. Their view is that low data quality comes from poorly designed processes and/or from insufficient organisation. The analysis of data quality deficiencies can help a manager to identify the origins of data quality problems and enables them to initiate an improvement project.

It is obvious that the best data quality activity is useless when the way and the order in which data is acquired, entered and maintained in the database application is not defined and processes are not managed. Therefore, management and computer science must be seen as complementary players in a data quality improvement project.

Applying a data management that includes the modelling of data quality

If a food composition research institute decides to implement data quality functionalities such as an analysis module or a prevention module, a data quality model is helpful for the following reason. As seen above, data quality requirements can be numerous, can have different assessment schemas, and must be evaluated at data entry or with a data quality monitoring functionality. A model can help to gain an overview of these different properties and helps to represent what users mean by data quality in the real world.

The challenges are to abstract data quality requirements and their properties in such a general way that they can be used in a data quality model.

2.3 Definitions of Data Quality Terms

Having introduced the background and challenges of our work, we present existing definitions of the terms quality, data quality, information quality, data quality dimension and data quality framework. These definitions belong to the basics but the discussion is helpful to get the idea of these terms with regard to our own definitions of different data quality terms in Chapter 3.

2.3.1 Quality

Joseph M. Juran contributed to quality management with several books and publications. Among his ideas and concepts is the Juran trilogy that includes quality planning, quality improvement and quality control. Juran is often cited to give a short definition for quality as "fitness for use". But in [7] he clearly shows his antipathy by stating that it is unlikely that a short phrase can provide the depth of the meaning. He also makes, as one of the first authors, a certain differentiation in the meaning of quality [7]: Quality can be seen as related to product features that meet customers' requirements. Improving quality therefore means to provide more or better features. This improvement generally costs more but can also increase income. On the other hand, quality can be seen as freedom of deficiencies. From this point of view, "quality costs less" as no effort is needed for error corrections. As a consequence, fewer customers are dissatisfied and less effort is needed for customer care. The American Society for Quality [8] and the ISO 8402:1994 used Juran's differentiation in their definitions.

Another differentiation of the term quality is introduced by Garvin in [9]. Although he focuses on product quality, he provides five different approaches for the term quality:

- The transcendent approach of philosophy: Quality is defined by the innate excellence of a thing that is beyond the border of measurability. It can only be captured by experience and comparison.
- The product-based approach of economics: The quality of a product is measured by the quantities of attributes using subjective criteria.
- The user-based approach: Quality is defined by the fitness for intended use.
- The manufacturing-based definition: Quality is defined by the degree to which it is conformant to specifications.
- The value-based definition: Quality is related to a cost-performance ratio. It is the degree of excellence at an acceptable price.

A second aspect of the term quality is the context in which quality is considered. The American Society for Quality [8] adds this context dependency and defines quality as a subjective term for which each person or sector has its own definition. In ISO 9000:2005 [10], quality is defined as "The degree to which a set of inherent characteristics fulfils requirements". It further defines that the term "quality" can be used with adjectives such as poor, good or excellent. These two

definitions are context independent and neutral. The advantage of such definitions is that they are generally accepted but on the other side they lack details and precision.
The ISO definition also contains a third aspect of quality. Using the words “degree” and “fulfil”, it implies a range with a minimum and a maximum boundary. So quality is not a two state term which can be good or poor but there can be differentiated levels.
Crosby in [1] agrees with the ISO 9000:2005 that quality is a conformance to requirements but is more strict on quality levels in his definition of the Four Absolutes of Quality Management:

1. Quality means conformance to requirements, not goodness
2. Quality is achieved by prevention, not appraisal
3. Quality has a performance standard of Zero Defects, not acceptable quality levels
4. Quality is measured by the Price of Non-conformance, not indexes

A philosophy to achieve quality is called Total Quality Management, a term coined by William Edwards Deming, Joseph Juran and Kaoru Ishikawa [11], [7], [12]. As the name indicates, it is a management guideline to assure that products and services can get more quality. The management and improvement of quality is an on-going activity that runs in parallel to a main process like production or service. That means that the activity should run as long as the process is running.
The second point of the philosophy is that all persons that are involved in the process are also integrated in the quality management and improvement. Every manager, worker, supplier, customer and stakeholder is responsible for the quality of a product or service and should contribute so that the desired quality can be achieved. The main goal of the philosophy is the satisfaction of customers. English in [2] added a definition for quality to the Total Quality Management as "consistently meeting knowledge worker and customer's expectations". In his definition, a knowledge worker is a customer of information who requires data to perform their job.

2.3.2 Data Quality and Information Quality

After the definition review of quality, we will summarise existing definitions of the terms data quality and information quality.
Although most authors of data quality literature make a distinction between data and information [2], [13], [14], [15], [16], and [17], these distinctions are not continued in the definitions of data quality and information quality. Most authors use the two terms synonymously and hence provide one definition for the two terms.
There are two distinguishable definitions for the terms:

1. Data or information has quality if it is appropriate for the intended use ([18], [19], [20], [21]).

2. The quality of data or information depends on users' expectations and requirements ([17], [18], [22], [23]).

These two definitions are by their nature different because the set of requirements for a particular use can be smaller than a set of all user requirements. An aspect that is pointed out in both definitions is that data quality is inherently subjective and hence customers are the final arbiters of quality, Redman [17], Tayi and Ballou [21].

An addition to the first definition is made by Olson [18] and English [2] as data quality depends as much on the intended use as it does on the data itself (inherent quality). An addition to the second definition is that customers' expectations cannot only be satisfied but even exceeded, Kahn and Strong in [23]. Another addition to the second definition is called pragmatic quality, which is the value that accurate data has in supporting the work of the enterprise. If data does not help to accomplish enterprise's mission, it has no quality, no matter how accurate it is, English [2]. In addition to definition 2, Eppler in [24] defined that information has quality if it meets the functional, technical, cognitive, and aesthetic requirements of information producers, administrators, consumers, and experts.

A completely different approach for the definition of data quality is the differentiation by the problems that can arise (Garvin [9]). Three types of data quality problems are distinguishable: biased information, outdated information and massaged information. Biased information is information that is inaccurate or distorted due to the interests or motives of the source or information transmitter. Outdated information is information that is no longer current due to its tardy delivery or a failure to update it. Massaged information is related to inadequate format.

According to Garvin, massaging is the putting together of data in a manner that applies to a particular problem at hand. The problem is that different massaged information can lead to different interpretation.

For the rest of this thesis, we will only use the term data quality for the following reasons. We assume that a scientific database such as a food composition database contains data and not information because we follow the definition ([2], [13], [14], [15], [16], and [17]) that through analysis and interpretation, which can be based on a question, a user gets information. What information a user gets is dependent on user's understanding, experience, as well as on the used analysis and interpretation method and of course on the data. An appropriate evaluation of these factors is strongly user-dependent. In contrast, data quality "only" depends on samples, how value are measured or determined and on a careful data input. This meta-information can be stored together with a food composition value and we are therefore able to provide automated data quality evaluation methodologies.

2.3.3 Different Approaches to Data Quality Dimensions and Frameworks

So far, we have seen the definitions of different terms and used the terms data quality requirement or data quality dimension to describe a certain quality facet or a desirable quality attribute of data. In this section, we focus on these data quality terms.

The notion of data quality having multiple dimensions is used in many of the cited publications in this section. Thereby, a dimension is a certain data quality requirement such as accuracy, timeliness or completeness. The dimensions can be seen as degrees of freedom in a space in which quality of data can be placed.

These dimensions have been grouped into categories according to certain criteria. The result is called a data quality framework. The most likely source for the term framework is the approach to theory building of Porter [25] where he defines a model to be of limited complexity because the intention is to abstract a real world situation to isolate a few key variables. In contrast, a framework is defined to encompass many variables and tries to capture much of the complexity of real world situations. He states: "Frameworks identify the relevant variables and the questions which the user must answer in order to develop conclusions tailored to a particular industry and company."

A comprehensive review with a focus on data quality frameworks was done by Eppler in [16] where he identified 20 frameworks and investigated 7 from different application contexts (Newspaper, Corporate Communications, Data Bases, Information Science, Data Warehouse and Business Information and Web Pages) in more detail. His research showed that most of the frameworks are built for a certain domain and only a few general valid frameworks exist.

As there are many of these frameworks, we present only some selected frameworks to give an overview of the different approaches and refer interested readers to the two review works of Batini [26] and Eppler [16].

4 categories, 15 dimensions

The most often cited framework is that of Wang and Strong [27]. They interviewed data consumers about data quality and ended up with 179 dimensions. They condensed and summarised these into 15 dimensions and 4 categories that are presented in Figure 2.1.

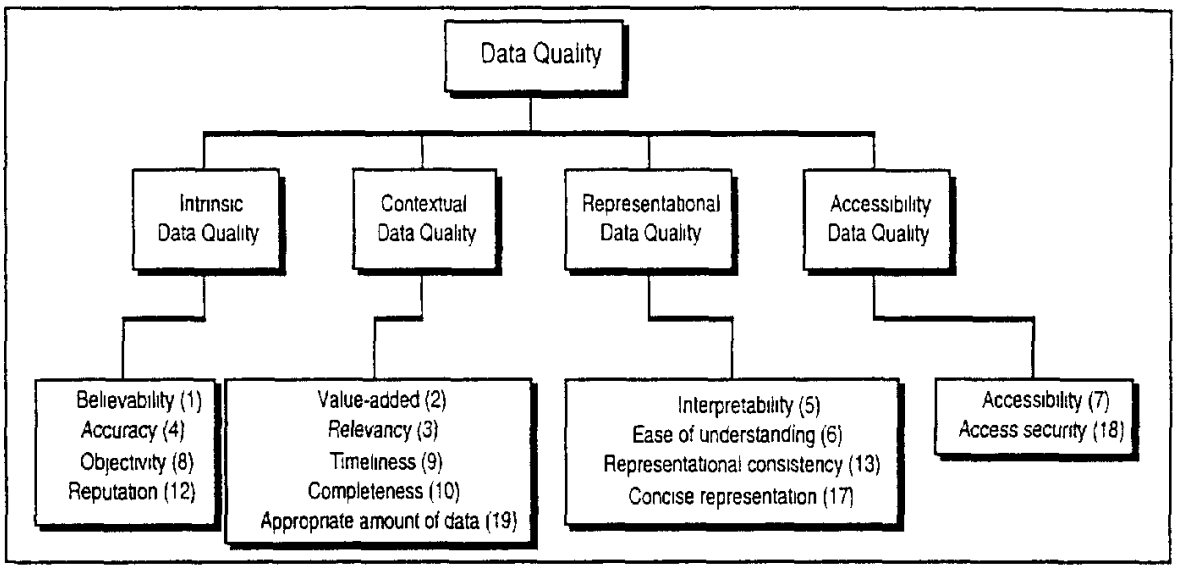


Figure 2.1: A data quality framework with 15 dimensions identified by Wang and Strong in 1996.

3 sets with 15, 4 and 8 dimensions

Redman in [17] categorises data quality dimensions in three sets: Those relating to the model or view, those relating to data values, and those relating to the representation of records. Redman defines a view as "part of the real world" to be captured in the data. The first category contains 15 dimensions (relevance, obtainability, clarity of definition, comprehensiveness, essentialness, attribute granularity, domain precision, naturalness, occurrence identifiability, homogeneity, minimum redundancy, semantic consistency, structural consistency, robustness and flexibility), the second category contains 4 dimensions (accuracy, completeness, currency and value consistency) and the last contains 8 dimensions (appropriateness, interpretability, portability, format precision, format flexibility, ability to represent null values, efficient usage of recording media and representation consistency).

4 levels, 4 phases and 16 dimensions

Eppler [16] proposes an information quality framework consisting of 16 dimensions that are categorised according to four levels and four phases. The four phases represent the information life cycle from a user's point of view: The information may be the answers that a user wants to find, understand and evaluate, adapt to his/her context (=allocate) and apply in the correct manner. The four levels describe the relation of information to the target community, the information product, information process and infrastructure. Figure 2.2 gives an overview of the framework. The framework describes also interdependencies, which can be potential conflicts, between the dimensions with simple arrows and contains also management principles.

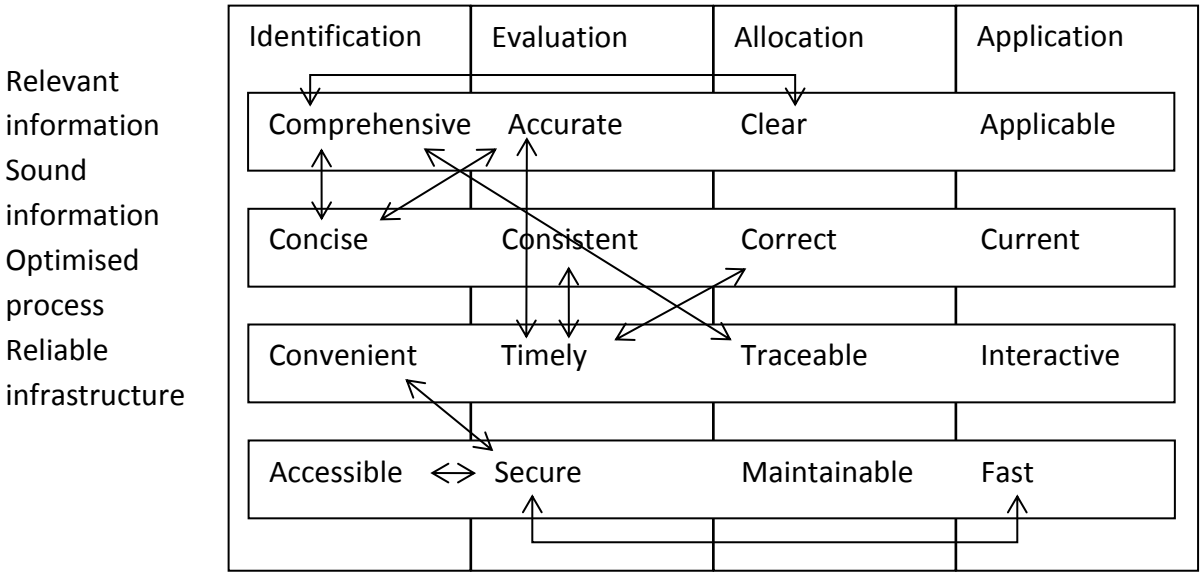


Figure 2.2: Simplified information quality framework of Eppler with 16 dimensions on four level and four phases. The arrows indicate potential conflicts of dimensions.

4 dimensions

A derivative approach defining a data quality framework was done by Wand and Wang [28]. Intrinsic data quality dimensions were derived by identifying possible deficiencies during the transformation from the real-world system into the information system. The resulting four dimensions Complete, Unambiguous, Meaningful and Correct reveals four generic data quality problems that can be observed in using an information system.

Classifying data quality problems

Rahm and Do [29] focused on data cleansing in that they classified data quality dimensions according to data quality problems that arise during cleansing. Figure 2.3 gives an overview of their categories. They distinguish between single-source and multi-source problems and between schema- and instance-related problems. Based on this classification, the authors provide an overview of the main solution approaches in data cleansing.

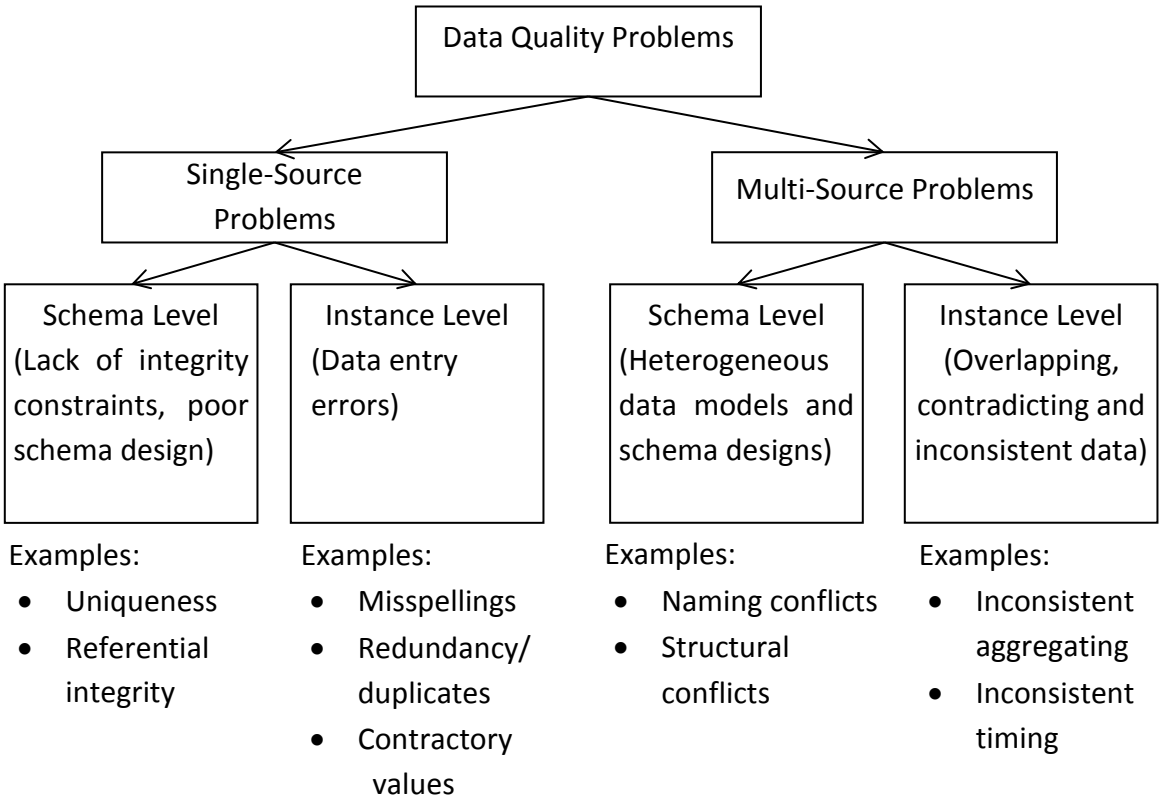


Figure 2.3: Classification of data quality problems according to Rahm and Do

Conclusion

Although these frameworks are well elaborated, data quality dimensions are mostly not described in measurable and formal manners. Instead, they are defined by subjective descriptions using substantives or adjectives for which the semantics are overlapping or fuzzy. This subjective description of data quality dimensions has the advantage that every person has a notion about the meaning. On the other hand, the approach leads to metric problems because it lacks details. For instance, consider the question to rate the reliability of a value such as 5 mg Vitamin C in an apple. As seen in Section 2.2, a dimension contains many data quality requirements that must be scored before the data quality dimension reliability can be rated.

In addition, as different descriptions of dimensions have overlapping notions, a data quality requirement can belong to multiple dimensions. For an assessment of a summarised data quality value from different dimensions, we get a serious problem: If a data quality requirement is involved in multiple dimensions, then it has more weight on the aggregated total data quality value than others as the following example shows.

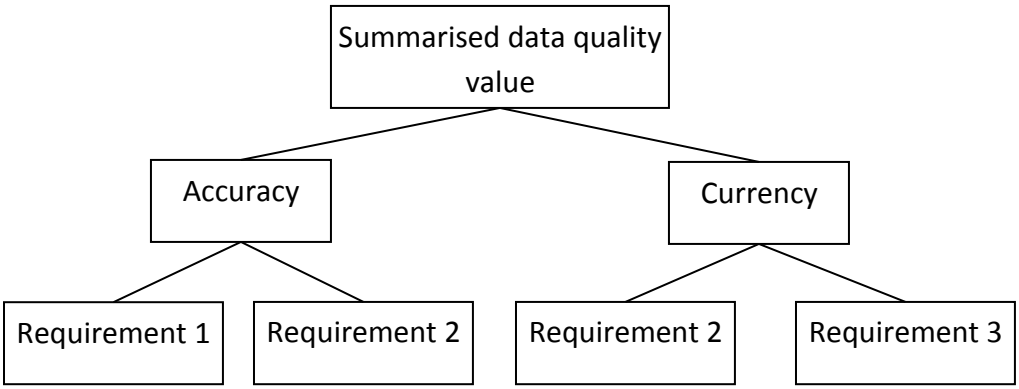


Figure 2.4: Example how a summarised data quality value can be calculated based on it data quality dimensions and their requirements

Requirement 2 is involved in both dimensions and if requirement 2 is fulfilled, 50% of the total data quality value is already reached although only 1 of 3 requirements is satisfied. That behaviour is probably wanted but if it is not wanted then the implementation of the assessment schema needs to countervail the influence of such a data quality requirement. Although some authors claim data quality dimensions to be independent of each other, it is difficult to satisfy this claim because subjective description leaves a certain space for interpretation.

A further issue of some frameworks is the dependency of certain data quality dimensions. Eppler in [16] shows some examples. A classic example consists of the four dimensions timeliness, accuracy, completeness and consistency. To have accurate, complete or consistent data, time and activities are needed. On the other side, if up-to-date data is needed, this can cause negative impact on accuracy, completeness and consistency.

2.4 Modelling of Data Quality

Once the dimensions for a given context are determined, the next step for system architects is to extend the information system conceptually to accommodate these dimensions. This section presents a summary of existing work on data quality extensions to existing models and schemas as well as new data quality models.

Existing data models such as the relational model or the abstract conceptual Entity Relationship model (ER model) contain implicit or explicit requirements to data such as type definitions, referential integrity and cardinality, which are also data quality requirements. The same is true for modelling languages such as UML. However, as such data models and modelling languages in general are not intended to focus on data quality, they only contain a subset of data quality requirements and are mostly not comprehensive. The most common data models are the relational model and the ER model. In particular, a wide range of database management systems adopt the relational model. Therefore, it is not surprising that most data quality extensions are built on these two models.

Using the ER model

A first approach to use the ER model schema is simply to add data quality attributes to an entity.

Food		
ID	Description	QualityDimensionValueOfDescription

Figure 2.5: Example of quality dimension represented in the Entity Relationship model schema

In this approach, the data quality value for the food description is added as an attribute in the Food entity.

Extending the ER model

Storey and Wang[30], [31] criticise that the entity is no longer normalized because the attribute QualityDimensionValueOfDescription refers to the description and not to the food. Another problem is that if we want to create several quality dimensions, we have to define numerous additional attributes. They solved the above problem by introducing the Quality Entity Relation (QER) model, which is an extension in the ER model schema. The model differentiates application requirements, application quality requirements and data quality requirements. Application requirements are requirements such as the presence of entities representing real-world objects, their relations as well as the functionality. For instance, an employee with attribute age and salary is a typical application requirement.

Application quality is equal to software quality and indicates how well an application conforms to non-functional requirements such as performance, usability or maintainability. If a functional requirement is not satisfied, then an application requirement is not fulfilled. If a functional requirement is satisfied but it takes more time to perform a task than specified, the corresponding application quality requirement is not fulfilled. Application quality requirements were derived from the “production of a product” [32],[10],[33]. The idea comes from the management area and compares the production of data to the production of products.

A data quality requirement refers to the quality of data stored in a database. Although the requirements above can be satisfied, we can have low quality data as it is outdated or inaccurate.

This distinction between application requirements, application quality requirements and data quality requirements is an interesting approach. It means that some data quality aspects are regarded in the design phase of the database application (application requirements and application quality requirements) but there are additional data quality requirements that are not covered by the first two categories.

Storey and Wang propagate "The Data Quality Separation Principle" which means that data quality requirements are modelled separately from application requirements and application

quality requirements. The model introduces two new types of entities: a Data Quality Dimension entity and a Data Quality Measure entity. The concept is explained using Figure 2.6.
The goal is to store different quality dimension scores for the attribute value description. In the classic ER model, there is no concept to connect an attribute such as description to an entity such as Data Quality Dimension entity. The QER model offers a solution by introducing the "food description has" relation. The Data Quality Dimension entity has the attributes “dimension-name” and “rating” in order to represent all quality dimensions and their possible ratings. As an example, it is possible to say that "food apple" has the quality dimension rating of [Accuracy, 1] and [Completeness, 2] for the attribute description.
To describe the rating in more detail, the Data Quality Measure entity is used, which has the attributes rating and description. Rating is associated with the attribute rating of the “Data Quality Dimension” entity and the description can be used to stipulate the meaning of the rating. For example, "the rating of 1 means lowest quality because 1 is the lowest possible value in a scale from 1 to 100".

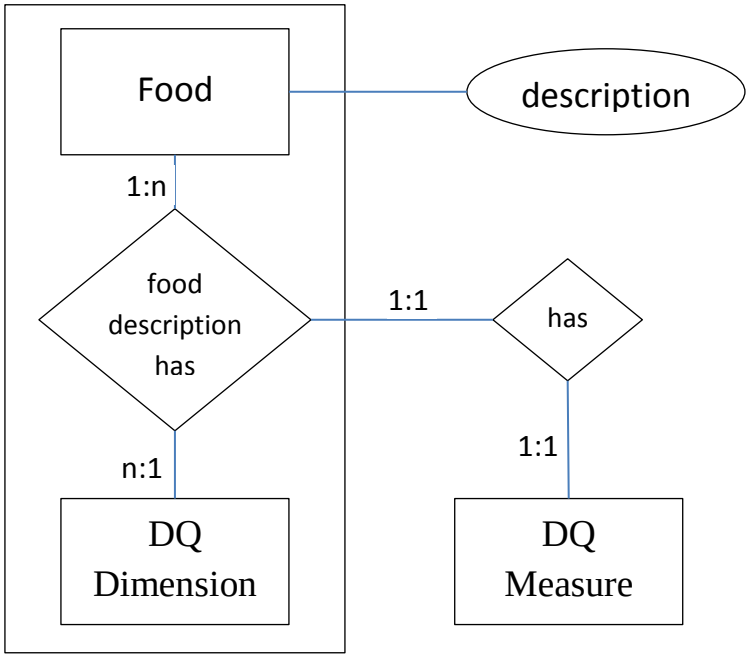


Figure 2.6: Example diagram with the Data Quality Dimension and the Data Quality Measure entities which are used to store and to describe quality dimensions for the food attribute description.

The DQ Dimension is denormalised because the entity contains the dimension, the attribute name and the value. In addition, all combinations of attribute and value must be listed in the entity.

Extending the relational model

Wang et al.[34], [35] extended the relational model by defining an attribute of a tuple as a quality cell. They called it the attribute-based approach model. A quality cell consists of two attributes: the attribute value, which is the ordinary attribute value, and the quality key value. The quality key value is a foreign key to a tuple of quality dimensions and corresponding value, which is called the quality indicator tuple. See Figure 2.7 from [26]:

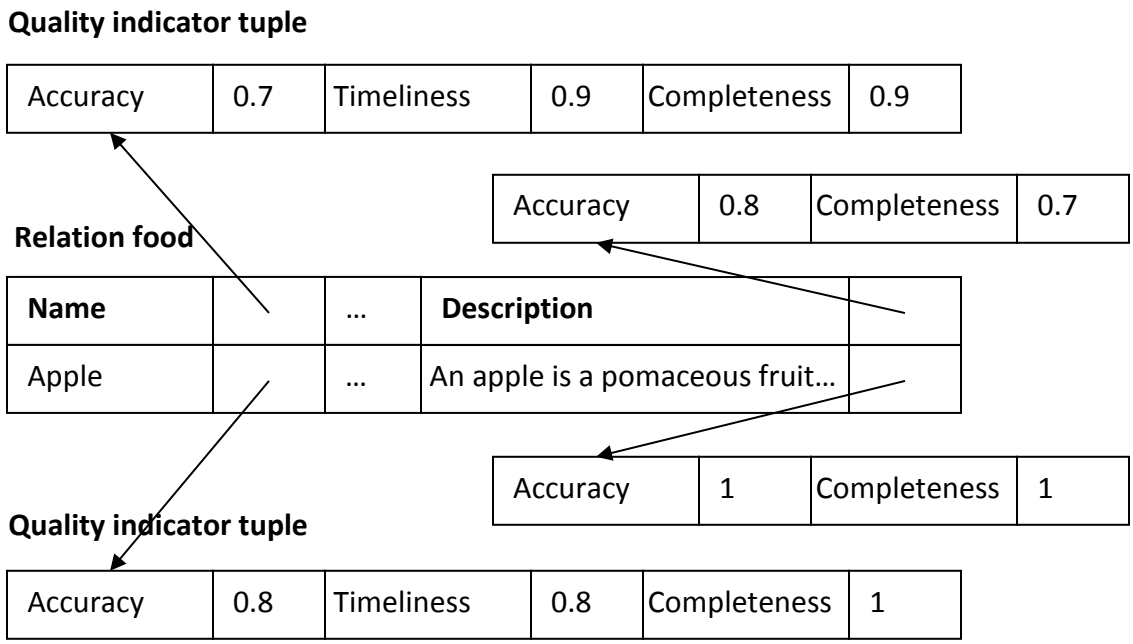


Figure 2.7: Example of attribute-wise referring to defined quality dimensions by the example of a relation food.

Every attribute in a tuple of the relation food has a foreign key to a quality indicator tuple (at the bottom of Figure 2.7). The quality indicator tuple contains quality dimensions that are applicable for referenced food attribute and contains the quality dimension values. In addition, the head row of the relation food contains also foreign keys to quality indicator tuples, which represent the attribute quality summaries over all tuples in the relation food.

The attribute-based approach postulates that every attribute should be quality evaluated using quality requirements. Conceptually this is correct but needs tremendous implementation efforts and the approach does not make a proposal for attributes that are involved in multiple data quality requirements. A second problem is the insertion of new dimensions or even new data quality requirements. In that case, the schema must be extended. The proposal also doubles the number of attributes and that can lead to a problem because some database products limit the number of attributes per table.

Source tagging models

Other models focus on data provenance, which is defined by Buneman [36] as "the description of the origins of a piece of data and the process by which it arrived in a database". When queries on multiple data sources generate output, users can be interested in information about the source. A common way is to use annotations on attributes of tuples, which can be investigated by users. These annotations can be remarks and can also contain data quality values. In the case where annotations contain data quality values, users have also information about the quality of the sources.

The Polygen Model [37], [38] by Wang et al is also an attempt to represent data provenance.

The idea came from the requirement that source information is needed to evaluate quality when using a distributed database system. The name Polygen expresses this situation because poly means multiple and gen means source. The model is designed to trace the sources and intermediate sources of data that contribute to a final query result. Databases are denoted as the different single databases from which data is queried while the query results are composited in the user application system. A Polygen domain is a set of ordered triplets and each triplet consists of three elements: A datum from a simple domain schema of a database, a set of originating sources denoting the databases from which the datum originate and a set of intermediate sources which contribute to the final query result. The model also contains a Polygen algebra for the operations projection, Cartesian product, restriction, union, difference, and coalesce, which all consider the originating and intermediate sources.

The idea that scientific data should be source proven is an important point because the source of data is a data quality requirement that, if available, must be integrated in the data quality assessment. Nevertheless, the source evaluation is not the only data quality requirement and so the model covers only a part of a comprehensive data quality assessment model.

Model for semi-structures

A model that is not designed for relational database systems but for semi-structure data such as XML, is called Data and Data Quality (D^2Q) [39]. The model is intended to be used in cooperative information systems (CIS), which means that independent information systems want to exchange data. The idea is to associate a data schema with a quality schema to enrich semi-structured data with quality information. The D^2Q model is made for XML and Figure 2.8 should help to understand the model:

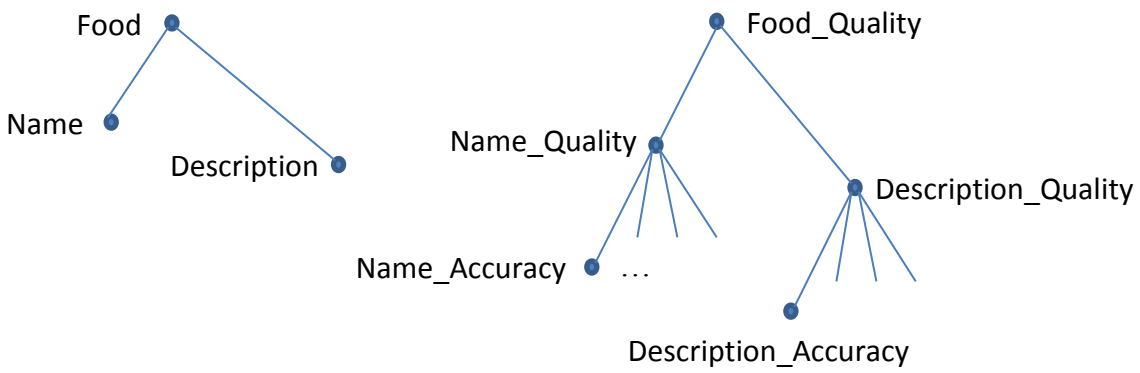
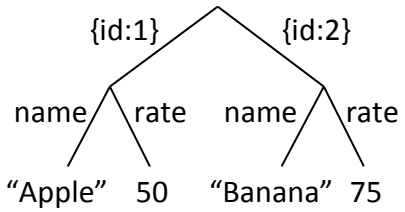


Figure 2.8: Example of D^2Q model with the data on the left side and the data quality on the right side.

Hierarchical data is represented in a tree. In Figure 2.8 the tree can be seen on the left side. For instance, there is a Food with two leaves Name and Description. The definition of the structure is called the data schema. Given that, for every subnode, a set of quality dimensions is defined, a similar hierarchical structure is created, that contains the quality dimensions for every node, in Figure 2.8 on the right side. The definition of this structure is called the quality schema. For example, the node Name has a corresponding node Name_Quality that in turn has all quality dimensions defined as leaves. These leaves have concrete values for the according quality dimension. A so-called mapping schema connects the data schema with the quality schema. The model can be easily translated into XML and then queried using XQuery.
The D^2Q model is made for semi-structure data and shows how attributes and quality values can be associated. One requirement in this model is again that data quality dimension are precisely defined and data quality assessment are defined.

Source tagging models for semi-structures
Buneman et al [40] introduced a model for semi-structured data where the source of any data can be uniquely described by a path. They used a tree model where the labels of edges are exploited to contain information. Two instances of an entity with id 1 and 2 that have the attribute name and rate can be presented the following way:



In this tree structure, a path to the attribute value “Apple” can be defined as {{id:1}:{Name:Apple}} where the part before the colon is the label and the part after the colon is the value. With the help of these paths, two types of provenance were defined by Buneman in

[36]: Why and Where provenance. The why provenance asks the question "why is a piece of data in the output?" and hence focuses on the reason for the output including the constituting sources. The where provenance asks the question "where did a piece of data come from?" and focuses on source from where data was propagated.

Business process based models

Another model is the Information Production Map (IP-MAP) model [41]. The idea is again to compare the cycle of information with the production cycle in industries. Data is seen as a product, and a graphical model is designed to help comprehend, evaluate, and describe how an information product is assembled in a business process. IP-MAP models are also designed to identify ownership of process phases, understand information and organizational boundaries, and estimate time and quality metrics associated with the current production process. The IP-Map consists of a set of construction block as can be seen in Figure 2.9 taken from [26]:

Concept name	Symbol	Description
Source (raw input data)		Represents the source of each raw (input) data that must be available in order to produce the information product expeceted by the customer
Customer (output)		Represents the consumer of the information product. The consumer specifies the data elements that constitute the "finished" information products.
Data quality		Represents the checks for data quality on those data items that are essential in producing a "defect-free" information product.
Processing		Represents any calculations involving some or all of the raw input data items or component data items required to ultimately produce the information block.
Data Storage		It is any data item in a database.
Decision		It used to describe the different decision conditions t obe avaluated and the corresponding procedures for handling the incoming data items,based on the evaluation.
Business Boundary		Specifies the movement of the information product accross departmental or organization boundaries.
Information system boundary		Reflects the changes to the raw data items or component data items as they move form one information system to another type of information system. These system changes could be inter or intra business units.

Figure 2.9: Constructs and symbols used to represent elements of the IP-MAP.

In a later work Scannapieco et all [42] translated the IP-MAP elements to UML to have the concept in a de facto standard language and called it IP-UML.

The IP-MAP model is a management tool to get an overview of data flows and processes. What is missing for our purpose is that the model does not care about database schemas.

Another model is proposed by Würthele [43]. He created a new model to represent processes and data quality. Therefore, he enriched every process step with information about data quality and performance. Process steps can be connected to build process paths and so process cycles in a company or institute can be represented. Using the probability of a path and the data quality decrease in each process step, a quality manager is enabled to calculate data quality information and the process performance along the path for a data quality dimension. The data quality decrease of a process step is a variable that needs to be determined.
The model of Würthele is designed to represent data management processes and how data quality decreases in these processes. It is also a model for management and does not deal with database schemas. As Würthele mention in his work, the most difficult task in his model is to find the appropriate values for the data quality decrease of each process step. The model is strongly dependent on these values, and data quality evaluation depends on the accuracy of these values. Moreover, the maintenance of these values must be sustained and hence the analysis for every process value has to be done continuously. In addition, a process graph for every single dimension is needed.

Conclusion
If we look at the presented data quality models, we can see that they are aimed at different objectives. They all postulate that data quality dimensions are explicitly defined and appropriate metrics are known. But as seen above this is not the case.
After having a set of data quality dimensions and a data quality model, the next step is to find an assessment of each dimension in the model to evaluate the degree to which data satisfy the dimension.

2.5 Data Quality Assessment

As we have seen above, concise definitions and context dependencies of data quality dimensions cause difficulties in finding assessments. A claim of some authors [44],[45] is the use of metadata to help to assess data quality. Rothenberg argued that information producers should perform verification, validation, and certification of their data and that they should provide data quality metadata along with the datasets[45].
Naumann et al [46] found a methodology to select data from different data sources according to their quality that is based on the metadata claim. Their work presented a framework where data quality is involved in the query processing in a multi database environment. This means that a mediator system that is queried for molecular biology information and that further queries other database systems to collect data, performs first a quality-driven source selection and then returns only most qualitative results to the user. Naumann took the data quality framework of Wang and Strong [27] and defined different simplified scores for the data quality

dimensions. The scores have not the same range and are later normalised to be able to build a weighted sum.

One work that proposes general assessment methods for all data quality dimensions is made by Pipino et al [47]. They presented three functional forms for developing objective data quality assessment: The simple ratio, the minimum or maximum operation and the weighted average.

The simple ratio measures the ratio of current outcomes to total outcomes. For instance, if a column of a table should contain at least one occurrence of all 50 states but it only contains 43 states, then we have population incompleteness and a ratio of 43/50.

The minimum and maximum operator can be applied where the aggregation of multiple data quality dimensions is required. One computes the minimum or maximum value from among the normalized data quality values of the individual data quality dimensions. The minimum operator is conservative in that it assigns an aggregate data quality value not higher than the value of its weakest data quality dimension. The maximum operator is a more liberal interpretation.

The weighted average is an alternative to the minimum and maximum operator. The different data quality requirements are weighted according to their importance and then averaged.

For data quality dimensions completeness, currency, timeliness, and accuracy, the following related works exist.

Naumann [48] describes a method for performing data quality aggregation in a database table. Given that a measurement for completeness is known, an aggregation can be performed over a table column and over a table row. Combining these two aggregations over all columns and all rows leads to a quality assessment of a whole table.

The data quality dimension of timeliness offers the possibility to use mathematical functions because the underlying data type is time. Ballou et al [49] proposed the following formula for the currency dimension:

$$\text{Currency} = (\text{Delivery Time} - \text{Input Time}) + \text{Age}$$

where Delivery Time is the time when data is delivered to the user, Input Time is when the data was entered into the database and Age determines how old data was when entered into the database. For timeliness, they defined the following formula:

$$\text{Timeliness} = \left\{\max\left[\left(1 - \frac{\text{Currency}}{\text{Shelf\_life}}\right), 0\right]\right\}^{s}$$

where Shelf_life stands for how long data is up-to-date. With the parameter s, it can be determined how strong the change of the quotient influences the timeliness value. Hinrichs [50] proposes the formula:

$$\text{Timeliness} = \frac{1}{\text{Update}(\text{A}) * \text{Age}(\text{w}, \text{A}) + 1}$$

where Update represents the update frequency of attribute A and Age determines how old the attribute value w of attribute A is.

Klier in [51] defines some constraints for data quality metrics based on the work of Even et al [52], Heinrich et al [53] and Hinrichs [50]. A data quality metric must be normalised, cardinal

scalable, able to use different weights, able to aggregate over attributes, tuples and relations, able to be operationalized to map the domain set to the range set, and interpretable by specialists. He criticises that the former two data quality metrics are not cardinal scalable and not interpretable by specialists. Therefore, he proposes the following function:

$$Timeliness = e^{-Expiry(A)*Age(w,A)}$$

where Expiry is the expiry rate of the values of Attribute A and Age is the age of value w of Attribute A.

In domains where the correct value is known, the difference between the correct value and the value stored in the database can be measured. Hinrichs [50] proposes:

$$\text{Accuracy} = \frac{1}{\mathrm{d}(\mathrm{w_I}, \mathrm{w_R}) + 1}$$

where d is a difference function between the value in the information system $\mathrm{w_I}$ and the real world value $\mathrm{w_R}$. For attributes of type string, the edit distance such as the Hamming distance or the Levenshtein distance can be calculated.

Klier [51] listed two examples Meierhore/Mayerhofer and Mayr/Wein. The accuracy function above has the same value of 4 taking the Hamming distance as difference function. Klier criticises that the number of matching characters is not taken into account and he therefore proposes to use the following accuracy function:

$$\text{Accuracy} = 1 - \left(\frac{|\mathrm{w_I} - \mathrm{w_R}|}{\max\{|\mathrm{w_I}|, |\mathrm{w_R}|\}}\right)^{\alpha}$$

where $\alpha \in \mathbb{R}^+$ can be seen as a weighting factor. Using the Hamming distance for $|\mathrm{w_I} - \mathrm{w_R}|$ gives the first example a much better accuracy function value than the second.

One discussable point of the function to calculate timeliness and accuracy is the usability in practice. The complexity can overburden some users and hence can be regarded as black box behaviour. Another point is that, in food composition, it is possible to formulate simpler timeliness requirements such as "value should not be older than 5 years". This requirement is satisfied for the first 5 years and unsatisfied afterwards. An assessment function that has a jump at 5 years cannot be represented with the presented functions.

There are only a few assessment approaches, model and framework contributions that helped us to implement a data quality framework in the Swiss food composition database system. Also the provided maps of the data quality research area, provided in [26], [54] and [55], did not help further. The proposals are all lacking in details or context tailored. The obvious reason is that data quality dimensions and their assessment is context dependent. Using high-level definitions for data quality dimensions does not improve the situation. We are convinced that going into a more detailed level such as concrete requirements, offers us more possibilities to formulate general valid assessment rules as in the case of timeliness and accuracy. In the following section, we change to the topic of food composition to see what data quality concepts are available.

2.6 Data Quality in Food Composition Databases

Until now, we considered data quality in general. In this section, existing data quality work in the area of food composition is investigated to discover the context specific approaches to improving data quality.

Greenfield and Southgate [20] published a book of guidelines to aid individuals and organisations involved in the analysis of food, the compilation of food data, the dissemination of food data, and the use of data. While the second half of the book is addressed to food scientists and food compilers because it focuses on different aspects of the food compilation process, the first half of the book covers general topics on food composition databases and gives some new concepts concerning data quality: Multilevel database, provision of thesauri, and value precision.

A multilevel database is one that classifies food composition data in different stages. The "Data source" stage contains original documents such as published research papers or unpublished laboratory reports. In the "archival data" stage, they define original data transposed to data records without amalgamation or modification. The written or computerised data records hold all data in the form they were originally published including details about origin, sample, sample handling, analytical methods and quality-control methods. The data records have a defined structure. The "reference database" stage is the complete pool of rigorously scrutinised data in which all nutrients and values are expressed uniformly, but in which data for individual analyses are held separately. Finally, the "user database" stage contains data that is selected or weighted to give representative values for the intentions of users.

The authors propose a provision for thesauri that limits the set of possible values for a certain attribute. They list different tables for food composition aspects and propose also a thesaurus of alternative food names.

Greenfield and Southgate provide also a table with numbers of significant digits that should be used in publications. This is necessary because laboratory instruments often deliver numbers with many digits that pretend a precision that cannot be achieved. Therefore, the limitation of significant digits helps to circumvent that problem. The book is the standard reference work for every person working on food composition.

Schlotke and et al, as a special working group in the EU project COST Action 99 [56], proposed a set of recommendations for food composition data interchange using electronic media. The recommendations include a description of attributes of food, component, nutrient value and data source. The thesauri collection of Greenfield and Southgate was extended and the set of possible values for certain attributes was harmonised and presented in the recommendation. To exchange data, they recommended using one text file per database table with one data record per line using semicolons as delimiters. In the recommendation, the plan to use XML is mentioned as: "It is therefore planned to translate the data structure presented in this report into an XML application once this Internet standard has been established" (2000).

Holden et al in [57] generalised and expanded their existing data quality evaluation system in the U.S. to be valid for all nutrients. The evaluation system consists of the five categories sampling plan, number of samples, sample handling, analytical method and analytical quality control. One modification was the extension of the rating scale for every category from 0-3 to 0-20 to have a more continuous scale. The sum of the five categories determines the so-called quality index of a certain nutrient value. The quality index is a data quality rating value that indicates the reliability of a food composition value. In a second step, nutrient data from several acceptable sources are aggregated to give an overall estimate of the nutrient content of that food. In this step also the Quality Indexes (QI) were aggregated into the Confidence Code (CC).

To explain the confidence code, consider again a database that has three Vitamin C values for an apple from three different sources. For the three values, their quality index was calculated as described above. To have one representational Vitamin C value that can be published, the three values are aggregated. For the aggregated Vitamin C value, a confidence code indicates how reliable the value is and is therefore similar to the quality index. The scale of the confidence code is simplified and consists only of the letters A, B, C and D, where A is the highest rating.

Specific rules are applied in the aggregation of the CC. For instance, if individual sources may not have received high rating because the samples were only regional, the CC may be higher as the simple sum because the regions were not intersecting and hence a bigger area of the country is covered. In addition, the summation of the confidence code was fitted to certain range of maximum 100 points to avoid that the aggregation of numerous mediocre data sets could, together, merit a higher CC, as was the case in the previous evaluation system.

The EuroFIR project [58] aimed to provide the first comprehensive pan-European food information resource, using state-of-the-art database linking, to allow effective management, updating, extending and comparability. In order, one task group revised the food composition recommendation of Schlotke et al. A proposal for structure and detail for a EuroFIR Standard on food composition data was one of the outcomes and was made by Becker et al [5]. The proposal contains food composition entities, their attributes with data types, extended thesauri lists as well as an XML structure for data exchange.

Another task group within the EuroFIR project defined a catalogue of questions and possible answers with which it should be possible to determine the EuroFIR quality index [6]. The quality index determines how reliable a food composition value is based on background information about a value. With the proposed metric system, all partners will have the same notion of data quality rating, and data exchanges including data quality values will be possible. The working group also elaborated guidelines on how the questions should be answered. The quality assessment consists of seven categories, which are equally weighted to build the quality index for food component values. Mostly yes/no and yes/no/not-applicable are used as question types. A category can have between 1 and 5 points dependent on the number of yes and not-applicable answers. For instance, the category sampling plan has 6 yes/no/not-applicable questions. If 4 questions are answered with yes and 2 questions are answered with not-

applicable then we get the maximal rating of 5 points. The reason is that only 4 questions are applicable and they are all answered with yes. A tree view of the quality index can be seen in Figure 2.10:

Figure 2.10: Schema of the EuroFIR quality index for single food component values. The seven categories are equally weighted to calculate the score of the quality index.

EuroFIR planned also to define a confidence code. At the time when this thesis was written, only the scope of the confidence code was defined but not the assessment schema. The scope is defined to lie between A and D, where A is the highest score and D the lowest score.

The concepts of Greenfield and Southgate are of all part of a data quality framework but do not cover data quality as a whole concept. The quality index and the confidence code are two approaches for a total data quality rating but they cover only the reliability of food composition values.

2.6.1 History of the Swiss Food Composition Database

The first general data model for a food composition database and its implementation in Switzerland was made in 1992 at the Institute of Computational Science at the ETH Zurich [59].
Between 1992 and 2010 6 different applications to manage the Swiss food composition data were built [60], [61], [62], [63]. In that time, different national and international cooperations to centralise, standardise and harmonise the Swiss food compositions database were performed [64], [56], [58]. Most research and implementation was made at the Department of Computer Science at ETH Zurich where Professor Hinterberger was involved. From that long-term experience, know-how was generated and used to build the different food composition database systems. The complete history about the Swiss food composition database system can be found in Appendix A.

2.7 Motivation: The Necessity for a Data Quality Assessment Model and a Data Quality Framework

What we are completely missing in existing literature is a data quality assessment model, automated data quality assurance/improvement concepts in information systems as well as their position in the data life cycle.
A data quality assessment model is a generalised approach to how the assessment of data quality is made. An important part of such a model is a precise definition of data quality requirements with respect to their measurement. Two useful consequences of such definitions are that information system users become aware of what data quality means and that the definition facilitates users to formulate their data quality requirements. If a set of data quality requirements by domain experts is given, a data quality assessment model allows information system architects to specify how data quality values are calculated from objects and attributes using a data quality assessment schema. As this schema is closely related to the database schema and higher order programming class definition schemas, it must be designed at the same time or be able to be added later.
Automated data quality assurance/improvement concepts are concepts that a database application can offer to users to support them by measuring the quality of data, help users to find data quality problems, assist users to enter data of high quality and offer a possibility to observe data quality values over time. As on every database application, a data life cycle can be

identified, the data quality assurance/improvement concept should cover the whole data life cycle.

The set that consists of a data quality assessment model including schema and automated data quality assurance/improvement concepts will be our data quality framework. The data quality framework should be applicable to new information systems as well as existing systems.

From the discussion in Section 2.3, it is obvious that we did not find a definition for the term data quality that fits our data quality model and framework. Therefore, in the next chapter we are going to investigate the term in more detail. Based on the proposed definition, Chapters 4 and 5 will present a data quality assessment model and a data quality framework that offer solutions to the problems and lacks described in Sections 2.4 and 2.5 and the missing points that are not covered by literature.

3

Definition of Data Quality

In Chapter 2 we have seen that different definitions for the term data quality exist and that data quality dimensions are good abstractions to categorise data quality issues. But we have also seen that data quality dimensions are defined on such an abstract level that it makes it hard or even impossible to measure them. The problem that we are going to solve in this chapter is to define data quality issues in more detail than data quality dimensions do, thus, making data quality measurements possible. The first step to achieve this objective is to understand what users mean by data quality by collecting and analysing data quality issues raised by data producers and data consumers. The second step is then to derive precise definitions for different data quality terms. A second objective of this chapter is to have a comprehensive collection of data quality issues from which the definition of the data quality model in Chapter 4 can be derived. In addition, we also investigated what functionalities a data quality framework is expected to contain for Chapter 5.

The chosen approach was to conduct two user studies. In Section 3.1 we present the results of an experimental user study that we performed in the specification phase of the database application FoodCASE. The objectives of the experiment were to analyse data quality issues and to see how users focus on data quality in a specification phase. The experiment discovered some insights for how data quality must be considered and we developed a concept for data reliability and a list of data quality framework functionalities which are both presented in Section 3.2. We also present in Section 3.2 the results of a workshop survey that we conducted during the implementation phase of FoodCASE. The objectives of the survey were to prove the concept of Wang et al [27], our data reliability concept and to prove our list of data quality

framework functionalities. Based on the results of the two user studies and our experiences during the specification and implementation of FoodCASE, we define the terms data quality, data quality requirement and data quality indicator in Section 3.3. These terms build the basis for the data quality model and the data quality framework in Chapters 4 and 5.

3.1 Experiment to Extract Data Quality Issues from Users' Feedback

3.1.1 Goals

As in many other software implementation projects, the goal of a software requirement specification (SRS) is to investigate whether users and application designers have a common understanding of what and how the software should work. This was also our first goal. A second goal was to find out to what extent and in what form data quality is regarded in users' feedback in the specification phase to investigate what users understand by data quality. We wanted to investigate if data quality issues are formulated as separate issues or if data quality issues are covered in application features where software designers have to extract them. Finally we investigate to what extent an SRS accommodates the 15 dimensions of Wang and Strong [27].

3.1.2 Approach

We prepared an SRS for the food composition database application FoodCASE. The SRS described graphical user interfaces using print screens as well as their functionality. The SRS was written for food compilers and hence no technical details were included. Parts of the database design and the entity definitions were given by a EuroFIR proposal [5]. We gave the document to 62 participants in the FoodCASE project and asked them to give feedback in written form. We made no guidelines on the form of the feedback has to look like and asked some implementation specific questions at the end of the SRS. The specification can be found in Appendix A. The participants knew that we were researching data quality on scientific database applications but we did not include data quality concepts or functionality in the SRS. The idea behind this open question form was that users should not be encouraged to focus on data quality feedback but should give feedback as in any other software development project.

We then summarised the feedback, classified them into groups of similar requirements and counted the number of requirements in every group. The groups were further classified into data-quality and non-data-quality issues. Finally, we analysed some of the requirements in more detail.

3.1.3 Results

The 62 participants represented 32 institutions and companies in the area of food composition from 23 countries around the world. We received 32 feedback items in electronic form, E-Mails

or documents, from 17 institutions and companies from 16 countries. This is an effective response rate of 51.6%. Because of the open question the feedback was very different, from one sentence to 19 pages. From some institutions, multiple persons discussed the SRS and sent one mail as the feedback. For the analysis, these feedback items were multiplied by the number of persons because we assume that these persons agreed on the feedback.

We classified the feedback into four categories: General feedback, application quality, data quality, and new application features. General feedback is not about the database application or its functionality but is general feedback about the SRS. The application quality category contains existing GUI components and functionalities in the software that can be improved, for instance, with longer fields, fields that should be at a different place or with additional functionality that simplifies usability. The category can also be called usability feedback. In contrast to the application quality category, the application feature category does not improve existing GUI components or functionalities but defines new ones. To the category data quality belongs every feedback that cares about the quality of data. As a data quality requirement should be supported by database application functionality, it can be seen as a subcategory of application quality and, at the same time, as a new application feature.

The feedback and the number of mentions are listed in the following four tables:

General feedback

	Feedback	**Number of mentions**
1	SRS is very good	12
2	SRS is well written	4
3	Presentation of screenshots are not uniform	1
4	It is difficult to give detailed feedback on the screenshots. It would help if we could try out the software.	6
5	Minimum screen size is not provided	1
	Total	**24**

Application quality

	Feedback	**Number of mentions**
6	Add possibility to control every GUI with the keyboard	7
7	Use longer fields for names	7
8	All tables and their columns must be resizable to be able to read all data	8
9	It should be easy to change from one GUI to another	1
10	Menu structure is illogical	1
11	Recipe without ingredient amount information is good	3
12	Store statistical information like mean, max, min etc. is good	2
13	Keep history of food component values is good	4

14	Order in which aggregation/recipe calculation and calculated components are made, must be defined	5
15	Retention factors and yield factors that are used in recipe calculation are located at a wrong calculation step	6
16	Add possibility to delete data	6
17	Recipe calculation must be able to handle non-numeric values	1
18	Add possibility to copy food composition values from one food to another	5
19	Add possibility to distinguish which data is newly entered and which data is already evaluated by a second compiler	5
20	Use a more sophisticated user rights and roles concept	5
21	Add possibility to enter common used amounts like tea spoon, egg(s) etc.	1
22	Possibility to copy quality assessment because it is often similar	1
23	Add possibility to limit value quantification, under which the value is zero	1
	Total	**69**

Data quality

	Feedback	**Number of mentions**
24	Aggregations of values and recipe calculations must be performed automatically to reduce mistakes	5
25	Data checks like finding data falling outside a specific range, checking if the sum of the component values is ±100 g, check total fatty acids versus total fat, sum of individual sugars compared with the component "soluble sugars"	9
26	Add new field to describe recipe source	1
27	Add possibility to group food to more than one classification	5
28	Add picture of a person	1
29	Add fields genus, species and subspecies	1
30	The field matrix unit is missing	1
31	Add own country-specific fields	5
32	Add possibility to have more than one picture for a food item	1
	Total	**29**

New application features

	Feedback	Number of mentions
33	New import/export functionality	3
34	New search functionality	5
35	New functionality to make not all food composition values public	2
36	Connection to a reference tool	2
37	Extra tab for data supplier management (not under persons)	5
38	Food overview table is needed	3
	Total	**20**

3.1.4 Discussion

Although we made a clear definition of the categories application quality and application feature, some of the feedback items such as 16, 19 and 20 could be placed in both categories. As the objective of this experiment was not a unique categorisation but to analyse data quality issues, the classification of the four feedback items does not influence the outcome. In the following, we discuss 5 conclusions derived from the above results. The first conclusion discusses that users have difficulties to provide feedback on functionalities that are not described in a SRS. As a result, the second conclusion is that users have also difficulties to provide feedback on data quality issues. The third and the forth conclusions present that not all users seem to have the same notion of data quality and that for some data quality issues users have even contradicting opinions. The final conclusion is that data quality feedback is sometimes obviously mentioned in feedback but sometimes needs to be extracted from an application quality or feature describing feedback item.

Difficulties to provide feedback

A first observation is that the list of application quality is the longest and also the total number of mentions is the highest. This is not surprising considering the fact that a SRS simply describes GUIs and functionalities of software and users can give feedback based on their experiences in the daily work with their software, with the focus on the application rather than on data. The number of application quality feedback items indicates that users are more capable of rating presented GUIs and functionalities than to formulate data quality issues and new application features. This statement is based on the fact that we found more feedback for these categories during the implementation phase. In addition, feedback number 4 shows that 18.75% of the users found it difficult to give feedback and wished to be able to try out the software. This is the well-known chicken and egg problem. If software must be built, designer and programmers must know what the user needs but users can first say what they need when the software is built.

More difficulties to provide data quality feedback

If we look only at the data quality feedback items, it is conspicuous that some users have a notion of what good data quality is. 28% of the users wish to have data checks but only one person was able to give concrete examples, which are shown in the table under feedback number 25. Most data quality feedback items concern the completeness of data (26, 27, 28, 29, 30, 31 and 32) where missing attributes are mentioned. If we look at the number of mentions of these 7 feedback items, we can see that they are small. An obvious mistake in the specification was feedback number 30, which was detected by only one person.

Data quality is not the same for all users

Feedback number 27 gives an indication that data quality aspects are only important for certain users. From discussion with other food compilers, we found out that some food compilers take up a position that a food item can only be assigned to more than one classification while other food compilers take up a position that a food item belongs to only one classification. There seems to be data quality aspects that are user dependent. In Section 3.2 we investigated that observation in more detail.

Contradicting data quality issues

Feedback number 18 and 22 are controversial points from the data quality point of view. The quality assessment that the user means in feedback number 22 is the EuroFIR defined quality assessment for food composition data which was described in Section 2.5. If data and even quality assessments can be copied, users are able to work faster. On the other hand the danger is that a user forgets to modify the differences of the copied data record, and, as a result, decreases data quality and falsifies the quality assessment. It is possible to perform duplicate checks on the data but equal quality assessments are common for values that are measured for the same food in the same laboratory.

Form of the feedback

The forms in which users mention data quality issues differ. Some feedback items such as number 25 concern only intrinsic data quality issues while other feedback items such as number 24 target data quality but are embedded in feature or functionality descriptions. Incompleteness of data is mentioned by missing fields on GUIs. Hence application designers need to carefully analyse SRS feedback items and to extract data quality issues.

Conclusion

The feedback showed that some points should be improved but that our SRS is accepted by the users. An important observation was that although we had the SRS, an ER schema and a UML schema for the food composition database application FoodCASE, we did not have enough

information on what users mean by data quality. Specifically, we had no information on how we could model data quality and their assessments and we had little information about what concepts a comprehensive data quality framework should have. The EuroFIR assessment schema covers only the reliability of data. The ER model, the UML model and the SRS guarantee only that every entity has the attributes that are required, that the correlation between the entities are correct according to the context of food composition data and that the workflows of the GUIs represent reality. But users are still able to enter data of low quality.

Because of these restrictions, we concluded that a need of data quality model exists if only to support the design of data quality in the specification phase. We also concluded that a data quality framework would be helpful to design where and how data quality must be considered in an information system. The data quality model and the data quality framework should also sensitise users to data quality.

However, the above observations that not all users define data quality in the same way and that contradicting issues can exist, were interesting for further investigations in a second user study. But before the second user study is presented, we apply the data quality framework of Wang and Strong [27] to our SRS.

3.1.5 Data Quality Dimensions applied to a Software Requirement Specification (SRS)

As the quality framework of Wang and Strong [27] contains not only intrinsic data quality dimensions, we investigated to what extent an SRS without a set of data quality requirements and data quality concepts satisfies the 15 dimensions.

Approach

It is assumed that the implementation of software is equal to its accepted SRS. Hence, no difference should exist if only the SRS is analysed instead of the SRS and its implementation. We briefly discuss the four categories of the quality framework in this section and present a general discussion at the end.

The first category “representational data quality” contains the following dimensions and their description:

- representational consistency
  "data are continuously presented in the same format, consistently represented, consistently formatted, data are compatible with previous data"
- ease of understanding
  "easily understood, clear, readable"
- interpretability
  "interpretable"

- concise representation
  "well-presented, concise, compactly represented, well-organized, aesthetically pleasing, form of presentation, well-formatted, format of the data"

As the dimension “interpretability” is described with its adjective, we assume that this means that the presentation on a GUI is uniquely understandable.
If we consider that the intention of an SRS is to show that graphical user interfaces have a given structure, follow a defined logic and reflect business processes of a given domain, then a user accepted SRS must be rated to fulfil all 4 dimensions and completely fulfil the whole category.

The second category “accessibility data quality” consists of the following two dimensions and their description:

- accessibility
  "accessible, retrievable, speed of access, up-to-date"
- access security
  "data cannot be accessed by competitors, data are of a proprietary nature, access to data can be restricted, secure"

As an SRS stipulates the user-and-right concept, software normally is secured on a company or institute server, and the SRS defines how publishing over the Internet works, an accepted SRS also satisfies these dimensions.

The third category “contextual data quality” contains the following dimensions and their description:

- value-added
  "data is giving you a competitive edge, data add value to your operations"
- relevancy
  "applicable, relevant, interesting, and usable"
- timeliness
  "age of data"
- completeness
  "breadth, depth, and scope of information contained in the data"
- appropriate amount of data
  (without description)

When users evaluate new database applications like FoodCASE, two objectives of users are to simplify and speed-up their work and that the software is able to manage data that satisfy the dimensions of this group. In particular the ER schema should show that value-added and

relevant entities and attributes are present. But data maintainers, such as food compilers, are responsible for what data is entered into the database. This category of dimensions targets the acquisition of data where users must have defined standard operation procedures on how new data is evaluated and on what data are entered in their database. The acquisition policy must also include the timeliness dimension of data.

The fourth category “intrinsic data quality” consists of the following dimensions and their description:

- believability
  “believable"
- accuracy
  "data are certified error-free, accurate, correct, flawless, reliable, errors can be easily identified, the integrity of the data, precise"
- objectivity
  "unbiased, objective"
- reputation
  "reputation of the data source, reputation of the data"

As in the contextual data quality category, the SRS can only show that entities and attributes are present that help users to improve the four dimensions. Again data maintainers are responsible for the data in the database and again standard operation procedures are most important to meet concerns of these dimensions.

Discussion

We could see that an accepted SRS without a set of data quality requirements and a data quality concept should accommodate the 6 dimensions of the representational and accessibility data quality categories. For the two other categories, contextual and intrinsic data quality, the SRS can only give partial support and appropriate data quality constraints must be defined. The result of the presented analysis is that an accepted SRS has impact on data quality according to the dimensions defined by Wand and Strong. Expressed as a number, we have more than 40% of data quality that is covered by an accepted SRS. The question is, if certain dimensions are more important than others so that the coverage of 40% must be calculated differently. Wang and Strong found in their study that accuracy and correctness are the two dimensions that were rated to be the most important, but in their further analysis, they found that the categories are more or less equally weighted.

However, if we consider that our SRS contained no data quality requirements but only described the information system FoodCASE, more than 40% coverage seems to be optimistic. As mentioned in the categories intrinsic and contextual categories, users are responsible for the

data and their quality that is entered if no data quality requirements exist. That would mean that an information system can contain worth possible data quality but an accepted SRS increases the quality on 40%.

Finding 1: For our definition of data quality later in this Chapter, we will not regard the categories representational and accessibility data quality. The dimensions in these two categories are somehow related to data quality in that unsatisfied dimensions can cause wrong user understanding of data or harmful modification of data, which in turn can decrease intrinsic or contextual data quality. But these dimensions do not directly target the quality of data.

During the analysis of the categories intrinsic and contextual data quality, we realised that mostly it was not clear how a dimension can precisely be analysed on food composition data and what data quality requirements precisely belong to a certain dimension in the context of food composition. This uncertainty was caused by defining a dimension with a substantive and a brief description of the meaning. Therefore we gained the following finding:

Finding 2: Dimensions are good starting points to find data quality requirements. Their brief definitions make them simple and intuitive but at the same time they also lack for details. In that sense their definitions and meanings are abstract and therefore are inappropriate for our definition of data quality.

3.2 Study on Data Quality Dimensions, Data Reliability and Data Quality Functionalities

After the SRS, we started the implementation of FoodCASE. Some detail questions arose during the implementation phase and, during the discussions with food compilers, we tried to extract as many data quality requirements as possible. Data quality requirements range from simple ones such as “food name is mandatory” to requirements like “as every nutrient of a food is given as g per 100g edible portion, the sum of all nutrient values must sum to 100g”. We collected data quality requirements and functionalities, and noticed that user opinion differ in the importance rating of requirements and functionalities. We also noticed that there are two main categories that must be considered when designing for the development of a data quality framework. One category contains the above mentioned data quality requirements while the second category contains reliability requirements of empirical data where the exact value is not known. In this category belongs meta information such as “who did the measurement” or “how was the measurement performed”. The reliability of data replaces the dimension accuracy and is therefore an important data quality issue.

As seen in Section 2.6, EuroFIR defined a set of data quality questions for food composition values to determine how reliable the values are. Inspired by that work, we generalised that set

of questions to be generally applicable for the reliability of empirical scientific data collections. We defined a reliability classification that consists of 4 categories that are called sample, analysis, data acquisition and data processing.

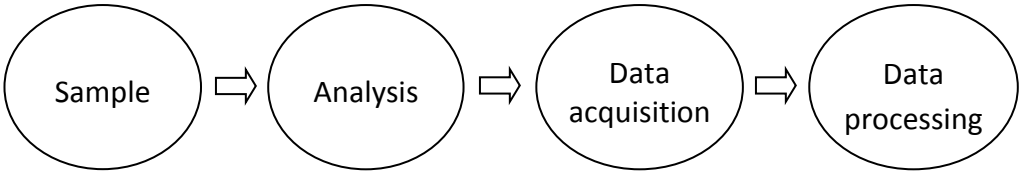


Figure 3.1: Presentation of four categories that can be used to determine the reliability of scientific data. The four categories represent four steps of the scientific data lifecycle.

The categories represent four selected steps of a data lifecycle where the reliability of empirical data is determined. The complete data lifecycle will be presented and discussed in Chapter 5.

- Sample
  The first category contains the two subcategories origin of a sample and sample handling. Information about the sample helps to answer the question if an adequate sampling was performed. If a sample was transported to the place where it was analysed, then also information about the sample handling influences the reliability of data.
- Analysis
  This category concerns the units that perform the analysis of a sample, such as instruments or laboratory, and the measurement method. The more appropriate the measurement methods are and the more information about the unit and the analysis steps exist, the more reliable are the results.
- Data acquisition
  The data acquisition is the next step that influences reliability. In the case of food composition data, food compilers look for publications and laboratory reports and enter the information into their systems manually. Information about the data selection criteria and the data input process such as whether entered data was double checked by a second person, have impact on the reliability assessment of data.
- Data processing
  This category is used when sample data is further processed to generate new data. This category concerns the selection of sample data and the methods used. When data is further processed such as calculating average values, the quality of selected data and the appropriateness of the aggregation methods impact the reliability of new generated data.

We also identified that 5 forms of questions appear in all of the four categories: What-was-done, who-has-done-it, where-was-it-done, when-was-it-done and how-was-it-done. The what-

was-done question clarifies what samples or data are considered. Depending on the categories and on the context, the where-was-it-done and how-was-it-done questions are obsolete and additional forms of questions can occur. As an example consider the category data acquisition and the issue if an entered value was doubled checked by a second person. The where-was-it-done question is obsolete because it does not matter if the person did it at home or in the office. On the other, if data is further processed and only selected values are used for further processing, then the form of question why-was-it-done must be added to make a more reasonable reliability rating.

As some implementation questions remained unanswered, we decided to organise a workshop to discuss these questions and to conduct a survey to validate our data reliability classification. The results of the survey are presented in this section.

3.2.1 Goal

To investigate what users understand by data quality, two approaches were taken: The data quality framework defined by Wang and Strong [27] as well as our reliability classification. In order, we investigated the importance rating of single dimensions and single data reliability requirements. The first goal was to find out if our reliability classification was correct and complete. The second goal was to find out if only single users differ in the importance rating of data quality requirements or if general differences between user groups can be identified.

We added questions about general data quality functionalities with the intention of investigating what functionalities users regard as helpful. Knowing these functionalities, we were able to develop concepts and include them in our data quality framework.

We also added questions about controversial data quality requirements with the goal of establishing if and how the notions of data quality are distributed and where users' notion of data quality ends.

3.2.2 Approach

At a EuroFIR conference, we organised a two hours workshop that had two parts. The first part was to find answers to implementation questions because either the specification did not cover them or we had contradictory information from food compilers. The results of these open discussions were not important for our research objectives and therefore are not presented here. The results are available at

http://www.foodcase.ethz.ch/research/ResultsWorkshopVienna/index_EN.

The second part of the workshop was a survey, in which every participant was asked to answer data quality questions for a food composition database. In contrast to the first experiment, we used a questionnaire with concrete questions. In the first part of the questionnaire, we investigated the dimension of the data quality framework defined by Wang and Strong and

added performance to the category Accessibility as an additional dimension. In the second part, we took the 4 identified reliability categories and added the following two categories:

- General data quality functionality
  Requirements concerning general functionality of software that support data quality management (data quality framework functionalities)
- Personal requirements
  Requirements that we assumed to have large variances because answers depend on individuals and on user groups. Some of the requirements can be assigned to one of the other categories but some cannot.

We formulated questions for the 4 reliability categories and for the 2 additional categories where the above mentioned forms of questions build the basis for the first 4 categories and asked the workshop participants to answer each question if that data quality aspect is important and if yes, how important it is.
To distinguish the different user groups, the participants were asked to give their job position and optionally their name on the survey. The document with the survey can be found in Appendix B.

3.2.3 Results and Findings

24 persons working in the area of food composition participated in the workshop and submitted answers to our questions. We identified four different user groups: Food compilers, computer scientists as creators or system maintainers of food composition software, managers as team leaders of food composition teams and ordinary users. Some persons belong to more than one user group. We decided to count these persons as one person in the overall analysis and as one person in each user group to which the person belongs. The disadvantage of this decision is that it leads to dispersion and the results of the user groups get closer to each other. But the advantage is that considerable differences are significant because the differences exist although some users are counted in both user groups.

The importance of Wang and Strong dimensions

We asked participants to rate the importance of the 15 dimensions by Wang and Strong and our own dimension performance. The scale used in the questionnaire was from 0 to 5, where

0 means not important
---
1 means less important
2 means rather important
3 means important

4 means rather very important
5 means very important

The scale was chosen for the following intention. The scale values 1 (less important), 3 (important) and 5 (very important) defines 3 distinctive values that a user should be able to differentiate. The value 2 (rather important) and 4 (rather very important) were meant to be values between the three distinctive values in the case that a user has the opinion that a dimension does not belong to one of the distinctive values. It is obvious that the scale has a relative meaning in that we wanted to compare the importance of dimensions to each other rather than to have an absolute scale.
Based on this relative scale, we defined also relative analysis ranges as follows: In a scale range from 0 to 5, a value of 0.5 corresponds to 10% difference. We regarded 10% and less to be as more or less equally rated whereas with a difference of 20% and greater, it is clearer that the rating is not equal. We regarded the range between 10% and 20% as intersection area where it is not clear if a difference exists or not.

In the following figure, we present the average importance rating value per dimension for different user groups. For the group of all users, the variance is additionally indicated.

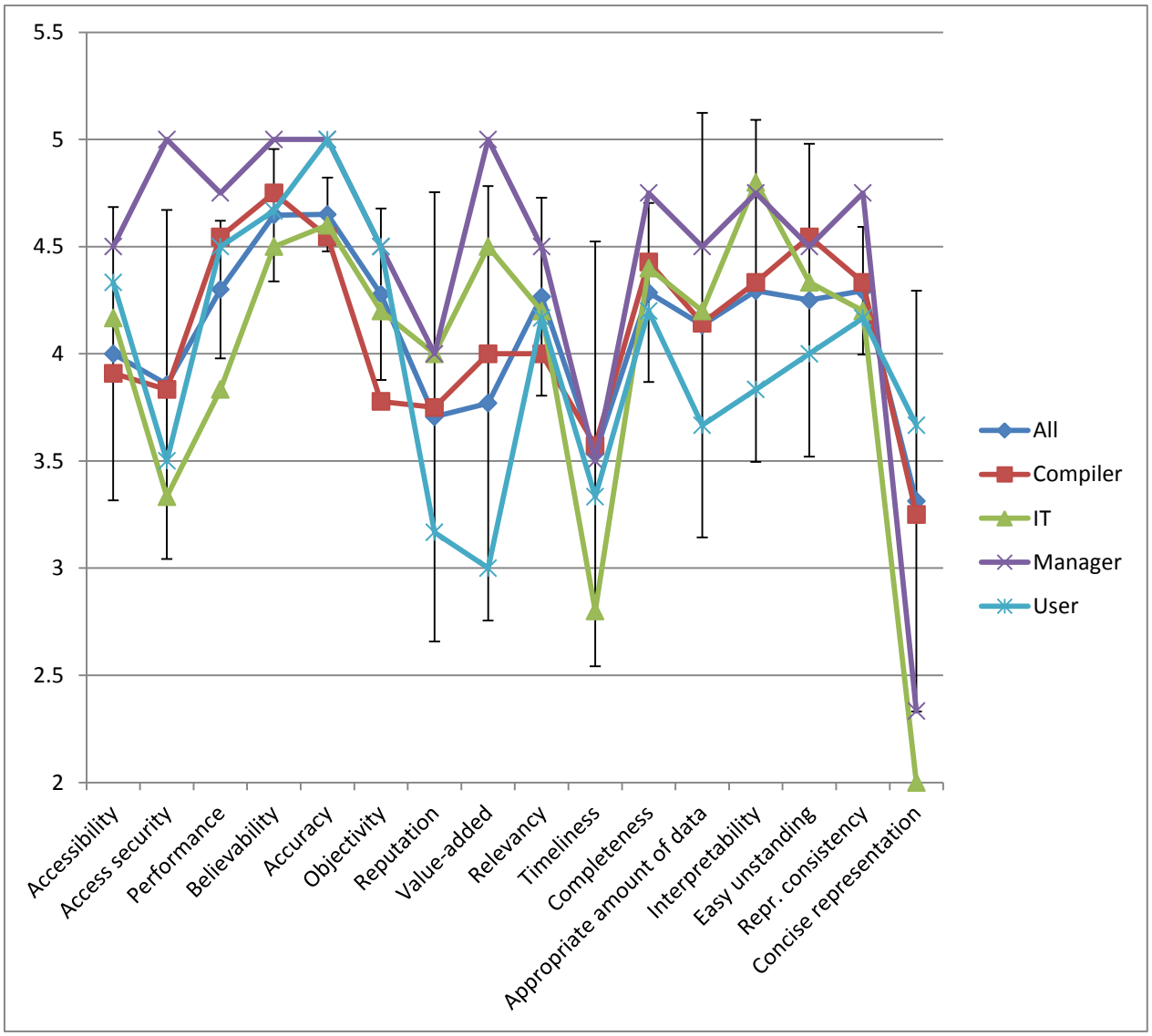


Figure 3.2: Assessment results of different user groups for the 15 dimensions of Wang and Strong and 1 own dimension from food composition database

As expected, the average rating over all participants for all dimensions, including performance, was at least "important" (3.3125 = lowest average value) and hence show that, in the opinion of the participants, the data quality framework of Wang and Strong contains important dimensions for food composition data. The differences of the average values for all users show also that some dimensions are rated to be more important than others. In the case of food composition database applications, accuracy and believability are the most important dimensions. It is interesting to note that although all participants were aware that, in food composition, real values of nutrients are not known, accuracy is rated to be the most important dimension. We must assume that participants mean that a measured value should be as close as possible to its real value and that a reliability test gives information on how accurate a measured value is assumed to be.

If we look at the different user groups it is conspicuous that the manager group has the highest average values for the dimensions accessibility, access security, performance, believability, accuracy, objectivity, reputation, value-added, relevancy, completeness, appropriate amount of data and representational consistency. We count 12 of 16 dimensions in total. On the other hand, the group of computer scientists has lowest average values for 5 of 16 categories and the group of ordinary users has 7 lowest average values. If we try to understand this result, it is clear that food compilers and users have specific intentions for using data. According to these intentions, some of the dimensions are more relevant than others. The main task of the IT-person group is to enable food compilers and users to optimally use the information system. As they are not experts in a specific context, it is not surprising that their ratings for some dimensions differ to the ones of food compilers and users. The group of managers has a more global point of view. Their task is to organise the compilation of food composition data so that food compilers follow well defined operation processes, users' needs are optimally satisfied and that the IT optimally supports the former two issues. We assume that an accumulation of importance rating takes place on the manager's side. If a dimension is important for 2 user groups, then the dimension is rated to be more than "important" in a global rating because more than one user group is affected. Therefore, it seems to be obvious that managers rate most dimensions with highest values.

For the dimensions believability, accuracy, relevancy, completeness and representational consistency the four user groups lay in a small range of about 10% difference, which can be interpreted as to be more or less the same. The dimension ease of understanding is also in this range but the variance for all users is bigger compared to other dimensions. While the dimension performance has a small variance, managers and the computer scientists have an average value difference of almost 20%. On the other side the dimensions access security, value-added and concise representation have user group differences of about 35 to 40%.

Finding 1: These observations lead to the finding that data quality aspects vary for different user groups between 0% and 40%.

Consider the dimensions with largest variances that are accessibility security, reputation, value-added, timeliness, appropriate amount of data, interpretability, ease of understanding and concise representation. The variance in each user group, which are not presented in the diagrams above, reveals that mostly one or two user groups also have high variances in these dimensions. Although we regarded participants that belong to more than one group in each of them, the variances are not caused by these participants alone.

For the dimensions accessibility, performance, objectivity and concise representation all user groups have variance values near the average variance value.

Finding 2: Users within the same user group do not assess some data quality requirements with similar importance.

We asked participants to give further dimension that they think to be important. From 24 persons only one person mentioned “precision” as an additional dimension. As seen in the specification phase, it seems to be difficult for users to formulate data quality dimensions. Although we gave 16 dimensions as examples, only one person mentioned an additional issue. This result is an indication that a helpful data quality model would be one that enables application designers and users to define data quality issues.

General data quality framework functionalities

We asked participants to rate every data quality requirement and functionality according to its importance. The scale was from 0 to 5, where

0 means not important
---
1 means less important
2 means rather important
3 means important
4 means rather very important
5 means very important

In the following figure we present the average importance rating value for the single categories and for every single user group. For the group of all users, the variance is additionally indicated.

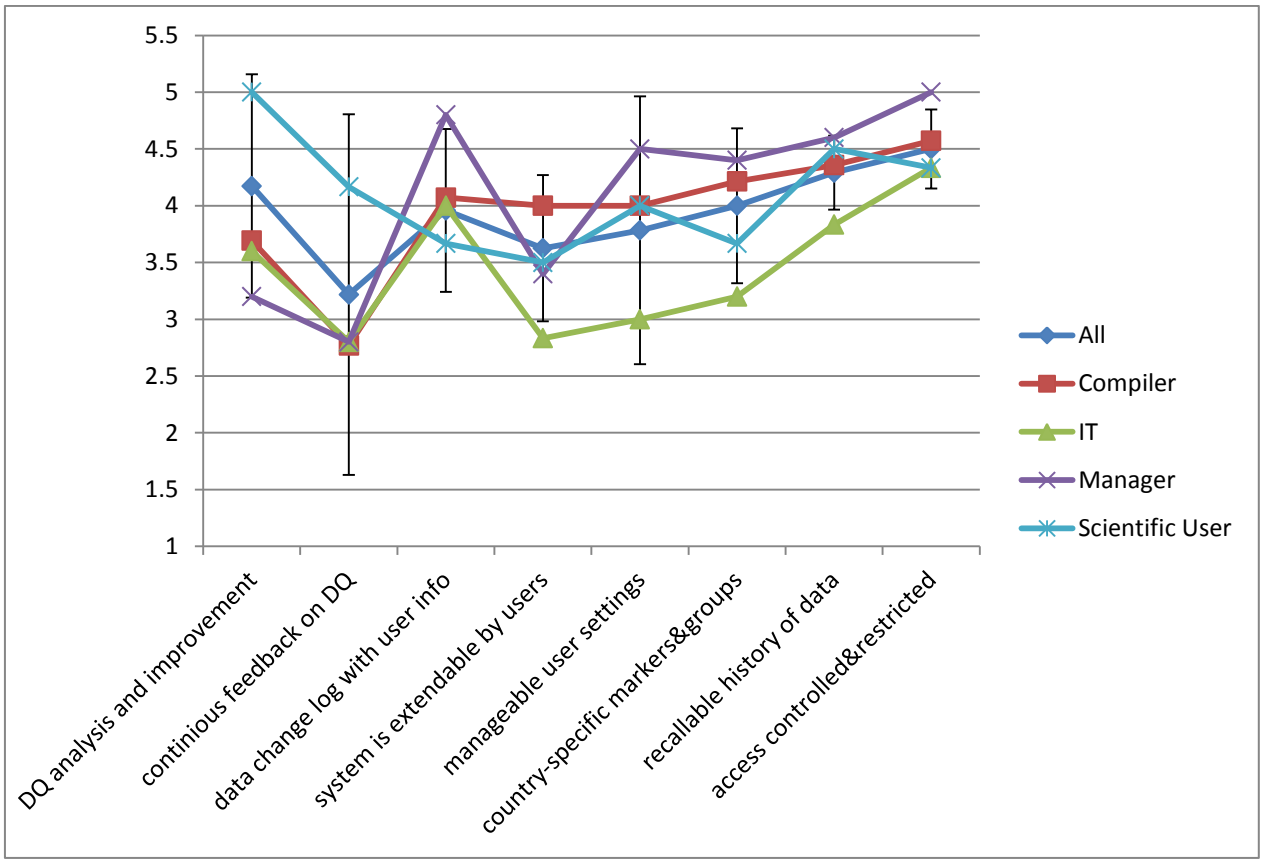


Figure 3.3: Importance rating results of different user groups for 8 concrete data quality aspects that software in general could support

The intention of the category was to get feedback what functionalities users regard as helpful. From these functionalities, we are able to develop concepts and include them in our data quality framework.

The functionality continuous feedback on data quality has the lowest average rating of all functionalities. With three “not important” ratings (13%), it is also the question that had the most “not important” answers in the whole questionnaire. But with an average rating of 3.28 the functionality is important and should be included in a data quality framework. Normally, users do not deny additional functionalities in software as long as they can be helpful. Nevertheless, the results show that a data quality framework should include concepts that cover all of these functionalities.

Reliability categories for scientific data

We asked participants to rate every data quality requirement and functionality according to its importance. The scale was from 0 to 5, where

0 means not important

---

1 means less important
2 means rather important
3 means important
4 means rather very important
5 means very important

In the following figures we present the average importance rating value for the single categories and for each user group. For the group of all users, the variance is additionally indicated.

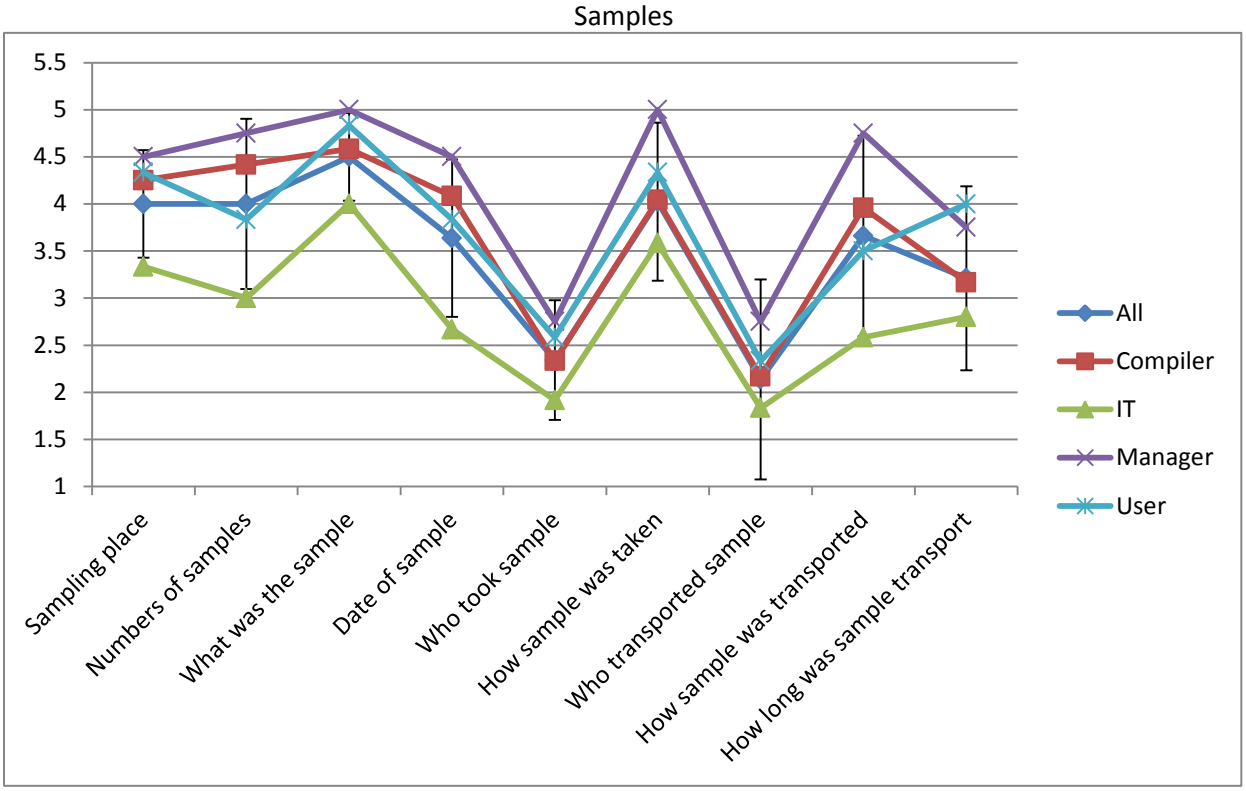


Figure 3.4: Importance rating results of different user groups for 9 concrete data quality issues of the category sample

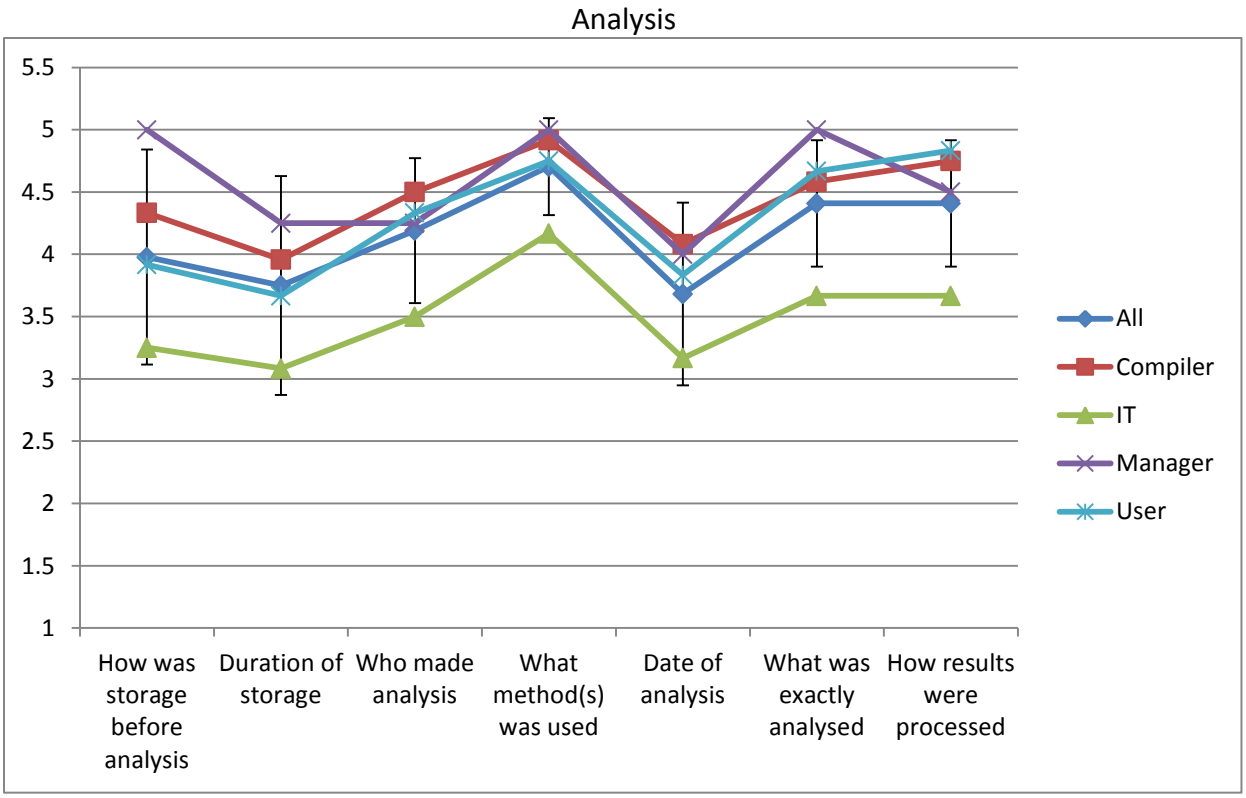


Figure 3.5: Importance rating results for 7 concrete data quality issues of the category analysis

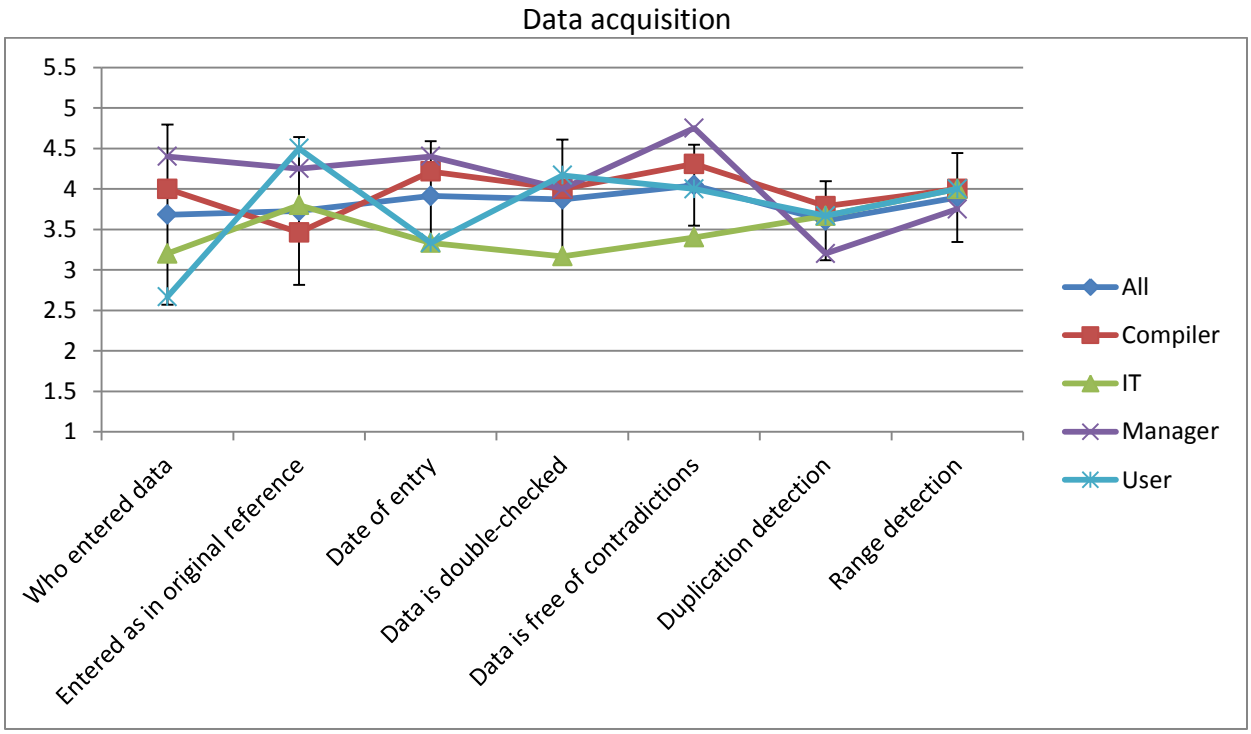


Figure 3.6: Importance rating results for 7 concrete data quality issues of the category data acquisition

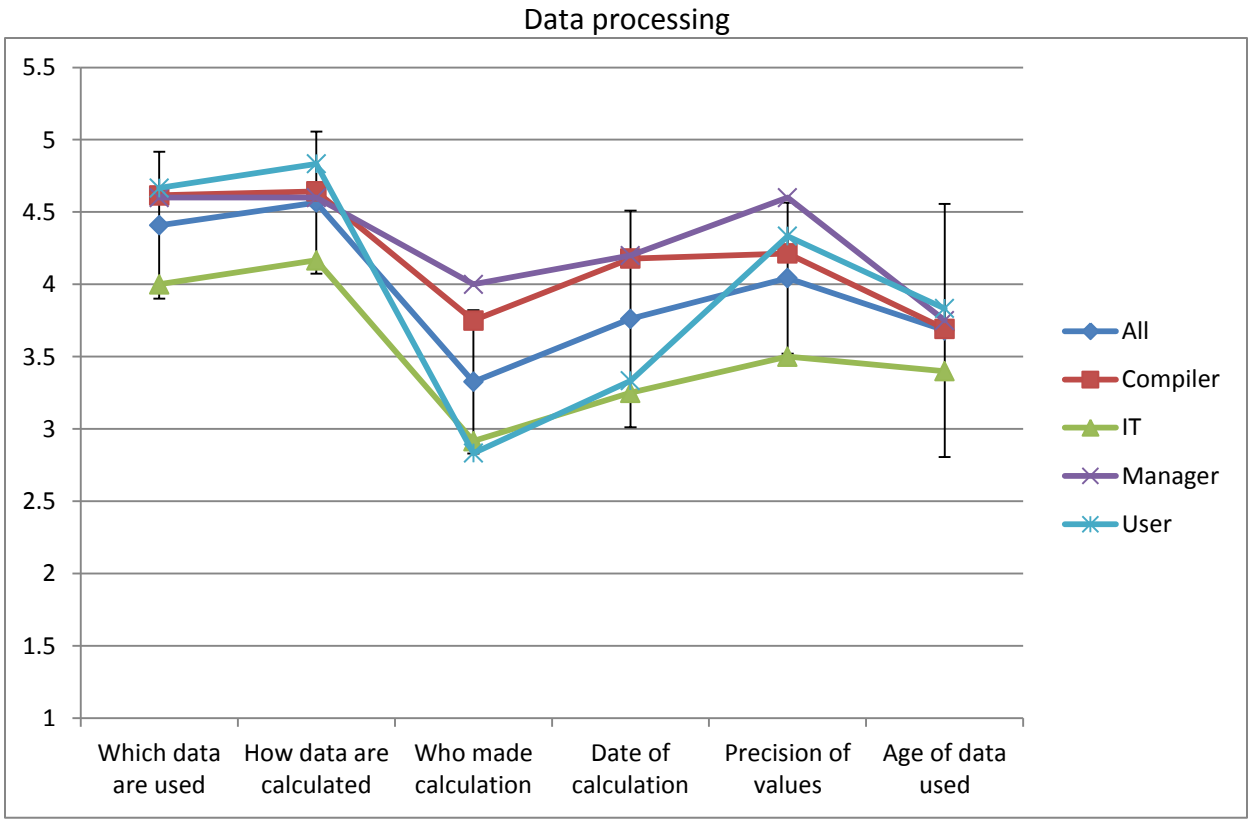


Figure 3.7: Importance rating results for 6 concrete data quality issues of the category data processing

The two questions "Who took sample" and "who transported sample" of the category sample were considered to be less important for the reliability of data. The question "who made the calculation" has also a rating under 3.5 but all other questions have a rating above 3.5.

Finding 3: The definitions of the 4 reliability categories and the 5 forms of questions are rated to be important for the reliability of data where the exact values are not known.

We also asked if there are other requirements that are important. Two users gave the following three answers:

- processes (in the software) need to be more automated (number of mentions: 1)
- approval of data after data entry, before and after processing needs to be a compiler controlled automated procedure (number of mentions: 1)
- the values have to be transferred to the units used in DB (number of mentions: 1)

From this small numbers of feedback items and the limited data quality feedback items in the previous experiment Section 3.1, we came to the following finding, which is valid for at least food composition database systems.

Finding 4: If we consider schematically what kind of persons are involved in database application development we get three persons: a programmer who implements the system, a middleman who speaks user language and programmer language and a user who orders the software. As our experience from various projects shows, a middleman is sometimes missing for whatever reason, and the programmer or the user has to take over the middleman's part. As users have difficulties to formulate data quality requirements, the programmer or the middleman, if one exists, is responsible for considering data quality. This is a dissatisfying finding because programmers seldom have context knowledge. In the case that a middleman exists, the situation is improved.

In all four categories we can observe different ratings for different user groups above 20%. An example can be seen in category "samples" for the question "number of samples" where the difference between managers and computer scientists is 35%. These observations substantiate finding 1.

On the other hand, there are requirements with small variances over all participants. For example, in category "analysis" the question "what method(s) was used" where the average values of user groups compilers, managers, and ordinary users lay within 5%. This finding also emphasises finding 1 in that not all requirements are equally affected. If a common rating scheme for the reliability of data records must be found for a group of food compilers, the questions with lowest variance are preferable because all persons agree on the importance of that issue.

An example that emphasises finding 2 is the data quality functionality requirement of "continuous feedback on data quality". The range of answers in the group of compilers, computer scientists and managers goes from minimum 0 to maximum 5. But also in other categories such examples can be found.

A category comparison shows that the category analysis with an average value of 4.16 is rated to be the most important category for the food composition database application. While the data quality requirements "what method was used to measure" and "how data was calculated in processed data" having 4.7 and 4.56 respectively, are the two most important requirements.

Personal bias

In the category of personal aspects, we asked participants to say to what extent they agree with a given data quality statements. The first line represents the number of answers and the second line the percentage.

Do you agree that data quality functionalities have an influence on data quality?

	no	slightly	partly	mostly	completely
Number of answers	2 8.7%	1 4.3%	8 34.8%	3 13%	9 39.1%

Rounding error of 0.1%

Do you agree that the number of functionalities has influence on data quality?

	no	slightly	partly	mostly	completely
Number of answers	5 22.7%	2 9%	6 27.3%	4 18.2%	5 22.7%

Rounding error of 0.1%

Do you think that data which is not updated can lose quality over time?

	no	rather no	indifferent	rather yes	yes
Number of answers	1 4.5%	1 4.5%	8 36.4%	9 40.9%	3 13.6%

Rounding error of 0.1%

Do you think that missing component values degrade data quality?

	no	rather no	indifferent	rather yes	yes
Number of answers	2 9%	3 13.6%	8 36.4%	6 27.3%	3 13.6%

Rounding error of 0.1%

Do you think that the more fields filled in the database the better data quality?

	no	rather no	indifferent	rather yes	yes
Number of answers	4 18.2%	3 13.6%	3 13.6%	8 36.4%	4 18.2%

Do you think that the number of accesses to your published data is an indicator of data quality?

	no	rather no	indifferent	rather yes	Yes
Number of answers	8 36.4%	3 13.6%	3 13.6%	8 36.4%	0 0%

Do you agree that the representation of data has an influence on data quality?

	no	slightly	partly	mostly	completely
Number of answers	4 21.1%	1 5.3%	4 21.1%	4 21.1%	6 31.6%

Rounding error of 0.2%

Do you agree that a specification has an influence on data quality?

	no	slightly	partly	mostly	completely
Number of answers	0 0%	2 15.4%	4 30.8%	5 38.5%	2 15.4%

Rounding error of 0.1%

As we assumed, some of the questions give no clear answer as to opinions about the aspects has influence on the quality of data or not and the opinions are strongly user dependent. This observation of this category also substantiates finding 2 and leads to a further finding.

Finding 5: Some requirements have a consensus on their importance while others have not. The common ones can be used to define a general assessment scheme for data records. Individual users and user groups need a possibility to extend that assessment scheme or need to agree on a general assessment scheme.

Conclusion

Summarising, we can state that although it is not clear how a data quality model or a data quality framework should look, we gained a better understanding of what data quality means to users. We have seen that users have different opinions about data requirements that must be satisfied to get high quality. Based on the findings we can define the terms data quality, data quality requirement and data quality indicator which were all used in designing a data quality model and framework.

3.3 Definition of Data Quality, Data Quality Requirement and Data Quality Indicator

Data quality dimensions are a starting point when data quality must be measured, secured or improved in database applications. We saw in Chapter 2 that dimensions are based on empirical or logical methods or on surveys where respondents are asked to describe data quality in terms of adjectives or substantives like accessibility, performance or accuracy. The dimensions are therefore on a high level of abstraction whereas concrete measurements can better be applied on relations and attributes. For instance, reconsider the dimension reliability and an apple's vitamin C value of 5 mg. Food compilers would ask: "Who and where was the value measured and what measurement method was used". A further question would be: "Is the laboratory accredited for the measurement method used". Of course they would ask more questions and the example shows that numerous questions can belong to one dimension. Hence, dimensions, as they are defined in literature, tend to be categories rather than concrete data quality requirements.

Another point that was already discussed in Section 2.6 is the dependency of dimensions on each other. The example questions above also affect the dimensions believability, objectivity and free of error.

However, a definition for a data quality dimension is given by Wang and Strong [27] as:

A data quality dimension is a set of data quality attributes that represent a single aspect or construct of data quality.

Using the findings in this chapter and the experience from the specification and implementation phase, we define the following data quality terms:

Data quality requirement
A data quality requirement is a concrete intrinsic or contextual constraint to data that helps to satisfy a data quality dimension. The constraint is related to at least one data attribute value or at least one data tuple. Each data quality requirement must contribute to at least one data quality rating with a certain importance. The importance depends on the user group or even on a single person. Normally, a data quality dimension consists of multiple data quality requirements.

Data quality indicator
A data quality indicator is a concrete intrinsic or contextual constraint to data that can help to satisfy a data quality dimension. The constraint is related to at least one data attribute value or at least one data tuple. Each constraint is an indication about data quality but in contrast to data quality requirements it is likely that the constraint affects data quality but the affection is not proven. Each data quality indicator must contribute to at least one data quality rating with a certain importance. The existence of a data quality indicator and its importance depend on the user group or even on a single person. A data quality indicator is a subtype of a data quality requirement.

Data quality and data quality value
Data quality is the degree to which a set of data quality requirements and data quality indicators is satisfied. Data quality is determined by the data quality value that is the result of an aggregation function on the set of data quality requirements and data quality indicators. As the importance of data quality requirements and the existence and importance of data quality indicators are dependent on a user group or person, the aggregation function is also dependent on a user group or even on a person.

In our definitions only intrinsic and contextual requirements are included. Therefore, the representation and system quality aspect belong to other quality definitions that are not further investigated in this thesis. As an example, we define that system security does not decrease data quality per se. A weak system security offers the possibility to falsify data what in turn can decrease data quality. But it is possible to have high quality data in an unsecure system and that is why system quality does not belong to our data quality definition. A similar argument is used

for the representation aspects. They have influence on user's interpretation of data but they do not decrease data quality per se.
A constraint is usually formulated by a question or a statement. In general it is possible to rephrase a question into a statement and a statement into a question. The definition says that a question like "is the reputation good?" is not a data quality requirement because reputation is a data quality dimension and is not concrete enough. An expert in the area of food composition would ask more detailed questions like the examples above. Hence, the statement or the question of a requirement has to be atomic so that no sub-requirements are possible. Otherwise we are talking about dimensions or sub-dimensions.
To also oblige requirements and indicators to be concrete, they are defined to target an attribute's value or a data record/object instance. Requirements and indicators are not targeted to an attribute or relation but on the content of them. If an attribute or relation does not exist, then a design deficiency is detected. Therefore, requirements that do not concern concrete data values are no data quality requirements.

We define a data quality requirement and data quality indicator to be dependent on a user group or individual user which means that the user group or the individual user needs only a subset of the requirements and indicators to be satisfied for his intended data use. This is reflected in the definition that every data quality requirement and every data quality indicator must at least belong to one data quality rating. But a total data quality value is determined by an intention-independent set of data quality requirements and indicators.
The user-dependent definitions also allow contradictory requirements to be specified as they are used for different intensions. In the total data quality value assessment, the data quality designer has to decide how these two requirements should be taken into account. The definitions are not restricted to the relational paradigm. It is also valid for the object-oriented paradigm where the term data attribute represents also an object attribute and where data tuple must be replaced with object instance.

3.3.1 Data Quality Indicator where Real Values are Unknown

In addition to the above definitions, we have to discuss why data quality indicators, as seen in Section 3.2, make sense in combination with values for which the exact and real value is not known.
As we have seen in Chapter 2, data quality indicators are used, for instance, in empirical environments such as food composition, to indicate how reliable a value is. But for some of the values such as nutrient values, the exact and real nutrient value is not known. Empirical measurement instruments and empirical measurement procedures were developed to determine these values. How close these empirically measured values are to the real values is an unknown variable. In statistics, this unknown variable is called an "error" and has two

additive parts: Systematic error, which always occurs when measurement is performed in the same way, and random error, which can vary from measurement to measurement. Expressed in a mathematical manner:

$$\varepsilon = \dot{x} - x$$

where ε is the error, $\dot{x}$ the true value and x the empirical measured value. If we have more than one measured value and assume that they are independent and normally distributed with mean μ and standard deviation σ, then we have

$$x_1, \dots, x_n \sim N(\mu, \sigma^2)$$

and the sample mean is

$$\bar{x} = \frac{x_1 + \dots + x_n}{n}.$$

The statistical errors are then

$$\varepsilon_i = x_i - \mu$$

A data quality indicator gives an indication of the quality of a value in terms of how well it is documented, how old it is or how values are measured, for instance. They determine how reliable a value is, which means that it is assumed that the error is small or in the optimal case zero. But we have to admit that neither the real value nor the error is known. It is possible that we have a highly reliable value but the error is large. On the other hand, we can have measured data that is of low reliability but fits exactly the real value. In a statistical sense, the conclusion should be that a data quality indicator and the error are independent variables. This conclusion is not wrong but we prefer another conclusion because otherwise data quality indicators would not make sense. We assume that data quality indicator and the error can be independent but mostly are not independent. We think that our assumption is not completely wrong because in some cases good data quality goes along with serious measurement and serious documentation which means that the measurement was performed to the best of laboratory workers' knowledge. Hence we assume that the error is minimised. If we have more than one independent measurement, we then can even argue that data is evidence based. So we assume that a data quality indicator has a good reason to exist but users have to be aware that indicators are used to estimate unknown variables.

3.4 Summary

The overall goal of this chapter was to find out what users mean by data quality and we presented our way of achieving this goal. We first have presented an experiment in which we extracted data quality issues from the feedback of the SRS of FoodCASE.

Based on the outcome of the experiment, we introduced a reliability classification for food composition data and have collected a list of data quality features. We then presented a user study in which we evaluated our reliability classification and our collection of data quality features. Through the experiment and the user study we have shown that data quality

dimensions are a starting classification for data quality requirements but that they have disadvantages in overlapping meanings and in the abstraction of details that are important for the assessment of data quality requirements.

We also presented some fundamental findings from this user study that lead to the definitions of the term data quality, data quality requirement and data quality indicator.

4

Towards a Requirement-Oriented Data Quality Model (RODQ Model)

In an information system, data producers and data consumers can have numerous quality requirements of data. A measurement is needed for each of these requirements to determine if a data record satisfies the requirement or not. In addition, these requirements have a certain association with each other, for instance, one requirement is more important than another. Having all these requirements, it is also useful to have a summary rating over all data records and over all requirements. A data quality model helps to circumvent these challenges and helps also to design data quality requirements into an information system. It can therefore help to prevent data quality deficiencies and their time consuming corrections.

In this chapter we present such a data quality model called Requirement-Oriented Data Quality Model (RODQ Model). In Section 4.1, the objectives of the RODQ Model are derived from the analysis of what existing data models do not satisfy. In Section 4.2 we present our classification of the data quality requirements that we collected during the implementation of FoodCASE and which will be used in the RODQ model and in the RODQ framework. The elements of the RODQ model are defined in Section 4.3 and, in Section 4.4, the schema definition for the RODQ schema is presented. The model and the schema contain a data quality assessment element

that can be used for quality rating of different data types. This general data quality assessment element is described in Section 4.5. Finally, some advantages and disadvantages of the RODQ model and schema are discussed in Section 4.6.

4.1 Objectives of the RODQ Model

In 1975, the Data Base Management Study Group of ANSI/X3/SPARC published a journal article in which they emphasised three levels of a data model schema corresponding to the external, conceptual and internal view of a database [65]. In a later report [66], an external view is summarised as "... used to describe the data as seen by the programmer", the conceptual view as "... describes how the information of an enterprise is modelled in the database" and the internal view as "...used to describe the data to the system". Nowadays the three levels are referred to as the conceptual schema, the logical schema and the physical schema. In the conceptual schema, real world objects as well as their relations are modelled and examples are the ER model [67], UML [68] or the Object model [69]. The logical schema is used to map the conceptual schema into the DBMS used that can be, for instance, relational or object-oriented. The physical schema is used to describe physical issues to capture data in a storage medium such as partitions and index structures. The physical layer is implemented in a DBMS. Physical models are not being considered in this thesis.

To understand how far conceptual and logical models cover data quality requirements, we distinguish two categories: syntactic and semantic requirements.

Syntactic requirements request data to be in a certain form. A typical example is an attribute that should be of type integer. Syntactic requirements are also data quality requirements because they validate or invalidate data and hence increase or decrease data quality. For instance, if an attribute is of type integer then an attribute value "a" invalidates the data record and at the same time decreases the quality of the data record because the value does not make sense for that attribute. This example can be covered by a type definition for the attribute in a logical model. But let us consider the example of a zip code in Switzerland that has 4 digits. We analysed programming languages and SQL dialects such as Java, C#.NET, Transact-SQL and PostgreSQLDialect and found that data type definitions are limited in that an integer type or float type cannot be restricted to a certain length as it would be requested in the case of a zip code. In such a case a check constraint must be used. A counterexample that is not possible to design using constraints, is when two or more attributes from different tables are involved. Except foreign keys, the above DBMS do not support such constraints.

A semantic requirement specifies that attribute and instance values follow a certain context logic. An example in the area of food composition is that a food item should have at least four nutrient values for energy, fat, carbohydrate and protein. This requirement cannot be covered by a conceptual or logical model. The reason is that a cardinality constraint (4:n) on the

association between food and nutrient value cannot be used as the input of data for food items and nutrient values are not at the same time.

From the examples above, we conclude that syntactic requirements can partially be satisfied by conceptual and logical models whereas semantic requirements mostly cannot because they are mostly context dependent. The reason is that the objectives of conceptual and logical models are to support the design of information systems by abstraction and definition of entities or objects, their attributes and associations. They are used to design data objects to have the right form but they are not designed to guarantee that data content has the right semantics. Data type definitions focus on possible values that can be assigned to attributes and what operations can be performed on attributes. Constraints are used in database management systems to restrict the domain of attributes. By conception, they can restrict attributes, tuples and even whole relations or collections. But in many database management systems such as PostgreSQL, SQL Server and MySQL constraints can only be applied to data records. Constraints are checked on the data operation insert, update and delete. A timeliness requirement like data should not be older than 5 years (outdated) cannot be handled with constraints as the outdating happens over time and not during data manipulation. Normally a constraint checks the integrity of data and the storage of that data is not allowed if the constraint is violated. But the timeliness requirement example does not disallow data storage but wants to say that data should not be used for further calculations and publication. Another disadvantage of existing constraints is that they are either satisfied or violated. It is sometimes necessary to have a scale between satisfied and violated to make an assessment. As an example consider an attribute that can have 3 values “a”, “b” or “c”. Value “a” would have best quality assessment, whereas “b” would have 50% of quality and value “c” would have 0% quality. Hence, constraints are not helpful in situation where we want say that good quality data should have at least value b for a certain attribute.

The RODQ model is aimed at filling this lack in modelling of an information system. It is a conceptual model that describes data quality requirements and different types of assessments. The model can be used in parallel to any established models such as the ER model or UML but is not dependent on them. The objectives of the model are:

- Separate data quality consideration from database modelling and higher order language modelling to focus only on data quality requirements
- Provide a definition to design a Requirement-Oriented Data Quality schema (RODQ schema)
- Provide a formal concept to define functions that measure data quality
- Abstract data quality consideration in graphical form to discuss with users

4.2 Classification of Data Quality Requirements

In Chapter 3, it is mentioned that data quality requirements were collected during the specification and implementation of FoodCASE. We found 156 data quality requirements and investigated their properties to determine appropriate categorisation. We analysed properties that influence the modelling and implementation of data quality requirements in an information system. We identified 6 classifications: scope, type, data dependency, user dependency, time dependency and answering. Every data quality requirement can be assigned to a category in each of the 6 classifications.

The scope defines if a data quality requirement is a hard constraint, a soft constraint or an indicator. A hard constraint is a necessary condition that the quality is sufficient to understand and/or use data. A violated hard constraint means that data quality is deficient. Reasons can be, for instance, data is not understandable due to its insufficient form, the context in which data can be used is not defined or data contains contradictory information. We often use the term "data is invalid" or "data is deficient" which in fact means that the data's quality is not sufficient to understand and/or use it. Hard constraints are typically used to define what form data must satisfy to be stored in the information system and therefore are syntax checks on data. Hard constraints are mostly implementable in an information system because they represent the classical satisfy or violate check constraints.

A soft constraint influences data quality in that it increases or decreases quality. In contrast to a hard constraint, a violated soft constraint does not make data quality deficient but only decreases data quality. Soft constraints are typically used to assign different quality ratings to different attribute values or to increase quality if certain attributes provide information. As defined in Section 3.3, a data quality indicator is a data quality requirement that relates to some kind of ideal. Like a soft constraint, an indicator does not make data quality deficient. It is often used when real values are not known. A typical example is the age of data that indicates that the data can be outdated. The impact of these classifications can be seen in the RODQ model, Section 4.3.

A data quality requirement can be of type atomic, composite or quality key. An atomic requirement concerns attributes or object instances in a database whereas a composite requirement concerns also other requirements. For instance, an atomic requirement is that a "food name must have at least two letters" or "a food item must have at least 4 nutrient values". A composite requirement could be, for instance, a requirement that postulates that the two atomic requirements are satisfied. Such a composite requirement is used to calculate a summarised data quality assessment value as we will see in Section 4.3 and 4.5. The type quality key requirement is used to calculate some sort of summary or aggregate data quality assessment value that is normally disseminated together with the data. In the model, Section 4.3, the impact of a data quality requirement's type is discussed in more detail.

The classification data dependency has the two categories data general and data individual requirement. The data individual category represents requirements that are only valid for

certain data. As an example consider a nutrient value and its measurement method. If certain methods are used, additional parameter information should be provided. In consequence, the quality of nutrient values cannot be compared as some of them have more requirements than others. In contrast, a data general requirement can be applied to all possible values or all possible instances. The classification data dependency has impact on the assessment scale of data quality and will be discussed in Section 4.5.1.

The classification user dependency is directly derived from the findings in Section 3.2.3 that user groups and even users within the same user group have different data quality requirements. It is clear that these different requirements are intention independent as the task of the interviewed food compilers is to collect food composition data independently of later use of that data. The classification user dependency contains the categories user general and user individual requirements. For user individual requirements, the community of users does not find a consensus and hence some of the users support the existence or a certain importance while others deny the existence or define another importance for them. If users agree on the existence and importance of a data quality requirement, it belongs to the user general category. The classification user dependency has also impact on the assessment scale, Section 4.5.1.

The last classification time dependency has the two categories *during input* and *after input*. The *during input* category contains data quality requirements that can be validated during data input. The requirement "the sum of all nutrient values must be close to 100 g" is not in this category but is in the *after input* category. The reason is that all nutrient values must have been entered before this requirement can be validated. At the input time of a new food, this requirement cannot be satisfied. The impact on the quality assessment is presented in Section 4.5.1.

In the classification answering there are the two categories automatic answer and user answer. In the automatic answer category are requirements for which the answer as to whether a requirement is satisfied can be automatically evaluated using a predefined rule. In contrast, the user answer category contains requirements for which the answer cannot be automatically done but needs user input. For instance, a data quality requirement that an English name must have two digits can be evaluated automatically. In contrast, a requirement like "is it clear what part of an animal was taken for sampling" is hardly determinable by the system and needs user input. The impacts of the two categories are further discussed in Section 5.4.2.

4.3 Model Elements

The elements of the RODQ model are data manipulation activities, data quality objects, data quality attributes, and data quality requirements. There is also one type of association called uses/includes. All elements and the association are presented in the following subsections.

4.3.1 Data Manipulation Activity

Data quality must be managed on the four basic database operations create, read, update and delete (CRUD). In the case of create, data quality is generated along with the data, in the case of update and delete, data quality can change, and in the case of the read operation, quality must be evaluated to be displayed to users.

On the application layer and the presentation layer different user and automated activities can be identified. For instance, typical human computer interactions are data entry, data update and data retrieval. Typically automated activities are interfaces to other applications from where data is imported or the automatic generation of new data through calculations. Data quality must also be managed on each of these activities because these activities consist of multiple CRUD operations.

The data manipulation activity element represents such a user or automated activity where data manipulation is performed. In a flowchart, such an activity can be seen as a single step. The name of the activity is dependent on the application and on the context.

4.3.2 Data Quality Object

A data quality object corresponds to a real world object for which data quality checks need to be performed and a data quality object is associated to one or more data manipulation activities. Normally, the set of data quality objects is a subset of the entities that are defined in a conceptual model such as the ER model. It is a subset because often other entities that are related to a data quality object are quality checked in the context of the data quality object. For instance, consider an entity Nutrient value, an entity Food and an entity Synonym that have one-to-many associations between them:

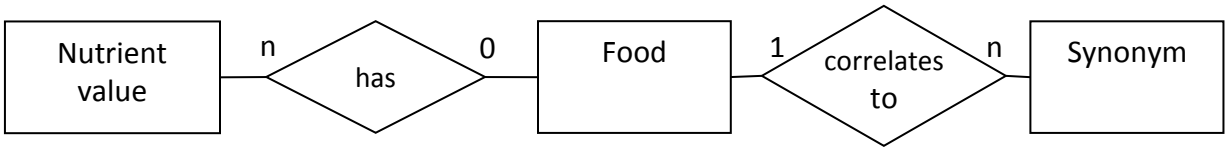


Figure 4.1: An example of three data quality objects having associations between them.

Assume we have only two requirements:

1. A food should have at least one synonym in English and
2. A food should have at least one nutrient value for energy, fat, carbohydrate and protein.

These two requirements are checked for every food instance and not for the entity Nutrient value and Synonym. Hence, only Food needs to be defined as a data quality object.

The set of data quality objects is strongly dependent on the existing data quality requirements and on the context. Mostly, entities become data quality objects that have many relations to

other entities because the related entities contain additional information and, with an increase in the number of relationships, the possibility increases for data quality requirements. But we assume that this rule is not generally valid.

4.3.3 Data Quality Attribute

A data quality attribute is a property of a real world object that is included in a data quality requirement. Similar to the data quality object, the set of data quality attributes is a subset of the attributes that are defined in an ER model. It is a subset because often not all attributes of a conceptual model are quality checked and hence there is no need to define them in the data quality schema.

In the example above, we have seen that a food should have at least one synonym in English. This means that the attribute language of entity Synonym is needed to evaluate the requirement. This attribute becomes a data quality attribute although the entity Synonym is not a data quality object. This is often the case when only a few entities of a conceptual model become data quality objects but other associated entities are used to evaluate data quality. As will be shown later, we use a special annotation in the schema to indicate to which real world object a data quality attribute belongs.

4.3.4 Data Quality Requirement

A data quality requirement can be as simple as “ID must be of type integer” or can include many records like “the sum of all nutrient values must be close to 100 grams”. A data quality requirement has a name, a question or statement, a quality value, an assessment schema, a scope and a type.

The name is used to define a unique identification of the data quality requirement.

The question or statement is used to formulate the requirement and focuses on the data quality attributes, data quality objects or on other data quality requirements. In the example of “ID must be of type integer”, only the data quality attribute id is associated with the data quality requirement and hence the focus of the question or statement is targeted on that data quality attribute. The question or statement has a set of possible answers or possible statements to answer. An answer or statement can be of any type.

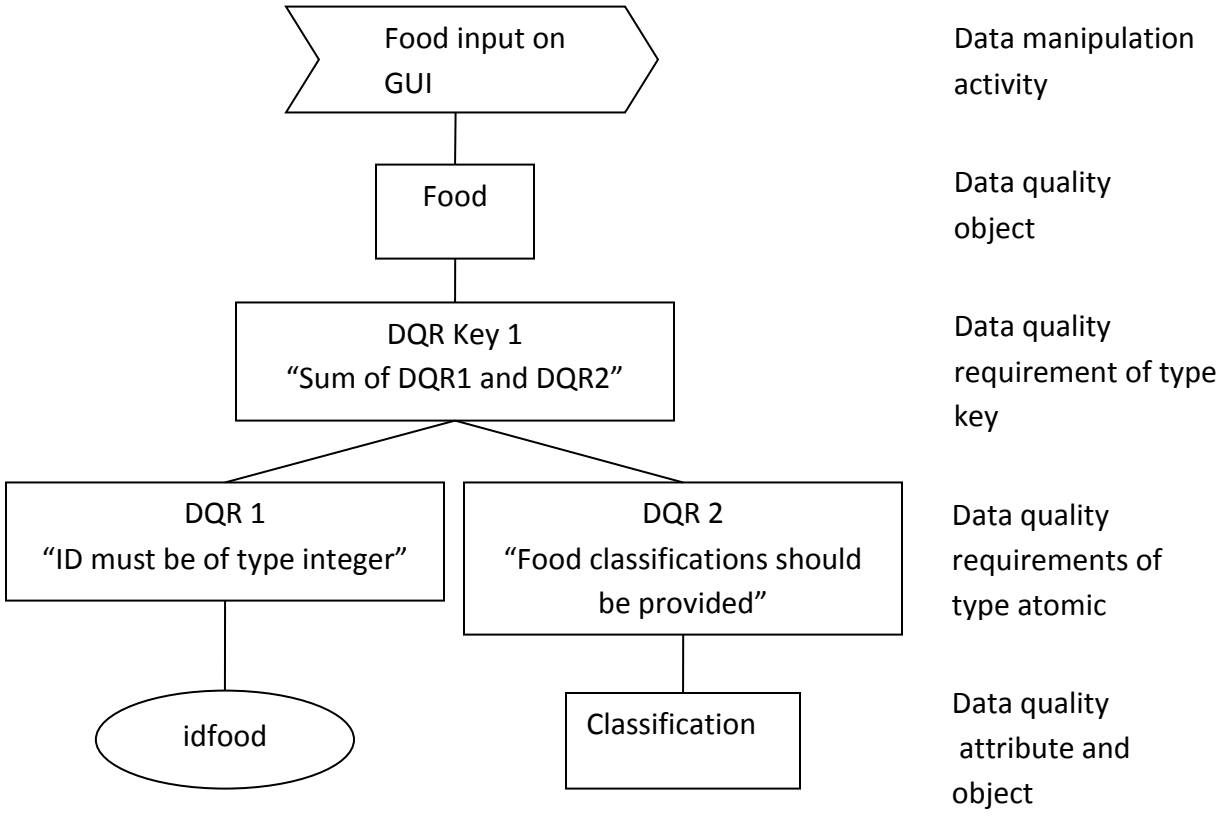
An assessment schema is applied to the set of possible answers or possible statements to get a quality value for each answer or statement. The quality assessment is a function that transforms possible answers into the domain of quality values. The assessment schema can have any form from prose text to mathematical functions. An assessment schema is valid as long as the schema can be implemented using a programming language.

In general, quality values can be of any type. Nevertheless, we propose to use integers or floats as it simplifies later calculations of summarised data quality values in the way that mathematical functions, such as weighted sums, can be applied.

A general valid approach for a quality assessment schema will be presented in Section 4.5.

The scope corresponds to the scope classification defined in Section 4.2. The set of answers to hard constraint questions or statements is typically two-valued: satisfied and not satisfied whereas the set of soft constraints and indicators are context dependent.

The type of data quality requirement corresponds to the type definition in Section 4.2. An atomic requirement is only associated with data quality attributes or data quality objects. A composite requirement is the opposite and is dependent on other requirements which must first be assessed before the assessment function can be performed. The type quality keys means that the quality value of the requirement represents a summarizing quality value and that it is associated with a data quality object. It is possible to have more than one quality keys for an object. As seen above, the objective of a quality key is to have a simplified representative quality value that can be published with the data. A quality rating with many of data quality requirements would be complex and time consuming for some users and therefore a key can be regarded as hiding quality details behind a simplified quality value. The following Figure 4.2 shows an example:



DQR = Data Quality Requirement

Figure 4.2: A simplified RODQ schema that shows the different RODQ model elements. The schema is used to model the quality assessment of a food item.

In Figure 4.2, we can see on top that the RODQ model schema focuses on the input activity of food data. Data quality requirement key 1 (DQR Key 1) consists of two data quality requirements DQR 1 and DQR 2. Let us assume that DQR 1 is a hard constraint and DQR 2 is a soft constraint. These constraint types are not yet visible in the schema. A Food object has an n:m relation to Classification and so the data quality object Classification is associated with DQR 2.

4.3.5 Uses/Includes Association

A uses/includes association is a connection from a data quality attribute, a data quality object or a data quality requirement to a data quality requirement. As seen in Figure 4.2, an association means that the requirement has a relation to the referenced object, attribute or requirement in the sense that they have to be present in the question or statement and hence are also regarded in the assessment schema.

The same association is used to connect a data quality requirement to a data quality object and a data quality object to a data manipulation activity.

4.4 Schema for the Requirement-Oriented Data Quality Model

Having defined the elements of the Requirement-Oriented Data Quality Model, we are able to define the schema elements of the RODQ model schema.

Element Name	Element Symbol	Description
Data manipulation activity	Name	Represents any kind of user or automated activity on the representation or middle tier of an information system
Data quality object	Name	Represents a real world object that can be distinctly identified
Data quality attribute	[Name:] Name	Represents a property of a data quality object. If necessary, the name of the data quality object can be indicated in front
Data quality requirement	Name Question/Statement Quality value = Assessment-schema()	Represents a requirement for quality data. The question or statement describes the requirement. The quality assessment schema is a function that returns a quality value. The scope and type of a requirement is indicated by the surrounding border, see the following table.
Uses/includes	→	Represents a relation from a source element to a target element where a source can be a data quality object, data quality attribute or data quality requirement and a target can be a data quality requirement, a data quality object or a data manipulation activity.

Table 4-1: Listing of the elements in the RODQ model schema.

The border of the requirement defines what scope and type classification a data quality requirement has. Whether a type is atomic or composite can be seen by the association arrows and does not need a visual differentiation. The following tables show the different border styles.

Type Scope	atomic or composite	quality key
hard constraint	Name ...	Name ...
soft constraint	Name ...	Name ...
indicator	Name ...	Name ...

Table 4-2: Type and scope combinations of data quality requirements together with their graphical representation in a RODQ schema.

To simplify a schema it is possible to build data quality schema without a data manipulation activity.

The RODQ schema applied to food composition data

As an example for the RODQ schema we consider a food composition value such as 5 milligram of Vitamin C in an apple. In Figure 4.3 we focus on the food composition value object and its attributes id, value, unit and measurement method. The measurement method is the method that was used in the laboratory to evaluate the value. To get a better overview, no data manipulation activity is used and the food composition value object is indicated twice.

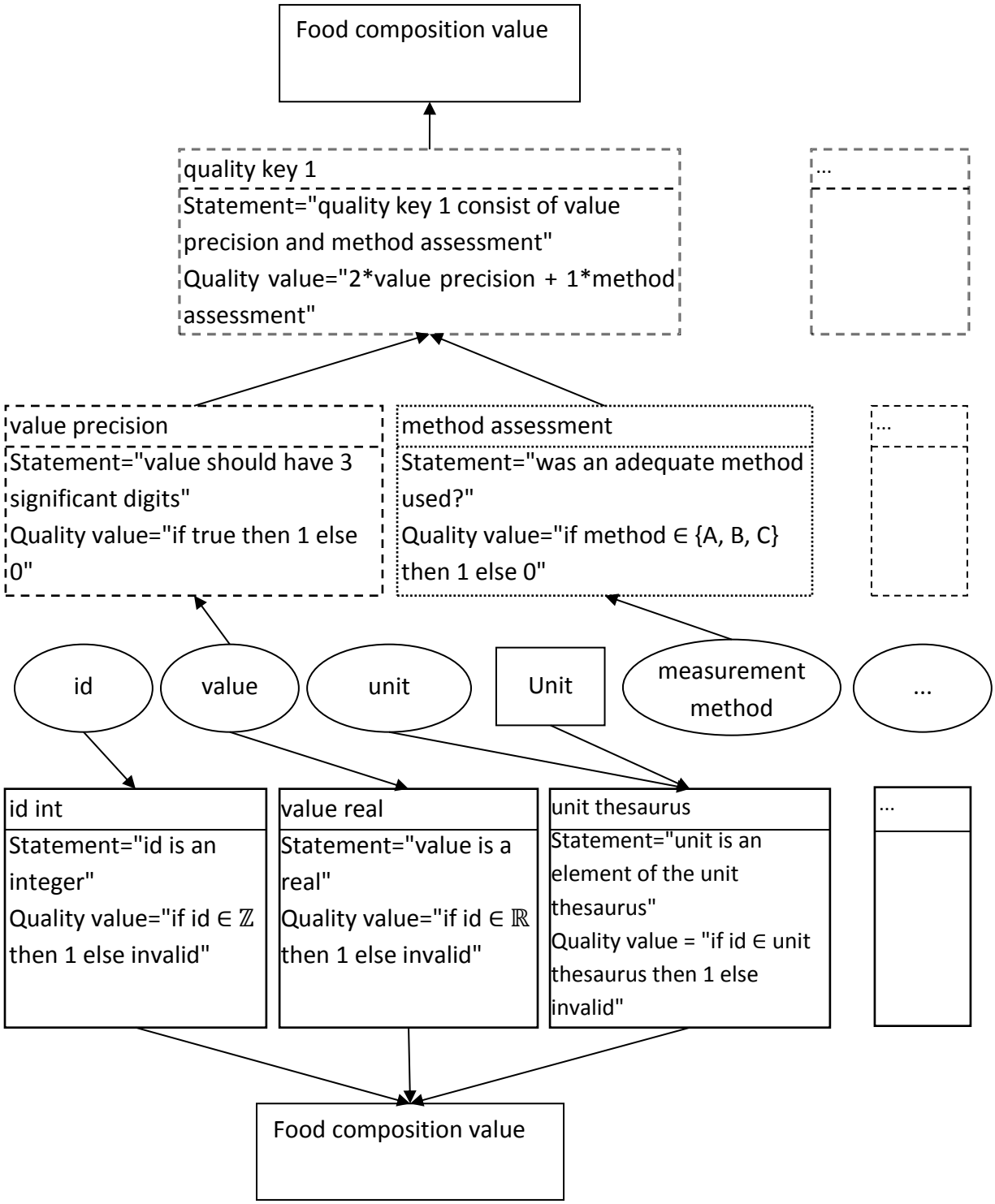


Figure 4.3: An example of a RODQ schema for a food composition value. Hard constraints are below data quality attributes and object and soft constraints and one indicator are above.

The food composition value object in Figure 4.3 has 4 attributes and 6 data quality requirements. The lower 3 requirements are hard constraints and can directly be implemented in a logical schema while the upper 3 requirements cannot. The upper 3 requirements build a

tree with a key and two leaves. One leaf requirement is of scope soft constraint while the other leaf is of scope indicator. The quality key requirement is a soft constraint. See Section 4.6.1 for how different scopes and types are combined. Although the upper 3 requirements build a tree, it is not generally a tree structure. A data quality requirement can have more than one parent and, in a mathematical sense, it is a directed acyclic graph. But, in most cases, a data quality requirement has only one parent and therefore we call it a tree schema.

4.5 Toward a Data Quality Assessment Schema

Until now, the assessment schema was described to have any form from prose text to mathematical functions as can be seen in Figure 4.3. If we consider that an information system has an underlying database and we are able to access the attributes of the objects stored in the database, we are able to define a general assessment approach. In order, we first show how an assessment scale is assembled in Section 4.4.1 and then show a formal approach for an assessment schema in Section 4.4.2.

4.5.1 Assessment Scales

The factors that influence how an assessment scale is assembled are the scope of a data quality requirement and the finding in Chapter 3 that users of the same user group can have different weightings for certain data quality requirements. Therefore, 4 of the 6 requirement classifications defined in Section 4.2 must be considered to determine an assessment scale. Figure 4.5 and Figure 4.5 are used to describe the ranges of an assessment scale.

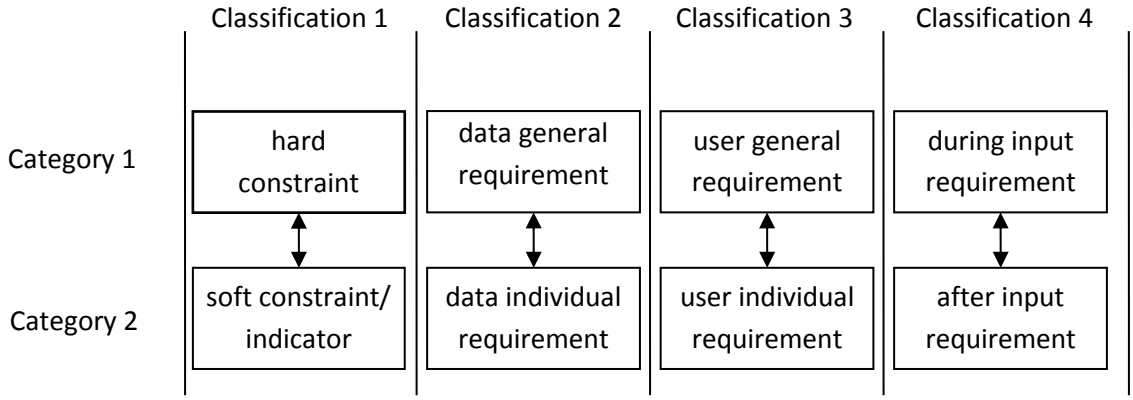


Figure 4.4: 4 of the 6 data quality requirement classifications that influence the data quality assessment scale.

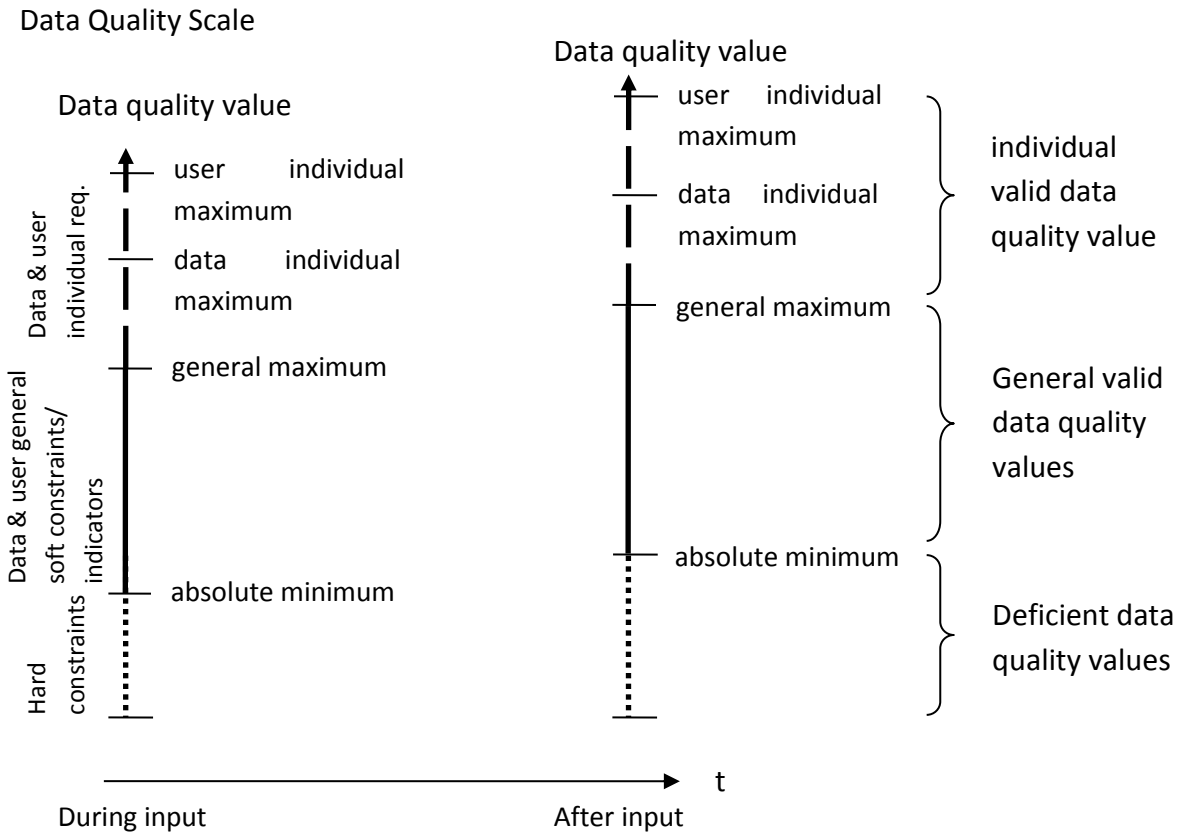


Figure 4.5: Assessment classification and their categories' impact on the data quality assessment. The assessment scale increases over time as some requirements can first be evaluated when all data is stored.

The two data quality value axes (bold arrows in Figure 4.5) go from bottom to top and represent the data quality rating of a data record. The higher a data quality value is the more requirements are satisfied with respect to their importance. To explain why there are two scales (arrows) having different sizes, we consider each of the classifications and their categories in Figure 4.4.

Each data quality requirement belongs to one category of each classification presented in Figure 4.4. The first classification has the categories hard constraint and soft constraint/indicator which correspond to the scope classification. We have seen in Chapter 3 that hard constraints invalidate data if they are not satisfied. In Figure 4.5, this fact is represented in that a scale has an absolute minimum, under which the quality of data is deficient. It can be argued that the area under the absolute minimum is not a range but a single point for deficient data. But as there is mostly more than one hard constraint, we see it as a small range, on which the amount of unsatisfied hard constraints influences how far the quality of a data record is from the absolute minimum. It is only a small range because the main message is that data is deficient and therefore the range under the absolute minimum does not need to be distinctive. If a data record satisfied all hard constraints, it has the absolute minimum data quality value.

The next categories of requirements that influence a data quality value of a data record are soft constraint/indicator, data general requirements and user general requirements. As seen in the description of the classification, soft constraints and indicators do not make data quality deficient. Their impact on the assessment scale is the same and therefore these two scopes are summarised. The three categories together determine the section on the data quality scale from the absolute minimum to the general maximum. General means that these requirements are applied against all data records and are not dependent on data or users.

In contrast, data-specific requirements can only be applied to certain data. Hence for individual data records, it is possible to reach a higher data quality value than for other data records. This behaviour is represented as dashed line in Figure 4.5 from the general maximum to the data individual minimum. There is a similar situation with user individual requirements. Certain users or user groups can have more data quality requirements so that data records can reach higher data quality values than they can for other users or user groups. This behaviour is also represented as dashed line from the data individual maximum to the user individual maximum.

The last categories that influence the data quality scale are the categories of the time dependency classification. The during input category determines the assessment range in which a data record can be during data is entered into the information system. In Figure 4.5 the arrow on the left side represents the whole during input scale. The requirements of the after input category can first be satisfied after some data is entered. In Figure 4.5 this behaviour is indicated by the second data quality scale on the right side. It can be seen that these requirements can have impact on all other categories on the data quality scale.

4.5.2 Data Quality Matrices

As defined, each data quality value is evaluated using a quality assessment function and the data quality value can be of an arbitrary type. A function can take data quality attributes, data quality objects and, if other requirements are associated, quality values as parameters. For instance, the following description uses attributes as input parameters.

If A_x is an attribute and a_x is a value from the domain of A_x and if qv is a quality value and an element from $\mathbb{R}$ then the formal description is:

$$Quality\ Function\ (Qf): A_1 \times A_2 \ldots \times A_k \rightarrow \mathbb{R}$$
$$a_1, a_2.., a_k \ \mapsto qv$$

A quality assessment function is an assignment of a numeric value to each element in the Cartesian product of attribute values.

The first assumption is that only persistent attribute values are allowed to guarantee that quality assessment function can be performed at any time. A second assumption is that the outputs of assessment functions are always the same if using the same input values. That means that quality functions are deterministic.

It is worth mentioning that an attribute can be of any type that an existing database management system supplies. This can be an integer, a string or even a byte array for images.

Quality assessment functions can be logically grouped into the following categories:

- single attribute quality functions
- multiple attributes/multiple objects quality functions
- more complex quality functions
- composite quality functions

In the following sections these functions are discussed and it is shown how a general valid approach using quality matrices can be used for these functions.

4.5.2.1 *Single Attribute Quality Functions*

Single attribute functions take a single attribute as parameter and can formally be expressed as:

$$quality\ value = Qf(A)$$

Following mathematical definitions, a function can also be specified as a relation:

$$Qf = \{(\mathrm{a}_1, \mathrm{quality\_value\_1}), (\mathrm{a}_2, \mathrm{quality\_value\_2}) \ldots (\mathrm{a}_\mathrm{k}, \mathrm{quality\_value\_k})\}$$

where a_x is an attribute value in the domain of A and quality_value_x is a representative of the data's quality.

If attribute A is of type integer, float or other numeric types then it is possible to use arithmetic operations. This would help to simplify the definition of the function. But for all other data types, mathematical operations can seldom be used. The advantage of the relational representation of the function is that it focuses on the mapping from the domain to the codomain. We use a data quality matrix to make the relational representation of functions more readable. A data quality matrix has the following form:

Attribute A	a_1	a_2	...	a_k
quality value	quality_value_1	quality_value_2	...	quality_value_k

Consider the example of an attribute that documents the method, with which a nutrient value was measured. The quality value is randomly chosen to be within 0 and 100, where 0 is the worst rating and 100 is the best rating. If the attribute value corresponds to method A, we get the best possible quality value. If the attribute value corresponds to method B, we get 50% of the maximum rating and if the attribute contains method C, we get the worst rating. The quality matrix has the following form.

Attribute method	method A	method B	method C
quality value	100	50	0

A data quality matrix can also be applied for syntax requirements as the following example shows. Consider a must field requirement which means that a certain attribute M must be filled with content. The quality value is randomly chosen to be within 0 and 100 (0 worst rating, 100 best rating) and m is an element of M. This hard constraint leads to the following data quality matrix:

Attribute M	m is empty	m is not empty
quality value	deficient	100

4.5.2.2 *Multiple Attributes/Multiple Objects Quality Functions*

Multiple attribute functions take more than one attribute value from multiple objects as parameters. The attribute parameters will be denoted as attribute dimensions. We start with a consideration of a quality function on two different attribute dimensions and then generalize to multiple attribute dimensions.

Again, such a function can formally be written as:

$$Qf: A_x \times A_y \rightarrow \mathbb{R}$$
$$quality\ value\ = Qf(A_x, A_y)$$

Again, it is valid that if attribute A_x and attribute A_y are of numerical types, a mathematical expression can be found. To keep the formalism more general, we follow the idea of the data quality matrix. A data quality matrix has the following form:

A_x / A_y	$value1_{Ax}$	$value2_{Ax}$	$value3_{Ax}$	...
$value1_{Ay}$	quality_value_1	quality_value_1	quality_value_2	...
$value2_{Ay}$	quality_value_3	quality_value_2	quality_value_4	...
...	quality_value_5	quality_value_5	quality_value_5	...

where quality_value_x is a representative of the quality of data concerning attribute A_x in combination with attribute A_y.

As an example consider an attribute A_x of type integer and an attribute A_y of type Boolean. The soft constraint assessment rule is defined as: If A_y has the value true then attribute A_x must be above 50 else below 50. The range of the quality value is again randomly chosen to be between 0 and 100, where 0 is the worst quality value and 100 is the best quality value. The data quality matrix has the following form:

A_x / A_y	< 50	>= 50
TRUE	0	100
FALSE	100	0

Now we change to multi attribute dimensions. The question is how to represent multiple dimensions on a two dimensional plane like paper or an electronic document. The solution is to divide and conquer the problem into multiple quality matrices. Assume four attributes are involved in the quality value assessment. Let the attributes be denoted as A_w, A_x, A_y and A_z. We start with attribute A_w and A_x and construct a data quality matrix for them. Instead of writing quality values into the matrix we write representatives into the cells:

A_w / A_x	$value1_{Aw}$	$value2_{Aw}$	$value3_{Aw}$	...
$value1_{Ax}$	rep1	rep2	rep3	...
$value2_{Ax}$	rep4	rep5	rep6	...
...	...	...	...	...

The representatives can be of type integer or type string. Important is that every representative occurs only once if the combination of its attribute values is unique. Unique means that no other combination of values lead to the same quality value regardless what values A_y and A_z have. If there are some combinations of A_w and A_x values that lead to the same quality value, it is advisable to use the same representative to reduce complexity of a data quality matrix.

We do the same for A_y and A_z:

A_y / A_z	$value1_{Ay}$	$value2_{Ay}$	$value3_{Ay}$	...
$value1_{Az}$	repA	repB	repC	...
$value2_{Az}$	repD	repE	repD	...
...	...	...	...	...

Here the combination of $value2_{Az}$ and $value1_{Ay}$ as well as the combination of $value2_{Az}$ and $value3_{Ay}$ will result in the same quality value independent of the values of A_w and A_x. Therefore the representative repD occurs twice.

The next step is obviously to combine the representatives of the two matrices and assign quality values.

Rep A_wA_x / Rep A_yA_z	rep1	rep2	rep3	...
repA	quality_value_1	quality_value_2	quality_value_3	...
repB	quality_value_1	quality_value_4	quality_value_2	...
...	...	...	...	...

Using quality matrices, any type of attributes can be combined to generate quality values.

4.5.2.3 *More Complex Quality Functions*

So far we have looked only at assessment schema using attributes and their values as input. There are more complex situations where missing rows or quality measurements over all existing attributes want to be performed. For this we need an attribute algebra that is more expressive to build more complex constructs. The attribute algebra has the following elements:

Element Name	Element Symbol	Description
Object	O^{Name}	A data quality object with name
instance	e_i^{Name}	A possible instance of a data quality object. The name is the name of the object and the index is used to identify a certain instance.
existing instance	$\tilde{e}_i^{Name}$	An existing instance of a data quality object. The name is the name of the object and the index is used to identify a certain instance.
Attribute	$A^{[Name:]Name}$	A data quality attribute with name. Optional the name of the data quality object can be indicated.
possible attribute value	$a_i^{[Name:]Name}$	A possible attribute value that is of the type defined by A. The name is the name of the attribute and optional the name of the data quality object can be indicated. The index is used to identify a certain attribute value.
existing attribute value	$\tilde{a}_i^{[Name:]Name}$	An existing attribute value of the type defined by A. The name is the name of the attribute and optional the name of the data quality object can be indicated. The index is used to identify a certain attribute value.
Min, max	Min(), Max()	Aggregation functions for the minimum and maximum of a collection of numbers
Sum	$\sum_{i=0}^{n}$	The sum of existing attributes value having a numeric type.
Quantity	N	The number of terms, where a term can be an existing instance or an existing attribute value

And	∧	Logical "and" operation
Or	∨	Logical "or" operation
Mathematical comparisons	=, <, <=, >, >=, <>	Operation equal, smaller than, smaller or equal than, greater, greater or equal than and unequal
Group	()	Brackets are used to indicate in which order expressions must evaluated
Set	{}	A set of attribute values or instances
Element of set	∈ {}	An attribute value is an element of a given set
association "of"	/	The condition expresses that the object or attribute on the left side has an association to the object on the right side. If it is a hierarchical or relational connection does not matter.
Length	Len()	Number of alpha-numeric letters in a string
Unique	isUnique()	The existing attribute value or instance is unique
Empty	isEmpty()	Checks if a certain existing attribute value is empty
Type	isOfType()	Checks if an existing attribute is of a certain data type
Age	A	The age of an existing instance or an existing attribute value.

Table 4-3: List of algebra elements for data quality matrices to express more complex constructs.

With this set of elements it is possible to formulate more complex constructs. The following three examples show how these elements can be used to formulate more complex quality requirement and can be used in quality matrices:

Example A:

Every food can have 0 or more nutrient values where food and nutrient value are existing objects having a 1:n relationship. The summation of all nutrient values for a single food should be close to 100 grams. Some discrepancy is allowed but not more than 3 grams. The range of the quality value is randomly chosen to be between 0 and 100. The compact notation for the data quality matrix looks like:

$\sum_{i=1}^{n} \tilde{a}_i^{Nutrient\ Value:value} / \tilde{o}_x^{Food}$	97-99g	100g	101-103g	else
quality value	90	100	90	0

Example B:
We use again the food and nutrient value objects. It is claimed that every food should have at least 4 nutrient values which are fat, carbohydrate, protein and energy.
The data quality matrix looks like:

$\boldsymbol{N}\tilde{a}^{Nutrient\ Value:value} / \tilde{o}_x^{Food} \geq 4$ $\wedge \{(\tilde{a}_i^{Component}, \tilde{a}_j^{Component}, \tilde{a}_k^{Component}, \tilde{a}_l^{Component}) / \tilde{o}_x^{Food}\} = \{fat, carbohydrate, protein, energy\}$ Quality Value	true	false
quality value	100	0

The requirement is represented by a complex expression and gives not much possibility to use differentiated quality values. A better way would be to split the quality function into two functions:

$\boldsymbol{N}\tilde{a}^{Nutrient\ Value:value} / \tilde{o}_x^{Food}$ Quality Value	1	2	3	4	5	6	> 6
quality value	0	10	20	100	100	100	100

and

$\{(\tilde{a}_i^{Value}, \tilde{a}_j^{Value}, \tilde{a}_k^{Value}, \tilde{a}_l^{Value}) / \tilde{o}_x^{Food}\} = \{fat, carbohydrate, protein, energy\}$ Quality Value	True	false
quality value	100	0

It is also possible to split the second quality function into four different functions so that the nutrients fat, carbohydrate, protein and energy are each used in an own matrix. The advantage would be the introduction of more different quality values.

Example C:
The Age function is a function that includes time as a parameter. The idea is that data decreases in quality over time because they get outdated. Reconsider the example of a nutrient value that is up-to-date only for a certain time period. In Switzerland, a nutrient value should not be older than 5 years. After 5 years, data quality is decreased and food compilers should search for and enter new values.

4.5.2.4 *Quality Value Functions*

Quality value functions take one or more quality values as input parameters. If we consider a tree-like structure of data quality requirements, as seen in the RODQ schema definition example, quality value functions can first be calculated after the input quality values are calculated. Quality value functions are used in data quality requirements of type composite and quality key.
Quality value functions can be used to summarize and to logically group data quality requirements. To do this, quality values are taken as input parameters from those requirements that belong to a logical group. We call this approach: grouping and aggregating.
Again quality matrices can be used as introduced. But for the moment, let us assume that the input parameters are numeric and also the resulting quality value, that we will call composite quality value, is numeric.

$$input\ quality\ value\ \in\ \mathbb{R}$$
$$composite\ quality\ value\ \in\ \mathbb{R}$$

In a logical group, quality values can be summed using different weights. The mathematical representation is:

$$composite\ quality\ value\ =\ \sum_{i=1}^{n} w_i * input\ quality\ value_i$$

The weights w_i correspond to the importance of a data quality requirement. The idea is to give a higher weight the more important a requirement is. Another advantage of using weights is the ease of customization. Users who assess requirements with different importance only have to change the weightings.
The composite quality values can further be grouped and aggregated. This step can be performed arbitrarily many times. As a final step, a quality value of a quality key requirement, called key quality value, can be determined using again the weighted sum:

$$key\ quality\ value\ =\ \sum_{k=1}^{n} w_k * composite\ quality\ value_k$$

To use a weighted sum is only one possibility. Other mathematical aggregation functions such as minima and maxima are also possible. In our project, the weighted sum was the only aggregation function used.

Grouping and aggregating is also possible without the assumption that quality values are numeric. In the case of non-numeric quality values, quality matrices can again be used and the weightings must be integrated in the quality matrices. To generalize this idea, we need to extend the quality function with quality values and weights.

$$Qf: (QV_1, W_1) \times (QV_2, W_2) \ldots \times (QV_k, W_k) \rightarrow CompositeQualityValue$$
$$(qv_1, w_1), (qv_2, w_2) .., (qv_k, w_k) \mapsto cqv$$

where Qf stands for quality function, QV_x for the domain of quality values, W_x for the domain of weights, qv_x for a quality value, w_x for a weight and cqv for a composite quality value in the domain of CompositeQualityValue.

The data quality matrix must then combine all possible quality value and weight pairs. An example with two quality values would have the following data quality matrix:

(QV_x, W_x) (QV_y, W_y)	(qv_1, w_1)	(qv_1, w_2)	(qv_1, w_3)	...	(qv_m, w_n)
(qv_1, w_1)	cqv_1	cqv _1	cqv _2	...	cqv_2
(qv_1, w_2)	cqv _3	cqv _2	cqv _4	...	cvq_5
...	...	...	...	...	...
(qv_o, w_p)	cqv _6	cqv _6	cqv _6	...	cqv_7

If we have m values for QV_x and o values for QV_y then the data quality matrix has at maximum m*o composite quality values. If W_x has n values and W_y has p values then we get at maximum m*n*o*p values.

To further see what amount and complexity non-numeric quality values will cause, the example in Figure 4.6 is used where 3 composite data quality requirements and one data quality key requirement exist. If we consider that this structure leads to only one single quality key value for one object instance, then we can get a notion of the effort required to calculate quality key values for all instances of a data quality object and for all objects in a software system. Hence, the idea of our model is to assign numeric values.

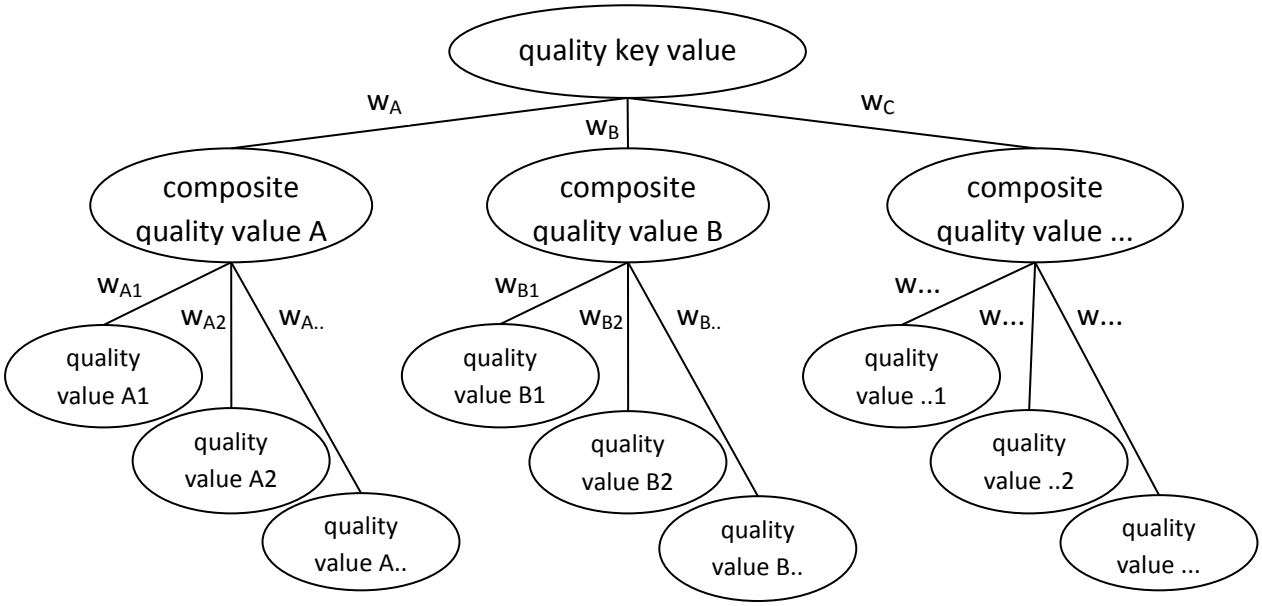


Figure 4.6: The quality key value of a single data records is composed of a number of data quality requirements organised in groups. Each requirement has a certain weight w_{xy} and the groups have also weights.

4.6 Evaluation of the Model for its Use in FoodCASE

The presented RODQ model and schema aim at modelling data quality requirements for an information system. It helps to translate user formulated requirements into a formal model that can be used in the implementation of an information system. During the implementation of FoodCASE, we encountered some advantages and disadvantages of the model and schema that are discussed in this section. To the discussed topics belong the combinations of scopes and types, individualism of users, complexity of a RODQ schema, benefits of a data quality matrix, incompleteness of attribute algebra and normalisation of data quality assessment ranges.

4.6.1 Combinations of Scopes and Types

According to the definition of the RODQ model, a data quality requirement has different combinations of the two properties scope and type. The case where a quality key is also a hard constraint is not obvious. The situation can arise if a separate RODQ schema for only hard constraints is designed. In such a case composite and quality key requirement can be of scope hard constraint.

Another question that has arisen during the design of a RODQ schema is what happens if different scopes are aggregated in a composite requirement? What scope does the composite requirement have?

The resulting composite requirement is per se a mixture and a directive cannot be given. Hence, the designer of the RODQ schema has to decide what scope the composite requirement will have. The designer also has the freedom to decide if a schema for indicators, a schema for soft constraints and a schema for hard constraint is designed or if they are mixed together. In the former way, a mixture is excluded and separate analysis is enabled.

4.6.2 Individualism of Users

The definition of a requirement's scope shows clearly the delimitation between a hard constraint, a soft constraint and an indicator. The survey in Section 3.2 and the experience with FoodCASE showed that different users choose different data quality requirements and categorise the same data quality requirement in different scopes. A schema therefore can be user and purpose dependent. The model does not distinguish between individual and general valid requirements. The model allows data quality schemas to be created that can be made on an individual basis or generally valid and different RODQ schemas can exist. We think that it is reasonable to design a general valid RODQ schema but also to allow different user groups or even users to have their own schema that fits to their intentional use. In Section 5.2, we will see how different schemas can be used in the data quality framework to analyse data.

4.6.3 Complexity of the RODQ schema

Figure 4.3 in Section 4.4 shows that with the amount of attributes also the amount and complexity of the RODQ schema increases. The effort that is necessary to design a RODQ schema could be argued to not be in balance with the outcome. Our experience with FoodCASE showed that this is partly true and we found an approach to overcome this problem. At the beginning of a specification phase of an information system, a conceptual model and a tabular list of data quality requirements is sufficient. When the logical models are designed, hard constraints should be designed in the logical database schema and in the logical application schema, if possible, and can be excluded from that tabular list of requirements. From the remaining list, the RODQ schemas can be designed.

We also had the practical experience that not all data quality requirements are known from the beginning. Many requirements came into the list after users started to work with FoodCASE. An advantage of the model, and also of the later presented framework, is that they can be extended with additional requirements. So the implementation of an information system is not delayed because the RODQ schemas are not finalised.

4.6.4 Benefits of a Data Quality Matrix

A second benefit, beside the applicability to all data types, is that undefined combinations of attribute values can easily be detected. In the FoodCASE project, we had some situations where

certain attribute value combinations were not defined by a given data quality requirement. With the help of a data quality matrix, the problems could be shown to users and solutions found.

An issue of quality value matrices is the fact that the number of attributes values can be large; particularly in the case of integer and float values the amount is infinite. In such cases, simplification must be applied, where possible, such as in the following example. Consider an attribute of type integer and an assessment rule that defines that an attribute value between 1 and 10 returns a quality value of 1, a value between 10 and 20 returns 2, and else 100 is returned. A simplified data quality matrix will have three attributes value columns named: 1 <= a<=10, 10<a<=20 and 'else'.

4.6.5 Incompleteness of Attribute Algebra

The presented attribute algebra is far from being complete. The algebra covers all cases that we could find in FoodCASE but we do not believe that it also covers all cases in other contexts. The attribute algebra is an attempt to formalise quality requirements for data in a way that users are able to understand them. Using a programming language or a database language would be other possibilities but would probably overburden users. Real life context problems are much more diversified and cannot be covered by our simple attribute algebra. But the attribute algebra is only used to allow requirements that are formulated in a natural language to be expressed formally. Instead prose text can be used in a data quality matrix. In the area of attribute algebra future research is necessary to find a good solution how context requirements can be formulated. One possibility is to extend the relational algebra which is a good basis. A possibility for strings is to use regular expressions and a possibility for functions is to use predicate logic, Lambda calculus, functional programming and all kinds of mathematical functions. The challenge is to find a simplified form so that ordinary users are able to comprehend or even to design their requirements.

4.6.6 Normalisation of Data Quality Assessment Ranges

Most probably, a data quality assessment will change over time as can be seen in the history of the Swiss food composition database in Appendix A. There will be newer measurement techniques, new knowledge about how reliability can be assessed and other reasons for change. A point that must be discussed with respect to assessment changes over time, is if an assessment schema should be normalised to a certain range or if it should be an open range. The advantage of a normalised range is that the minimum and maximum is fixed while an open range will increase or decrease over time. The advantage of an open range is that it is more obvious how many rating points a data record received based on the past or current assessment schema. But, in the end, the ranges can be easily exchanged and so it seems to be more a

question of preference than of evident arguments. We therefore leave it open to the designer whether they would like to normalise the assessment range or if they prefer to use an open range.

4.7 Summary

In this chapter, we have introduced 6 basic classifications (scope, type, data dependency, user dependency, time dependency and answering) in which data quality requirements can be categorised. These 6 logical classifications have properties that need to be regarded in the design and implementation of data quality in an information system.

The elements of the RODQ model were then introduced to which data manipulation activity, data quality object, data quality attribute, data quality requirement and the uses/includes association belong. With these elements, RODQ schemas can be built to design and summarise data quality.

For data quality requirements, we also introduced a basic data quality assessment that can be used in any kind of requirement and presented an algebra that can be used within this basic assessment.

We finally presented some advantages and disadvantages of the RODQ model that we discovered during the implementation of the model in the food composition management system FoodCASE.

5

Towards a Requirement-Oriented Data Quality Framework (RODQ Framework)

In this chapter we present the RODQ framework which is a conceptual software framework that focuses on the implementation and management of data quality in an information system. The objective of the RODQ framework concept is to show what elements a data quality framework must contain for optimal data quality assurance and hence simplify the implementation of data quality assurance functionalities for system architects and application programmers. In order, the RODQ framework provides a set of data quality assurance concepts, a mechanism with which users are able to define their own data quality requirements and RODQ schemas. Using the RODQ framework, system architects and application programmers have therefore more time to devote to the main requirements of an information system instead of dealing with implementation details of data quality requirements and RODQ schemas.

The RODQ model in Chapter 4 concentrates on the abstraction and simplification of data quality requirements, how data quality requirements can be measured and how RODQ schemas are

composed. The RODQ framework is based on the RODQ model and the basic concept, as in the RODQ model, is a data quality requirement.

To give an overview, Figure 5.1 presents a concept map of the RODQ framework elements. The chapter starts with two elements, namely data quality prevention and data quality analysis, which information system users will mostly use when dealing with the RODQ framework. Data quality prevention is presented in Section 5.1, which aims at helping users to prevent that low quality data can enter an information system. As additional concepts of data quality prevention, the usage of thesauri for attributes is investigated in Section 5.1.1 and, in Section 5.1.2, we present how quality information for attribute values can be indicated. Data quality analysis is used to analyse data and to find quality deficiencies and is described in Section 5.2.

We then focus on the data quality controller, in Section 5.3, which is the controller when an RODQ schema must be evaluated for data quality prevention and data quality analysis. The effective data quality evaluation is done in a separate framework element, described in Section 5.4. The evaluation process has influencing factors, which we call parameters and are discussed in the same section. Among these parameters are legacy problems when data are migrated from an old DBMS to a new one, Section 5.4.1. Data quality requirements where user input is necessary, is another parameter described in Section 5.4.2 and, last but not least, missing values is also such a parameter, Section 5.4.3. Having the awareness that requirements exist that need user input, we are able to discuss advantages and disadvantages of possible data quality values and their ranges in Section 5.4.4. All these parameters must be as recent as possible and hence the activity of continuous schema adaption is presented in Section 5.4.5.

As data quality can also be interesting for long-term analysis, the RODQ framework also contains a concept for data quality versioning, Section 5.5. Particularly data quality values are targeted in versioning, which is presented in Section 5.5.2. Because data quality versioning can be bound to older data versions, we also discuss data versioning in Section 5.5.1.

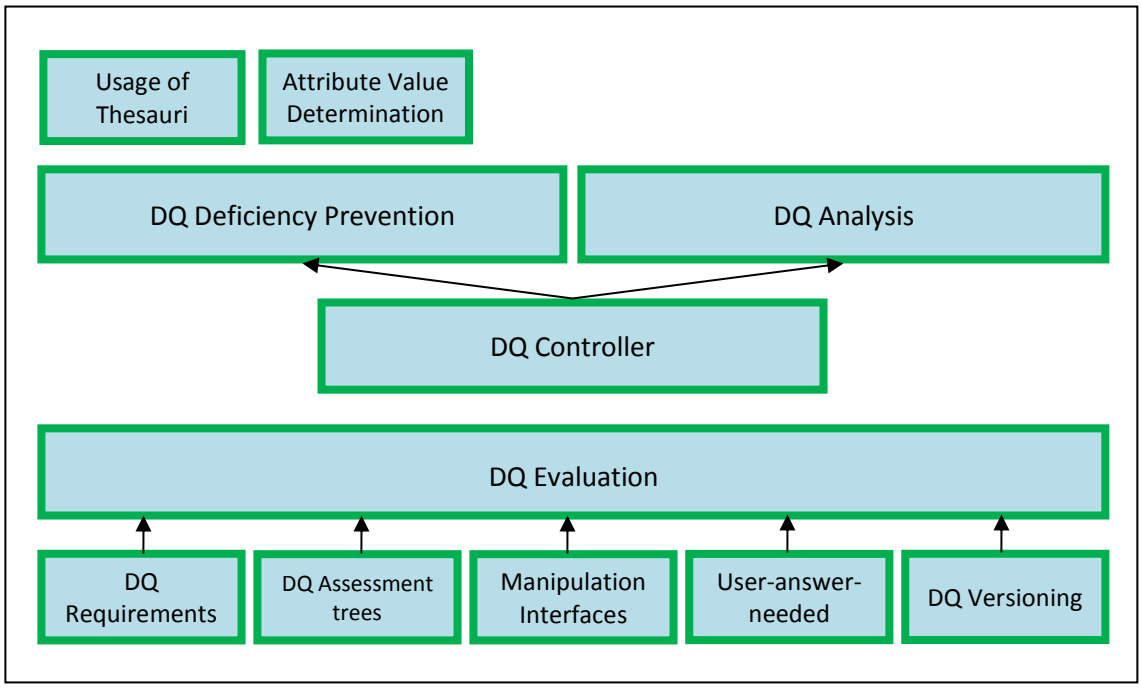


Figure 5.1: Concept map of the RODQ framework main elements. Whereas the data quality evaluation is the core element of the framework, data quality prevention and data quality analysis are the main elements for information system users.

5.1 Data Quality Deficiency Prevention

The basic idea behind data quality prevention is to help users secure data quality before data is saved in an information system. In order, the RODQ framework contains a data quality prevention concept that is designed to detect data deficiency as early as possible and hence can be seen as an early data deficiency detection system. In this way, data quality prevention improves data quality in an early stage and decreases time and cost efforts for later data corrections. It is clear that the more effective the prevention step is, the more time and costs can be saved. An additional benefit is that data that should be stored can be requested to fulfil a desired level of quality.

As seen in Section 4.4, a data quality requirement has an assessment schema that can be used to measure data quality. This assessment schema can also be used to prevent data quality deficiencies: Prevention is the concept in which manipulated data is monitored during data input. Manipulated data is data that is newly generated, changed or deleted by users or by automated processes. Monitoring means that an RODQ schema is evaluated and, according to the assessment, corresponding feedback to users is provided. The feedback should motivate users to correct data quality deficiencies which in turn trigger a new evaluation. It is therefore an interactive process.

Unfortunately prevention is not possible for all data quality requirements and in any situation as we will see in the following example. Consider again the example that a food should have at

least four nutrient values for energy, fat, carbohydrate and protein and where the input of each nutrient value is implemented on a separate GUI and hence cannot be performed at the same time. The RODQ schema is not the same at the input time (before a save operation) of a food item and after the input time (after a save operation) of the last of four nutrient values. Hence we need a separate RODQ schema at the input time of data that involves only those data quality requirements that can be evaluated.

Prevention also has to take place when data is automatically manipulated in automated data generations or data import. An automated data manipulation process can have some requirements to data such as completeness or having certain attribute values. To prevent that manipulation produces deficient data or that software runtime exceptions occurs, these requirements must be checked in advance.

Data quality requirement classification

Let us consider the two requirement classifications time dependency and scope of the RODQ model, introduced in Section 4.2, and see how they are involved in data quality prevention. The classification of time dependency is particularly made to distinguish which requirements are involved in the before save operation validation (category during input) and in the after save operation validation (category after input). Hence data quality requirements that belong to the category during input must be included in data quality prevention.

The scope classification of hard constraints is also involved in the prevention concept. Data that has invalid quality should be prevented from being stored in an information system. In other words, the hard constraint requirements for manipulated data can be used to decide if data can be stored or not. The other two classification categories soft constraints and indicators can be used to increase data quality but if an action is taken on the prevention feedback is under the control of the user who enters data.

Data quality prevention panel

To give prevention feedback to users, the RODQ framework defines a data quality prevention panel as a concrete implementation solution:

A data quality prevention panel is a graphical user interface component that presents evaluated data quality requirements and evaluated RODQ schemas of data that should be stored. A data quality evaluation panel should provide a bird's eye view and a worm's eye view. The bird's eye view, in graphical form, for instance, gives an overview of the RODQ schema whereas the worm's eye view, in tabular form, for instance, provides detailed information about the causes of the quality value.

In addition, the RODQ framework has a definition for the placement of the data quality prevention panel:

A data quality prevention panel must be present in every graphical user interface that deals with data manipulation operations (insert, update or delete).

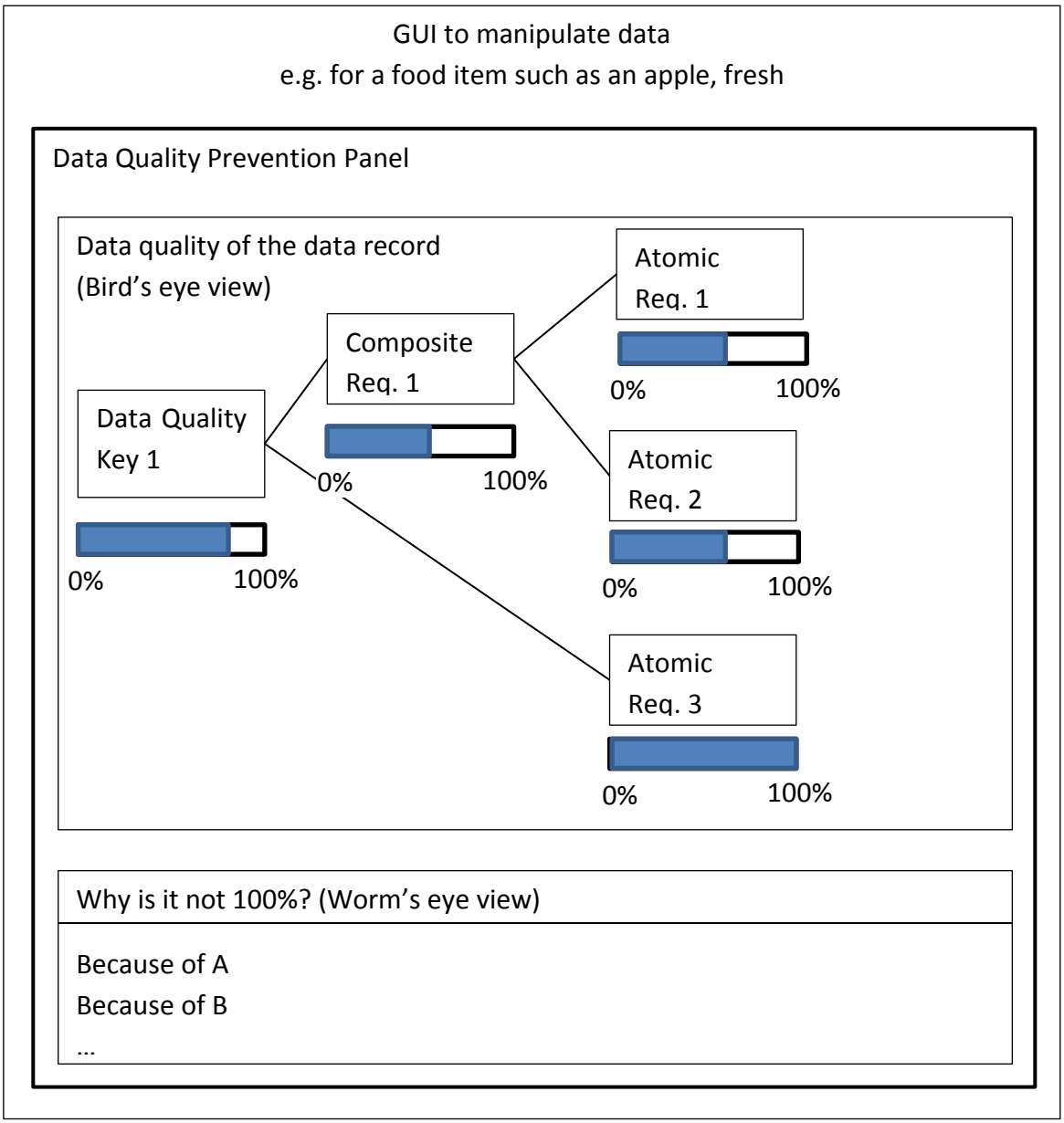


Figure 5.2: Schematical presentation of a data quality prevention panel that is integrated on a graphical user interface (GUI) and offers a bird's eye and a worm's eye view.

The definition's claim to have a quality panel on every GUI that performs data manipulation operations means that, as in FoodCASE, most GUIs have a data quality prevention panel. The evaluation of data quality requirements must be made in real time and additional graphical support on the GUI can be used. Typical examples that are often used are a red background colour, a star at the end of must fields or auto completion of the place where a zip code is

entered. Consider that this additional graphical support is what applications use instead of a data quality prevention panel and hence soft constraints and indicators are often neglected. The definition also states that before every automated data manipulation operation such as import or synchronisation, a data quality prevention panel must appear where a user is informed about the quality of data that should be generated.

The bird's eye view should contain the whole RODQ schema in graphical form in combination with all data quality values, which can be also represented in graphical form such as a progress bar, see Figure 5.2. If the whole RODQ schema is visible on one screen, it can be easily identified which subtree causes decreases in data quality. The tool should also offer to analyse a subtree in more detail, worm's eye view, by showing the requirement details like weights and the attribute values and data records that causes the deficiency. The objective is that users get enough information to understand the data quality value.

5.1.1 Domains of Values - Thesaurus

To data quality prevention belongs also the idea of using a limited set of domain values, also called thesaurus. The proposal for the usage of limited sets of domain values in food composition was made in two pan-European project to standardise and harmonise food composition data [56] [5]. Limited domains of values are known to help assure data quality in that attribute values, which must be chosen from that domain, are guaranteed to be identifiable in the case that referential integrity exists and the problems of typos and related terms are avoided. The second advantage of such domains of values is the possibility to standardise and harmonise data as all users use the same vocabularies. The RODQ framework contains not only the concept to use a predefined set of domain values but also propose to use them whenever possible.

The proposal is based on an analysis that we performed on the database of FoodCASE. We went through every attribute in the database to see if it is possible to use a limited set of domain values. We also investigated data types and constraints of attributes and their influence on domains of values. To the researched data types belong Integer, Real, Float and other numeric data types, Boolean, Datetime, String, Text and Image and to the research constraints belong primary key, unique, auto-increment, foreign key, not null and default.

We investigated 82 tables and 1016 attributes and found that for 869 attributes (85.53%) it would theoretically be possible to define a limited set of domain values. We will see what theoretically means in the following discussion.

Basically, domains of values can either be defined purely by types, types with restrictions/rules or explicitly using foreign keys. The data types String, Text and Image are harder to combine with limited set of domain values. They are designed to give the user the freedom to enter whatever they want and are therefore in a data quality consideration the worst case because intelligence can be necessary to understand and interpret the content. In addition, the number

of possible values for one character, depending on the character encoding, can be big. In FoodCASE we used UTF-8, which theoretically provides 8 bytes for one single character. The essence is that the concept of a limited set of domain values is not applicable and a content-oriented validation must be applied.

Attributes of data type String, Text and Image that serve for descriptions, remarks, comments, suggestions, titles, messages, additional information, filenames and files(images), web addresses, emails, job positions and so on belong to the round 14.5% for which a thesaurus cannot be applied. For the other attributes of data type String or Text we found that a predefined domain of values at least theoretically can be found. A food name for instance can be used together with a thesaurus where all existing foods are listed. The applicability of such a thesaurus can be limited by practical issues like curation effort, frequent update and so on. That is why we declare these domains of values as theoretically possible.

Check constraints like primary key, unique, auto-increment, foreign key, not null and default are designed to make data satisfy a condition. In that sense, they support data quality in that data must have a form to be inserted, updated or deleted. Especially a foreign key constraint demands that an attribute has a value of a given domain.

Summarising it can be said that theoretically most database attributes of a food composition database can be used in combination with a limited set of domain values. But, for practical reasons, the effort to create all domains of values can be time consuming. The advantage that a single attribute would not have a typo, an invalid value or can be easier evaluated for data quality could not be balanced with the creation and maintenance effort for all these domains of values. Seldom two or more attribute can be used together with one domain of values as in the case of zip code and place. But as introduced, the RODQ framework proposes to use as much as possible thesauri that are within the balance of effort and advantage.

5.1.2 Attribute Value Determination

During the implementation of FoodCASE, we found that food compilers sometimes have to interpret scientific publication or laboratory reports to be able to fill in a value for an attribute. We also found out that attributes have missing values for which values were available in the publications. The reason was that these data were imported from another information system that did not provide values for these attributes. A second reason was that it was simply forgotten to fill in the value. Because the information about whether an attribute value is clearly indicated in a publication or was interpreted could in some cases be helpful, the RODQ framework contains the concept of attribute value determination. Consider the following extract from a GUI:

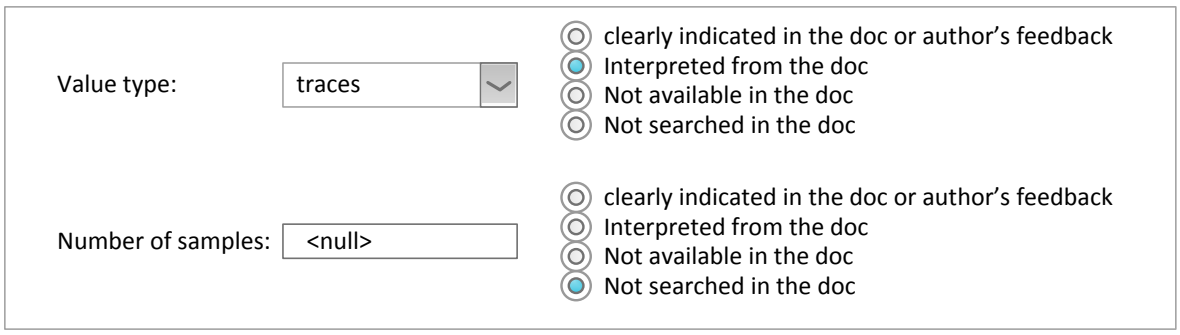


Figure 5.3: Extract of a user GUI having to input fields and four possible attribute value determination options.

In our example the attribute value determination has the following options:

Option	**Description**
Clearly indicated in the document or author's feedback	The attribute value is written in the underlying document (publication) or author(s) of the document was asked because the value was not indicated in the document and the author(s) provided feedback
Interpreted from the doc	The attribute value is not explicitly indicated in the document but can be interpreted from the context
Not available in the doc	The attribute value cannot be found nor interpreted within the document
Not searched in the doc	It is not clear if the attribute is indicated, can be interpreted or is not available in the document

Table 5-1: List of options can be assigned to an attribute value.

The value type in Figure 5.3 indicates that traces of a nutrient are detected for a food item. The attribute value determination indicates that the value was interpreted from the document. If and how much the reliability of the value is reduced, must be decided by a food compiler or by a data user.

The attribute "number of samples" was not searched in the document and there is a possibility to increase data quality in that the document is investigated to find the attribute value.

The attribute value determination is additional information for every attribute that indicates how the attribute value was determined. In our example, it is an indicator to distinguish between facts and assumptions and between no information available and real null values. Which options the attribute value determination can have is dependent on the attribute and on the context.

Although this determination can in some cases be useful, the cost to implement such information must be compared to its benefit. The example above shows that the space

requirements of GUIs increase and the number of database attributes doubles. In addition, the time for users to add data is increased if they also have to fill in the determinations. If this additional effort is high, it can happen that the determination is not accurately entered and data quality assessment for the attribute value determination becomes useless.

5.2 Data Quality Analysis

We have seen that data quality prevention cannot include all data quality requirements if not all necessary data is available at evaluation time. We therefore need a second concept that is called data quality analysis. In data quality analysis, the quality of existing data in the database is assessed and can be analysed to find low quality data. The analysis takes place on data that is permanently stored on any kind of medium. This activity must be performed periodically and, with appropriate functionality, low quality data can be identified as well as the reason for the low quality. In contrast to data quality prevention, this activity can be more time consuming because sets of data quality objects must be assessed.

Data quality analysis can be compared to Online Analytical Processing (OLAP) in data warehouses where data is aggregated to cubes and on which analysis queries are performed. For both, the goal is to collect data that is analysed for a given purpose. The difference is that, in OLAP databases, attribute values are aggregated and analysed, whereas, in data quality analysis, meta-information about attribute values are aggregated.

Data quality analysis panel

Data quality prevention and data quality analysis has in common that they optimally provide a bird's eye view and a worm's eye view and the following definition is similar to the data quality prevention panel:

A data quality analysis panel is a graphical user interface component that presents evaluated data quality requirements and evaluated RODQ schemas of stored data. A data quality analysis panel should provide a bird's eye view and a worm's eye view. The bird's eye view, in graphical form, for instance, gives an overview about RODQ schemas whereas the worm's eye view, in tabular form, for instance, provides detailed information about which data records have what data quality deficiency. It therefore helps to find data quality problems.

In contrast to data quality prevention, the data quality analysis contains two different quality panels:

a. *A data quality analysis panel must be present on every graphical user interface that presents single data quality objects.*
b. *A data quality analysis toolkit must be present to perform analysis over sets of data quality objects.*

In addition to the data quality prevention panel, a data quality analysis panel should also be present on GUIs that present single data quality objects (select operation) so that users are aware of the data quality as well as encouraged to investigate data quality values.
The data quality analysis panel shows evaluated data quality of existing data whereas the data quality prevention panel considers data before storage. If a GUI represents a single data quality object (select operation) and offers the functionality to manipulate attribute values, then the data quality prevention panel and the data quality analysis panel can be combined in one panel.
In contrast to data quality prevention, data quality analysis can not only be performed on a single data quality object but must also be performed on entire sets of data quality objects. To find deficient data quality objects in a set, additional functionality is needed, hence we call it a data quality analysis toolkit. A data quality analysis toolkit must be able to present a RODQ schema with evaluated data quality values in a graphical form. Users should be able to graphically analyse which data quality requirements in the RODQ schema have low quality values and which have high quality values. As in data quality prevention, the bird's eye view should show the RODQ schema in graphical form and the worm's eye view should present requirement details such as attribute values, data records and weights.

Data quality requirement classification

Let us consider the three classifications scope, data dependency and user dependency of the RODQ model in Section 4.2 and see how they are involved. Hard constraints that belong only to the "during input" group and prevent data from being stored, must not be included in a data quality analysis as they are already fulfilled at data input. All other requirements must be included in the analysis.
We have seen that different RODQ schemas can exist for the same data quality object because there are user-specific and data-specific requirements and hence the data quality analysis panel and the analysis toolkit must offer the possibility to analyse different RODQ schemas.

5.2.1 Data Life Cycle

With the two concepts for data quality prevention and data quality analysis in mind, we focus in this section on the placement of these concepts on different process steps in an information system. We first define the process steps by introducing a data life cycle, which is mentioned in [70] as the quality chain. We then present how the data quality prevention and data quality analysis concepts are connected to the data life cycle.
We define the following life cycle steps for food composition data as can be seen in Figure 5.4: sampling, analysis, digital data acquisition, digital data maintenance, digital data processing, dissemination and data usage.
In the sampling step, samples are taken according to sampling plans which describe, for instance, that different food items from different representational places must be chosen. If

applicable, also sample transportation and storage were taken into account. The samples were then analysed using measurement instruments following appropriate measurement methods and guidelines. Only sampling and analysis are backed by professional quality control methods, the former on mathematical statistics, and the latter through industrial standardisation. Of course a step for identifying what data are needed precedes these two steps. But as this step is part of the planning activities, it is not of interest in our considerations.

In the acquisition step, data are collected from one or different sources and are validated against pre-defined selection criterions. If data pass the selection criteria, they are entered in the database system and skipped otherwise.

In the data maintenance step, data are sustained and prepared for further processing or for dissemination. This includes for example, data cleansing or data double checking. Only maintained data should be disseminated because at least a data double check and a data quality analysis should take place for food composition data.

Data can be further processed to calculate aggregated values, for instance. It is also possible that data are processed in multiple cycles. With the example of food composition, data are first aggregated and can then be used to perform a recipe calculation. Newly calculated data is the same as newly acquired data because data is generated, stored in the database and should be again maintained. For this reason, the data processing step is connected with the data acquisition step.

Disseminated data are results that are made available to users. In Figure 5.4 the data usage step is represented by a cloud because there are further processes in which data are used but which are not the subject of our data live cycle. Dissemination is a special kind of processed data in that data is marked for publication and a separate data quality analysis can take place.

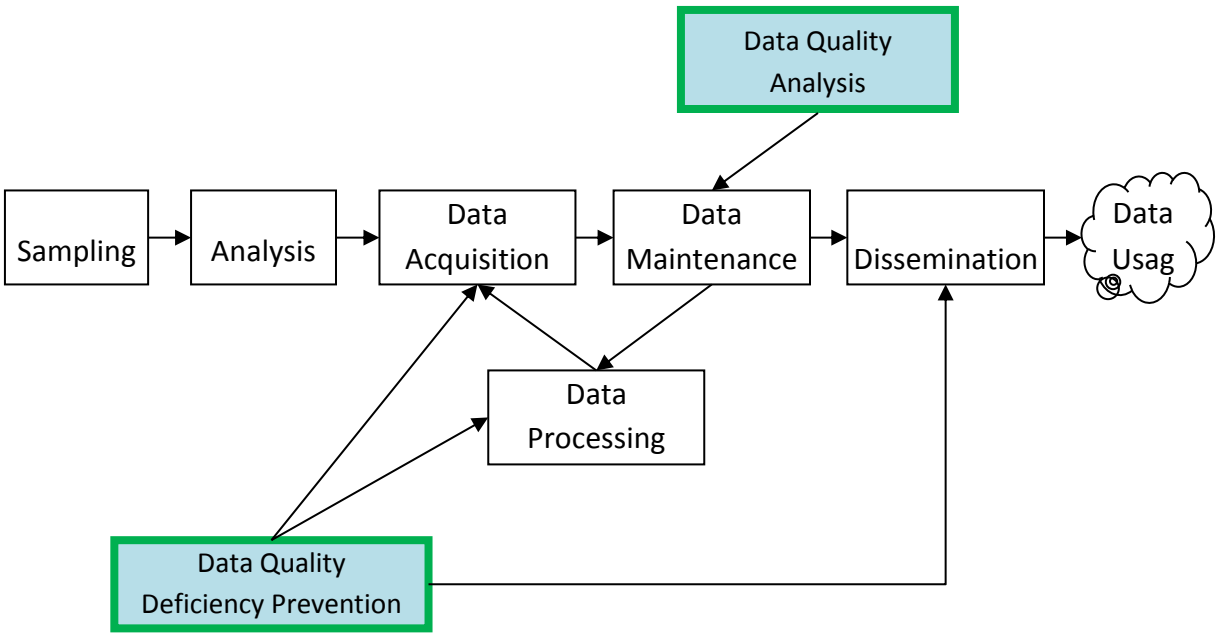


Figure 5.4: Data life cycle for food composition data. Until data acquisition the data are generated and collected outside of an information system. From the acquisition phase data are entered into the system and from then on must be regarded in data quality management.

If we take the data life cycle into account, we can clearly identify that data quality deficiency prevention must be implemented at the beginning of the data acquisition step and at the beginning of data processing where new data is generated and stored as well as at the beginning of the dissemination step.

On the data life cycle, data quality analysis is located at the data maintenance step because data needs to be collected and stored so that also requirements can be analysed which were not possible during data quality prevention.

5.3 Data Quality Controller (DQC)

As seen in the definition of the data quality prevention panel and the data quality analysis panel, they present evaluated RODQ schemas with additional information. If the panels have to evaluate RODQ model schemas on their own, application developers must invest time to keep all the evaluations synchronised during the life time of the information system. Consider the following example, illustrated in Figure 5.5. Applications can have an n-tier architecture. To simplify the example, we consider three tiers: presentation tier, middle tier and data tier. The presentation tier contains data manipulation interfaces, mostly GUIs, which in Figure 5.5 are illustrated as lollipops labelled DM. A data manipulation interface is an interface over which data is inserted, deleted or updated on the database. The middle tier is normally an application server that provides business logic operations such as recipe calculations or value aggregation in the context of food composition. It can also offer some data manipulation interfaces for

machine-to-machine communications with external applications. In the case of relational databases, the data tier has a data manipulation language (SQL) which is send over one data manipulation interface of the DBMS where it is interpreted.

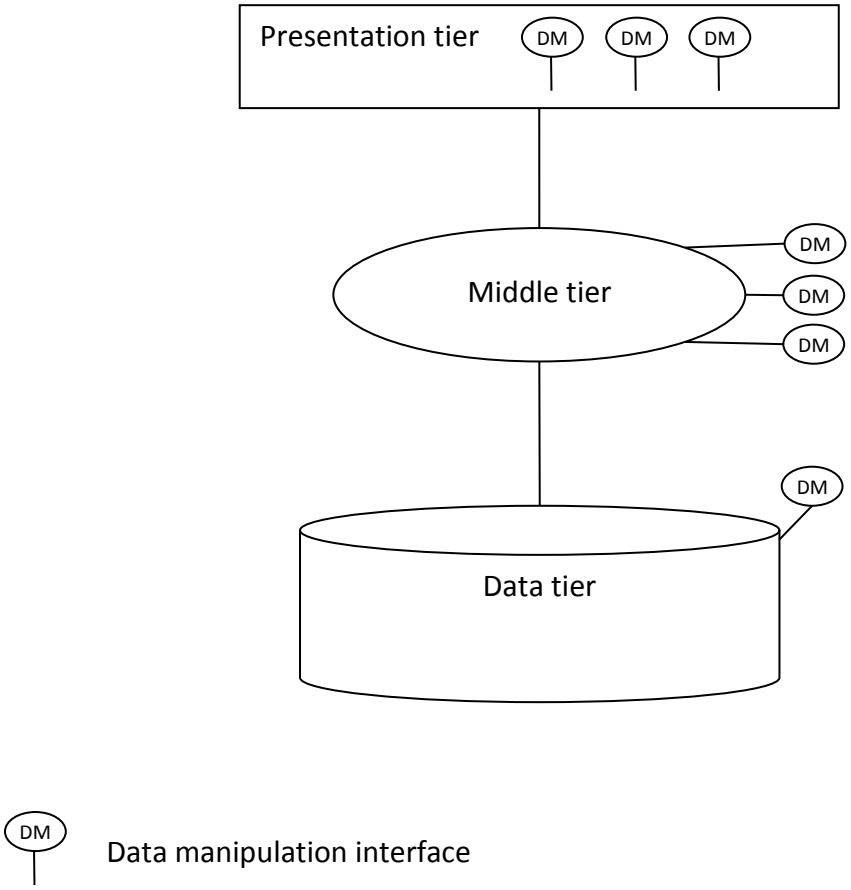


Figure 5.5: Schema of an n-tier architecture. The data manipulation interfaces are located on different tiers and therefore different data quality requirement implementations can be necessary.

There are special data manipulation interfaces that must be considered separately. Consider recipe calculation as example on the middle tier. The recipe calculation is a data manipulation interface because a new food item and new nutrient values are generated. The data manipulation interface needs a GUI on the presentation layer for users where they can select ingredients for the recipe. As users are not directly manipulating the recipe item or its nutrient values but the parameters for the recipe calculation, we call these interface a preceding data manipulation interface and the GUI a preceding GUI, whereas the recipe calculation is called an automated data manipulation interface because the direct manipulation of data records is done by an automated calculation.

After the preceding GUIs, the automated calculations are performed. If then the calculated data is stored in the database without user interaction, data quality prevention cannot take place. The data quality prevention has to take place on the preceding GUI. In that sense these

automated data manipulation interfaces are not equal to manual data manipulation interfaces on the presentation tier.
With this background, the following challenges can be identified:

- RODQ schemas that are defined for prevention must be evaluated for every data manipulation interface. A data quality framework should not only prevent user input from being of low quality but also automated data manipulation interfaces on the middle tier.

When a new data quality requirement arises, there are two tasks:

- The data quality requirement must be implemented
- The data manipulation interface has to be adapted to include the new requirement and must be recompiled.

In addition, there are two further challenges:

- If the different tiers are not written in the same programming language, the evaluation of a requirement needs to be implemented in every language and programmers have to take care that they are always synchronised.
- The data input interfaces of a DBMS are difficult to control. If a username and password are known and the user has modification rights on the database, then an external application can read and write data into the database. The only data quality control mechanisms are the limited set of constraints that a DBMS offers.

In Section 4.6.2 we have seen that a RODQ schema can be user dependent and therefore different schemas can exist. This mean that it must be known what data quality requirements and what RODQ schema must be evaluated.
To solve the problem, a data quality controller is needed that has the following tasks:

- Evaluate and return data quality values to the data quality analysis toolkit
- Registration of data manipulation interface
- Deliver data quality requirements and RODQ schemas to data manipulation interfaces

The first task of the data quality controller is to trigger the evaluation of RODQ schemas for the analysis toolkit and to return the results. The toolkit will then present the results as seen in the section above. Parameters for this evaluation task are which RODQ schema(s) has to be evaluated. The management of these parameters are discussed in Section 5.4.
The second task of the controller is to manage GUIs on which data quality prevention or data quality analysis should be shown to users. These GUIs must register themselves with the controller, see Figure 5.6. A GUI must indicate upon registration which data quality objects it

can manipulate. This information is needed in the administration of the data quality controller in Section 5.4. Now there are two possibilities:

1. Every GUI that gets data, sends them to the data quality controller. The data quality controller triggers the evaluation of the RODQ schema. The resulting data quality values are then sent back to the GUI.
2. Every GUI that gets data, asks the controller which RODQ schema must be evaluated. The controller sends the RODQ schema and the GUI evaluates all data quality requirements in the schema. In Figure 5.6 this case is presented.

In both cases, data quality values must be presented as results to users using the data quality panel. The results contain additional information such as whether data is allowed to be stored based on the evaluated hard constraints.

The first solution is the one that better fits to the idea that the data quality controller is the singleton element where quality evaluation is triggered. The disadvantage of the first solution can be seen in some web applications. The delay to send data to the server, evaluate it there, and receive an answer is a critical aspect. In some web applications this delay will annoy users. The reason can be the amount of data that is sent over a limited Internet connection or the limited server power that needs time to answer requests. The second solution transfers the evaluation to the client where the RODQ schema must be evaluated. This solution has generally smaller delays because the dependency on the network is decreased. But it must be mentioned that for some data quality requirements such as “food name must be unique”, a database connection is required to evaluate it.

In both cases, there is an engineering task that must be solved: If GUI elements want to offer graphical support, for example colouring the background of a text field to indicate that the mandatory field is empty, the GUI needs to interpret evaluated results and needs to find the appropriate GUI element to apply the graphical deficiency aid. Depending on the binding method used between the data quality objects and GUI elements, an appropriate solution for the implementation must be found.

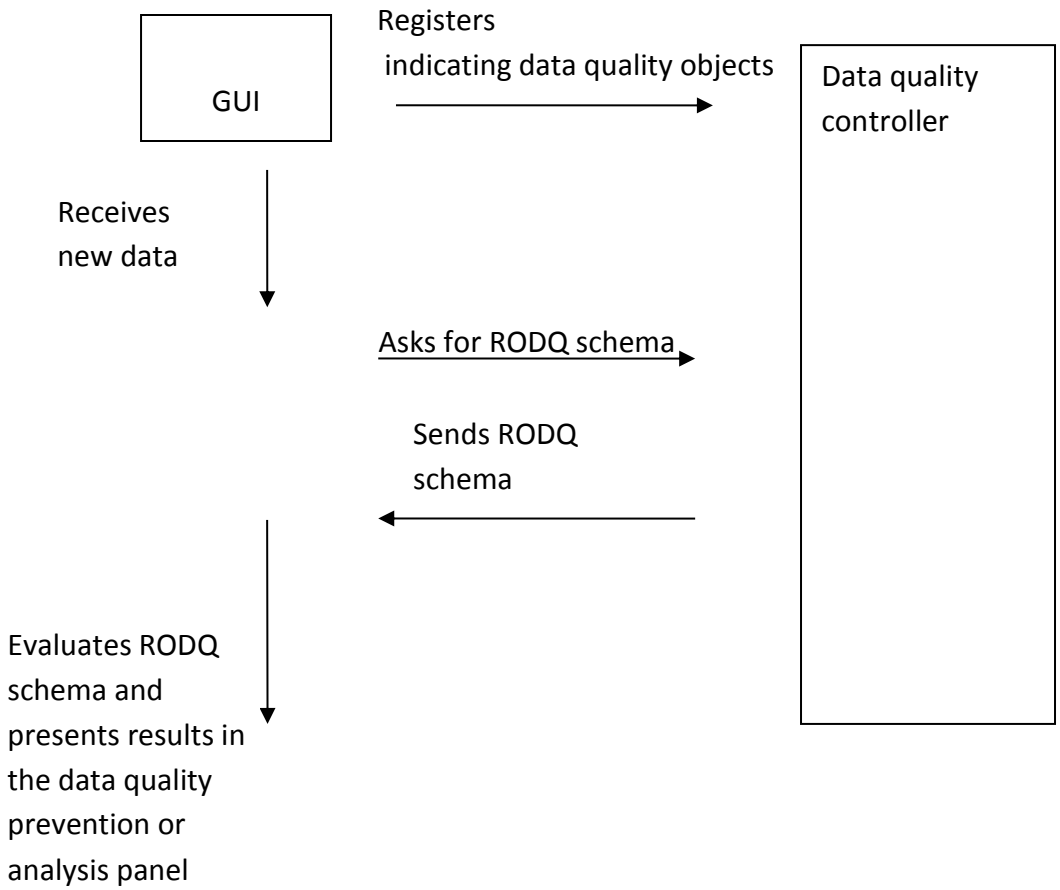


Figure 5.6: Simplified sequence diagram for the data quality communication between a GUI and the data quality controller.

The application architect has to decide which solution best fits to the information system at hand. With the proposed data quality controller concept, we have a solution that not every data manipulation interface must implement all RODQ schemas on its own and a defined RODQ schema can even be used for multiple prevention and analysis panels. In addition, these GUIs do not need to be recompiled if a new data quality requirement is added because sending or evaluating a RODQ schema has a static implementation. A web service interface of the data quality controller also solves the problem of how different programming languages can communicate.

The above solutions are not valid for data tier interfaces. We did not find a way in which insert, update and delete statement of relational DBMS can be redirected to the data quality controller. For the data tier manipulation interfaces there are two possibilities:

1. Restrict modification rights to only the middle tier or presentation tier so that no external application can directly modify data in the database.
2. Allow direct database access and use the data quality analysis to identify and correct data deficiencies in the imported data.

5.4 Data Quality Evaluation

We have seen that the data quality controller triggers the data quality evaluation. In this section, we briefly present the two steps of the data quality evaluation and then what parameters must be considered and how these parameters are managed.

The evaluation of a data quality requirement has two steps:

1. An answer to the question or statement of a data quality requirement is searched automatically (answering), and
2. the answer is mapped to a data quality value using the defined assessment schema in the requirement (scoring)

These two steps correspond to the concept of a data quality matrix in the RODQ model. The reason for separating these two steps can be seen in the Sections 5.4.1 and 5.4.2.

We now consider the different parameters that a data quality evaluation has and see how these parameters can be administered.

Parameter data quality requirements and administration

It would be helpful if a new data quality requirement can immediately be added to the information system by users so that data can immediately be analysed with the new requirement. Waiting for the software producer to implement the new data quality requirement and to disseminate it in the next release or service pack is not desired.

For this purpose, data quality requirements must be administered in that they can be defined and stored in the information system. The definition of a data quality requirement contains also the definition of the assessment schema for which a simplified syntax is needed. The syntax allows users or at least administrators to define how an answer to a data quality question/statement can be determined and what data quality values result from the answer. This definition of the assessment schema can then be interpreted at runtime in the data quality evaluation.

Hence the parameters to the evaluation are which data quality requirements must be evaluated as well as their definition of the assessment schema.

As this implementation is not a simple task, it is also possible to use a set of hard coded data quality requirement but the advantage to immediately add data quality requirements without recompilation will be lost. A set of hard coded requirements makes sense if the requirements can be formulated independent of attribute values or data records as in the case of data type checks, valid E-Mail address checks and so on.

Parameter RODQ schema and administration

In Section 3.2.3 we have seen that different users not only prefer different data quality requirements but also different RODQ schemas. The RODQ framework foresees that users are able to define their own RODQ schema for every data quality object. In order to achieve this, a tool must be offered where a user is able to choose data quality requirements and to build an RODQ schema. In Section 6.5 the implementation of such a tool in FoodCASE is presented. The RODQ schema that must be evaluated is another parameter for the data quality evaluation.

Administration of data manipulation interfaces

We have already seen that data manipulation interfaces have to register with the data quality controller. The main reason is that an overview of all data manipulation interfaces can be generated in an administration tool where also RODQ schemas can be assigned to the interfaces. Such functionality simplifies the synchronisation of requirements between the interfaces and the updates of interface and RODQ schema assignments. The administration tool must consider that the assignments are user dependent and that users can even assign more than one RODQ schema for a data quality interface. The administration tool is important for the controller to know which RODQ schema must be sent to the requesting data manipulation interface.

With the concepts for definable requirements, assessment schemas, RODQ schemas and the assignment of RODQ schema to data manipulation interfaces, we have a completely customisable data quality framework and centralised management from where the different data manipulation interfaces are served. This solves the challenges that data quality requirements and RODQ schemas are implemented at different places and that data manipulation interfaces need to implement them. In consequence, an application recompilation is not necessary.

We have discovered three special data quality requirements during our investigation of the RODQ framework that have influence on the administration of requirements and on the data quality evaluation. These three special requirements, which are legacy problems, user-answer-needed requirements and missing values are discussed in the following three sections.

5.4.1 Legacy Problems

Special requirements for legacy problems emerge when data is imported from another DBMS. We have seen in Section 4.1 that existing DBMS can only implement a part of requested hard constraints because they offer only limited functionality. But there are also situations where the underlying DBMS offers the required constraint functionality but it cannot be used as the following example shows: Consider an attribute “English name” that is requested to be not null but the data that should be inserted does not provide values. This happened when data was migrated from the old system into FoodCASE.

This situation cannot only happen for single attributes but also for entities or combinations of attributes and entities. The situation only becomes a problem if the importing information system has a hard constraint defined whereas the source information system has no values. In the case of soft constraints and indicators, the situation is not problematic. We therefore consider only hard constraints.

The concept that solves legacy problems has two steps: The first step is to find an immediate solution so that data can be imported and the second step is to get the data quality requirement implemented as a hard constraint.

As an immediate solution there are two possibilities: The use of a default value or to define the attribute as nullable in the database. Independent of the chosen solution, it must be guaranteed that this data quality lack does not allow data manipulation interfaces to enter data that do not satisfy the requirement.

Using the immediate solution we can get inappropriate data values in the database. This problem must be solved with the second step of the concept. The second step is to define a data quality requirement that checks data for inappropriate values. The data quality requirement must be available for data quality evaluation to prevent that new data with invalid quality can be entered and must also be available in the data quality analysis so that users are able to identify data records that cause problems. The data quality requirements must be tagged as a legacy problem so that it can be easily identified when a user is searching for legacy problems and the requirement should be clearly documented. When all legacy problems are solved, the immediate solution of step 1 can be undone and the data quality requirement is only needed for data quality prevention. A data quality requirement for legacy problems is only a temporarily needed in the data quality analysis.

5.4.2 User-Answer-Needed Data Quality Requirement

The second special data quality requirement that influences the evaluation process is a requirement where a user answer for a requirement question/statement is needed. In Section 4.2 we have seen the classification user dependency of data quality requirement, to which these requirements belong.

As already mentioned, two outcomes of the EuroFIR project were guidelines for quality index attribution to original data from scientific literature or reports [6] and an entities schema for food composition data including their attributes [5]. The guidelines consist of quality questions about the reliability of food composition data and some of the questions cannot be answered using the data that a food composition database should contain according to the entities schema. Three examples are:

- Was the part of plant or part of animal clearly indicated?
- If relevant was the analysed portion described and is it clear if the food was analysed with or without the inedible part?

- If the food was cooked, were satisfactory cooking method details provided?

The examples show that attribute values and even scientific publications or laboratory reports need to be analysed by a food compiler to answer the questions of these data quality requirements. The reason why such questions cannot be derived from existing data attributes such as food name or method, is because intelligence is needed, that is, a user has to interpret or to understand data in order to be able to answer the questions. In contrast to data quality requirements like "an attribute A must contain a value of type Integer", the question or statement of these data quality requirements is difficult to answer automatically. The RODQ framework defines these data quality requirements as user-answer-needed data quality requirements to indicate that user input is necessary. We found an additional problem of user-answer-needed requirements during discussions with food compilers: Some of the questions are open to a freedom of interpretation so these questions can be answered differently by two food compilers.

However, these user-answer-needed data quality requirements must be incorporated in the RODQ framework and the concept has the following four points:

- GUI elements need to contain the questions of these data quality requirements and the information system needs to store the answers of users together with food data. The automated evaluation of data quality requirements can then take the stored answers of the user and evaluate the data quality value according to the data quality assessment function.
- For machine-to-machine data manipulation interfaces, the information system designer needs to define where in the data manipulation process a food compiler has to answer these questions if imported data do not already provide answers.
- For the creation of RODQ schemas, the RODQ framework proposes that these user-answer-needed data quality requirements should be recognisable as users can pay attention to the weightings of these requirements depending on the interpretation discrepancy.
- Unanswered questions must be handled. Either the lowest data quality value is chosen for these particular requirements or they are specially marked in the data quality analysis but not regarded during evaluation.

5.4.3 Missing Value

The third special data quality requirement is when data are missing or were supposed to be missing. An aspect that users know is the problem when values are missing. Missing values is a common problem in information systems as every empty attribute value or non-existing data record can be seen as missing value. Missing values become a problem when a user needs data that is not present in a database. In the questionnaire in Section 3.2.3, we have seen the

question of whether users agree that the more fields are filled in a database, the better the data quality. The answers were quite equally distributed over all possible answers with a peak at the answer “rather yes”. The 31.8% of persons who said “rather no” or “no” can be interpreted that they generally accept missing values in such a way that they would not think of a data quality decrease. 36.4% who said “rather yes” can be interpreted that data quality is decreased if attribute values are missing but not all attribute values causes a decrease. Only 18.2% who answered “yes” see every empty attribute as a data quality problem. Hence, we need a concept that rigorously handles missing values but take also into account that attributes can be empty without quality decrease.

Importance of a missing value

It is theoretically possible to implement a missing value requirement for every attribute and entity in a database but practically it would be a time intensive task to implement and manage them. The bottom line is the importance of a missing value: Database users have to decide if an attribute value or data record can be missing and if it belongs to the most important attributes or data records in a particular information system. If the first question can be answered with yes and the second question with no, then a user can decide to not implement that data quality requirement because the importance of a missing value is so small that the impact on the data quality key is negligible and therefore set to zero. The difficult context dependent task is to take a decision on the two questions.

Beside the decision to implement a missing value requirement or not, there are other issues that influence missing values as we observed in a field study.

Initial situation of a missing value

We did a field study on missing values on course registrations at a university of applied sciences which will be described in Section 7.3. During the field study, we found that the data quality evaluation element that we implemented was able to identify some missing values but also identified erroneously missing values. As an example consider that a student in the third semester should have course registration for about 30 ECTS (European Credit Transfer System) credits with an accepted deviation of 4 ECTS credits. After the time for applications for course registration, a data quality assessment was performed to see which students did not have about 30 ECTS credits with the intention to find missing course registrations. A student was found to have only 10 ECTS credits, which was a candidate for missing values. But after a detailed analysis of the student, it turned out that the student reduced his study in this semester and so the data record was analysed to not have missing values and hence data quality was not decreased.

The problem was that the initial situation for that particular requirement was not considered. The challenge and the corresponding concept will be explained in the following. We logically derived that the initial situation of missing values in an information system can be one of the following:

1) it is known that data exist and data is missing
2) it is not sure if data exist and if data is missing
3) it is known that no data exist but “real life” has values
4) it is known that no data exist and “real life” has also no values

The first situation is the simplest and the particular data can be searched. The data search can be time consuming but the missing values are known to exist. The second situation is more complicated because the existence of data is not known or not sure. The example given above has this initial situation. As an example for situation 2 and 3, consider a food item that should have nutrient values for all 792 components that are defined in the component thesaurus, even if they are 0. The Swiss food composition database has only 29.45 nutrient values on average for one food item. It is assumed that for some of the missing nutrient values no measured data exist (initial situation 2), and that for some values data exist but are missing in the database (initial situation 2). For some component food compilers knows that no data exist because nobody has ever measured them (initial situation 3).
The initial situation of missing values has the following impacts for a data quality requirement:
Situation 1: If it is known that data exist and data is missing in the information system, then data is incomplete and data quality is decreased. As seen before, food compilers or data users must decide the importance of a missing value.
Situation 4: If there is no data and also “real life” has no values, then data quality is not decreased because data cannot exist.
For situation 3, the impact is not so obvious. If it is known that no data exist but “real life” has values, then there are basically three possibilities:

a) It can be argued that the digital dataset contains all existing data and hence no existing data is missing. As a consequence there is no data quality decrease.
b) It can be argued that “real life” is the only reference point and hence data quality is decreased as much as in situation 1.
c) It can be argued that data quality is decreased but the decrease is smaller than in b because it is impossible to gather existing data.

How situation 3 is treated must be decided by the domain experts for specific problems. Even a user dependent decision can be a possibility for the data quality requirement. In a second step the importance of missing values must be determined.
For situation 2, there are two possibilities

- Use a similar differentiation as in situation 3 if uncertainty cannot not be eliminated
- Perform a missing value analysis to eliminate the uncertainty before the assessment can take place

As the second possibility is the evidence based approach, let us consider this case in more detail.

Missing values analysis

As seen in the field study, some assessments of missing course registrations returned incorrect results. Data quality requirements that are based on the initial situation 2 must first be analysed to decide if there are missing values because the automated answering (first step in the data quality analysis) of a requirement question is not reliable. So for a data quality requirement, for which it is not 100% sure that data is missing, the automatically generated answer can be used as a proposal but user input is required. These requirements are similar to user-answer-needed requirements but differ in one point: In the case of missing values, only a subset of attribute values or data records that do not satisfy a criterion must be analysed whereas, in the case of user-needed-answers, every attribute value or data record must be analysed. The answer of the user must also be stored with the data and must be regarded when the automated scoring takes place.

Borrow other data for missing values

Another observation made in the food composition database concerning the initial situation 2 and 3 was the following: Some food compilers decided to not have missing values for a set of nutrients. To avoid missing nutrient values, they decided that a nutrient value from another similar food must be borrowed. Whether this contextually makes sense, must be decided by food scientists. The idea shows only that other data can be borrowed to prevent missing values, where applicable, and that a borrowed value can have a better quality rating than a missing value.

5.4.4 Data Quality Values and Ranges

An important part in the management of data quality requirements and the RODQ schema is the choice of the possible data quality values and the assessment range. We discussed this topic briefly already in Section 4.4.6. But with the knowledge of user-answer-needed requirements we made new findings.

In general, system designers are free in the choice of possible data quality values and their ranges and designers can make it dependent on the requirement. Nevertheless, we found an example in the EuroFIR data quality assessment for the reliability of data [6] that raises a problem: Consider that a method to measure a nutrient value consists of several so-called key method steps and has a headline method which describes the core of a method. The EuroFIR method scoring is described as having 5 points if the headline method and all key method steps are appropriate, to have 1 point if the headline method is not appropriate, and to have between

2 and 4 points if the headline method is appropriate but key method steps are not appropriate or unclear.

The last scoring point means that a food compiler, based on his opinion, can choose to give 2, 3 or 4 points. In a range from 1 to 5 where only Integers are allowed and 1 (20% of the whole range) and 5 (20% of the whole range) are clearly defined, we have a fuzziness range of 60%. If all data quality requirements have such a fuzzy scoring, a data quality key is unpredictable and not reproducible. The reason is that users are free to assign a data quality value if the values in the given range are not clearly defined and, as a consequence, the rating for the same data quality requirement can be different when evaluated by two different users. We see this problem as inadequate quality of the data quality assessment.

Experimental Approach

The data quality assessment function should be disambiguated to make the assessment value reproducible. Quality matrices or any other used quality assessment function should be validated to check that assignment from the domain to the codomain is precisely defined.

The assessment range is strongly dependent on the data quality requirement and on the overall scoring system for a RODQ schema that is used in an information system. During the implementation of RODQ framework in FoodCASE, we tried the following approaches and observed some advantages and disadvantages:

1. One approach was to formulate a requirement question/statement as a yes-no question where the resulting codomain consists of 0/1 or no/yes respectively. Where possible, we broke a requirement into several requirements having yes-no questions/statements. The obvious advantage is the simplicity of the requirement question because the set of possible answers is reduced to the minimum. The disadvantage of a yes-no question is that some questions need a differentiation, which cannot be mapped to a yes-no question.
2. Another approach was the use of a thesaurus. The entries in a thesaurus form the domain of an assessment function and, for every entry, a quality value of the codomain was defined. The advantage of this approach is that differentiations can be mapped on the thesaurus. A disadvantage can occur if the thesaurus entries allow an interpretation and therefore the thesaurus entries must be chosen carefully and must be enriched with explanations if necessary.
3. Another approach was to use only mathematical functions for the scoring which by definition have clearly defined mappings from the domain to the codomain. The limitation of this approach is that a mathematical function can only be applied if the data types of the domain (answers) are Integer, Real, Float or other numeric types.
4. User-answer-needed data quality requirements, Section 5.4.2, are the most problematic ones because they contain per se an ambiguity as user interpretation of written text may not be reproducible. For these questions, we did not find a satisfactory solution.

Even if they are reformulated to yes-no questions, the answers may not be guaranteed to be free of user interpretation. We therefore propose to make a policy as to the rating value to be chosen if the request can be completely answered with yes, completely answered with no, if the answer lies somewhere in between, and if it is not clear whether the answer is yes or no.

Conclusion

From the implementation of the RODQ framework in FoodCASE, we saw that it cannot be avoided that fuzziness comes into the assessment of data quality, particularly if user-answer-needed requirements are involved. To get an estimation how fuzzy data quality assessment can get, the minimum and maximum scoring of requirements that could potentially contribute to fuzziness, such as user-answer-needed requirements, can be analysed in comparison to the total minimum and maximum of all requirements in a RODQ schema. If the fuzziness range lies within 3%, we have better reproducibility of the assessment as when the fuzziness lies within 60%.

From the above experiences, it is advisable to break down requirements to yes-no questions as much as possible and to keep the fuzziness range as low as possible. In addition, an open range, which means to have a minimum data quality value but no upper boundary, will avoid limitation when RODQ schemas are evolving over time as we will see in Section 6.5.

5.4.5 Activity of Continuous Schema Adaption

A last point that is important in the administration of the RODQ framework is an activity that data quality requirements, RODQ schemas and their assignments to data manipulation interfaces must be kept up-to-date. The idea is not new but, in the context of the RODQ framework, we can concretely define what has to be done. The idea of this element was inspired by the PDCA (Plan-Do-Check-Act) process which is a four-step iterative business improvement concept. It was developed by Deming [11] referring to the work of Shewhard [71]. The concept was based on the steps that a scientific method must have (hypothesis-experiment-evaluation), Bacon [72]. In that context, Kaizen must also be mentioned, which is a Japanese philosophy or practice to continuously improve products, processes and management [73]. All ideas have the objective to continuously analyse data quality and improve it using adequate actions. This means that quality cannot be a one-time improvement project but must be continuously made. In quality management the overall concept is nowadays called continuous improvement process (CIP).

We include this aspect also in our RODQ framework in that we insert a time component for the RODQ model as in the classification time dependency. The idea is that every creation of an RODQ schema is only a temporal snapshot. Every data quality requirement and every RODQ schema must be revised periodically. The reasons for this are:

- Quality conscious users will find new requirements over time
- If the management supports a continuous improvement process, then new requirements can arise periodically
- As context and circumstances change over time, the structure of the composite keys, quality key and single requirements will change
- The history of the Swiss food composition database, Appendix A, showed that the reliability assessment of data and the number of database attributes changed over time

The administration of data quality evaluation should help data users and data maintainers to periodically check the current parameters and enables them to take actions. The periodic rework is also a possibility to check if existing attributes are plausible with respect to the context logic and if attributes are missing or obsolete that are based on new knowledge and experiences. In addition, the data quality controller can also periodically trigger the data quality evaluation and give feedback if some time dependent data quality requirements have changed.

5.5 Data Quality Versioning

Until now, we assumed that the data quality analysis just calls the data quality evaluation element which in a few seconds evaluates all data quality requirements over all data records. Unfortunately, there is a performance issue: In the Swiss food composition database 49 soft constraint data quality requirements on 1282 foods and 38'003 nutrient values were evaluated during a nightly job. We measured the time to complete the evaluation and performed the measurement once a week over 12 weeks. During the measurement, no backup or other operations were performed. The database was running on a new quad-core server with 4 GB RAM and SAS (10'000 RPM) disks.

The answering and scoring needed on average 2 hours and 7 minutes to complete. This means that all requirements for a single food or nutrient value can be calculated within a third of a second. To analyse a single record, this performance is adequate but, for an analysis over all data records, a just-in-time calculation is not applicable. The situation gets even worse if we consider a scientific database with terabytes of data.

As a solution, the RODQ framework includes a versioning of data quality values. If such performance problems arise, the analysis toolkit does not start the data quality evaluation process directly but instead uses data that is available in the data quality versioning element. Before we present this assessment versioning, we first have to introduce data versioning since it is connected to the assessment versioning.

5.5.1 Data Versioning

The already mentioned time aspect must also be regarded for food composition data itself. Food composition values have to have timeliness because otherwise they would lose quality over time as we have seen in Section 3.2.3. The reason is two-fold: Firstly, the environment is changing over time, especially in empirical sciences, and secondly measurement methods improve over time. Thus, it becomes necessary to pay attention to the timeliness of data to assure data quality.

In the Swiss food composition database, it is defined that a nutrient value is out-dated after 5 years and that the value must be replaced after 10 years. These are two data quality requirements for which data must be analysed. We have seen that laboratory measured food composition data (source data) are collected by food compilers and that they aggregate them to representative nutrient values (aggregated data). These aggregated nutrient values are processed data. After a while there are new source data in the information system and the aggregated data need to be updated to provide the most current values. For long term investigations, data should not be deleted but source data as well as aggregated data should stay available in the information system. In addition, scientific data, in particular food composition data, must be traceable and reproducible. Therefore, not only nutrient values but also all available meta-information about the values such as measurement method, sampling information and aggregation function must be stored so that old data is documented.

For this purpose, aggregated data can be versioned. In a new version the contributing values that are used in the aggregation function can be redefined and so most current representative values can be generated. To do this, a copy, also called snapshot, should be taken from the old version including all meta-information and should be marked with a new version tag and a timestamp. It is worth mentioning that the timestamp of a version can also be used for dissemination to show data consumers the timeliness of a data version as an additional data quality indicator.

The fact that data must be documented has the following consequence for source data: Source data that is not contributing to the new version cannot be deleted if it is contributing to an old version. Source data can first be deleted if it is not contributing to any existing version in the information system. Practically, this means that source data that is not allowed to contribute in an aggregation function must be excluded from the list of selectable source data to avoid that users accidentally choose them. The same must be done for thesaurus entries that are used in an attribute.

The concept also defines that data versioning should only be used where necessary as data versioning can be a storage intensive functionality. Therefore source data must not be versioned in the case of food composition data.

5.5.2 Assessment Versioning

Not only data should be versioned but also data quality assessments. A challenge that we have seen in the introduction of Section 5.5, is the performance of collecting and aggregating data. Depending on the amount of data, this can be a time consuming operation that cannot be performed in real-time when users would like to analyse data quality. The questions are what should be evaluated and when should an evaluation take place. The two protagonists are currency and performance which are counterparts and cannot be optimised at the same time. The idea is to have a continuous data quality evaluation and to provide users with the most current data quality assessment values. As databases change over time and data quality analysis can take place at any point in time, it is important to have an adequate strategy when to perform an evaluation. For this reason, the RODQ framework includes a concept with solutions for automated assessment and presents the advantages and disadvantages of each of them.

A data quality assessment run

As we have seen, different user groups and even different users can have different RODQ schemas. All data quality values that are used in at least one RODQ schema needs to be calculated in advance to provide adequate performance for data quality analysis. The RODQ framework introduces the following two terms:

An assessment snapshot is a set of data quality values from all data quality requirements that were evaluated without data modification in-between and can be used in data quality analysis. The evaluation and storage of an assessment snapshot are summarised in the term data quality assessment run, shortly run.

There are two types of runs, partial and complete runs. In a partial, run not all data quality requirements are calculated but only those that have changed. It is like a delta operation that exploits the fact that data quality changes can only happen on the database operation insert, update and delete. In contrast, a complete run recalculates quality values for all requirements. For both types, quality values have to be stored permanently in the database to be able to reuse them.

The advantage of a partial run is that it is less time and resource consuming than a complete run and evaluates only those requirements for which values have changed. One disadvantage of partial runs is that it is necessary to detect if a value change happened that influences a RODQ schema and therefore additional event handling is needed. Another disadvantage of partial runs is that the fetching of data quality values for the data quality analysis takes more time because the most current quality values must be searched. Also a long term analysis that wants to compare quality values at time x with those at time y is more costly because complete snapshots for time x and time y must first be generated from different partial runs.

The advantage of complete runs is that all quality values can be used after a calculation without additional search for the most current. The data fetch delay for data quality analysis is

improved. The disadvantage of a complete run is clearly that quality values are calculated that did not change and that it is more time and resource consuming.
Which of these strategies is used is an engineering task that must be decided by the application architects who have to take the context of the database application into account. A change of strategy over time is possible but needs additional effort.

Versioning of data quality assessment

As mentioned above, data quality requirements are not just interesting for an ad-hoc analysis of the data quality but also for long-term observations of the development. It follows that data quality assessment runs must be distinguishable. This can be achieved by tagging a run in such a way that the date and time when the run was performed is obvious using a timestamp, for instance.
The second point in versioning data quality assessments is the question of what should be stored for long-term analysis. A concept that offers a solution for this question has to take two aspects into account: The first aspect is if assessment schemas remain the same or not and the second aspect is to distinguish between atomic and composite/key value requirements. If the assessment schemas for all atomic, composite and key value requirements remain the same, then a run needs only to store quality values of at least atomic requirements. The quality values of composite and key value requirements can also be stored or can be calculated using the atomic quality values. If assessment versioning is used on a data version, then even the atomic quality values are obsolete. Basically, the RODQ framework does not prohibit combining data versioning and assessment versioning and to perform them always in conjunction. But it must be considered that assessment versioning is done for performance reasons and data versioning and assessment versioning must not follow the same time stamping strategy. If the two concepts are divided, then they are independent from each other and the timestamps of data versioning can be different from the timestamps for assessment versioning.
If at least one assessment schema changes, then there are two situations: First, quality assessments and quality values are still comparable or they are no longer comparable. Comparable means that the questions/statements of requirements stay more or less the same and that the new assessment schemas are a kind of extension to the old one. This imprecise description of the term comparable is caused by the fact that the domain expert has to decide if the assessment schemas and the quality values are still comparable or not. If they are comparable, we have the same situation as above and at least quality values of atomic requirements need to be stored. For documentation purposes, it is advisable, particularly for scientific data, to create new assessment schemas having all changes included, so that the old assessment schemas and the new assessment schemas are stored. This implies an update of RODQ schemas. If assessment schemas and quality values are not comparable, data can be called to enter a new data quality period. For these new assessment schemas, the old versions of quality values are not applicable and hence not usable. The same is valid for the old

assessment schemas. Domain experts have to decide if old assessment schemas and old versions of quality assessment snapshots should be archived or deleted.

When to perform a data quality assessment run

Consider the following situation: Assume that a partial run can immediately be performed after a save operation of a user. A first user changes an attribute value that is involved in a requirement and stores his change. A few seconds later, a second user changes another attribute that is involved in the same requirement. We can then have two runs that calculate the same RODQ schema whereas the second run is more recent. Figure 5.7 illustrates three cases that can happen when two or more partial or complete runs have to be performed that contain common requirements:

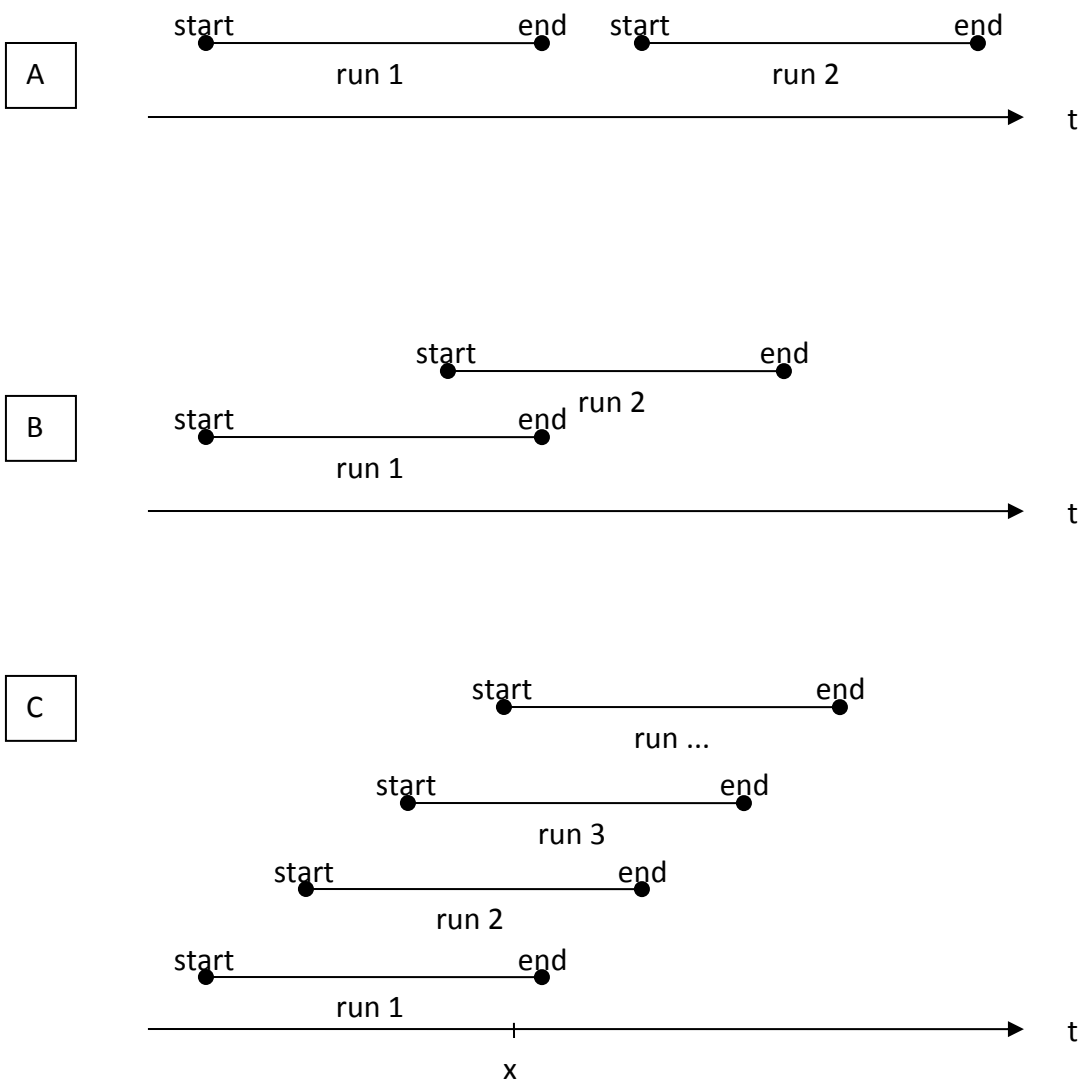


Figure 5.7: Three different cases with data quality assessment runs that can occur on the time axis.

Case A is normally the best situation because the load of the database is more or less equally distributed. Tagging using timestamps is clear because the end of run 1 is before the start of run 2. Case B already produces an overhead because the two runs can be summarised to one run to reduce load. In case C we have the worst situation at time slot x that can happen. Multiple runs are performed at the same time on the same requirements and produce a non-negligible load on the database.

Information system architects or administrators have to decide how runs are scheduled. Depending on the data change frequency and the duration of each run, architects and administrators have to decide

- on partial runs or complete runs (short or long runs)
- if runs are performed immediately after data change or are scheduled
- and if scheduled, the interval between two runs depending on the timeliness that data quality assessment runs should provide

The RODQ framework therefore includes the concept of a data quality timer in which the implementation of the above decision is made.

5.6 Summary

In this Chapter, we have presented the RODQ framework which is a conceptual data quality framework to manage, analyse and assure data quality in an information system. We therefore introduced the concepts of data quality prevention with the objective of increasing quality of newly entered data. We also introduced the concept of data quality analysis in combination with RODQ schemas which can be used for a top-down analysis of data quality.

We have seen that an information system can have several paths on which new data can come into the system and we presented the concept of the data quality controller that is aware of these interfaces and requests them to check a predefined RODQ schema. The objective of the data quality controller is to have a central place where data quality is evaluated and that the RODQ framework can be extended with new data quality requirements without the need for system recompilation.

We have presented data quality requirements that need separate handling in the evaluation process mostly because some requirements cannot be automatically answered but instead need input from users. When users come into play, it could be that the quality evaluation of data is not reproducible because the evaluation gets dependent on the user and his/her interpretation.

Finally we presented a concept for data versioning with the main objective of enabling long-term data analysis. We introduced data quality versioning with the same idea to study long-term behaviour and which should be regarded independent of the data versioning.

6

Implementation of the RODQ Framework and Model in FoodCASE

After the definition of the RODQ model and the presentation of the RODQ framework, we provide evidence in the following two chapters that these concepts can be implemented in at least two different contexts and that the concepts are rated to be useful to improve data quality. Whereas Chapter 7 will present two user studies, in this chapter we present the implementation of the RODQ framework in the food composition information system FoodCASE. As the RODQ model is used in the RODQ framework, it will also be shown how RODQ schemas are used in the implementation. We start in Section 6.1 with an introduction to the project FoodCASE and the associated software system. We describe the parts of the FoodCASE software that are necessary to understand the implementation description of the RODQ framework and model. We then provide an implementation overview of the main RODQ framework elements in Section 6.2 and discuss implementation details and issues in the remaining sections of this chapter. In Section 6.3, we discuss implementation challenges that are related to the data quality evaluation concept. These are the idea and architecture to make data quality requirements definable by users, Section 6.3.1, the handling of unanswered requirements that needs users' feedback, Section 6.3.2, and the GUIs to define the data quality

assessment schemas in Section 6.3.3. Further implementation issues are discussed on the concept of data quality prevention, in Section 6.4, data quality analysis as well as the data quality controller, in Section 6.5. Finally some implementation issues of the concept data quality versioning are discussed in Section 6.6.

6.1 FoodCASE - A Research Project and a Food Composition Information System

In cooperation with SwissFIR and EuroFIR, we developed a food composition database system called FoodCASE. FoodCASE is an acronym for Food Composition And System Environment. In the history of the Swiss food composition database outlined in Appendix A, it can be seen that FoodCASE is not the first software developed in Switzerland to manage food composition data. The reason for a new development of the tool was the new standards for food composition data defined by EuroFIR and the need for a flexible system where we would be able to conduct our data quality research on scientific data collections.

We started the implementation of FoodCASE in 2007 when we first implemented a database according to the entity specification developed in the EuroFIR project and in the COST Action 99 project [56]. After a presentation of our work in progress at a EuroFIR conference, the interest of food compilers outside Switzerland began to grow and we decided to start cooperation with EuroFIR and non-EuroFIR partners. We contacted all EuroFIR partners and other interested parties and asked them to participate in the FoodCASE project which means that they had to give feedback on the FoodCASE specification and to participate in a survey. In return, they were allowed to use FoodCASE for free. With this agreement, it was assured to get feedback whether the design was useful for food compilation and whether the design was compliant to different national food compiling activities. We got 28 participants from 22 countries around the world.

We distributed our 68 pages long software requirement specification for FoodCASE with some initial questions and asked the participants to give feedback. The feedback on the specification was presented in Section 3.2.3. We not only used the feedback for data quality research but also used it to improve FoodCASE. At that time, we also came in close contact with the Food and Agriculture Organisation FAO, a specialised agency of the United Nation UN. Since then, the FAO has also participated in the FoodCASE project.

We implemented FoodCASE and started as soon as possible with the SwissFIR team to test the software in daily food compilation work. In 2010, version 1.0 was finished and we conducted a survey with all FoodCASE project participants. The details of the survey can be found in Section 7.2. During the implementation, we worked in parallel on the concepts of the RODQ schema and added the RODQ framework after the implementation. In this way, we have shown that the RODQ framework can be integrated with existing applications, which we consider to be important because a redesign of existing database applications are cost intensive. The whole implementation took about 2 person years until we got version 1.2.0. The biggest challenge in

the FoodCASE project was to implement the information system leading post graduate IT-engineers and computer science students while collecting user requirements with food compilers around the world and conduct data quality research at the same time.

General architecture and food compilation

In Figure 6.1, we can see the application architecture of FoodCASE. The core component of FoodCASE is an application server, which abstracts the access to the database and performs business logic operations such as recipe calculation. The main working tool for food compilers is the food compilation application whereas the admin application is used for administrational tasks in the information system such as managing user rights or thesauri. The website and the web interfaces are used to disseminate food composition data.

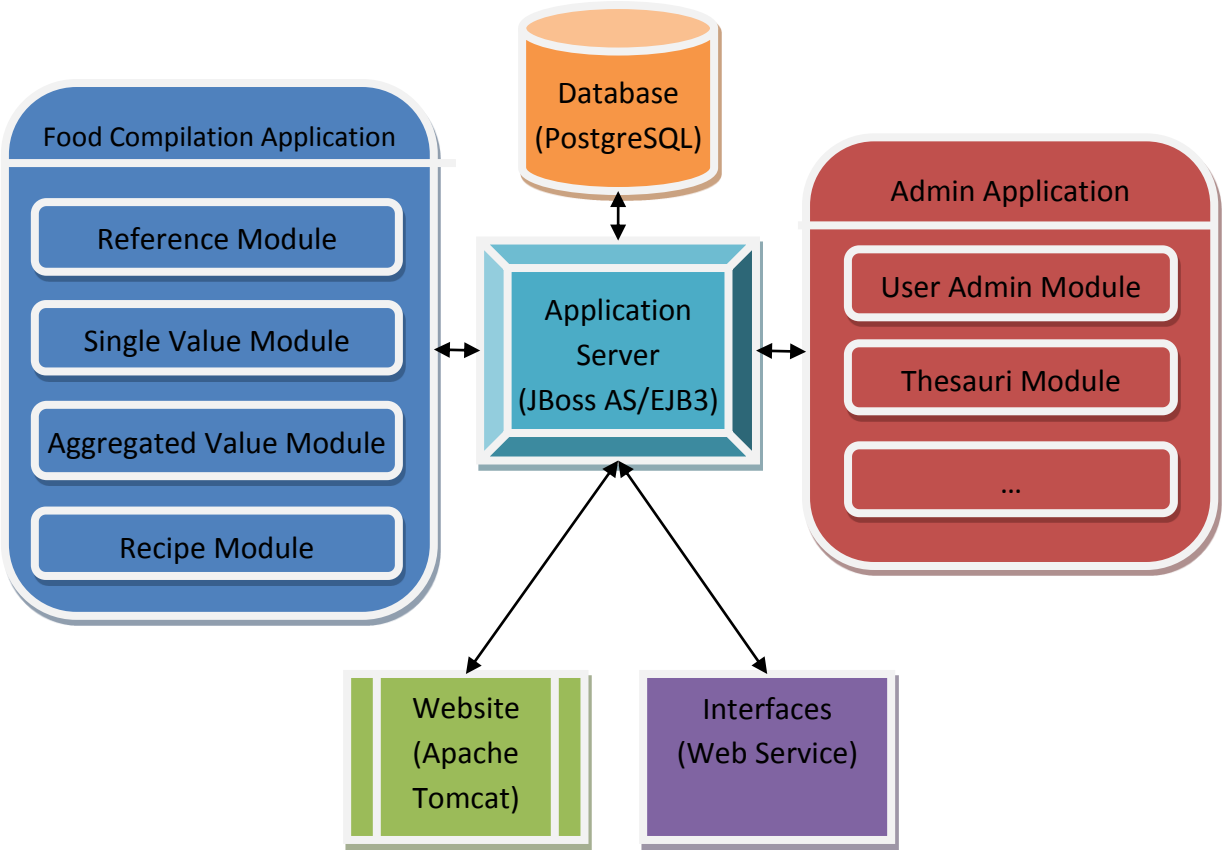


Figure 6.1: Architectural structure of FoodCASE. Every component has a different colour and in brackets used technology is indicated. The arrows show possible data flows.

To describe the modules of the food compilation application in Figure 6.1, consider the following example of a typical data set that needs to be managed in FoodCASE:

Date of Report: March 4, 2010
Author: Mr PerfectLab
Organization: PerfectScience Lab, University of Zurich, Switzerland
Description: 10 samples of an apple were purchased in the supermarket PerfectFoods in the city of Zurich on March 3, 2010. The samples were transported to the university laboratory and then stored in the laboratory as recommended by PerfectFoodTips. The protein analyses were carried out the day after the purchase. Protein was analysed according to a correct AOAC method (biological assay) and using all relevant certified reference materials and other standards. The analyses were carried out in triplicate for each of the 10 samples; the mean and standard deviation for the 10 samples was 0.3 ± 0.1 g / 100 g edible portion.

Such a scientific publication or laboratory report as described above is called a reference in FoodCASE. The reference module manages such publications about nutrient values or recipes. The module has the single task of storing original documents for the three other modules - single value, aggregated value and recipe. These modules correspond to data levels that we defined based on the definitions of Greenfield and Southgate, [20], Schlotke [56], Becker [5] and on the working practice of food compilers. Figure 6.2 is used to explain how the reference presented above would be processed in the other data levels.
On the single value level, food composition data are entered and stored as they can be found in the literature. It is common in this step that multiple nutrient values for one food item exist. In Figure 6.2, this fact is indicated by the two protein entries for an apple.

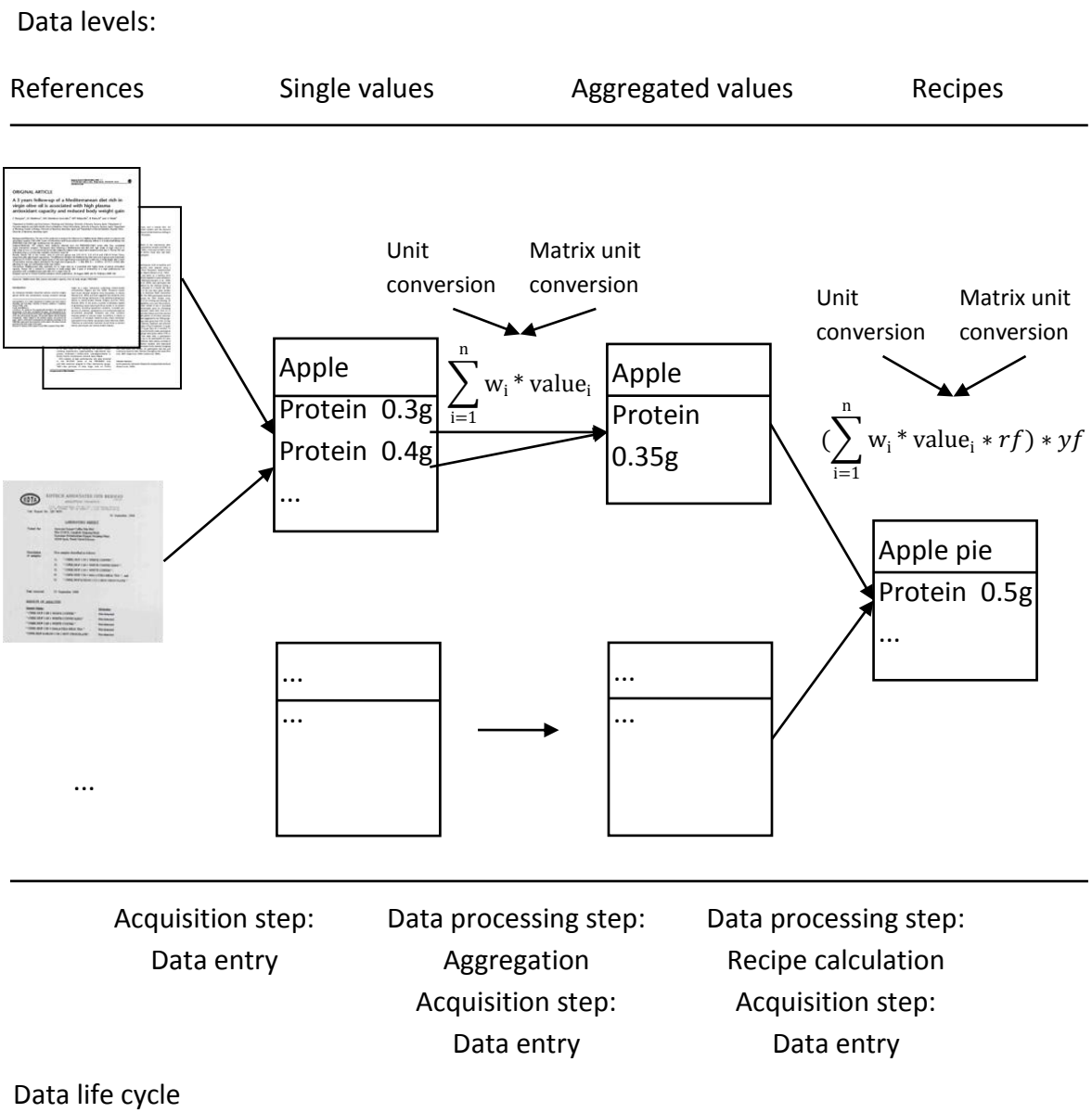


Figure 6.2: The 4 data levels in FoodCASE and how data is generated in these levels. On the bottom it can be seen which data life cycle steps are passed to generate data.

Food compilers' knowledge and experience is needed to evaluate and rate the different data sources in the evaluation of data quality. For publication and other usage of data, single evidence-based representative values are required. Food compilers produce these aggregated values by weighting the single values according to a predefined rule such as data quality. Because single values need not have the same unit, a unit conversion could be necessary and FoodCASE therefore takes care of a preceding conversion before aggregation takes place. The same is done with matrix units. A matrix unit is an indication like "g per 100g edible portion". It is a relation of the nutrient value to a weight specification.

What people mostly eat are prepared meals, which are called recipes in FoodCASE. Recipes are made with different amounts of ingredients, which are aggregated foods. It is again a weighted sum, but this time, the weight corresponds to the amount of the ingredient. Recipes involve some cooking methods such as boiling, deep-frying, baking and so on. Some nutrients gain or lose in value depending on the cooking methods, for example the reduction of vitamins during baking or the water gain when boiling pasta. To accommodate this situation, a retention factor (rf) and a yield factor (yf) are used in the calculation. The retention factor is applied to every single ingredient component while the yield factor is applied to the summed recipe nutrient values. The second difference between retention factor and yield factor is that retention factors are applied to different nutrient values than yield factors. For the recipe calculation, the unit and matrix unit conversion must again be regarded, particularly if liquids and solids are cooked together. The result of this rather complex calculation is one value for each nutrient and therefore recipes have the same form as aggregated foods. Data on the aggregated level and on the recipe level can then be published.

In Figure 6.3, a screenshot of the food compilation application is shown with the three tabs Single Values, Aggregated Value and Recipe representing the levels of food composition as well as the tab References representing the reference module described above. As it is possible to have different versions of food composition data, there is also a tab Versions to manage them. On every tab, the interfaces offers on the left side different search functionalities, in the middle the search results, and the clipboard on the right side is used to perform aggregations and recipe calculations.

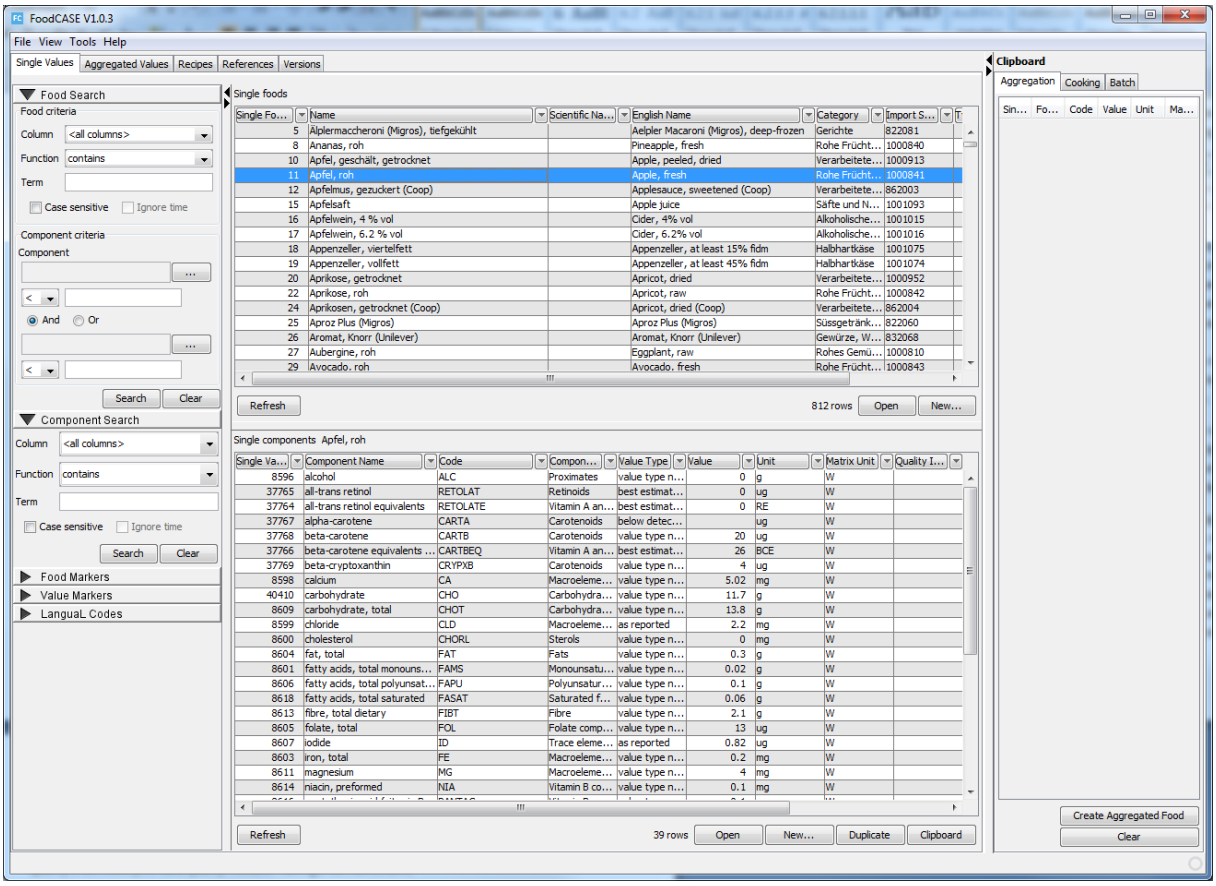


Figure 6.3: Print screen of the food compilation application of FoodCASE with the registers on the top representing the food composition data levels.

The following two tables give some facts about FoodCASE in version 1.0.3:

Database

Number of tables	74
Number of attributes	956
Lines of Java code	50’772 (without blank lines and comments)
Lines of XSLT (for webpage)	6’346

Table 6-1: Facts about the database structure and line of codes in FoodCASE.

Data in the database

Number of food items (single value level)	1’282
Number of components	791
Number of single food component values (single value level)	3’8003
Average number of components per food item	29.64

(single value level)	
Number of food items (aggregated value level without recipes)	822
Number of recipes	190
Number of aggregated food component values (aggregated value level without recipes)	25’578
Number of recipe components values (recipe level)	6’568

Table 6-2: Facts about the amount of data in the FoodCASE database filled with Swiss food composition data version 3.1.

With this short overview of the FoodCASE application and the process of food compilation, it is understandable why FoodCASE serves as good research object. Food composition data is entered from different sources and hence data provenance must be mastered. Food data having different levels of quality is then further processed to aggregated data and recipe data. How data quality is handled must be defined by users and supported by the system. Data can then be disseminated over a webpage or via web services and data quality should be provided to data consumers. In addition, EuroFIR defined a standard XML format to exchange food composition data called FDTP (Food Data Transport Package) [74]. FoodCASE also provides an automated interface to import such XML files and therefore data is not only entered by users over GUIs but also automatically imported. The RODQ framework must also be able to manage this situation by avoiding that users are forced to enter high quality data whereas the XML import interface allows deficient or even invalid data to be stored.

6.2 Overview of the RODQ Framework in FoodCASE

Because the context of scientific food compilation contains some challenges concerning data quality, a conceptual framework as proposed in Chapter 5 is necessary to master all of the data quality issues. Figure 6.4 is the visual summary of the RODQ framework and all of its elements.

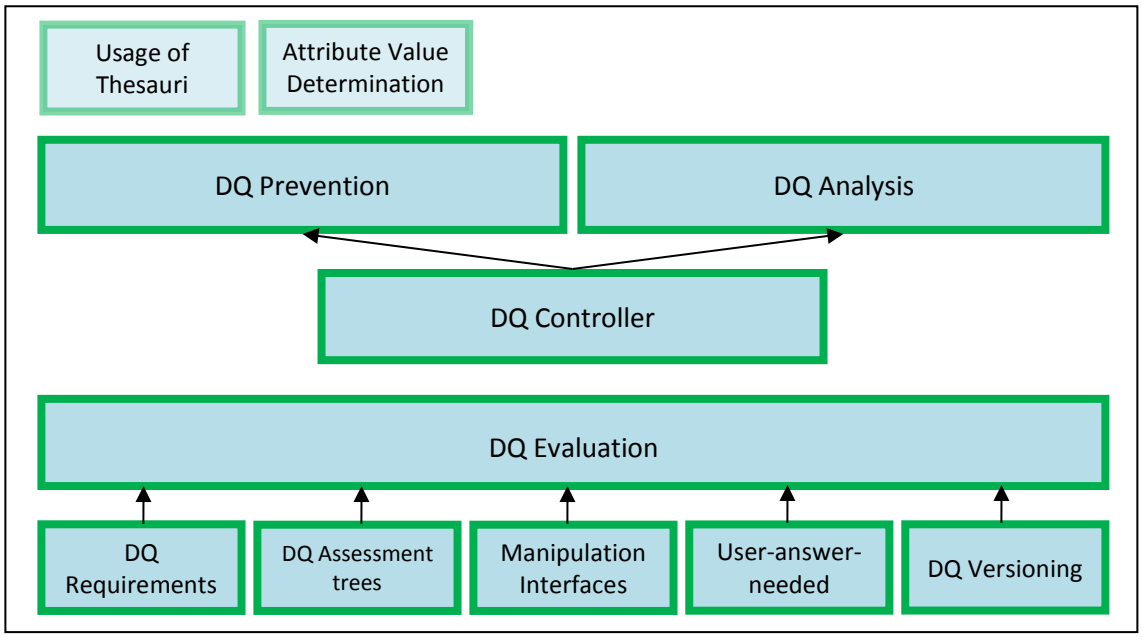


Figure 6.4: Concept map of the RODQ framework main elements.

The implementation of the saturated elements in Figure 6.4 will be presented in the following sections. The concept of "usage of thesauri" will not be presented because it is a simple concept and the implementation is obvious. The following list gives an overview of the functionalities that are implemented in FoodCASE:

- As mentioned in the Chapter 4 and 5, the central element of the RODQ model and the RODQ framework are data quality requirements. A feature was implemented in FoodCASE, with which new data quality requirements can be added without having to recompile the software.
- Functionality was inserted where data quality requirements can be used to build data quality assessment trees.
- A further functionality was implemented for the concept of data quality prevention. The concept is implemented on every GUI where users are able to enter data to ensure an adequate quality of data that is stored. This functionality includes the concept of the data quality evaluation.
- The concept of data quality analysis is realised in two different ways in FoodCASE. On one hand, data quality analysis is also necessary on every GUI where users can manipulate data. On the other hand, it must be implemented as a separate functionality to analyse data quality requirements where multiple steps of data input are necessary. As an example, consider the requirement "a food should have at least four nutrient values for energy, protein, fat and carbohydrate".

The concept of data manipulation interfaces are incorporated in the functionalities for the data quality prevention and analysis.

- Additional functionality is needed for data quality requirements that cannot be evaluated automatically but need feedback from users. Such functionality must not appear on each manipulation GUI and is strongly context dependent. It therefore needs a separate implementation and cannot be implemented as standard functionality in data quality prevention or analysis.
- The concept of the data quality controller is implemented as a separate functionality as it is a central component to which all data quality prevention and analysis functionalities need access.
- A final functionality is implemented for the data quality versioning.

In the following sections, the challenges and architectural design issues are presented and discussed based on the RODQ framework structure presented in Figure 6.4.

6.3 Data Quality Evaluation

6.3.1 Data Quality Requirements

As the starting point for the implementation of the RODQ framework, we collected data quality requirements. We used the technical annex (version 2008) [5] from the EuroFIR project with its entity definition and the document containing guidelines for quality index attribution [6] also defined in the EuroFIR project. In these two documents, we extracted the following number of requirements that are of type atomic:

EuroFIR document	**Number of data quality requirements**
Technical annex	295
Guidelines for quality index attribution	34

Table 6-3: Number of data quality requirements defined by EuroFIR in [5]and [6].

The technical annex document contains a list of food composition entities with their attributes and an indication about whether an attribute is a primary key, what data type an attribute has, whether it is mandatory or optional, and if an attribute is a set-value which means that it contains a 1:n relation. We counted each of these definitions as a single data quality requirement. The technical annex also specifies data levels but it does not stipulate what entities and which attributes belong to which data level.

The document with guidelines for quality index attribution contains a list of 34 questions, mostly with the possible answers “yes”, “no”, and “not applicable”, that are organised in 7 groups. Some questions are only for brand foods and some questions are dependent on each

other in that a question is unnecessary if the forerunning question was answered with no. Each of these questions was identified as a data quality requirement.

As the technical annex contains requirements for each attribute, it focuses on single attributes whereas requirements that depend on more than one attribute or more than one entity are not included. The guidelines for quality index attribution concern mainly the reliability of nutrient values. During the implementation and discussions with food compilers, we identified 122 additional data quality requirements for all three data levels that are of type atomic. Summarising we found 451 atomic data quality requirements for all data levels. The 295 (65.41%) requirements coming from the technical annex were implemented as database constraints as well as Java data types and all of them are hard constraints. In the remaining 156 requirements (34.59%), we found 61 hard constraints, 58 soft constraints and 37 indicators. The 34 EuroFIR reliability questions are contained in the 37 indicators. These remaining 156 requirements are only the atomic ones, which means that they are the leaves in the RODQ schemas.

GUI to define a data quality requirement

A GUI with which users are able to enter new data quality requirements in FoodCASE was implemented as a master thesis by Mock [75] and is presented in Figure 6.5. The idea is to store assessment schemas in an SQL syntax as data in the database, which must be interpreted during data quality evaluation. The advantage is that the assessment schema can be updated at runtime but has the disadvantage of a complex syntax so that users may not be able to define their own data quality requirements.

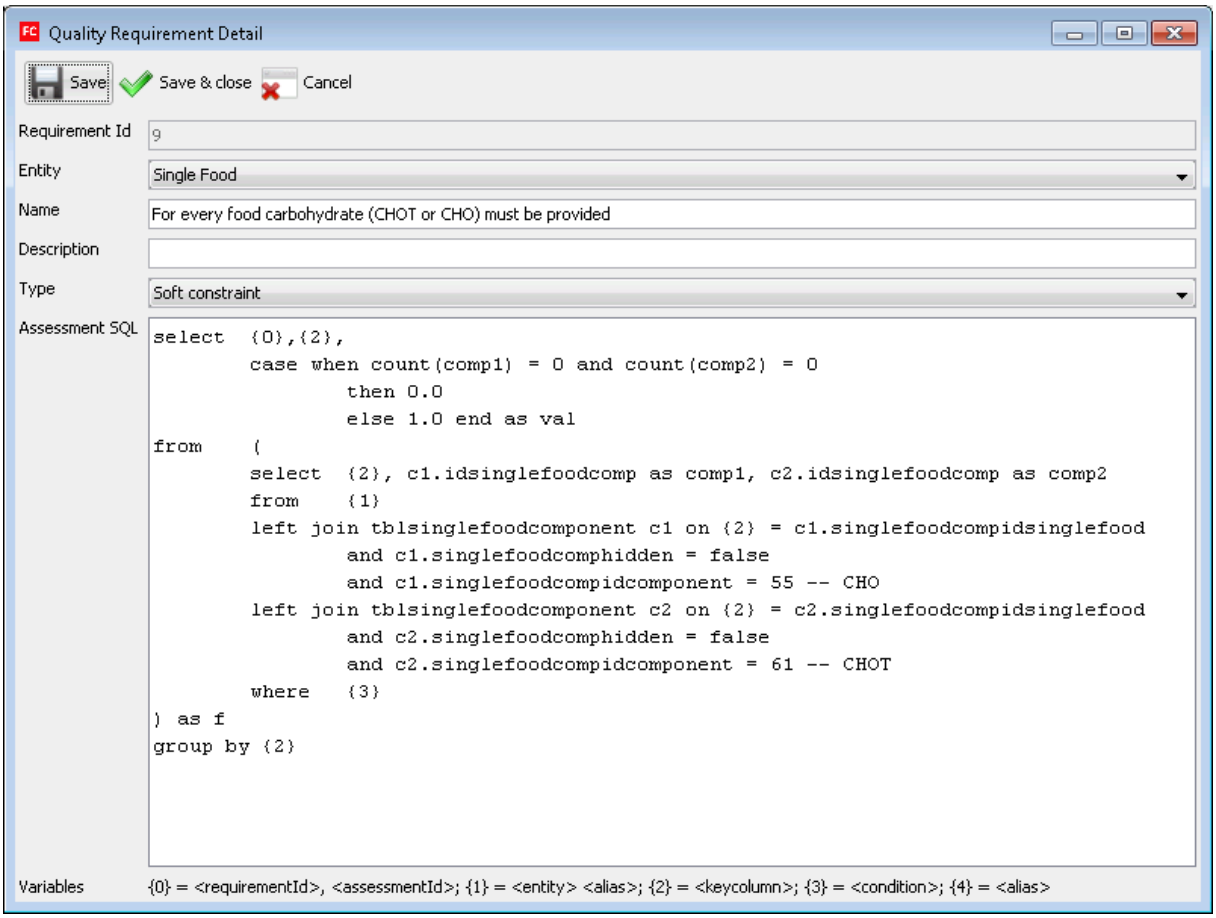


Figure 6.5: A data quality requirement (scope soft constraint) for a single food where the nutrient value carbohydrate (CHO) or carbohydrate, total (CHOT) should be provided [75].

SQL-like syntax is used because of its widespread use and the fact that it should be easy to integrate the implemented framework in existing environments.

An example with selected requirements

In Figure 6.6, an RODQ schema is presented for a selected number of soft constraint requirements for a food component value on the single value level.

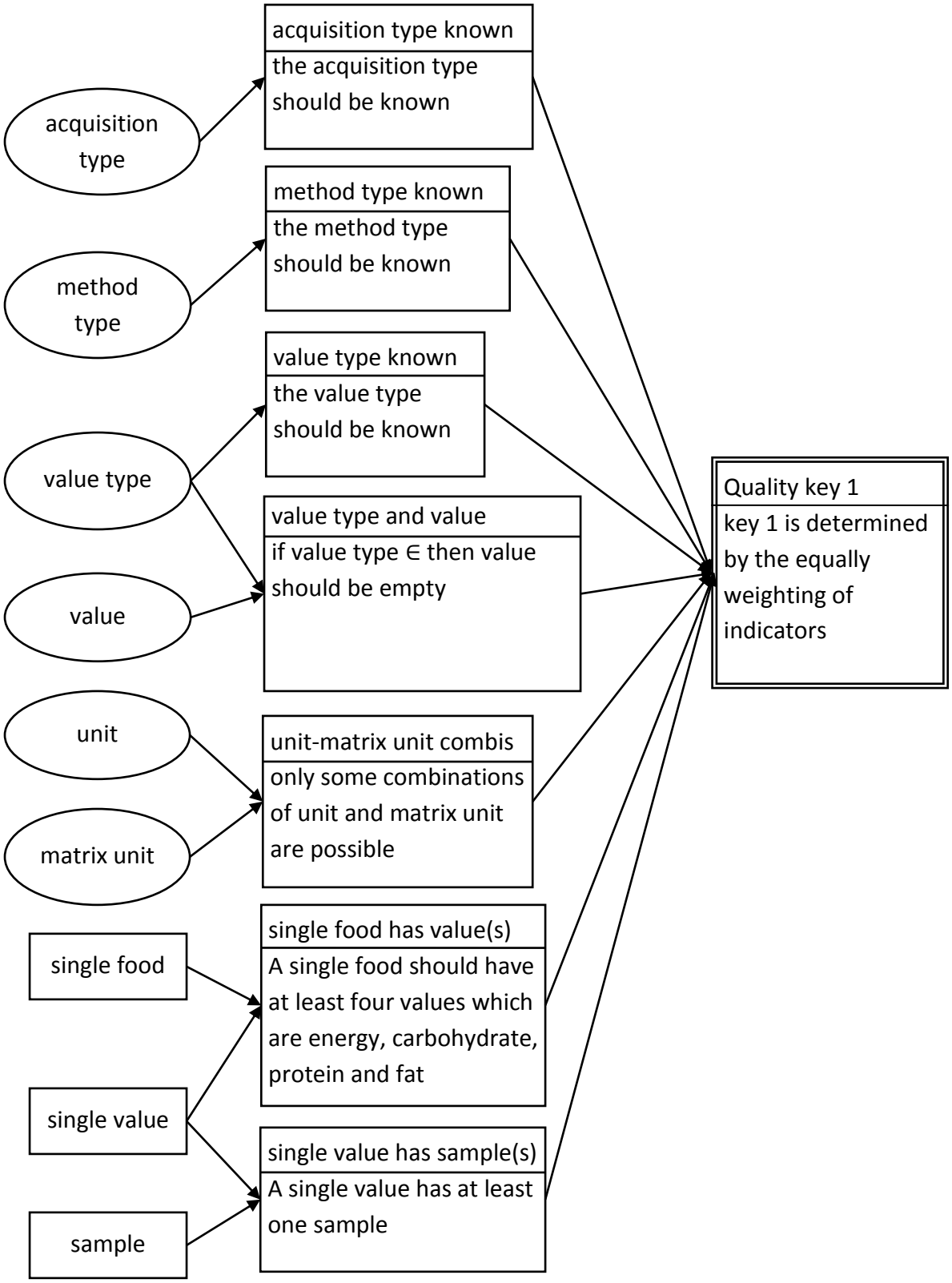


Figure 6.6: RODQ Schema with selected data quality requirements that form the quality key 1. To give a simple overview the assessment-schemas are not indicated.

The first three requirements focus on the possible values that can be assigned to the corresponding attributes. The values of all three attributes come from thesauri and some of the values decrease data quality. In particular, the thesaurus entry "unknown" lowers data quality because context important information is not indicated. The "value type and value" requirement and the "unit-matrix unit" requirement are examples for requirements that include multiple attributes. They should be defined as hard constraint requirements in FoodCASE but because of legacy problems and imprecise publications they were turned into soft constraints. The bottom two composite requirements include multiple entities and are soft constraints for the same reason.
The quality function for the first three requirements are similar and has the following form which is presented by the example of the assessment-schema of the requirement "acquisition type known":

Value	Scientific communication	Independent laboratory	Other acquisition type	Food composition table/database	Industry laboratory	Acquisition type not known
Quality value	1	1	1	1	1	0

The header row contains all possible entries of the attribute acquisition type. The quality value range from 0 to 1 was chosen, where 0 is the lowest quality rating. If the data quality requirement “acquisition type known” is evaluated, we get a quality value of 0 if the value is equal to “Acquisition type not known” and 1 if the attribute has another value.

For the quality matrix of “value type and value” we needed a two-dimensional quality matrix:

Value type Value	trace, below detection limit, undecidable, unknown	Else
0 or 0.0	1	0
Else	0	1

Again the quality value range is between 0 and 1 and the quality matrix covers all cases for that requirement. In consequence, we have no undefined attribute value combinations.
The quality matrix for “single food has value(s)” was already presented in Section 4.1.1.3 and has the following form:

$\boldsymbol{N}\tilde{a}^{Nutrient\ Value}/\tilde{o}^{Food} > = 4 \wedge \{\tilde{a}_i^{Value}, \tilde{a}_j^{Value}, \tilde{a}_k^{Value}, \tilde{a}_l^{Value}\} = \{fat, carbohydrate, protein, calory\}$	true	false
Quality value	1	0

The rather complicated quality matrix is reformulated in a true/false question. The formal description of the data quality requirement simplifies the implementation into a database script or higher order language and the quality matrix helps to take care of all possible attribute values. The range for the quality value is again from 0 to 1.
The requirement "Quality key 1" is a summation of the quality values from the seven children requirements and has the range from 0 to 7, where 0 is the lowest quality key value and 7 is the highest quality key value.

Architecture and implementation of data quality requirements
We now describe the database schema used to store data quality requirements and their quality values. As these database schema snippets look the same on all three food composition data levels, we only present the schema snippet for single foods, single values and the corresponding data quality values. We used the following database schema in a relational database:

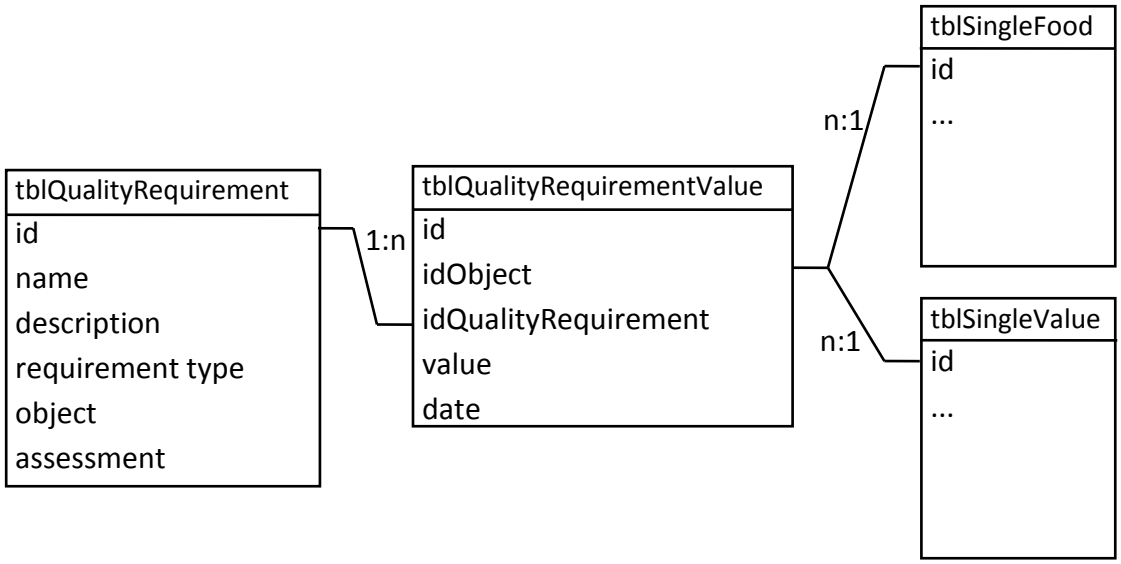


Figure 6.7: A database schema snippet for data quality requirement, single food, single value and quality values.

All entities in Figure 6.7 have a unique identifier which is represented by the attribute "id". The table tblQualityRequirement contains the attributes "name" and "description" to provide information about the requirement. The attribute "name" is mandatory and must be unique. The attribute "requirement type" indicates if it is a hard constraint, a soft constraint or an

indicator. The attribute “object” indicates to which data quality object (single food, single value and so on) the requirement is connected. And finally, the attribute “assessment” contains the quality matrix or any other form of assessment.

The table tblQualityRequirementValue has an association to the table tblSingleFood and tblSingleValue using the attribute “idObject” as well as a foreign key to the Requirement table. The “value” attribute is the data quality value and the attribute “date” is the timestamp when the measurement was made.

Alternatively, the entity tblQualityRequirementValue could be duplicated so that each data quality object has its own requirement value table. The advantage of this idea is that the number of data quality values per table is decreased. During data quality analysis, the table tblQualityRequirementValue is joined with other tables and data records of interest are selected. Having multiple smaller tables would yield better performance for data quality analysis. This fact is based on the 4.9 million data quality values that are generated within FoodCASE in a single data quality assessment run over all data quality objects, all data records and over all requirements. After several assessment runs, the table tblQualityRequirementValue is by far the biggest table in the database. The disadvantage of this approach is that new data quality objects would also need such a table. If, for example, a new data quality object composite food would be inserted, an additional table between requirement and composite food would be needed. This approach is therefore not as generic as the one presented above. Another disadvantage is that the multiple requirement value tables have all the same structure in term of attribute and hence we would generate some sort of redundancy if these tables were duplicated.

Both implementation alternatives of the data quality requirements satisfy the RODQ framework’s claim, introduced in Section 5.4.4, to define open ranges for quality keys: The data quality key value range is defined by the number of quality requirements in the table tblQualityRequirement, by the minimum possible and maximum possible data quality value of each requirement and by aggregation weightings in the RODQ schema. If new data quality requirements are added or removed, then the according ranges change.

6.3.2 User-Answer-Needed Requirements

Data quality requirements that belong to the category “user answer” of the classification answering need a separate implementation. If we recall that these requirements need answers from users to their questions or statements and only then can be evaluated by a data quality function, separate implementations on the data manipulation GUIs are necessary where users are able to provide answers to these requirements. In Figure 6.8 such an example of the 34 data quality requirements to determine the EuroFIR data quality index is presented.

There are two issues that must be considered when implementing user-answer-needed requirements: The evaluation of unanswered requirements and user notification for such unanswered requirements.

As proposed in the RODQ framework, there are two possibilities if no answer for a user-answer-needed requirement is provided: Either the lowest data quality value is chosen or the data record is not taken into account during data quality evaluation. The main question is how the requirement is considered. If it is said that the data quality requirement is in an undefined state because it cannot be evaluated, then the data record should not be taken into account. But if it is argued that not answering a data quality requirement's question or statement is an omission on improving data quality, then the lowest data quality value must be assumed. Although the definition of the 34 EuroFIR requirements in [6] do not specify how this situation must be handled, we decided to use the lowest data quality value in accordance with some food compilers. The main reason is that the EuroFIR questions also contain the possible answer "not applicable", which can be used if a requirement's question does not fit to the regarded nutrient value and this answer means that the requirement should not be considered in the data quality evaluation.

The second issue is that a data quality framework has to notice which questions are answered and which are not answered and notify users about the unanswered requirement questions. The main discussion points are where and how users have to be notified. In the FoodCASE implementation of the RODQ framework, the notification is incorporated in the data quality prevention and data quality analysis panels. If a user opens a data manipulation GUI, the notification is shown to the users in the same way as unsatisfied data quality requirements. In a sense, the data quality panels are misused for user notifications but as we consider unanswered requirement's questions as omissions on improving data quality, we also see unanswered questions as unsatisfied requirements.

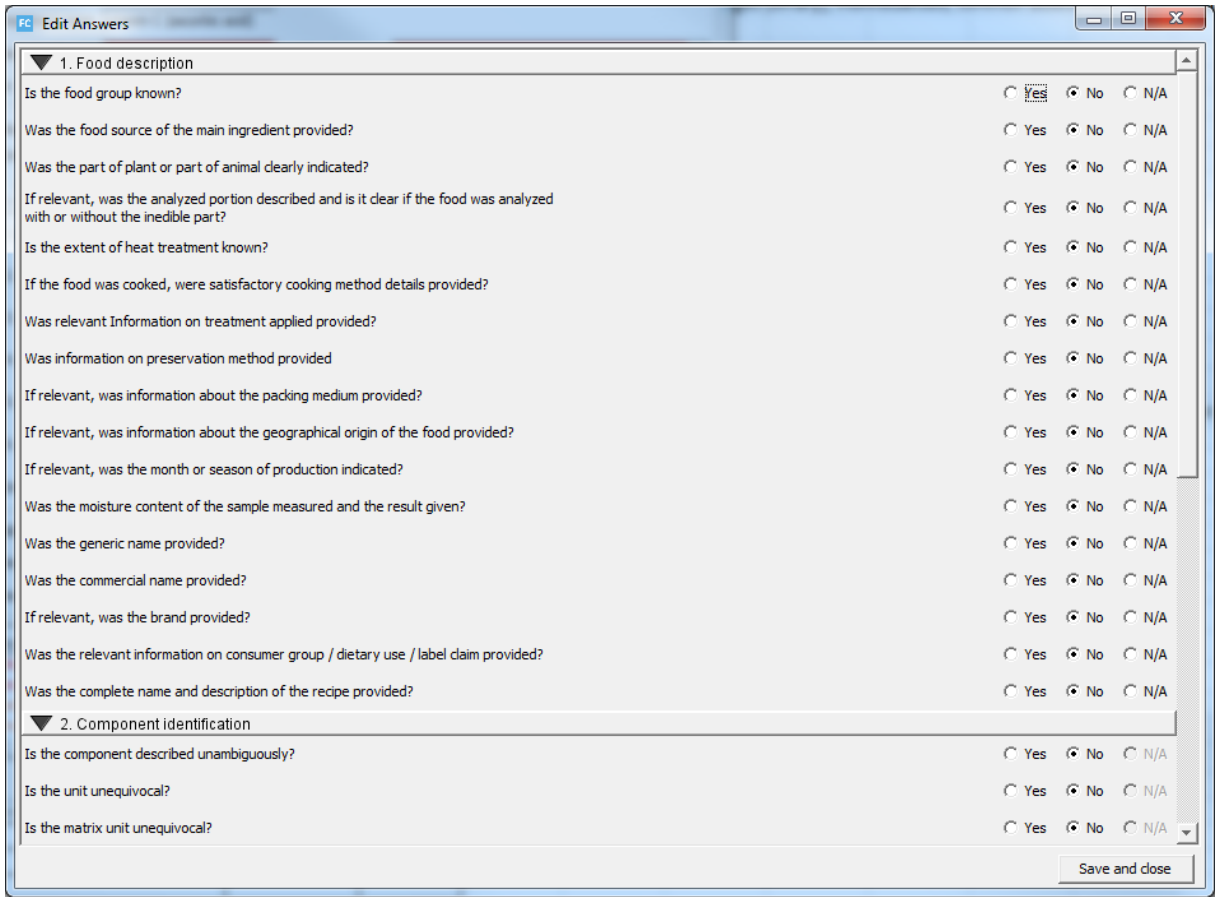


Figure 6.8: The 34 EuroFIR reliability questions are presented in a single window group by the 7 categories. The answers determine the number of points that are achieved for each category.

6.3.3 Data Quality Assessment

The GUI to build RODQ schemas were more challenging to implement given that the schemas should also be stored in the database and should be updatable during runtime. Figure 6.9 shows the GUI, in which only a small part of a RODQ schema is visible.

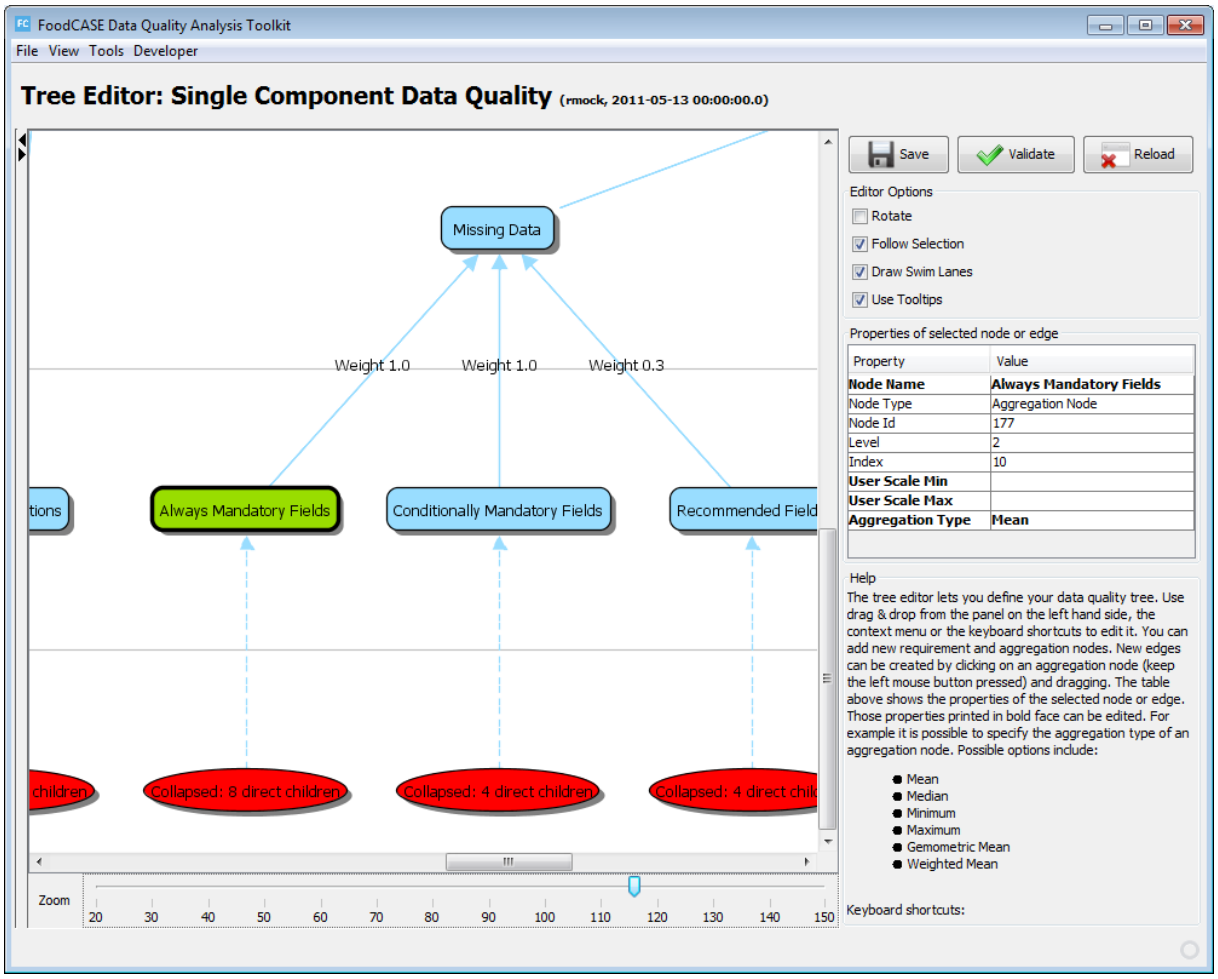


Figure 6.9: A screenshot of the GUI to define RODQ schemas in FoodCASE.

The composite requirement “Always Mandatory Fields” is selected and contains 8 requirements, which are collapsed in Figure 6.9 to give a better overview. The right side of the GUI, which is used to administer assessment details, indicates that the composite requirement is calculated as the mean of the 8 underlying data quality requirements.

Architecture and implementation of RODQ schemas

Two tables are needed to store the RODQ schema: a first table to store nodes in the schema and a second table to store the edges. The edge table has an attribute parent and an attribute child which both are foreign keys to the node table. We then linked the requirement table with the node table so that the basic data quality requirements can be used as leaves of a tree. As there can be different RODQ schemas for users, user groups and data quality entities, an entity representing data quality schemas is necessary. The node table is associated with this schema table to define to which RODQ schema a node belongs.

The advantage of this architecture is its simplicity so that all RODQ schemas can be represented using 4 entities.

6.4 Data Quality Prevention

After the implementation issues of the data quality evaluation, we now focus on data quality prevention. Because data quality prevention has to take place before data is permanently stored in the database, input masks and import interfaces are the places where prevention must be performed.

The most common way for prevention of data deficiencies is to add so called validators to user controls. Typical examples, as can be seen in Figure 6.10, are to add stars or colours to text fields to indicate that the attribute is a mandatory field or to add additional labels to text fields if the attribute values have not a given format. These example validators are single attribute related. For multiple attribute related validators, a typical example is the zip code and the place where a person resides. Such a validator automatically fills the right value for place when the zip code is entered or the validator intervenes when zip code and place do not match to an internal matching table.

As these examples are limited to attributes that are visible on the GUI, the RODQ framework concept of data quality prevention extends the facilities of validators in the following way: Figure 6.10 shows a GUI to enter single nutrient values into FoodCASE. On top, we find the most important information such as the food item and the component for which we want to enter data. Because nutrient values have additional information items, it is necessary to add sub-tabs labelled “Value”, “Samples”, “Methods” and so on. As not all fields are visible at the same time, the concept of the data quality prevention panel, discussed in Section 5.1, is used and shown on the bottom of the GUI, where a list of violated requirements is presented. We combined the RODQ framework concept with the standard ideas of using colours for mandatory attributes, where possible, to facilitate faster finding of data deficiencies.

The data quality prevention panel uses different colours according to the scope classification of the RODQ model. Hard constraints are coloured red and soft constraints as well as indicators are coloured orange. As proposed by the RODQ framework, the policy is to not store any data if hard constraints are violated. In this way it is guaranteed that data has a valid and hence measurable data quality.

As seen above, we made some exceptions on hard constraints because of legacy problems. An example is the English name of a food item which, according to EuroFIR, is defined as mandatory. The old food composition database did not provide an attribute and hence no attribute values are available. So we defined the database attribute to be not mandatory but on the GUI layer the attribute is mandatory. It follows that users are not able to store a modified food item on the GUI before the English name is not provided. But the GUI mask can be open and closed without having to enter the English name as long as no data is changed.

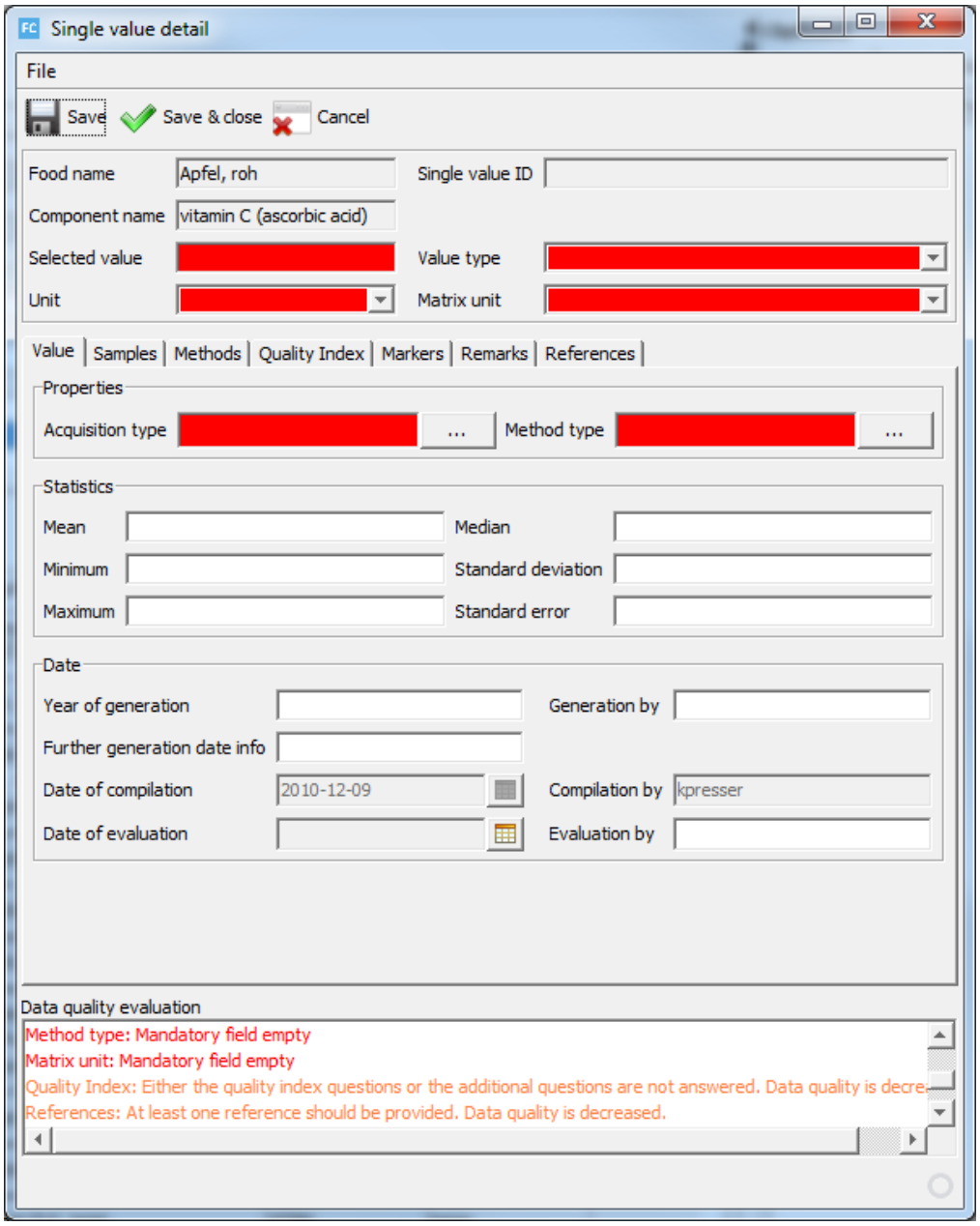


Figure 6.10: Data input GUI for a single food composition value in FoodCASE. The data quality evaluation panel at the bottom is the central place where data quality issues can be analysed. Red coloured entries are hard constraints and orange coloured entries are soft constraints and indicators.

Architecture and implementation

As defined in data quality prevention and in data quality analysis, every GUI on which data can be inserted, modified, deleted and, in the case of analysis, selected, a data quality prevention panel and an analysis panel are required. As the two panels on the GUI work in the same way in that they notify users if requirements are not satisfied, we used only one panel, the so called data quality panel, with information from both concepts. We therefore implemented a default

GUI that contains the data quality panel. Every GUI in FoodCASE, on which data can be manipulated, has to inherit from this default GUI to obtain the prevention panel.
For the data quality panel we used the following approach presented in Figure 6.11:

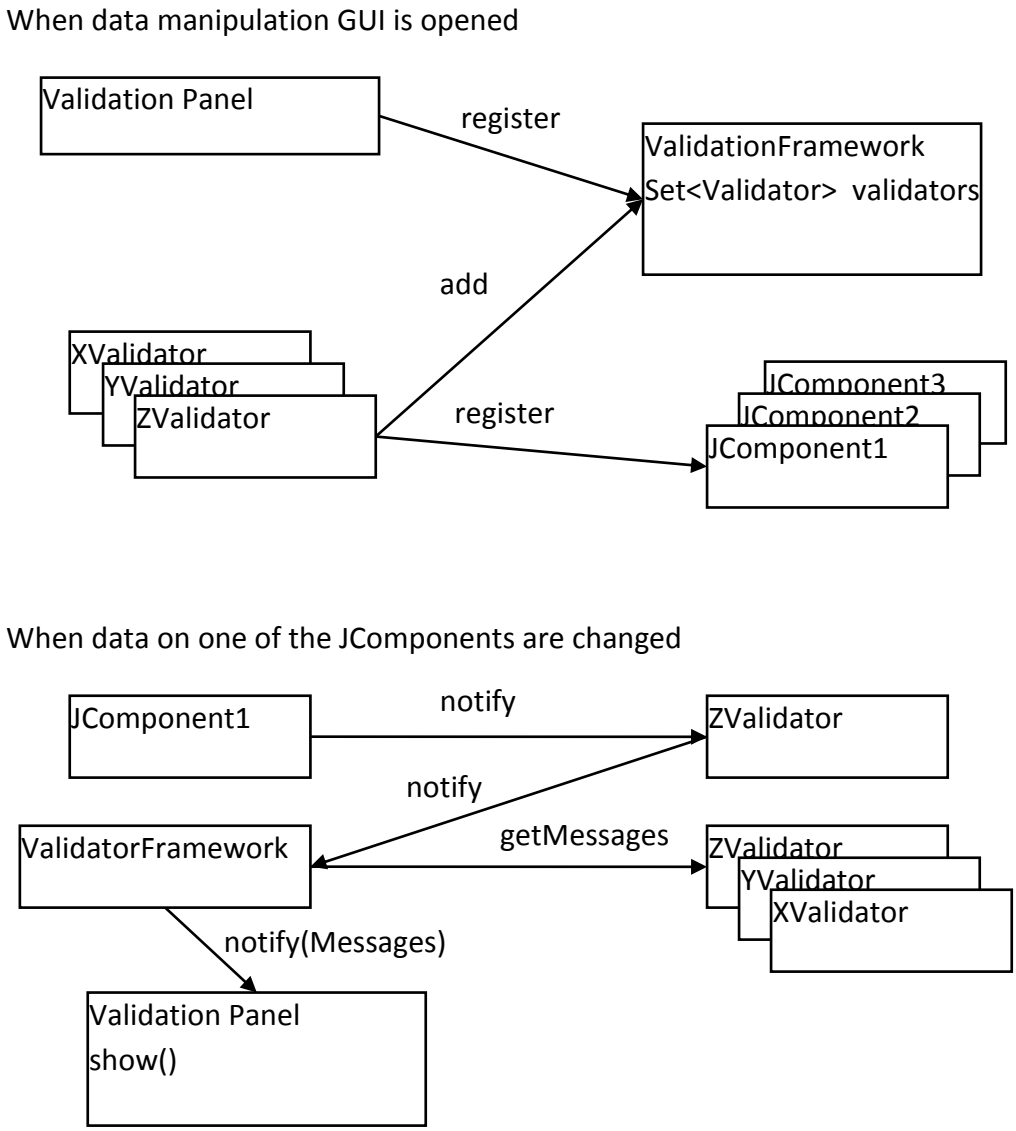


Figure 6.11: Activity diagram when a new data manipulation GUI is opened and when a user modifies data in Java JComponent that has influence on one of the data quality requirements.

When a GUI with a data quality panel is called, a ValidationFramework class is initialised and the data quality panel registers itself to the ValidationFramework. In the constructor of the data manipulation GUI, all preferred Validators are initialised, added to the set of Validators in the ValidationFramework and registered as an observer to a GUI component for which they check a data quality requirement.

If a user changes a value on a GUI component that is observed by a Validator, the JComponent notifies the Validator, which in turn notifies the ValidationFramework. The ValidationFramework asks all Validators to provide their error messages or, in the case of soft constraints and indicators, their suggestions. The ValidationFramework assembles the messages and notifies the ValidationPanel with the well formatted list. The ValidationPanel then presents the messages on the GUI.

Alongside these two activities, Validators of hard constraint requirements colour the JComponent background red if the requirement is not satisfied and white if the requirement is satisfied. If multiple hard constraints observe the same JComponent, all Validators are first asked for errors before the background is coloured.

6.5 Data Quality Analysis and Data Quality Controller

We have seen that the data quality panel is also an implementation of the data quality analysis concept. To be precise, it is only a partial implementation because it implements only the worm's eye view. In this section we present the complete implementation approach of the data quality analysis concept, which is the second important concept for users to assure and improve data quality. In addition, we present how the concept of the data quality controller was implemented.

In the master thesis of Mock [75], the data quality analysis panel was designed, implemented and the concept of the data quality controller was added. A precondition of the data quality analysis is a pre-performed assessment run, which correspond to the data quality versioning concept that is presented in the next section. For the moment, we assume that all data quality requirements were evaluated so that there are valid data quality values that we can analyse.

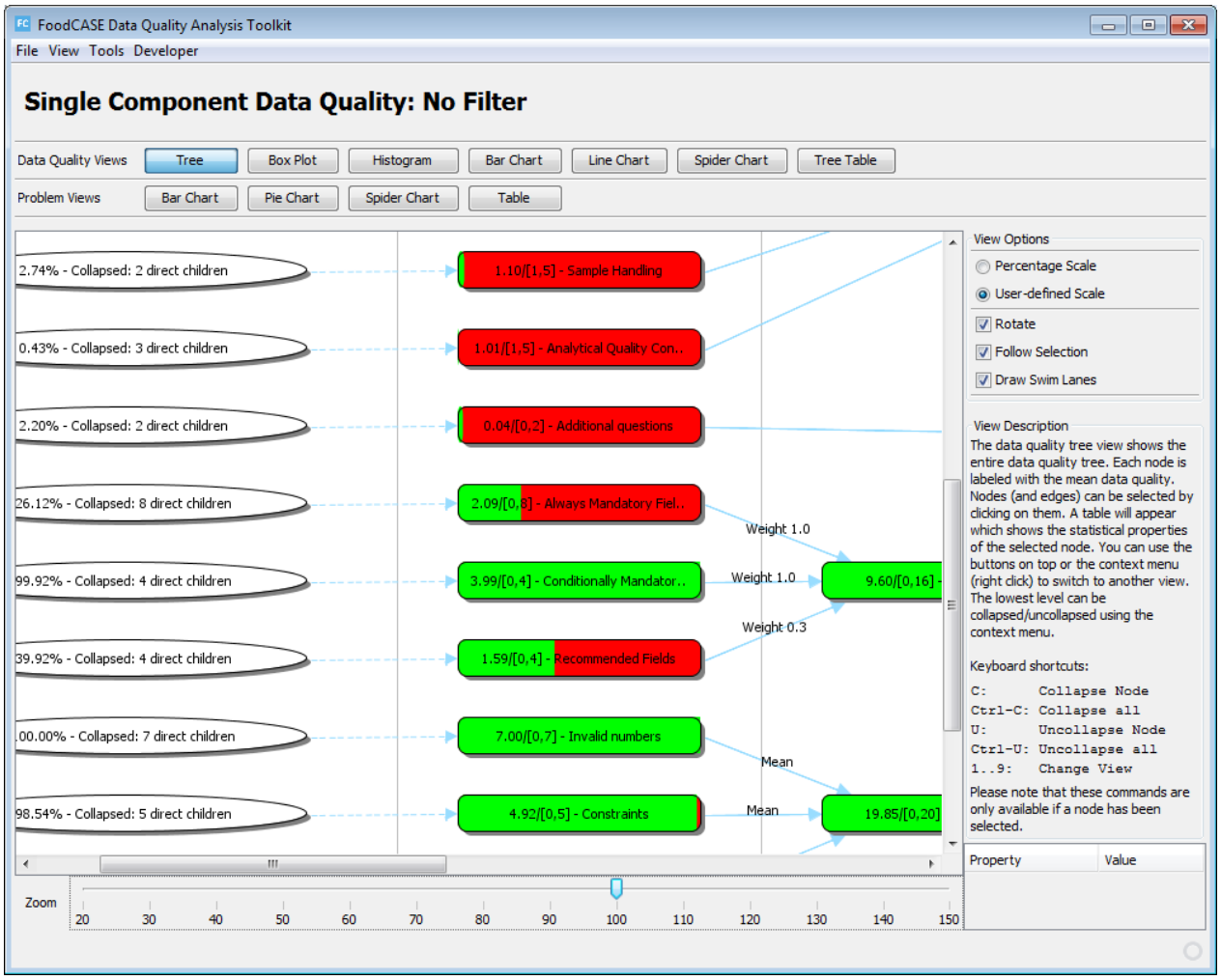


Figure 6.12: The data quality analysis panel to analyse RODQ schemas. The implemented tool is called data quality analysis toolkit.

The print screen in Figure 6.12 presents a part of a RODQ schema that is used to analyse nutrient values on the single value data level. On the left side, the basic data quality requirements are collapsed to give a better overview. The data quality requirement "Always Mandatory Fields" is a composite requirement and consists of 8 requirements. Every red and green box represents a data quality requirement and the border of the box indicates what scope the requirement has. The colour red and green are used to indicate how much of the single values satisfy the requirement: The green part of the box represents the data records that satisfy the requirement whereas the red part represents the data records that do not satisfy the requirement. The number within a box is the numerical representation of the above and means in the example of the requirement "Always Mandatory Fields" that the average data quality value of all nutrient values is 2.09 in a range from 0 to 8.

On top of Figure 6.12 we can see that there are some buttons in the group of "Data Quality Views" and some buttons in the group of "Problem Views". The first group correspond to the concept idea of the bird's eye view. Using the graphical RODQ schema and/or other graphical

components, users get an overview of the data quality over all data records. The data quality analysis panel offers the functionality to select a node in the RODQ schema and analyse the data quality with different graphical schemas such as plots, charts or histogram. It also offers the functionality to step down in the analysis process to the worm's eye view (group of "Problem Views"), which means that all data records are listed that do not satisfy a data quality requirement.

In addition, the concept of the data quality controller was implemented as part of the data quality analysis panel. On every data manipulation GUI in FoodCASE, there is a link to the data quality analysis panel. Using this shortcut only the data presented on the GUI can be analysed in the toolkit. More information about the architecture and implementation can be found in Mock [75].

Performance issue

The approach presented above is the second approach in implementing the data quality analysis concept. In a first approach, a real-time data quality analysis was implemented in FoodCASE. The decision was made at the expense of performance. The reasons were that we weighted real-time analysis higher than the performance loss and that the delay to evaluate all data quality requirements for Swiss food composition data was reasonable. In Figure 6.13, the first approach of the data quality analysis panel is presented.

In the upper part, a user can select from all possible data quality requirements, weight them and see the results in the lower part graphically and detailed in a table. The idea to give the user the feasibility to select and weight data quality requirements, gives the analysis additional flexibility.

The evaluation of all data quality requirements was implemented as a stored procedure in the database. The advantage of this approach was that the requirements as well as the evaluation code were stored in the database and an update at runtime was possible. The disadvantage was the SQL specific implementation and the slow performance of this implementation. Real-time analysis was not reasonable because the calculation of 4.9 million data quality values in FoodCASE version 1.2.0 needed about 30 minutes. Hence, the current implementation needs a pre-performed assessment run before data quality analysis is possible.

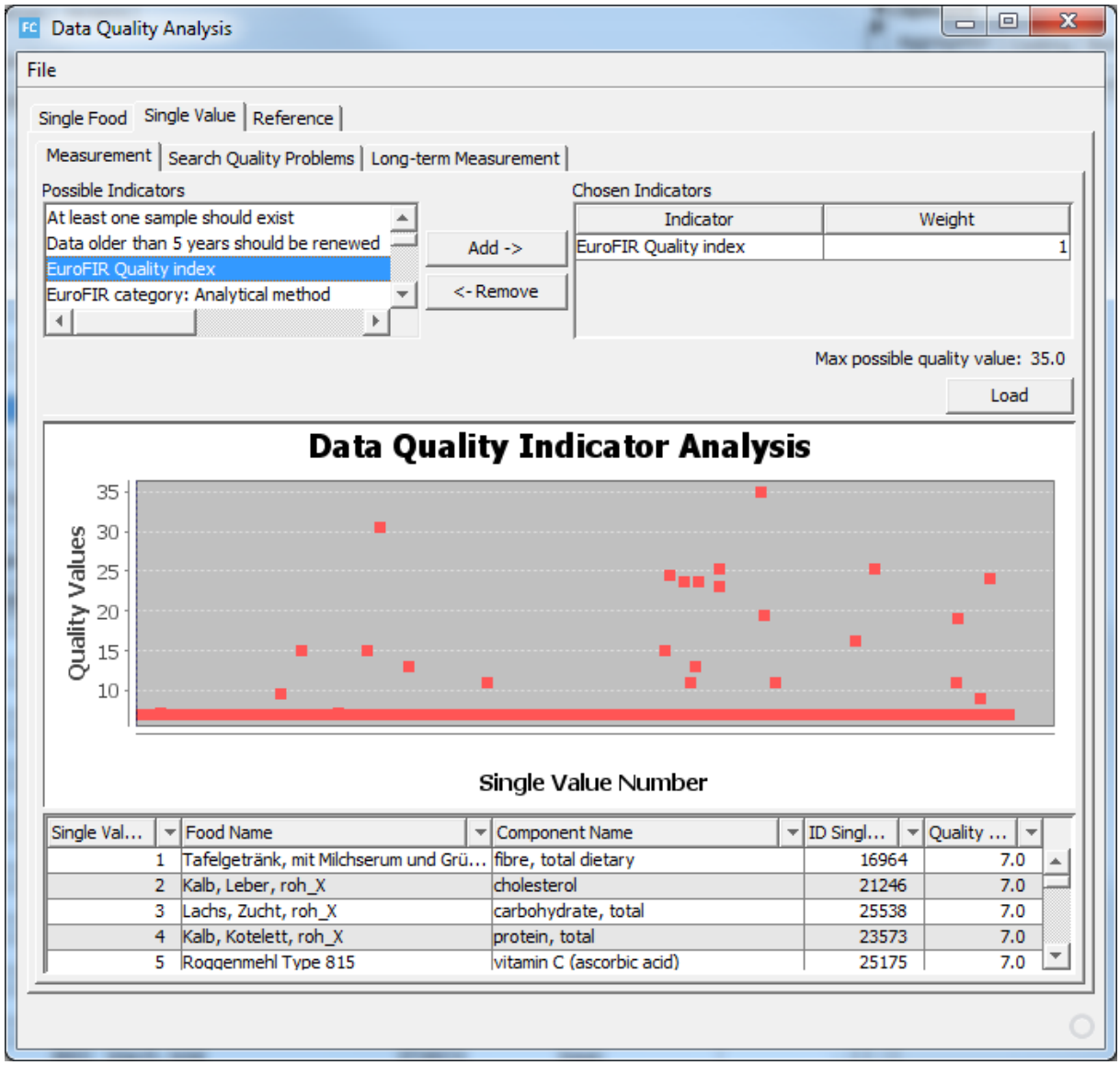


Figure 6.13: The data quality analysis panel of single value is presented. Users have the possibility to choose data quality requirements, see the maximum possible quality value and get a graphical and tabular representation of the measurement.

6.6 Data Quality Versioning and Data Quality Assessment Run

To be able to use the data quality analysis panel also for long-term analysis, data quality requirements need to be automatically evaluated and stored over time. The implementation is simple in that the table tblQualityRequirementValue, presented in Section 6.3.1, must be extended with an attribute version and an attribute timestamp and an automated job must be defined to perform the answering, scoring and storing in the table. More difficult is the management of the amount of data that is generated, which will be discussed below.

In a first approach, the RODQ framework measured all basic data quality requirements once a week. We soon realised that the size of the backup files will grow rapidly. The backup size of the

Swiss food composition database after migration from the old system was 2.3 MB. Different data adjustments were then made either by script or by hand. Until the first snapshot, the backup file grew to 6 MB. After the first snapshot the backup file was 12 MB, which doubled the size of the database. We observed the growth of the backup file over some months and give an overview in the following table:

Date (yyyy-mm-dd)	Size in MB
2010-06-10	6.0
2010-06-11	12.0
2010-06-21 (later this day a new version of aggr. values was made)	46.3
2010-07-15	70.4
2010-08-22	90.1
2010-09-27	113.5
2010-11-19	137.1

Table 6-4: The development of the backup file size for Swiss food composition data version 3.1 over 5 months.

In the first snapshot, 14 data quality requirements were measured on 1282 single food item and 21 requirements on 37933 single value items. We got 17804 data quality values for single food items. This number is not equal to 14*1282 because not all data quality requirements were measured for every single food because, depending on the data, not all requirements are applicable. We got also 722'273 data quality values for single value items, which is again not equal to 21*37933 for the same reason. In the following the number of data quality requirements grew together with the size of the backup file. The following table shows two data quality assessment runs (2010-06-11 and 2010-10-28) to see the growth:

Date (yyyy-mm-dd)	Number of food items	Number of nutrient values	Number of references	Number of requirements	Number of data quality values
2010-06-11	1282			14	17'804
2010-06-11		37’933		21	722'273
2010-10-28	812			14	11'248
2010-10-28		37’191		30	782'805
2010-10-28			307	2	614

Table 6-5: The development of food composition data and their requirements in FoodCASE for Swiss food composition data version 3.1 over 4 months.

References were not measured in the first assessment run but only in the second. For this reason, there are no data for references at 2010-06-11. The Swiss food compilers worked on the

data between the two runs and hence the number of food items and the number of single values were reduced. At the same time, more data quality requirements were identified and implemented. Totally we had almost 800'000 data quality values in the second data quality snapshot. We have seen in the preceding sections that in FoodCASE version 1.2.0 we had already 4.9 million basic data quality values because the total number of data quality requirements increased to 156.

What needs to be measured in an assessment run?

With this amount of basic requirement values, the question arose whether it is sufficient to only store atomic data quality requirements or if also composite value and quality key values should be stored? From our experience in FoodCASE version 1.2.0, it is sufficient to store only atomic requirements because less effort is needed to calculate the composite values and quality key values. For atomic requirements, multiple objects and attributes must be checked whereas for composite values and quality key values only atomic quality values must be taken and an aggregation must be calculated, which is a simpler operation than all join operations on the database. In FoodCASE, it was additionally the case that fewer requirements exist that are of type composite or quality key than atomic quality values. Hence, as long as the delay to calculate composite and quality key requirements in a RODQ schema is accepted by users, only atomic values must be stored.

6.7 Summary

In this chapter, the food composition information system FoodCASE has been introduced. FoodCASE was built to manage Swiss food items and their nutrient values as part of this thesis. As the system has to administrate scientific data, it has a strong focus on value documentation and on data quality. The three data level single value, aggregated value and recipe have been presented, which are used to prepare scientific data for publication.

We also have presented how the RODQ framework together with the RODQ model was implemented as proof of concept in the research object FoodCASE. In order, we showed the implementation of the concepts data quality requirements, user-answer-needed requirements and data quality assessment, which all belong to the data quality evaluation. In addition, we discussed the architecture issues of data quality requirements and data quality assessment as well as the issue what data quality value should be chosen for unanswered requirement's question. We then presented the implementation of the data quality prevention panel with the data quality panel on every manipulation GUI. We showed an architecture to register data quality validators to the data quality panel. When these validators are notified about changed data, they send messages for users to the data quality panel. We presented also the implementation of the data quality analysis concept and discussed the performance issue of real-time analysis of data quality requirements. The critical point is that the amount of data

quality values in applications such as FoodCASE, can make a real-time data quality evaluation almost impossible. This finding was evaluated with two approaches of the data quality analysis panel. The analysis panel have also been presented together with a variant of the data quality controller that enables the customisation of data quality requirements in such a way that recompilation of the software get obsolete. We also have shown how data quality versioning and data quality assessment runs simply can be implemented but that the amount of data needs a careful storage strategy.

7

Evaluation of the RODQ Framework and Model

We have seen in the last chapter how the RODQ framework including the RODQ model was implemented in FoodCASE. This implementation was the first step in showing that our framework and model are applicable. The second step of evidence is presented in this chapter and includes further investigations of our concepts including user studies. In Section 7.1, we show why the idea of a flexible data quality requirement and data quality RODQ schema are necessary by using the example of change in reliability assessment in the Swiss food composition database. In Section 7.2, we describe a usability study conducted with national food compilers on FoodCASE. Finally, we present a field study performed at the University of Applied Science ZHAW in Winterthur in Section 7.3. The objective of this field study was to investigate missing values on course registrations and collect user feedback for our proposed concepts for data quality analysis and data quality evaluation.

7.1 Evolution of the Data Quality Requirements in the Swiss Food Composition Database

We have seen that users are able to enter their own data quality requirements. Why such flexibility is necessary was already indicated in the last chapter where version 1.0.0 had 21 requirements for single values whereas version 1.2.0 had 82. Obviously, the main reason is that FoodCASE and the RODQ framework were newly introduced and new requirements were

formulated during the usage of FoodCASE and its data quality framework. Hence the growth in data quality requirements is likely to be greater at the beginning than after 5 years of usage.

Two other reasons why such flexibility is used can be found in the history of the Swiss food composition database. The first evidence is the evaluation of the reliability assessment and the second is the evolution of hard and soft constraints, which both were discovered during our study on the history of the Swiss food composition database.

We found the data reliability assessment schema from the project "Aufbau einer schweizerischen Nährwerttabelle" lasted from 1998 to 2001. We compared this schema with the schema implemented in FoodCASE, which was defined by EuroFIR in the project from 2004 to 2009 in cooperation with the Swiss food compilers. Both assessment schemas used a grouping of concrete questions by introducing categories. In Figure 7.1 we can see that most categories match but there are also differences between them. The more interesting point is the investigation of the concrete requirement questions and the assessment schema.

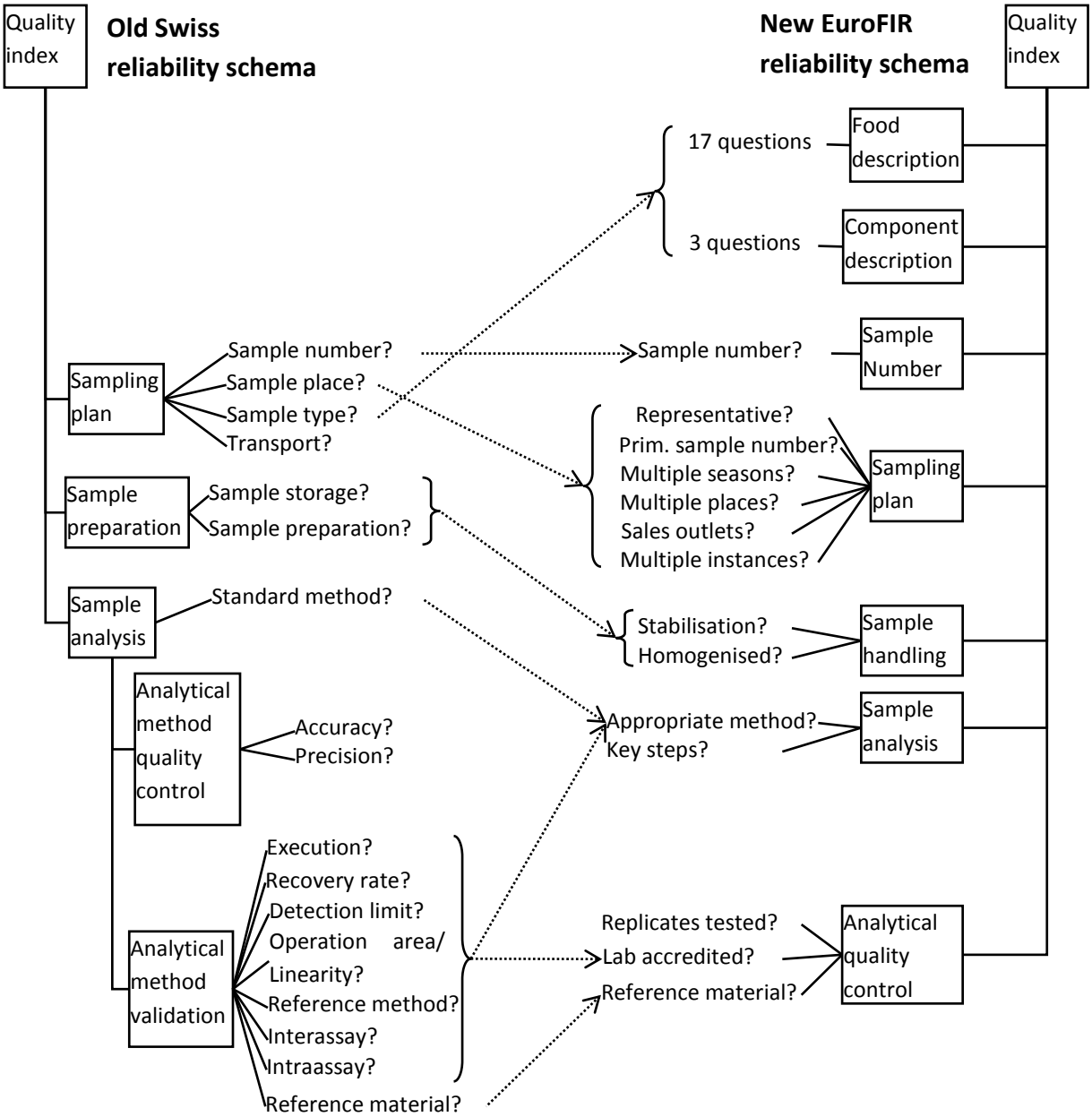


Figure 7.1: The schema compares the old and the new data quality reliability schemas in Switzerland and indicates the evolution of data quality requirements from the old version to the new version.

In Figure 7.1, the old Swiss quality index with its categories and their requirement questions are listed on the left. On the right, the quality index schema is presented and the arrows connect the old questions to the new ones. The matching is not perfect but must be seen as an approximation. The reason must be sought in the variability of natural languages. Some questions were reformulated, condensed, multiplied with adjustments or slightly changed in a sense that the questions do not mean exactly the same.

In addition, the differences in Figure 7.1, there are also differences in the assessment schema. The old schema had 16 questions which are mostly of type YES/NO questions whereas the

EuroFIR schema has 34 questions mostly of type YES/NO/Not applicable. The maximum score for the old schema was 19 and is 35 for the new schema. If we consider the years in which both projects ended, we can see that, within 8 years, the number of question more than doubled and the most common types of questions changed from a binary to a three-state question.

The second evidence that we found in the history of the Swiss food composition database was when comparing requirements of the forerunner software of FoodCASE with the requirements in FoodCASE. In particular, we compared hard and soft constraints when entering a new food item and a new food nutrient value. The old system had only the single value level and the recipe level. As an example, we will compare data quality requirements of the single value level.

The old system had two hard constraints on the single value level: A food must have a name and for a nutrient value the component must be given. Users were able to enter a new single food nutrient value entry without a value. FoodCASE had 6 hard constraints for a food items and 11 hard constraints for a single food component values in version 1.0.3.

Soft quality constraints were not included in the old system whereas FoodCASE has 9 soft constraints for food items and 18 soft constraints on single food component values in version 1.0.3.

The number of hard constraints is higher in FoodCASE which means in consequence that new data enters FoodCASE with higher data quality than in the old system. That was one goal of the RODQ framework.

7.2 Usability Testing on FoodCASE

After a first implementation effort on FoodCASE and the evidence that the set of data quality requirements can change over time, a usability test on FoodCASE was conducted. In addition to being able to show that the system could represent data quality requirements, it was necessary to evaluate usability. Investigated were the FoodCASE modules Single Value and Recipe to get a sufficiently deep insight into the system in order to evaluate its potential as a standardised food composition information system. The pilot version 0.8 of FoodCASE was usability tested, where the RODQ framework concepts data quality prevention and data quality evaluation were fully implemented. The usability study was published in the EuroFIR project as one of the deliverables [76].

7.2.1 Approach

The approach used for the usability testing of FoodCASE 0.8 was based on the usability testing of EuroFIR's eSearch prototype [77], the usability testing carried out on the Online-Version of the Swiss Food Composition Database [78] and considering the recommendations laid down by Hornbæk [79].

The users were instructed to download the Java client of FoodCASE on their institute computer. The FoodCASE client established a connection to the FoodCASE server at ETH Zurich. In this way, we had also a performance test when data is sent over the Internet.
The usability testing designed for FoodCASE 0.8 consisted of two specific, but everyday tasks of a compiler's routine work: Adding a nutrient value with meta-information to a food existing in the database and the creation of a recipe. Additionally, the participants were invited to have a general look and play and to give feedback about their experience, both positive and negative, whenever possible. They were also invited to give suggestions for improvements. We used an online feedback form on Surveymonkey.com and, all in all, the testing consisted of 40 questions, mostly multiple choice questions and some open questions.

7.2.2 Structure of the Questionnaire

For most of the multiple choice answers, we used a 5-points Likert scale. In addition, we also used Yes-No questions and open questions, where participants could freely state their problems, impressions and suggestions.
The questionnaire used in this study is structured into 6 sections. The first section contains questions about general user information and the second section contains questions about first impressions by investigating users' opinions about the look-and-feel, basic structure and navigation of FoodCASE. The next two sections contain questions about the two food compilation tasks. Then the section with questions about the second impression of FoodCASE, which are similar to the question about first impression, and the section about likeliness of using FoodCASE in participants' institution finalise the questionnaire. Some selected questions are described in the following, for which results will be presented in the next section.

User information

We asked users to estimate their skills as computer users and as food composition data compilers. In addition, we wanted to know how they currently managed their data. Therefore, we formulated the following questions:

- How do you estimate your general computer skills?
- How do you estimate your skills as food composition data compiler?
- Which software are you using to manage your food composition data?
- Are you using FoodCASE for the first time?

Task1: Insert nutrient value

We then asked participants to enter food composition data into FoodCASE according to the following hypothetical lab report:

Date of Report: March 4, 2010
Author: Mr. PerfectLab
Organization: PerfectScience Lab, University of Zurich, Switzerland
Description: 10 samples of the food were purchased in the supermarket PerfectFoods in the city of Zurich on March 3, 2010. The samples were transported to the university laboratory according to the information given on the package and then stored in the laboratory as recommended on the package. The magnesium analyses were carried out the day after the purchase. Magnesium was analysed according to a correct AOAC method (atomic absorption spectroscopy) and using all relevant certified reference materials and other standards. The analyses were carried out in triplicate for each of the 10 samples; the mean and standard deviation for the 10 samples was 35 ± 15 mg / 100 g edible portion.

After the task, we were interested in 4 questions that are also posed at the end of task 2. These 4 task questions are listed in Table 7-1 with their possible answers that the questionnaire offered to the participants and the answer number that is used for the graphical presentation on the X-axis in Figure 7.3.

Were you able to successfully complete the task, i.e. do you see the value just entered in the component list?		
Answer Options	**Yes**	**No**

Is FoodCASE built logically to complete this task?					
Answer Options	**completely illogical**	**somewhat illogical**	**neither illogical nor logical**	**rather logical**	**completely logical**
Answer Number	**1**	**2**	**3**	**4**	**5**

How simple/difficult was it to complete this task?					
Answer Options	**absolutely simple**	**somewhat simple**	**neither simple nor difficult**	**rather difficult**	**absolutely difficult**
Answer Number	**1**	**2**	**3**	**4**	**5**

How long did it approximately take to complete this task?					
Answer Options	**< 3 min**	**3 to 5 min**	**>5 to 10 min**	**>10 to 15 min**	**> 15 min**
Answer Number	**1**	**2**	**3**	**4**	**5**

Table 7-1: List of 4 task questions that are investigated on both tasks to compare the answers.

In addition, we ask participants, who are not able to complete the task to describe why they were not successful and we also ask the open question if there are any changes or improvements participants would like to suggest.

Task 2: Assemble a recipe in FoodCASE and start automated nutrient value calculation

As the second task, we asked participants to search and select ingredients for a given recipe and to let FoodCASE calculate all nutrient values according to the following instruction:

> **For the purpose of the testing we use the following shortened list of ingredients:**
> 2 raw eggs
> 1 tea spoon white sugar
> 3.75 dl whole milk pasteurised
> 125 g butter of choice
> 750 g wheat flour bakery type 550
>
> Everything has to be mixed in a specific order (sorry, can't name the order, it is a secrete). The dough is left to levitate for one hour at room temperature, the dough is woven in a traditional way, and then baked in the oven at 200 °C for 45 min. Name the recipe "Bauernzopf_yourInstitution", i.e. for ETHZ it is Bauernzopf_ETHZ. The yield factor for water is 0.9 and the other yield factors are 1.0.

Beside the same 4 task questions, we asked the open questions why participants were not able to complete the task, what problems they encounter and what suggestions they have.

First impression and second impression

Before users were asked to answer the questions of the section first impression, they were given time to take a look at FoodCASE. We did not define a specific task nor did we define how much time they should spend. We just let participants start FoodCASE and let them entirely free to do anything. We then asked users about their first impression of FoodCASE. After the participants performed the two tasks, we asked again for their impression. We asked the same questions in section first impression and second impression to be able to compare them. The two impression questions are listed in Table 7-2, where also possible answers are presented.

Select the choice which most closely matches your feelings. The first impression of the look-and-feel of Food-CASE V0.8 is...					
Answer Options	**terrible**	**bad**	**neither bad nor good**	**good**	**excellent**

Do you think that the basic navigation and structure of FoodCASE is intuitive in respect of performing some simple tasks? FoodCASE is...					
Answer Options	**not intuitive at all**	**somewhat intuitive**	**neither / nor**	**rather intuitive**	**completely intuitive**

Table 7-2: List of 2 impression questions that are investigated before and after the two tasks to compare the answers.

In addition, we asked the following question in the section second impression:

Considering the two tasks performed and the few minutes you looked at FoodCASE at the beginning, what do you think about the understandability of FoodCASE? FoodCASE is…					
Answer Options:	**in need of explanations**	**somewhat in need of explanations**	**neither / nor**	**rather self-explanatory**	**self-explanatory**

Table 7-3: Question about understandability of the FoodCASE usability study with the possible answers.

A few questions were integrated in the section about the second impression to collect additional feedback to investigate the usability of FoodCASE. Some selected multiple choice questions are listed in Table 7-4.

How do you like the design of FoodCASE? The design is …					
Answer Options:	**terrible**	**bad**	**neither bad nor good**	**good**	**excellent**

What do you think about the information density/number of fields on the two modules single value and recipe calculation? The information density/number of fields is ...					
Answer Options:	**too little**	**somewhat little**	**as it should be**	**rather high**	**too high**

FoodCASE was designed as a basic system for the everyday tasks of a compiler. How do you rate FoodCASE in this respect? FoodCASE is...					
Answer Options:	**less than a basic system**	**somewhat less than a basic system**	**a basic system**	**rather more than a basic system**	**more than a basic system**

According to your personal opinion, does FoodCASE in its V0.8 contain all features you need to manage your food composition database? FoodCASE contains...					
Answer Options:	**only few features I need**	**some features I need**	**about half the features I need**	**most features I need**	**all features I need**

Table 7-4: Additional usability questions in section second impression.

Likelihood of using FoodCASE

As part of the study, we wanted to investigate how likely it is that participants think that FoodCASE could be used in their institution. One prerequisite that somebody exchanges an existing system can be that the new system provides more or better functionality and therefore increases customer value. We designed the questions, listed in Table 7-5 to target that issue.

How do you rate FoodCASE, as it is now in V0.8, compared to what you use now to manage your food composition database? FoodCASE V0.8 is …					
Answer Options:	**much worse**	**worse**	**the same**	**better**	**much better**

If the first full release of FoodCASE V1.0 will contain the features you need (including any you mentioned previously that are missing), how will it compare to your current system? FoodCASE V1.0 will be …					
Answer Options:	**much worse**	**worse**	**the same**	**better**	**much better**

Table 7-5: Two selected likeness question with their answer options.

Open questions and RODQ framework

We already mentioned some of the open questions in the questionnaire sections above. We had additional open questions in the section second impression and Likelihood of using FoodCASE such as

- "What is missing according to your opinion?"
- "Can you describe things you especially liked about FoodCASE?"
- "Can you describe things you especially disliked about FoodCASE?"
- "If you have any additional comments that you would like to make, please record the here."

All in all, there were 11 questions where open feedback could be provided. All of them were indicated as optional. The objective was to give users several opportunities to make comments that cannot be expressed by multiple choice questions. The open questions were also spread over the whole questionnaire to minimise the danger of forgetting a comment.

7.2.3 Results

In total, 53 persons involved in compilation of food composition databases representing 29 institutions and 22 countries (with two non-European countries) were invited to participate in the usability testing. The invitation was accepted by 36 persons and all of them started the testing. However, only 25 completed the online feedback form. The Swiss compilers did not participate in the testing, because they were already accustomed to FoodCASE due to its use in their everyday work and because they helped to prepare the user study.

In the following, we present the results of this usability study in terms of selected questions and provide the full details of questions and answers in Appendix C.
The results are given according to the sections above, expect for the first impression, which is presented together with the second impression. Only the results of the 25 full responders are displayed. The differences between the 25 full responders and 36 partial responders for the first 8 questions, for which all 36 partial responders gave feedback, are smaller than 1% and are therefore negligible.

User information
The following figure presents the results of the first two questions as pie charts:

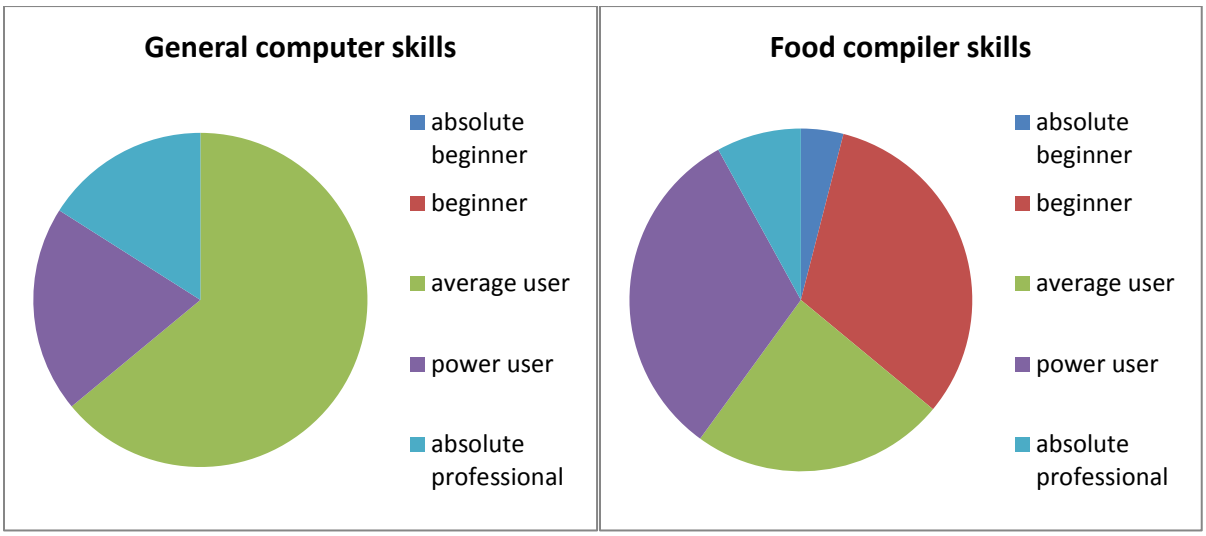


Figure 7.2: Self-estimated skills of participants on computer in general and on the food compilation process.

Whereas most participants estimate themselves as average computer users, there are 36% that estimate themselves as absolute beginner or beginner in food compilation. The number of power users and absolute professionals is smaller on the computer side (36%) than on the food compilation side (40%). However, there are less absolute professionals on the food compilation side than on the computer side.
11% use Microsoft EXCEL as their food composition management system whereas 84% of the participants answered that they have specific software based on a relational DBMS such as ACCESS, SQL Server, Oracle, MySQL or PostgreSQL.
FoodCASE was released some weeks before this usability study and 32% had already used the software.

Task1: Insert nutrient value

22 of 25 users (88%) were able to successfully complete the tasks whereas 3 persons were not able. One person from South Africa commented to have tried for 4 hours to complete the task and give up because of a very slow Internet connection. The second person stated supposed to have performed the task too quickly and had not have enough knowledge of the software. The last person commented to not have been able to store entered data.
The results of the 3 other task questions are visualised in the following figure:

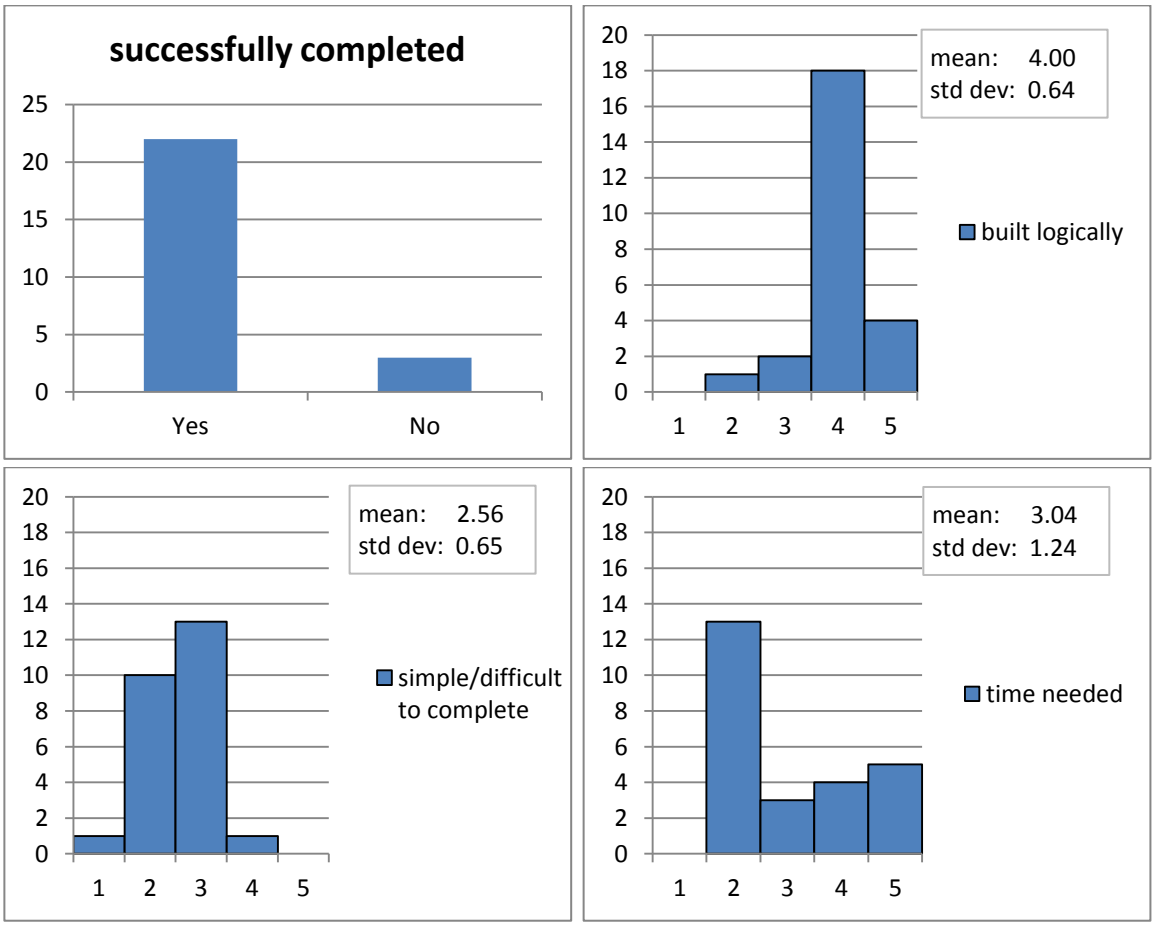


Figure 7.3: Answer profile of the participants' answers to 3 task questions for task 1.

18 users (72%) rated FoodCASE to be rather logically built to complete the task whereas 13 users (52%) of participants had the opinion that the task was neither simple nor difficult. Most users needed about 3-5 minutes to complete the task, which is faster than we expected after the trials with the Swiss food compilers.
The 16 feedback items for the open question to suggest changes or improvement mostly describe that a manual or instructions to use FoodCASE would be helpful. There are also some improvement suggestions such as import functionality for EXCEL files, keyboard shortcuts or advanced search functionality. But no feedback item was mentioned twice expect that the field "Date of sampling" should provide a calendar to pick a date.

Task 2: Assemble a recipe in FoodCASE and start automated nutrient value calculation

We asked the 4 task questions defined in Table 7-1 and the answers are displayed in Figure 7.4.

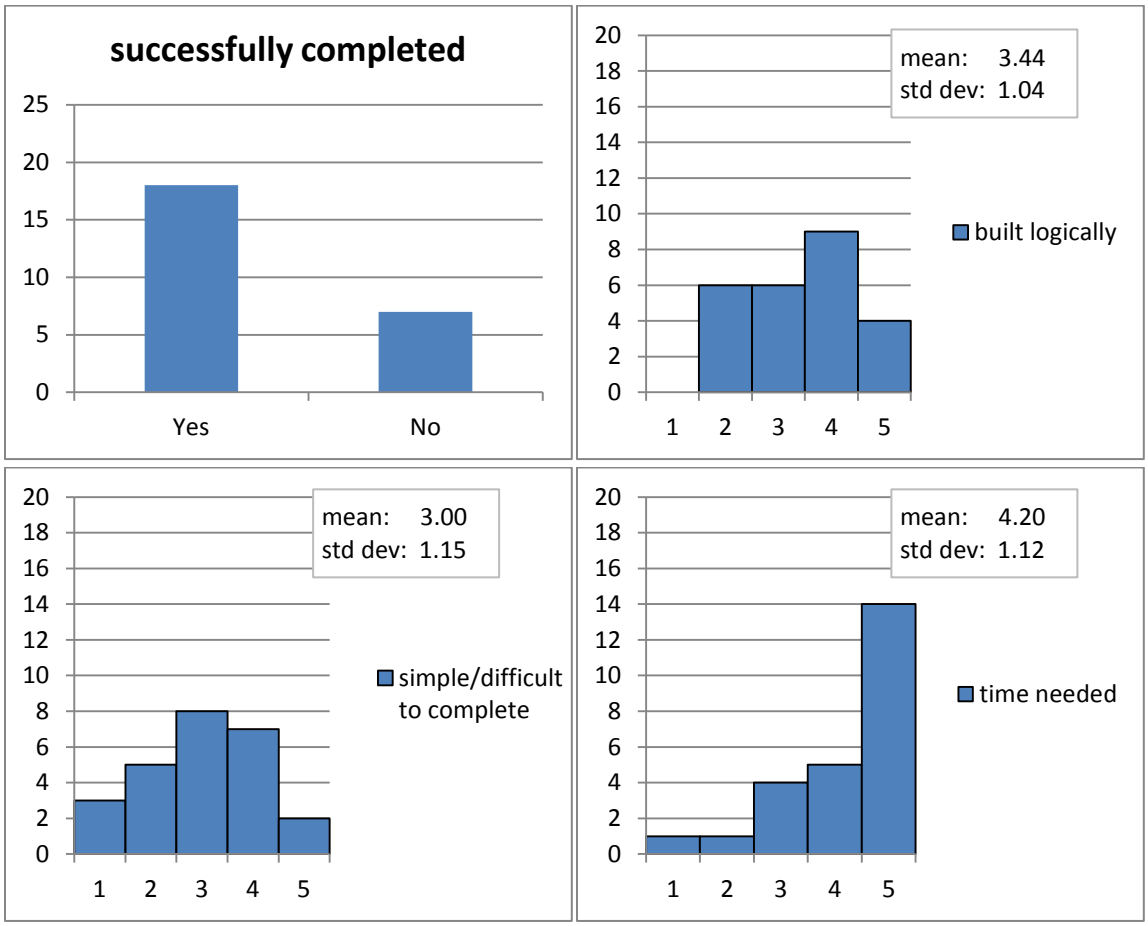


Figure 7.4: Answer profile of the participants' answers to 3 task questions for task 2.

The profile of the last 3 task questions in Figure 7.4 looks different than the profile in Figure 7.3. There is a clear shift in the participants' answers so that FoodCASE is built less logically for the second task and that the task is more difficult to complete than the first task. Also the time needed to complete tasks 2 is clearly higher than in task 1.

72% of the participants (18 persons) were able to successfully complete the task. The other 7 persons who were not able to complete, had problems to find requested ingredients, to enter the amounts of ingredients and encountered problems saving the recipe.

The open question showed again that some users would like to have a manual or instructions. Some users encountered a problem on adding an ingredient. The problem was that FoodCASE had a bug on the ingredient selection GUI. It was possible to add an ingredient but users had to find out how. Another point that was mentioned was the labelling of the two buttons on the recipe GUI. They were labelled as "Calculate" and "Cancel" and some participants wanted to

first save the recipe before starting the recipe calculation. A drag-and-drop functionality to add ingredients to a recipe is also an improvement that users mentioned. Beside these problems and suggestions, there were again points that are mentioned only once.

First impression and second impression

The following Figure 7.5 and Figure 7.6 compare the answers of the impression questions.

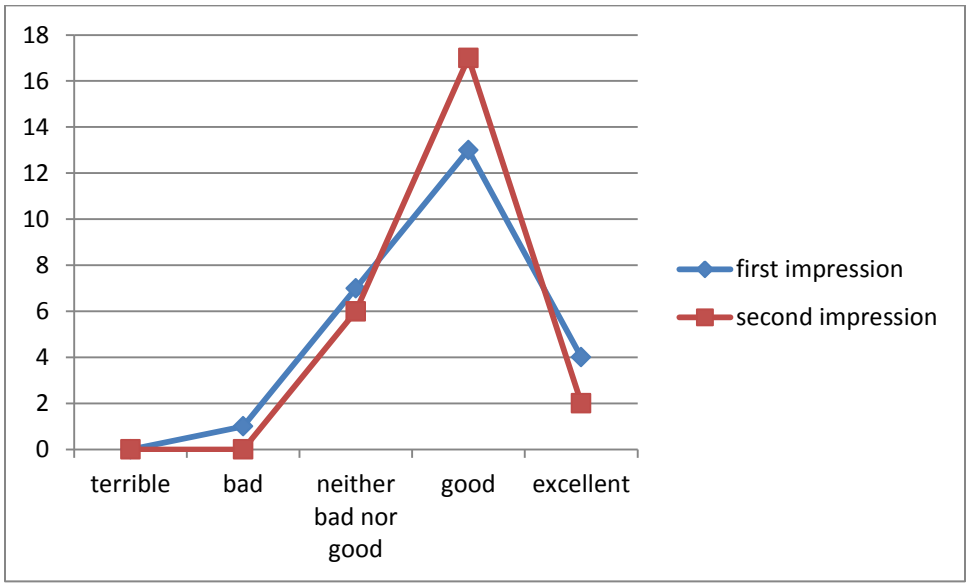


Figure 7.5: Comparison of the first and the second impression of participants about FoodCASE.

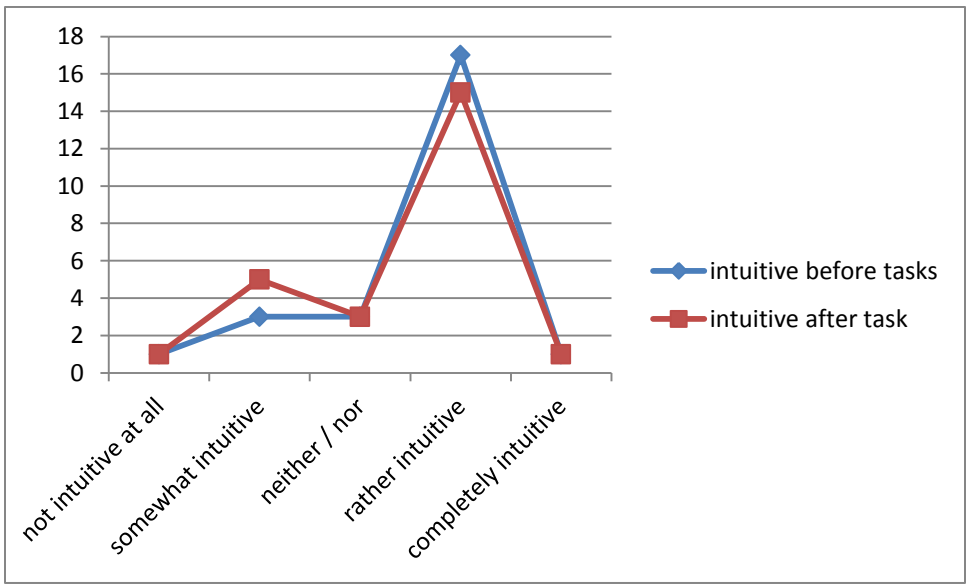


Figure 7.6: Comparison of participants' feedback about the intuition of FoodCASE before and after performing two tasks.

It can be seen that FoodCASE makes a better second impression than first impression whereas the intuition decreases a little after performing the two tasks.

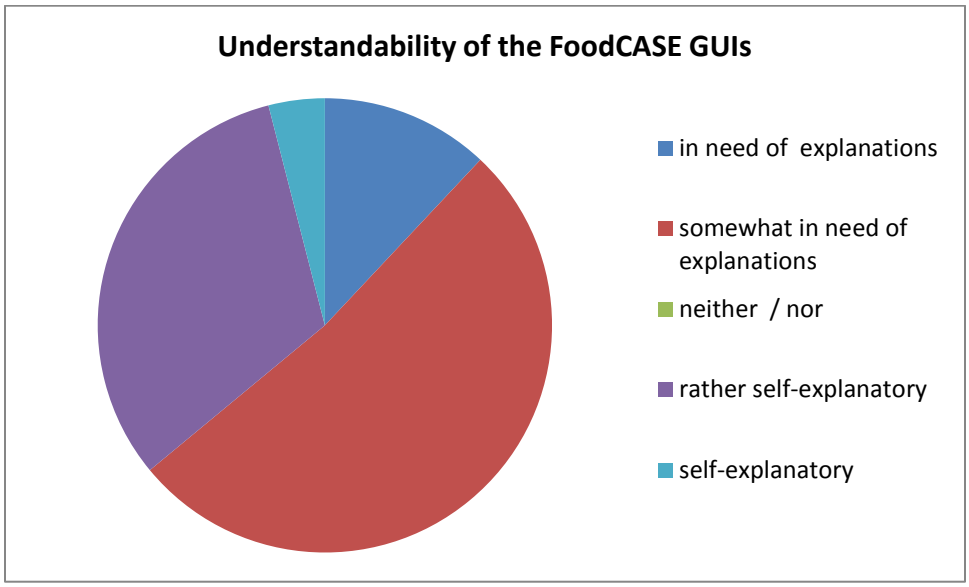


Figure 7.7: Participants answer to the question if FoodCASE is understandable.

This question is somehow related to the question about the intuition of FoodCASE. Obviously, there is a clear difference between intuition and understandability. We will discuss in the next section what this difference might be and from where this difference originates.

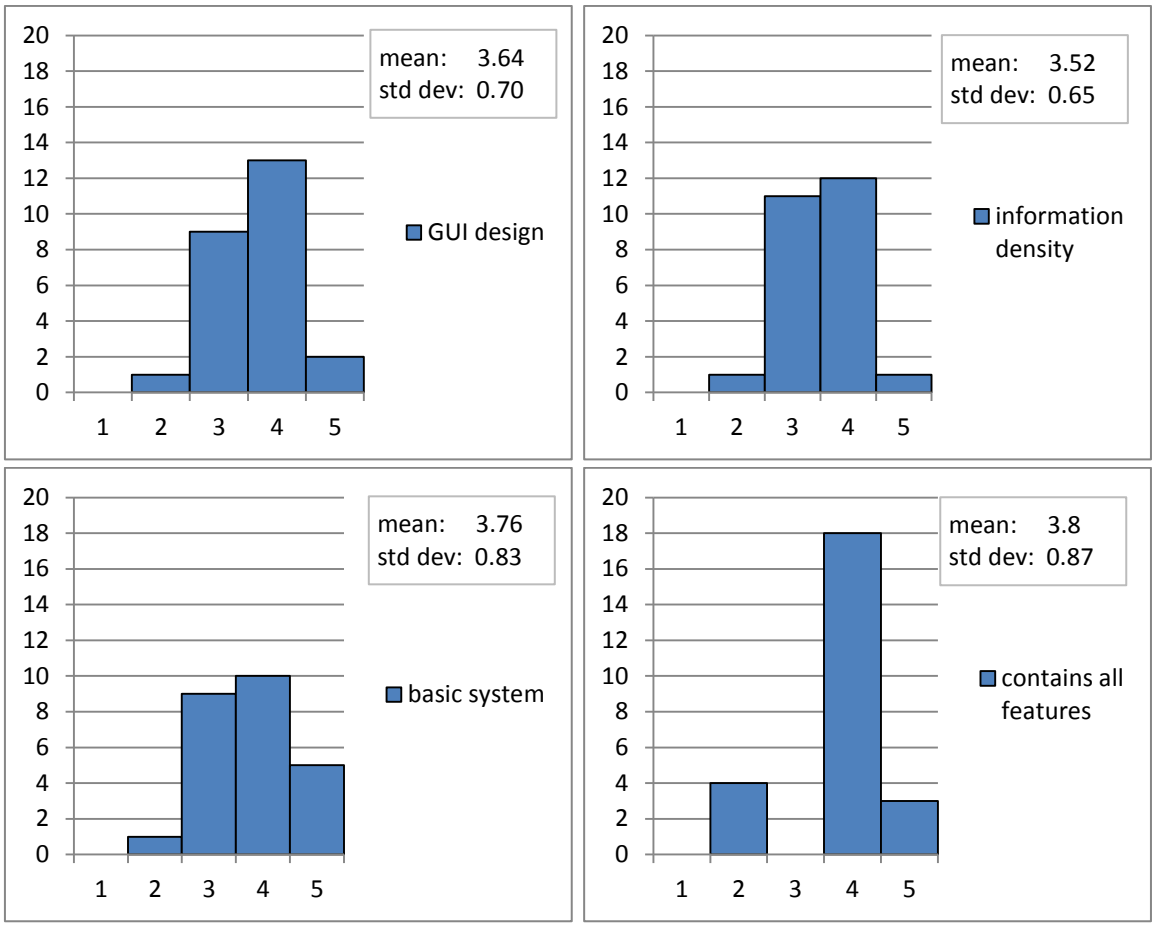


Figure 7.8: Participants feedback on selected question of the section second impression.

The upper two histograms in Figure 7.8 show that participants rated the GUI design of FoodCASE to be rather OK and the information density, which means the number of input controls per GUI, to be at the upper limit. The lower two histograms present a good result. FoodCASE is rated to be a basic system or more than a basic system for food compilation with 96% whereas version 0.8 is rated to contain most required or all required feature by 84% of participants.

Likelihood of using FoodCASE

Figure 7.1 compares the two likeness questions. Why one person rated FoodCASE V1.0 lower than V0.8 (much better), is not clear for us.

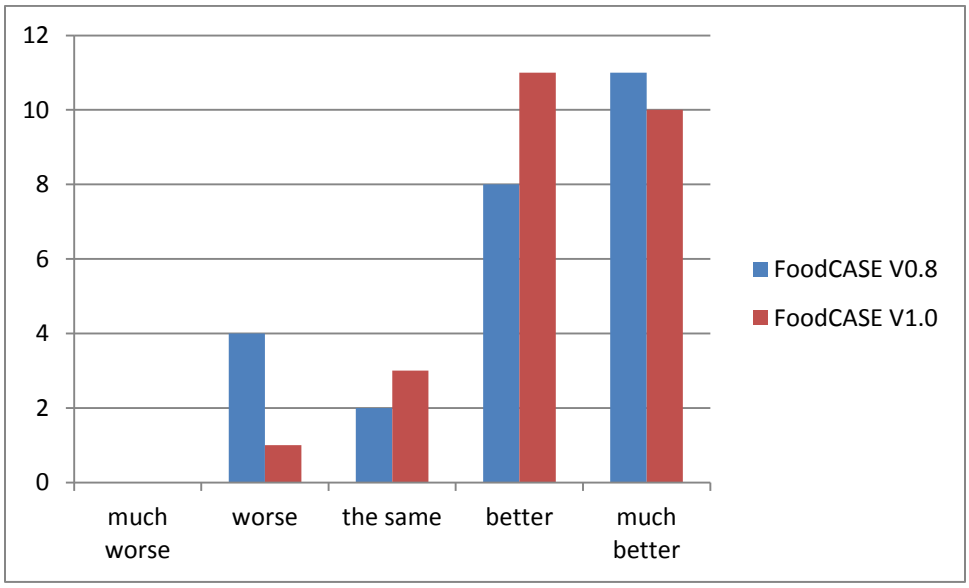


Figure 7.9: Participants' feedback if they compare FoodCASE to their current food composition management system.

Open feedback and RODQ framework

We got 137 feedback items for the open questions. As already mentioned, there were also feedback items describing problems where participants did not know how to proceed. Also mentioned were some feedback items concerning proposals for additional functionalities that would improve FoodCASE while some contained bugs that participants encountered during testing.

There were also feedback items on our RODQ framework although we did not ask specific questions about the data quality framework. We got 14 feedback items from 12 different users on the implementation of the data quality prevention panel and the data quality evaluation concept, such as:

1. "It would be helpful to receive instant feedback to be reminded on missing relevant parameters"
2. " … Then FoodCASE complained that some of the amounts were missing (Please fill in all values for all ingredients). Then later on I got the message that yield factors are invalid …"
3. "… I noticed duplicity in reference list of the record Mr Perfect Laboratory, there should be some monitoring of the existing reference list records and comparing with new entries, or at least notify user to check the existence of the reference before entering a new one …"
4. "… Are there any control mechanisms of the system that could warn the complier that something is wrong with the values (e.g. sum of mineral does not exceed the value for ash)? …"

5. "The Quality Panel is a great idea and works well. It helped me to complete my tasks when I was still learning the interface and was a bit confused. …"
6. " … The missing field were clearly marked - except once it was in an other tab - a tricky thing to solve. Mostly gives good feedback,"
7. "… Highlighting the mandatory fields in red helps not to omit mandatory fields…"
8. "… Sometimes I would need more feedback from the system …"

Although not all feedback items contain a clear rating if the participant like or dislike our implementation approach of the data quality prevention and data quality evaluation concepts, it can be seen that some participants were aware of the data quality panel and it helped some users to complete their tasks. We categorised the feedback in the following 4 groups:

Type of feedback	**Number of feedback items**
Positive feedback (e.g. 5 and 6)	6
Request for additional validation on data quality panel (e.g. 1 and 3)	4
The data quality is just mentioned in the feedback item (e.g. 2)	4

Table 7-6: Classification of the RODQ framework feedback items and the number of feedback item that belong to each group.

7.2.4 Discussion

Along with the general positive feedback of the testing, several comments indicated that some features were still missing and necessary to make the system complete. Nevertheless, 68% of the testers rated their impression of FoodCASE after performing the two tasks as good and further 8% as excellent. Most of the testers also thought that the system in its present state is rather intuitive, though several comments indicate that training and/or a user guide would be helpful.

The first remarkable point to discuss is that 84% of the testers thought that FoodCASE in its pilot version already contained most or all features they needed, and 76% of the testers stated it was better or much better than their system. The ratio increased to 84% of the testers feeling that FoodCASE in its first final version 1.0 would be better or much better than their current system. Also other usability questions from other sections showed similar good results and not a single question evaluated to a negative or rather negative result. Therefore, the most negative point in this usability study was the absence of a manual or other instruction.

Another point to discuss is the feedback on the data quality framework. At least 6 of 25 participants (24%) got aware of the RODQ framework functionality and made a positive feedback on it. If we add the requests for additional validation features because they agreed on the concept but were missing further validation rules, we get 40% of the participants to rate our

implementation of the data quality prevention and data quality evaluation concept as usable. It is also obvious that the 2 participants mentioning the RODQ framework in their feedback got somehow aware of the framework but unfortunately did not make a rating comment. A fact in this usability testing is that no participant made a negative comment on the functionalities of the RODQ framework in FoodCASE.

A further point to discuss is the different feedback of participants concerning the two tasks. The second task is a much more complicated process where more than one GUI is involved. This fact can be seen in the time that participants needed to complete the tasks. But more process steps in general do not justify that the task is rated to be more difficult to complete. We also assume that the self-estimations about participants' food compilation skills are not fully responsible for the shift in the results. There are some feedback items about problems finding the correct ingredients for recipes and there was also the bug to select ingredients. We therefore assume that the simplest solution is the correct one and that the GUIs for recipe calculation have a bigger potential to be improved than the GUI for the first task.

A point that is less obvious is the difference between whether FoodCASE is intuitive and self-explanatory. While after the second tasks 16 participants (64%) rated FoodCASE to be intuitive or rather intuitive, only 36% rated FoodCASE to be self-explanatory or rather self-explanatory. It is not completely clear to us why this difference exists. It could be that we asked if the basic navigation and structure is intuitive and that is only a part of the whole software, it could be that participants were influenced by the fact that they know what the software is able to do but did not know how the software does it, it could be that users understood something different than we wanted to know or a combination of these three reasons. But what most probably can be excluded, is that the results came out by accident because the two questions were close to each other in the questionnaire.

A final point to mention is the result of the self-estimated food compilation skills. We asked mainly national food compilers inside and outside Europe to participate in this survey. But only 40% of the participants estimated themselves as power users or professionals. The reason is not a general understatement of the participants, which is shown by the results of the forerunning question about participants' computer skills. We assume that the reason for the rather low food compilation skill estimations must be searched in the different levels of food compilation knowledge of the EuroFIR partners.

7.3 Usability Testing on School Administration Software Evento

After the usability testing on FoodCASE, the plan was to conduct a usability study on the RODQ framework concept of data quality analysis, for which users are preferred who are not only experts in their context but also have working experience in data deficiency corrections. The idea is that these persons have to find and correct data deficiencies as part of their daily business and, hence, that these persons are able to rate if our data quality analysis concept is

applicable and if it is value-added for their data quality improvement tasks. This study was not possible using FoodCASE because version 1.0 was not finished in that time and the evaluation of data quality requirements was in a state where such a study would be difficult. A good environment was found at the Zurich University of Applied Sciences (ZHAW) in their school administration software Evento. In Evento, the ZHAW managed all their students, teachers, courses, course registrations and school certificates. This data is changing frequently and so data quality corrections are common tasks. Because the ZHAW planned a project to make these tasks more efficient, we joint the project to elaborate a concept for our data quality analysis concept and to conduct a user study on the implemented concepts.

7.3.1 Introduction

The „Business Applications & Project Management“(BA&PM) team at ZHAW is responsible for the support of in-house managed software applications and for projects concerning these applications. In autumn 2007, the ZHAW was formed out of the four universities of applied science. Since this integration, the BA&PM team has encountered an increasing number of problems concerning the quality of data in the school administration software Evento. In addition, the former universities were all using Evento but the usage of the software and the processes in school administration were not harmonized and therefore were partly equal or similar but also partly different. Facing this situation, the BA&PM decided to start a quality assurance pilot project to find an applicable concept to circumvent data quality problems and implement this concept.

The goal of the project was to develop a concept for how the quality of data could be monitored and controlled continuously, for instance, every semester. The concept should pay attention to the not-harmonized usage of Evento and processes in school administration. It should also take into account that data quality improvement is an on-going process and therefore additions to the quality assurance framework must be possible. The pilot implementation should support the employees working with Evento in detecting data quality problems and support the users to solve the problems.

Because of the broad range to which quality assurance could have been applied, the first task in the pilot project was to find a major data quality problem which could be focused on in the pilot project. It was decided to focus on course registrations because they play a central role in school administration. Missing or incorrect registrations cause problems when marks should be entered, when transcript of records or certificates will be disseminated and when financial compensation from the home cantons of students must be calculated. Therefore, the data quality tool should be able to find missing or incorrect course registrations.

At Swiss Universities of Applied Sciences, a construct of module and sub-module is often used. The module can be seen as an abstract course group that a student can visit. The module consists of one or more sub-modules which are real courses. For instance, a module informatics

can consist of the three sub-modules Informatics lecture, informatics exercise course and informatics laboratory. The grouping of these three sub-modules means that they are associated and rules, including the three sub-modules, can be defined in what case an informatics module is passed. For example, the module informatics is passed if all sub-modules are visited (attestation) and the mark is equal or greater than 4. The students therefore need a course registration for every sub-module as well as the module.

We participated in this pilot project to find data quality deficiencies in module and sub-module registrations. Our tasks were to make a concept for data quality assurance functionalities in Evento and to help BA&PM with the implementation of the concept. The concept should contain an analysis of the current situation, find appropriate data quality requirements and to propose a feasible way to implement the concept of data quality analysis and data quality evaluation from our RODQ framework. From the research perspective, we wanted to achieve the following two goals: First, the implemented data quality analysis and data quality evaluation concept should be usability tested and, second, we wanted to investigate missing values in more detail. Missing values are a common data quality problem that can have different sources ranging from user input mistakes over incomplete automatic interfaces to organisational and management problems. We wanted to investigate missing values to get knowledge about characteristics and how they can be analysed.

All in all, we wanted to find answers to the following questions:

- Is it possible to detect missing value automatically?
- How well is our RODQ model and framework suited to a non-scientific database?

In the conception part of the pilot project, we analysed how registrations come into Evento and found out that different ways exist. Most of the data are entered by hand in other applications (satellite applications) and then imported into Evento. We also realised that an overview (bird's eye view) of data quality was not rated as important whereas finding problems (worm's eye view) was the focus for BA&PM. Therefore, we addressed only the later.

Four data quality requirements were identified for course registrations. We proposed to only implement three of them in the pilot project because the fourth was estimated to have a high implementation effort which would not fit into the project time constraints. As Evento is closed source software, it was not possible to implement a data quality analysis panel on existing GUIs. But Evento offers the possibility to define a report and associate it with a database query and an EXCEL file. It is then possible to call this report from Evento GUIs, so that Evento executes the database query and opens the result in an EXCEL file. The offered solution is so flexible that user-defined database queries could be used and the design of report can be done with Microsoft EXCEL. The solution using EXCEL can be seen as a simplified data quality analysis panel. The reports listed students for whom missing course registration were assumed. In Figure

7.10, a GUI from Evento is presented from where such a data quality analysis report can be started. Figure 7.11 shows the report with assumed data quality deficiencies.

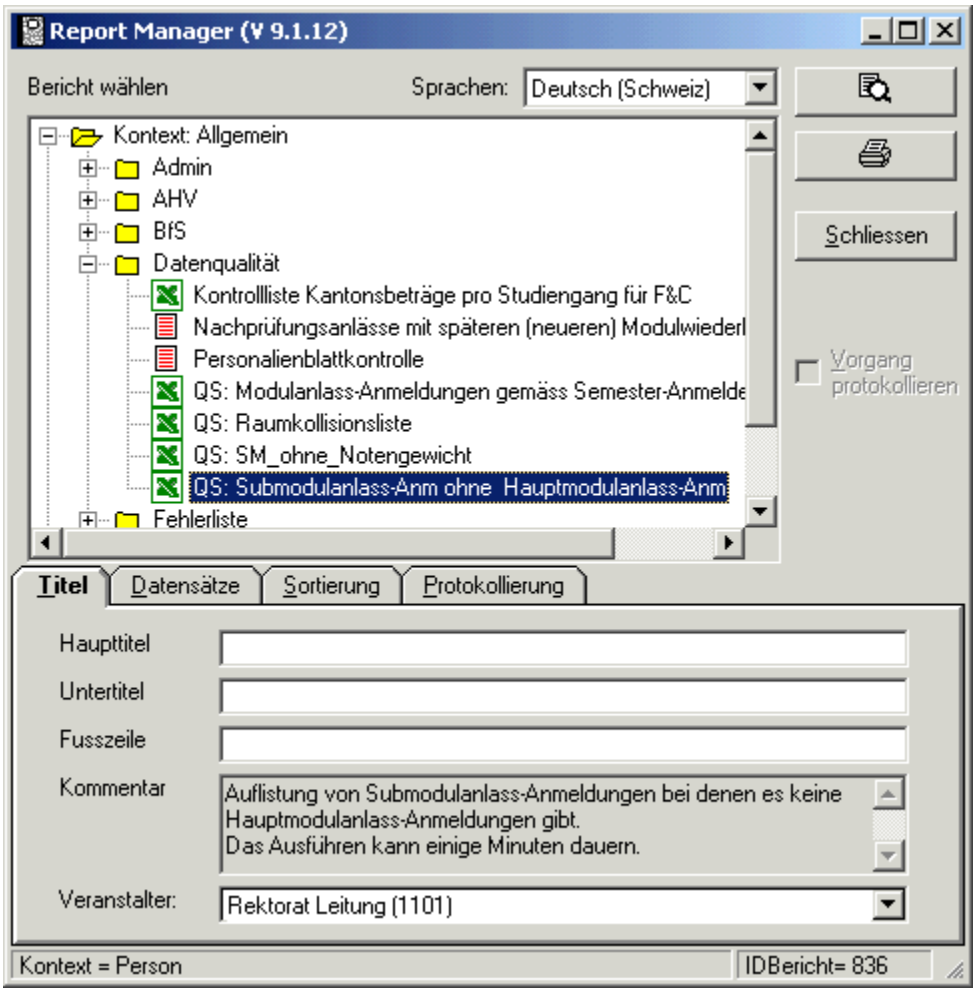
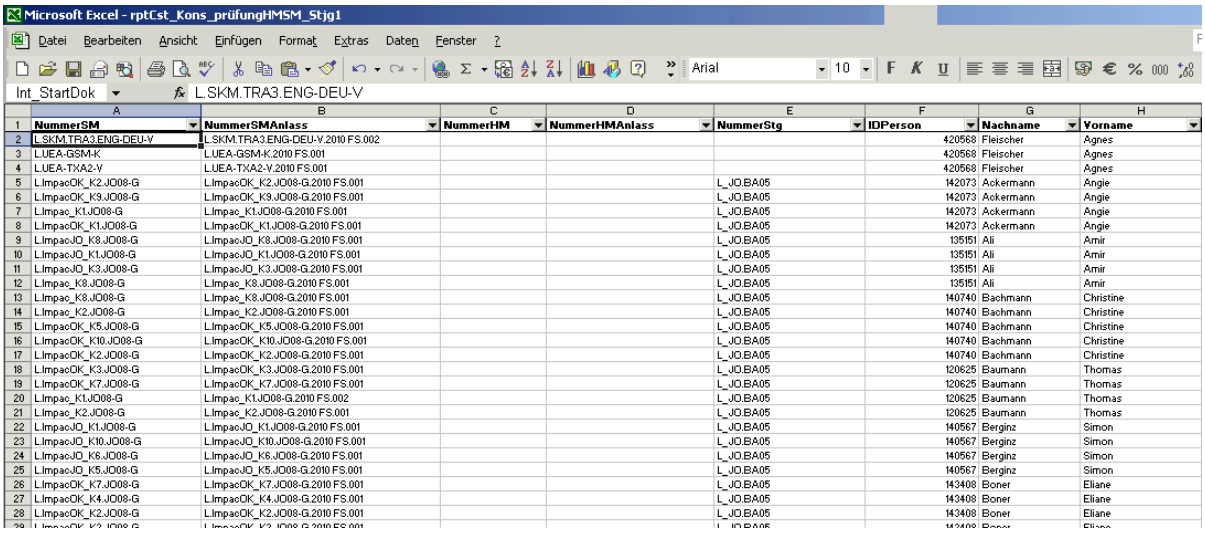


Figure 7.10: Screenshot of the Evento GUI where user-defined reports can be called.



Microsoft Excel - rptCst_Kons_prüfungHMSM_Stjg1

Datei Bearbeiten Ansicht Einfügen Format Extras Daten Fenster ?

Arial 10

Int_StartDok L.SKM.TRA3.ENG-DEU-V

	A	B	C	D	E	F	G	H
1	NummerSM	NummerSMAnlass	NummerHM	NummerHMAnlass	NummerStg	IDPerson	Nachname	Vorname
2	L.SKM.TRA3.ENG-DEU-V	L.SKM.TRA3.ENG-DEU-V.2010 FS.002				420568	Fleischer	Agnes
3	L.UEA-GSM-K	L.UEA-GSM-K.2010 FS.001				420568	Fleischer	Agnes
4	L.UEA-TXA2-V	L.UEA-TXA2-V.2010 FS.001				420568	Fleischer	Agnes
5	L.ImpacOK_K2.JO08-G	L.ImpacOK_K2.JO08-G.2010 FS.001			L_JO.BA05	142073	Ackermann	Angie
6	L.ImpacOK_K9.JO08-G	L.ImpacOK_K9.JO08-G.2010 FS.001			L_JO.BA05	142073	Ackermann	Angie
7	L.Impac_K1.JO08-G	L.Impac_K1.JO08-G.2010 FS.001			L_JO.BA05	142073	Ackermann	Angie
8	L.ImpacOK_K1.JO08-G	L.ImpacOK_K1.JO08-G.2010 FS.001			L_JO.BA05	142073	Ackermann	Angie
9	L.ImpacJO_K8.JO08-G	L.ImpacJO_K8.JO08-G.2010 FS.001			L_JO.BA05	135151	Ali	Amir
10	L.ImpacJO_K1.JO08-G	L.ImpacJO_K1.JO08-G.2010 FS.001			L_JO.BA05	135151	Ali	Amir
11	L.ImpacJO_K3.JO08-G	L.ImpacJO_K3.JO08-G.2010 FS.001			L_JO.BA05	135151	Ali	Amir
12	L.Impac_K8.JO08-G	L.Impac_K8.JO08-G.2010 FS.001			L_JO.BA05	135151	Ali	Amir
13	L.Impac_K8.JO08-G	L.Impac_K8.JO08-G.2010 FS.001			L_JO.BA05	140740	Bachmann	Christine
14	L.Impac_K2.JO08-G	L.Impac_K2.JO08-G.2010 FS.001			L_JO.BA05	140740	Bachmann	Christine
15	L.ImpacOK_K5.JO08-G	L.ImpacOK_K5.JO08-G.2010 FS.001			L_JO.BA05	140740	Bachmann	Christine
16	L.ImpacOK_K10.JO08-G	L.ImpacOK_K10.JO08-G.2010 FS.001			L_JO.BA05	140740	Bachmann	Christine
17	L.ImpacOK_K2.JO08-G	L.ImpacOK_K2.JO08-G.2010 FS.001			L_JO.BA05	140740	Bachmann	Christine
18	L.ImpacOK_K3.JO08-G	L.ImpacOK_K3.JO08-G.2010 FS.001			L_JO.BA05	120625	Baumann	Thomas
19	L.ImpacOK_K7.JO08-G	L.ImpacOK_K7.JO08-G.2010 FS.001			L_JO.BA05	120625	Baumann	Thomas
20	L.Impac_K1.JO08-G	L.Impac_K1.JO08-G.2010 FS.002			L_JO.BA05	120625	Baumann	Thomas
21	L.Impac_K2.JO08-G	L.Impac_K2.JO08-G.2010 FS.001			L_JO.BA05	120625	Baumann	Thomas
22	L.ImpacJO_K1.JO08-G	L.ImpacJO_K1.JO08-G.2010 FS.001			L_JO.BA05	140567	Berginz	Simon
23	L.ImpacJO_K10.JO08-G	L.ImpacJO_K10.JO08-G.2010 FS.001			L_JO.BA05	140567	Berginz	Simon
24	L.ImpacJO_K6.JO08-G	L.ImpacJO_K6.JO08-G.2010 FS.001			L_JO.BA05	140567	Berginz	Simon
25	L.ImpacJO_K5.JO08-G	L.ImpacJO_K5.JO08-G.2010 FS.001			L_JO.BA05	140567	Berginz	Simon
26	L.ImpacOK_K7.JO08-G	L.ImpacOK_K7.JO08-G.2010 FS.001			L_JO.BA05	143408	Boner	Eliane
27	L.ImpacOK_K4.JO08-G	L.ImpacOK_K4.JO08-G.2010 FS.001			L_JO.BA05	143408	Boner	Eliane
28	L.ImpacOK_K2.JO08-G	L.ImpacOK_K2.JO08-G.2010 FS.001			L_JO.BA05	143408	Boner	Eliane

Figure 7.11: Screenshot of the EXCEL report where registrations on sub-modules exist but not registrations on modules.

We conducted a usability study of the resulting system, which is described in the following sections.

7.3.2 Approach

The approach used for this usability study relied on the experiences gained in the usability study on FoodCASE version 0.8 [76], the usability study on the Online-Version of the Swiss Food Composition Database [78], the documented experiences of the usability study on EuroFIR's eSearch prototype [77] and considering the recommendations laid down by Hornbæk [79].

For usability testing, we used only two of the Evento reports because the third worked similarly and could be omitted. We asked the participants to perform two tasks in Evento to test the usability and to check the results of the new functionalities. The two tasks were:

1. Finding missing course registrations in your department for modules and verify the missing registration for at least one person.
2. Finding missing course registrations for sub-modules in your department and verify the missing registration for at least one person.

We prepared a user manual on how to call and use the data quality analysis reports in Evento and gave a copy to each participant. Additionally, the participants were invited to answer a paper questionnaire. The questionnaire consisted of 27 questions, mostly multiple choice questions and some open questions.

7.3.3 Structure of the Questionnaire

For most of the multiple choice answers, we used a 5-points Likert scale. In addition, we also used Yes-No questions and open questions, where participants could freely state their problems, impressions and suggestions.

The questionnaire is structured into four sections. The first section contains general user information. The second and third sections contain the two tasks to analyse quality deficiencies on course registrations and the fourth section asks questions about the impression of the new data quality functionalities.

The questionnaire was made in German and therefore the following English formulations of the questionnaire are translations.

General user information

We asked participants to indicate how many times a week they use Evento. We gave 5 possible answers where only three of them are labelled as indicated in Table 7-7.

How many times per week do you use Evento?					
Answer options	Never		2-3 times a week		5 times a week and more
Answer number	1	2	3	4	5

Table 7-7: User profile question with possible answers and answer number in the Likert scale.

Task 1: Find a missing module registration

We then asked participants to perform the following task:

> Find a missing module registration in your program of study using the provided instructions and check whether the registration is missing. If the report shows no missing registration in your program of study, please take another program of study.

After the task, we were interested in 5 questions that are also posed at the end of task 2. These 5 task questions are listed in Table 7-8 with their possible answers that the questionnaire offered to the participants and the answer number that is used for the graphical presentation on the X-axis in Figure 7.13.

Were you able to successfully complete the task?					
Answer options	No	Yes			
Answer number	1	2			

Did you find the data quality report at the place where you expected the report to be?					
Answer options	No				Yes
Answer number	1	2	3	4	5

Does the report contain enough information?					
Answer options	No				Yes
Answer number	1	2	3	4	5

How simple/difficult was it to complete the task?					
Answer options	Simple				Difficult
Answer number	1	2	3	4	5

Do you think that the time you needed to complete the task is reasonable?					
Answer options	No		50-50		Yes
Answer number	1	2	3	4	5

Table 7-8: List of 5 task questions that are investigated on both tasks to compare the answers.

In addition, we ask participants what problems they encountered and if there are any changes or improvements they would like to suggest as two open questions.

Task 2: Find missing sub-module registration

As the second task, we asked participants to do the following:

> Find a missing sub-module registration in your program of study using the provided instructions and check whether the registration is missing. If the report shows no missing registration in your program of study, please take another program of study.

Beside the same 5 task questions, we also asked the same open questions in the section for task 2.

Post task impression

After performing the two tasks, we asked the following question to mainly evaluate the concepts of the RODQ framework. The questions are again listed together with their multiple choice answers and the answer numbers.

Considering the two tasks performed, what do you think about the understandability of the new data quality functionalities? The data quality functionalities are …					
Answer options	Self-explanatory				In need of explanations
Answer number	1	2	3	4	5

What is you general impression of the report that should help to assure data quality?					
Answer options	--	-	-/+	+	++
Answer number	1	2	3	4	5

Do you think that the data quality functionalities will help you to find wrong or missing registrations more efficient?					
Answer options	No		50-50		Yes
Answer number	1	2	3	4	5

Table 7-9: List of 3 selected questions that are investigated after the two tasks are performed.

7.3.4 Results

In total, 5 persons involved in school administration participated in the usability testing. We were able to invite 3 secretaries from different departments of the ZHAW and to invite 2 persons working in the BA&PM, where they support secretaries in their daily work with Evento. Unfortunately some persons did not provide answers to all questions.

In the following we present the results of this usability testing in terms of selected questions and provide the full list with questions and answers in Appendix D.

General user information

The following figure presents the results of the first two questions as pie charts:

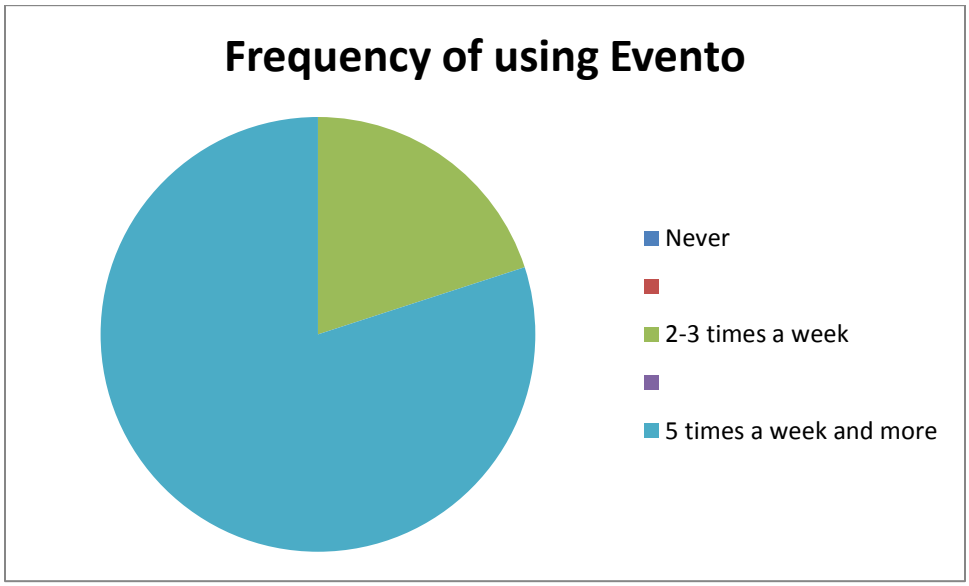


Figure 7.12: User feedback on how often they use Evento in their daily business. Only 3 of the 5 possible answers were labelled as presented in the chart.

As planned for this usability study, we asked frequent users.

Task 1 and Task 2: Find missing course registrations

All users were able to complete task 1. Four of five users were also able to complete task 2. One user could not finish the task because the report produced wrong information for her department caused by an implementation bug.

The results for the other four selected questions are presented in Figure 7.13 and Figure 7.14, where the 5 users are projected on the X-axis and the answers on the Y-axis.

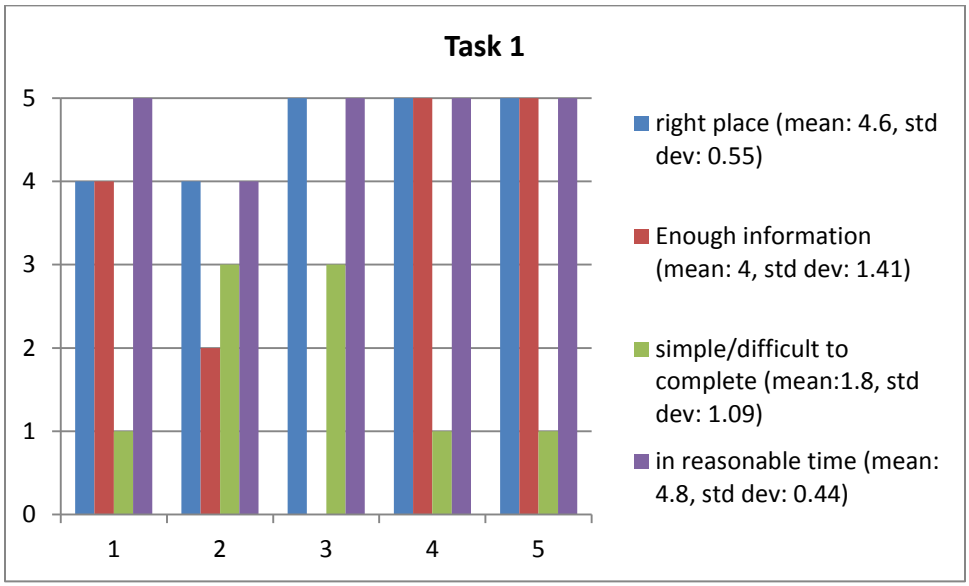


Figure 7.13: Participants feedback on four questions about task 1.

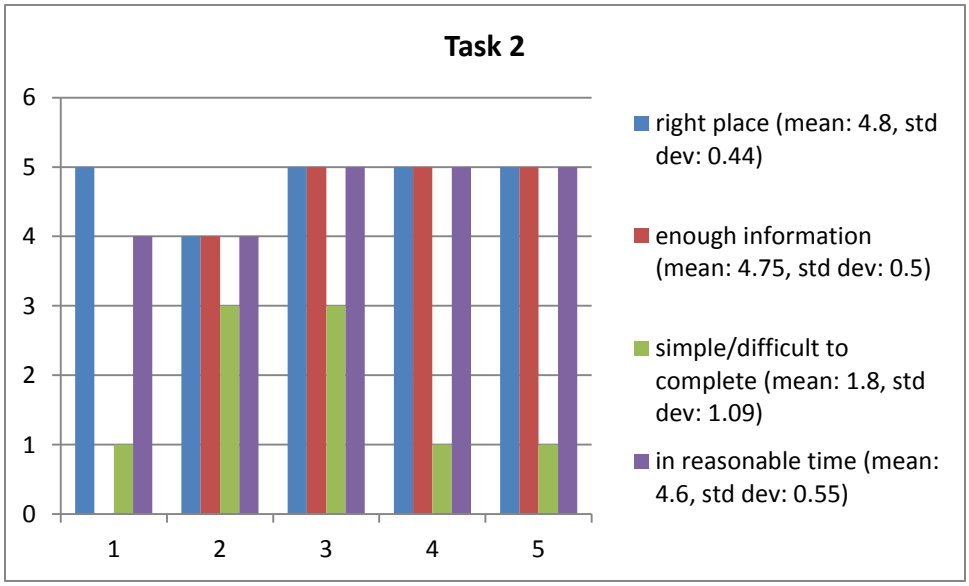


Figure 7.14: Participants feedback on four questions about task 2.

The question whether the report contains enough information, is the one having most missing answers. The reason is that participants, who are missing some information, did not answer the

question but made some notes in an open question what information they would like to have. All of these notes contain additional columns for the EXCEL report.
The majority of participants rated the tasks to be simple to complete and the time needed to perform the task as reasonable.

Post task impression

The answers for the three questions are graphically presented in Figure 7.15, where again the 5 users are listed on the X-axis and their answers on the Y-axis.

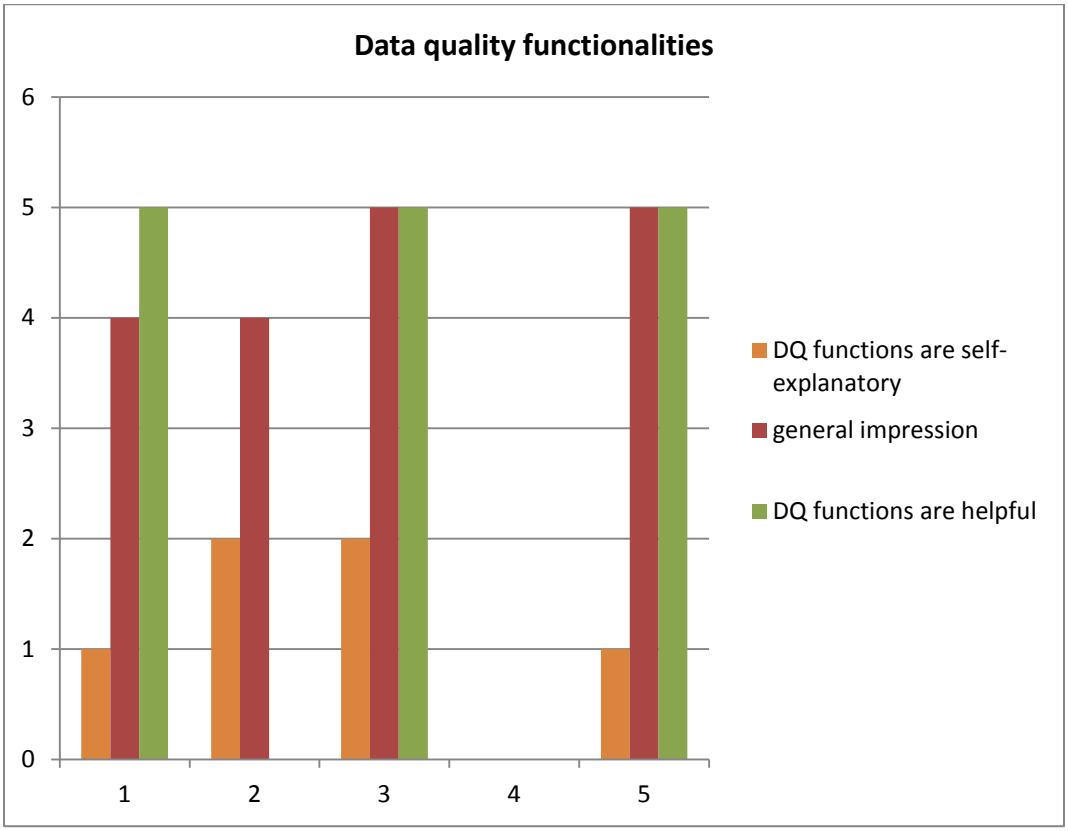


Figure 7.15: Participants feedback on the implementation of some data quality concepts of the RODQ framework.

The ease of understanding of the quality reports is rated as almost self-explaining. The general impression of the quality reports is rated as good or very good and 60% of participants rated the quality functions as helpful. Person 2 made a comment on this question that the reports are very helpful for persons having an adequate education/training.

7.3.5 Discussion

Unfortunately the number of persons is small to be representative. It was not possible to invite more persons in the ZHAW with corresponding knowledge as well as experience in school administration and Evento.

In contrast to the usability study within FoodCASE, the advantage is that these people have experience in data quality deficiency solving and know exactly how Evento works. The advantage is that these persons could concentrate on the new functionality and did not have the understandability issues we observed in the FoodCASE usability study. It was therefore no challenge for them to verify or to decline the results listed in the reports and to state if the data quality analysis panel is useful for their work or not.

One of the four identified requirements was the number of 30 ECTS credits that a student should collect, on average, during a semester. We had a situation where a student had course registrations for only 10 ECTS credits and hence missing registrations were assumed. It turned out that the student reduced his study for one semester for certain reasons. For the other three data quality requirements we found similar problems so that all four requirements must be classified as data quality indicators. The presence of three or four indicators and no hard or soft constraints leads to the situation that the data quality analysis panel can only automatically propose an answer to the requirement questions. But as the data quality analysis does not know any special agreements between students and the department, the answers must be validated by humans. Based on this situation, it becomes obvious that an automated scoring of data quality should not be applied in such situations.

In spite of the automated data deficiency identification, the data analysis was time consuming because identified data quality problems must be checked for every student. The root cause of the problem was that different satellite applications were used to generate course registration data and that this data was imported. The way to solve that problem is to strengthen data quality prevention. The satellite applications must implement data quality prevention functionality to produce data with higher quality. In addition, data quality analysis must be improved by providing more information about special agreements between students and the department and about study specialities in every semester. This information is also necessary for data quality prevention.

7.4 Summary

In this chapter, we have presented investigations to evaluate our proposed RODQ model and framework. We first verified why a flexible customisation of data quality requirements and the RODQ schemas is important. We used the history of the Swiss food composition database system where requirements as well as the RODQ schema for the reliability of scientific food composition data have changed over time.

We have also presented a usability study of FoodCASE as we had the possibility in the EuroFIR project to test the software with persons working in the field of food composition. We could show that FoodCASE in its pre-release version works as users expected, which is a first small step to assure data quality. We have also shown that our implementation of the data quality prevention and data quality evaluation concepts are rated as useful by some users.

Finally, a field study has been presented, in which the concepts of data quality analysis and data quality evaluation, in particular missing values were investigated in the school administration software Evento at the ZHAW in Winterthur. In the field study, a simplified version of a data quality analysis panel was implemented and a usability study was performed. The usability study has shown that the implementations of our RODQ framework concepts are rated to be helpful and that the participants would like to use the new data quality functionalities in Evento. In addition, a finding has been presented that missing values, depending on the context, can only be analysed using data quality indicators, which means that the automated answering in the data quality evaluation step must be verified by users. A complete automated answering and scoring would not be applicable in such a situation.

8

Conclusions

In the background and related work chapter, we have seen that some concepts already exist for categorising and measuring data quality but they are insufficient as the basis for the implementation of a data quality framework in an information system. We therefore investigated concepts and ideas for how data quality can be managed. In the previous chapters, we presented a definition for data quality, the RODQ model and the RODQ framework. We provide in Section 8.1, an overview of all concepts, their linkage and provide a critical analysis of the achieved work. In Section 8.2, we discuss open issues and directions for future works.

8.1 Summary and Discussion

Food composition is an empirical science and the documentation of nutrient values, such as information about the sampling plan or measurement method, is therefore crucial for researchers. In the context of food composition, we investigated what data quality exactly means, how it can be measured and what concepts a general implementation framework should contain.

We started our investigation in Chapter 3 with a software requirement specification of FoodCASE to understand business concepts of food compilation. In the review of existing definitions of data quality, we found out that they all point in the same direction but differ in two main focuses, which are the appropriateness of intended purpose and dependence on users' expectations and requirements. We therefore investigated what food compilers mean when rating their data to be of high or low quality with the help of the specification. We defined a reliability classification of empirical scientific data in which requirements about the sampling,

the analysis, the data acquisition and the data processing are included. We presented definitions for the terms data quality and data quality requirement. Based on these definitions, we put data quality requirements centre stage for all following concepts. In contrast to other definitions, the emphasis of our definitions is on the set of requirements with different scopes, in which aggregation functions are applied to determine data quality. Retrospectively, it must be admitted that this emphasis influenced our definitions and our concepts but we argue that a specific purpose is defined by requirements which is a subset of all requirements. In two user studies that we conducted, we found that not only user groups have different sets of requirements and different importance ratings of single requirements but also users within the same user group. Therefore, it must be assumed that data quality even without specific purpose of data usage is an individual aspect. On the other side, we have seen in the EuroFIR project that a common accepted set of data quality requirements for the reliability of nutrient values can be defined. As these two examples are basically different, the main contribution in our definition of data quality is that we put the focus on requirements and their evaluation to make data quality measurable and leave the rather abstract area of user intensions and user expectations. But it is not excluded that individual definition of data quality exists as well as commonly agreed definitions.

Based on these definitions, we presented in Chapter 4 a classification of data quality requirements according to characteristics that influence their implementation in an information system. In the field of data quality, there are other classifications from which we presented a selection in the background chapter. As these classifications are based on different concepts, we would not rate our classification to be better. We could observe in the area of food composition that there are also different classifications for food items, which are used in parallel. Hence, we see our classification to be complementary to existing classifications and having a strong focus on the implementation in information systems.

Our classification was then used in the definition of the RODQ model and the RODQ schema. The RODQ model and the RODQ schema are aimed at helping information system architects, programmers and users to specify data quality requirements, to sum them up to total values of data quality and to specify and implement appropriate data quality functionality in each data life cycle step. The RODQ model can be used in three different ways: First, architects and programmers can use an RODQ schema to specify what validators must be implemented on GUIs and in an analysis schema. They can implement the requirements in the application code but then users are not able to customise them. Second, the system offers a framework where RODQ schemas can be defined by users which the system uses to validate data. Third, a combination of the first and second ways can be used. For this reason, it seems unclear whether the RODQ model and schema is intended for architects and programmers or for users. The objective of the RODQ model and schema is to simplify the modelling of data quality requirements and their assessments. If all three groups are able to use the model, the objective is achieved and we have a model that can be used and understood by everyone. We see this as

an advantage when a data quality framework must be implemented. On the other hand, the RODQ framework claims that RODQ schemas should be customisable without re-compilation of the software because the changing environment and improvement of measurement techniques, for instance, make a modification of the RODQ schema from time to time necessary. This claim does not conflict with the three different ways of using an RODQ model above because it proposes to use the second way in contexts where frequent requirement changes occur and because it is a more flexible way. In the implementation of FoodCASE, we have seen that this way is the most time intensive and error-prone and we see it therefore reasonable when the other ways are chosen.

We also presented a basic assessment schema for data quality requirements by introducing quality matrices. The concept is quite simple and applicable in that an attribute value matrix is applied to summarise requirement questions/statements and their ratings. The advantage of a matrix form is that every combination of attribute values on the x-axis and the y-axis is regarded and missing combinations are easily detectable by empty matrix cells.

To simplify the question or statement of a data quality requirement, we also introduced an attribute algebra. For food composition requirements that have been collected so far, the presented attribute algebra was sufficient. But as requirements can have different occurrences and are dependent on the application context, we have to assume that set of operators in our algebra is not complete for every requirement and context.

In Chapter 5, we presented the RODQ framework which is a conceptual framework for the implementation of a data quality framework in an information system. To the RODQ framework belongs the concept of data quality prevention defining the data quality prevention panel, which helps users to prevent data being of low quality. The advantage of the data quality prevention panel is that they are central places of data quality issues. It is therefore possible to provide information about a GUI component that is currently not visible to a user as well as to provide information about soft constraints, indicators and requirements for which user answers are necessary. A prevention panel is therefore a flexible form to include all kinds of data quality requirements. An additional advantage is that some users are motivated to increase data quality as they can see the rating of the data that they are currently entering.

The data quality analysis panel is similar to the data quality prevention panel in that the quality of a data record is presented to the user. The difference is that the analysis panel works on existing data in the database and in addition is able to analyse sets of data records. The idea behind this concept is also to present quality information to users while they are working with the system. As a consequence, not only the data maintainer can use the quality information but also the data user. For the analysis of datasets, the analysis panel has to offer at least two stages of analysis. The top most level is where users can get an overview of data's quality in that, for instance, complete RODQ schemas are presented. The analysis panel can offer intermediates level over which a user can zoom in to specific requirements and finally on the bottom most level a user can see single data records that contain data deficiencies. In the case

of scientific data, a long-term analysis of data as well as their quality is an additional concept that an analysis panel should offer.

A data quality analysis panel can also be an effective help to argue pro or contra a dataset to groups of persons such as policy makers, decision makers, funding maker or to the public. On the other hand, there can also be a negative effect. In our implementation, grouping criteria can be used to compare datasets with each other. One of these groupings is according to usernames in FoodCASE and hence it is able to compare the produced data quality of different users. The idea behind this concept was to use the information to see if there are educational needs for users. But we realised that this information could also be used against data maintainers in that an employer could use the data to determine employment, salary or to setup a competition between maintainers.

We have also presented that data quality prevention panels and analysis panels must be present on every data manipulation interface. These interfaces are not only GUIs but also machine-to-machine interfaces. If data quality validators must be implemented on all these data manipulation interfaces, the maintenance of the validators will be a time consuming work. The concept of the data quality controller is made to prevent that problem. The data quality controller is the single quality management place where data quality requirements and RODQ schemas are defined and from where every data manipulation interfaces gets the RODQ schema that they must evaluate.

It was already mentioned that long-term analysis of data and their data quality is a useful functionality. In FoodCASE we have seen that the number of data quality values is bigger than the number of values for which the quality values stand. The concept of quality versioning foresees that data quality measurements can be performed as a periodical task. The amount of quality values can become a problem over time and the database owner has to find an appropriate solution. It is therefore advisable to reduce the amount of information that is stored for quality versioning.

All these concepts were implemented in FoodCASE and in Chapter 6 we have presented the architecture and some implementation details of FoodCASE. In a study, presented in Section 3.2.3, we asked how users agree that the representation of data has an influence on data quality. 31.6% of users answered completely, 21.1% answered mostly, 21.1% answered partly and the rest answered few or even no. That means that even before data quality requirements are evaluated GUI design can influence data quality in that data is not properly presented or data can be incorrectly interpreted. The general design and, in particular the GUI design, is a data quality point that must be considered. We therefore made a usability study with FoodCASE, presented in Chapter 7, where we let users perform two food compilation tasks without giving them a manual or any other instructions. The tasks are mostly correct completed (88% task 1 and 72% task 2) but users gave feedback that a manual is desired. The usability study showed us that the general design and the GUI design can be improved at some places but data deficiencies caused by the presentation of data are reduced.

FoodCASE was also used to implement the concepts presented in this thesis and we also presented implementation variants for our concepts. One difficulty during the implementation of FoodCASE was to find the minimum set of hard constraints. If mandatory fields exist and users feel restricted in their input attitude, the system could be rated as being restrictive. Where possible we tried to find a solution for that problem. For instance, when a new single nutrient value is created, first a kind of wizard is opened where the user has to select two attribute values, which are hard constraints. After this step, the single nutrient value GUI is opened and the two hard constraints must not be evaluated any more. Hence, we could reduce the number of hard constraints on the single nutrient input GUI. We also decided to not overfill the quality panel in that only data problems are presented. For the other data quality requirements we do not provide information as long as they are satisfied.

An idea that we had during the implementation of the RODQ framework was to find automated data correction routines that can be started after deficient data is identified. We talked to different food compilers and got the feedbacks that this kind of "black box" behaviour is undesired as the system is going to change data and users have not the possibility to intervene once the process has started. The decision on how and when data is changed must remain under the control of users because some food compilers do not trust an automated algorithm to correctly handle all problems in such as complex environment like food composition. In consequence, we define the policy in the RODQ framework of FoodCASE that the system will not autonomously correct data deficiencies.

As already mentioned, we performed two usability studies, presented in Chapter 7, on the implementations of our RODQ framework and model. In addition to the implementation, we regarded this second step of "proof of concept" as crucial for our research work. The second usability study was carried out on school administration data and not within FoodCASE. In this way, we could bring evidence that our concepts do not only fit to food composition data. As the general feedback was positive, we can state in summary that the RODQ model and RODQ framework proofed to be applicable concepts for data quality assurance automation in and outside FoodCASE. The cooperation with food compilers showed that such functionality is highly recommended in the area of food composition data. But we also realised that our concepts can be extended at some places.

8.2 Outlook

In this section we are going to present at which places the concepts of the RODQ model and the RODQ schema can be improved or extended.

A contribution that we wanted to do but were not able to do, was to conduct a usability study with persons experienced in using FoodCASE. As in the field study at the ZHAW in Winterthur, we wanted to perform a user study with experienced food compilers that understand their data and requirements as well as FoodCASE. But, at the time of this thesis, there was only one person

working with FoodCASE and some food compilers did not rate themselves as experts. We therefore split the user study into two parts: One study with food compilers, Section 7.2, and one study with experts knowing their software and experienced in data deficiency corrections, Section 7.3. We regard it as optimally to combine the two studies once there are enough experienced FoodCASE food compilers.

An additional evidence or counter-evidence for our RODDQ framework and model is a comparison of the produced data quality between users that have a RODQ framework in their information system and users that do not have a RODQ framework.

Finally, an analysis of quality evolution of data since the usage of the data quality framework implemented in FoodCASE would show if data quality increases, remains constant or decreases using our concepts.

As a possible extension, we see a simplification of the RODQ schema. The assessment-schema has a requirement for space. If a solution can be found that assessment-schemas can be presented together with the RODQ schema on one screen, then the requirement that a bird's eye view should give an overview is improved. One challenge for this solution will be the attribute algebra. The attribute algebra is the approach to translate natural language into a mathematical one. On one hand the attribute algebra must be a simplification and on the other hand unambiguous and easy to understand. A solution must be found with which every possible formulation in natural language can be transformed in a more formal one, which is at the same time understandable without much effort.

During the implementation of FoodCASE, we observed some points where extension would simplify the work of architects and programmers. The first point is that GUI elements like text fields, radio buttons or combo boxes almost offer no support for a data quality framework. A text field, for instance, must be seen as dummy GUI element where a user is able to enter whatever he wants to. It can be argued that this is the main objective of text fields but users work always in a context when using an application GUI and their input belongs to that context. That means that the input is not random or whatever comes to users' mind and hence constraints can be defined. It would be helpful if GUI elements offer functionalities in a way that constraints can be easily connected to them instead of writing the full constraint code. Ideally, the constraint framework uses one definition (including soft constraints and indicators) for object as well as for GUI elements. The constraints then need also to include user feedback and visual effects such as colouring of the GUI elements.

A further implementation challenge is that the data types offered by database products and higher order languages such as Java or C#, are in first order constraints for the machine code so that it is guaranteed that operations on variables do not produce runtime exceptions. A helpful extension would be to regard data types as quality restrictions to variables in that more constraints can be specified. Consider relational database products, they have an advantage in that simple constraints can be defined and on every data manipulation operation, triggers can

be implemented with user specific code. This idea with extended functionality would be helpful in higher order languages.

Another limitation, at least for relational databases, is that the data quality controller, which in FoodCASE is implemented in Java using JBoss Application Server, is not able to send a RODQ schema to the database where it can be analysed on a data manipulation interface or for stored data. For this reason the database must be encapsulated if the RODQ framework wants to quality control all data going into the database. A concept and an implementation for this problem would be another useful extension.

Appendix A: History of the Swiss Food Composition Database

In this appendix the history of Swiss food composition database is listed in chronological order. In Section 0 notable precursors are enumerated, which have influenced the history of Swiss food composition database. In Section 0, all historical events are mentioned that are linked directly with the database system or the used data. Specially mentioned are concepts that were developed and incorporated into the database systems.

Mentionable Ancestor

1944 During the Second World War, the first Swiss database about food composition values was created and published in table form. The editor in that time was the "Eidgenössische Kriegs-Ernährungs-Amt" [80]. 20 years later, a new version appeared in the "Schweizer Lebensmittelbuch" [81].

1984 A project was launched by the United Nations University (UNU), which bears the name INFOODS (International Network of Food Data Systems). The objectives were to promote activities in the field of quality and availability of food analysis data and to ensure that everybody can retrieve adequate and reliable data at any time. Since 1994 the project is co-financed by the FAO (Food and Agriculture Organization), which is a specialized agency of the United Nations.

1991 Martin Gander developed as part of a term paper at the Institute of Computational Science at ETH Zurich, two programs called MAKEDIET and PC DIET under the direction of Prof. W. Gander [82]. The goal was the electronic support of patients with Phenylketonuria (PKU) in

daily food consumption. PKU is a metabolic disorder in which the amino acid PHE cannot be broken down properly and can lead to severe developmental disabilities.

With MAKEDIET, medically trained personnel could create diets and determine the required nutrients that should be monitored during a diet. The diet and foods could then be transferred to PC-DIET. While MAKEDIET was intended for an IBM-PC, PC-DIET was designed for the use on a Pocket PC, in that time the ATARI Portfolio. Accordingly, the program code was programmed to save resources. The ATARI Portfolio had a 4.92 MHz Intel CPU processor, 128 KB RAM and 256 KB read-only memory. A user was able to enter food menus in PC-DIET by selecting foods with quantities from the database supplied. The program associated each menu with a date in order to make a day-analysis on all menus that could be graphically shown. The important nutrients for the diet were specially marked. In addition, other nutrients were also displayed so that the balance of the menus could be analysed.

The two programs were written in Turbo Pascal and ran under MS-DOS. Both programs used a dBase IV data base with about 940 foods having 65 nutrients each. The database came from Prof. Dr. B. François from the University of Diepenbeek in Belgium, a famous doctor in the field of metabolic diseases such as PKU.

Figure A.1: DIET-PC on an ATARI Portfolio. Kilocalories (kcal), protein content (PROT) and the phenylalanine (PHE) and tyrosine (TYR) are displayed for a menu. This information is important for PKU patients.

1992 A comprehensive book by H. Greenfield and DAT Southgate [20] was written in the project INFOODS, which is still regarded as a reference book for food composition databases. The book

explains the basics and defines guidelines for the analysis of foods, for the collection and monitoring of food composition data (also called compilation), and for the dissemination and the use of food composition data. Beside majoritarian scientifically nutrition guidelines, some concepts for the creation of electronic database systems are available. These include the four-stage model, the accuracy of values, and the input-validation principle. The four-stage model consists of four levels that are important for food composition information:

- Data source: Public and private technical literature containing analytical data, including published and unpublished papers or laboratory documents
- Archival records: Original data transposed to data record without amalgamation or modification; scrutinized for consistency
- Reference database: Data from all records for one food brought together to form the total pool of available data
- User database: Data selected or combined to give base mean values with estimates of variance for each food item

The accuracy of values defines for all components the number of significant digits.
The input-validation principle is a quality assurance process. The idea is that every entered value must be validated by a second person before the value may be used for the user database.

Developments and Concepts concerning the Swiss Food Composition Database

1992 Raphael Schenker developed the software Paracelsus 1.0 as part of his diploma thesis at the Institute of Computational Science at ETH Zurich under the supervision of Prof. Walter Gander and the direction of Prof. Hans Hinterberger [59]. The aim of the work was to design a data model for general food composition values and to implement a user-friendly interface. The dBase IV database of Prof. Dr. B. François had the disadvantage that the data was stored in a single table and had to be changed when extending the data model. In parallel to this database, two other food composition databases existed (in the Children's Hospital in St. Gallen and at the School of Nutrition in Zurich), which were used for a specific application. The newly generated data model should cover all three areas of application and should also be designed in such a general form that further applications were possible.
At the beginning of this work, Prof. Dr. B. François was contacted in Belgium as well as the Children's Hospital of Eastern Switzerland in St. Gallen and the School of Nutrition in Zurich. The goal was to learn how the food analysis data was managed, used, and what requirements were existing from user's side.

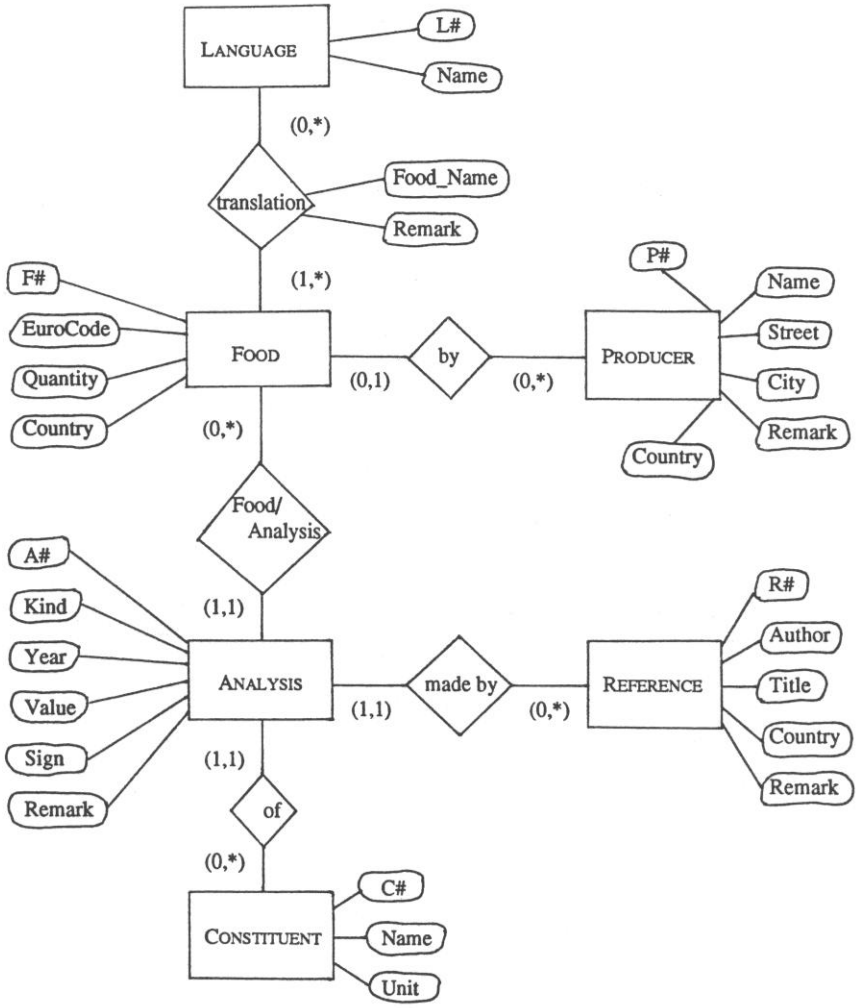


Figure A.2: data model of the Paracelsus database. The nutrient table (Constituent) was modelled expandable and food names in different languages were provided.

The new data model was programmed in dBASE IV. In a second step, user interfaces have been implemented in dBASE IV for data manipulation. The program ran on IBM-compatible PCs. For performance reasons, a 386-processor was recommended, and the program including an empty database had the size of 500 KB.

```
21.01.93                --< P a r a C e l s u s >--

Language: ENGLISH                  F O O D                  # Records:    3

                        FOOD
Name   : SPINACH                                      GOTO FIRST
Remark : RAW                                          GOTO LAST
Class  : vegetables, veg. products   EC2: 319         NEXT
                                                      PREVIOUS
                      PRODUCER                        SEARCH
Name   :                                              FILTER
Remark :            No producer.                  ------------------
Country:                                              INSERT
                                                      MODIFY
                       COUNTRY                        DELETE
Country: SWITZERLAND                          ------------------
                                                      ANALYSIS
                                                      LANGUAGE
                                              ------------------
F1 --> Help                                           PRINT
F2 --> Analysis survey                        ------------------
F3 --> Food name translations                         RETURN
```

```
21.01.93                --< P a r a C e l s u s >--

                                ANALYSIS                 # Records:    59

                        FOOD                          GOTO FIRST
 SPINACH RAW                                          GOTO LAST
                                                      NEXT
                      ANALYSIS                        PREVIOUS
Const. :  ENERGY     Qty. :    12.0 kc                SEARCH
Method : measured    Year :      87 Sign : rs      ------------------
Remark :                                              INSERT
                                                      MODIFY
                      REFERENCE                       DELETE
Author : MCCANCE AND WIDDOWSON'S                      DELETE ALL
Title  : THE COMPOSITION OF FOODS             ------------------
Country: NETHERLANDS                                  RETURN

F1 --> Help
F2 --> Analysis survey
```

Figure A.3: Schematic representation of the input screens of a food (above) and a nutrient value (below). The figures come from the user manual version 2.0 of Paracelsus.

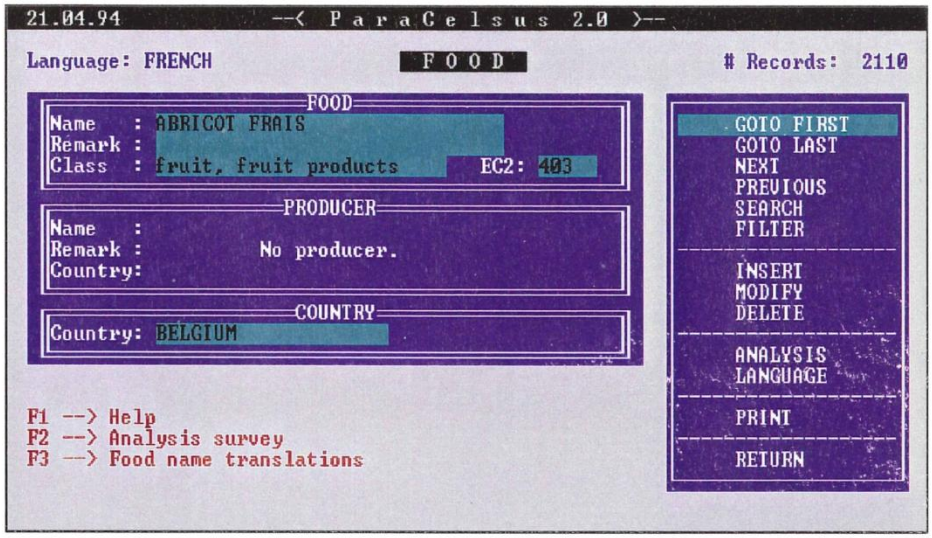


Figure A.4: Print Screen of Paracelsus version 2.0. To input data, the cursor was positioned in the right place. The user could navigate using the keyboard.

Towards the end of the diploma thesis Raphael Schenker made contact to John C. Klensin, the former director of the INFOODS secretariat at MIT, to discuss his data model.

MAKEDIET was integrated into Paracelsus after the diploma thesis, using a conversion program that prepared the data from Paracelsus to be usable for MAKEDIET. Finally, a tool for data exchange was created and various improvements and additions to the database were made. The version Paracelsus 2.0 was born.

Upon completion of the diploma thesis, Professor Walter Gander invited a selected group of nutrition experts from academia, industry and social institutions in order to present the results of this work.

This group included Professor R. Amado (Professor of Agriculture and Food Chemistry at the ETH Zurich), Mrs E. Carmenati (School of Nutrition Zurich), Dr. O. de Rham (Nestec Ltd., a subsidiary of Nestlé, with the main task of the preparation of technical knowledge and skills for the Nestlé Company), Dr. Hoffmann, Mrs Jacob S. (Institute of Food Science at ETH Zurich), Ms M. Honegger (nutritionist at Children's Hospital Zurich), Ms A. ter Velde (nutritionist at Children's Hospital St. Gallen).

1993 The "Eidgenössische Ernährungskommission (EEK)" founded a working group called "Arbeitsgruppe Nährwerttabellen" after the desire was expressed from various sides that a database should exist that is focused on Swiss food composition. In the working group were people from research, food industry, nutrition and computer science. In spring 1994 the group began the project " Aufbau einer auf die schweizerischen Verhältnisse ausgerichtete Nährwertdatenbank". The planned database should be managed centrally at the Federal Office for Public Health (BAG) and a scientific advisory board should deliver and validated data. The technical implementation should be done by the Institute of Computational Science at ETH Zurich. In a first step, the group was concerned with the gathering of data, with the correction

of data and the completion of missing data. The working group assembled in 1995 a list with for Switzerland relevant basic food. The list contained about 700 foods.

1994 Christian Vonäsch ported the dBASE database of Paracelsus to Microsoft Access during a term paper at the Institute of Computational Science at ETH Zurich under the direction of Dr. H. Hinterberger and the supervision of Prof. Dr. W. Gander [60]. The motivation was on one hand a more modern user interface and on the other hand the possibility to separate the database from the user application. At that time, it was possible to use the Microsoft ACCESS database without buying the program Microsoft ACCESS.

The data model, the table structure and data were incorporated in ACCESS. As new data quality assurance on the database level the referential integrity was added, which was offered by ACCESS. The GUI has been re-created in ACCESS, which was a great advance for the user interfaces. With buttons, text boxes, combo boxes, list boxes and forms sculptural elements were newly added.

In this work, also PC-DIET has been ported to ACCESS and so Paracelsus version 3.0 was created.

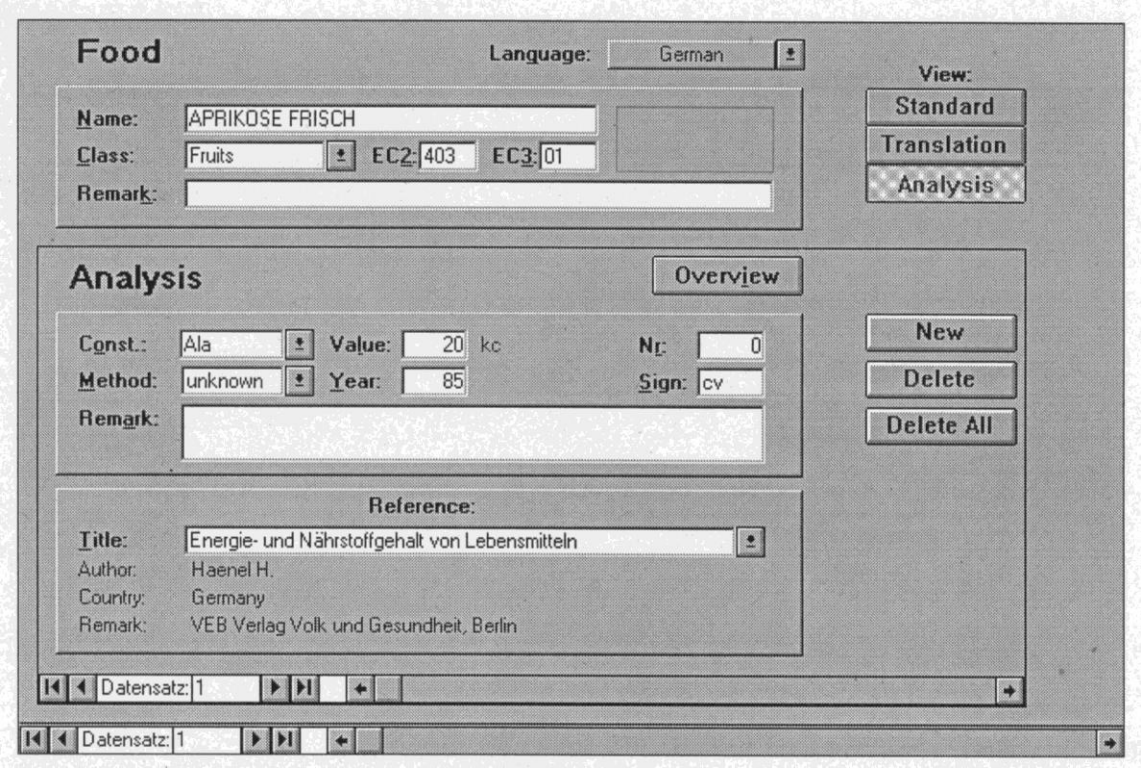


Figure A.5: Print Screen of Paracelsus version 3.0. The GUIs were implemented as ACCESS forms. With the computer mouse one could click on buttons and select input fields.

1994 Florian Schlotke developed as part of his diploma thesis at the Institute of Computational Science at ETH Zurich, the DIAKON program under the supervision of Prof. Dr. W. Gander and Prof. Dr. H. Hinterberger [83]. The work was implemented in cooperation with Prof. Dr. A. Teuscher of the "Stiftung Ernährung und Diabetes in Bern". The software was the electronic aid for the "Zehn-Gramm-Wertesystems" that Prof. Dr. A. Teuscher few years earlier had invented

and published. The "Zehn-Gramm-Wertesystems", that is designed specifically for diabetics, categorized food items into value groups such as bread values, vegetable values, fruit values, milk values , protein values, and lipid values. The idea is that a diabetic knows how many values she/he may eat from each group per day. She/he is free in food choice as long as it does not exceed the daily limits. Deacon helped users to plan the daily consumption and to manage the consumption. The software offered the possibility to replace food items with others by searching food items with equivalent values from the according value group. The software offered also the possibility to manage patients, blood sugar, insulin and activity data. From these data, calculations of energy consumption and energy needs were performed. A graphical analysis of the daily consumption compared to the diet was helpful in the diet analysis. According to Schlotke, Paracelsus 3.0 and DIAKON were prototypes and the data from Paracelsus were not practicable. Therefore, data were obtained from the "Schweizer Vereinigung für Ernährung", which had a circumference of about 600 foods and ready meals and was composed from different sources. The data were available in German and French. DIAKON has been completely implemented in Microsoft ACCESS and ACCESS forms were used as GUIs. The minimum requirements for the system were: DOS 3.1 and Windows 3.0, a 386 processor, 2 MB of free hard disk space, 2 MB of memory and EGA screen (640x350).

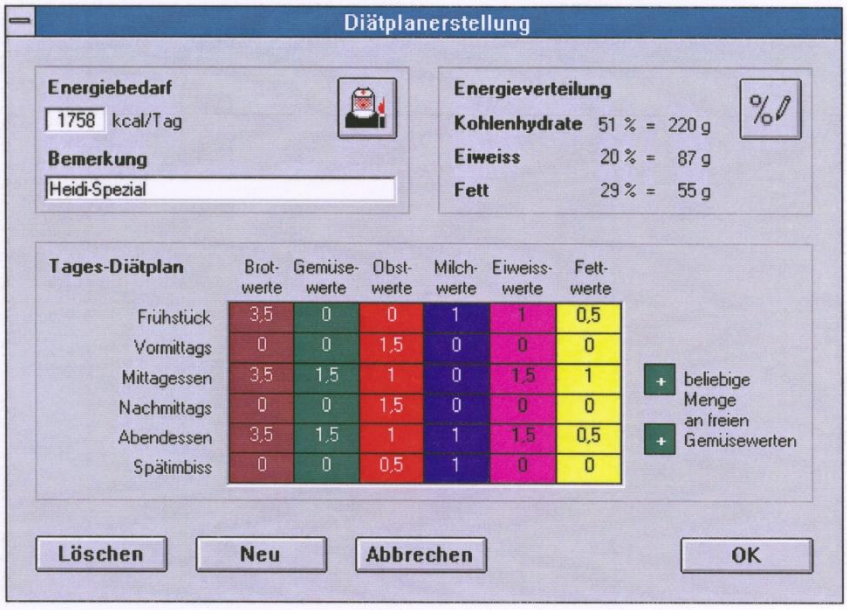
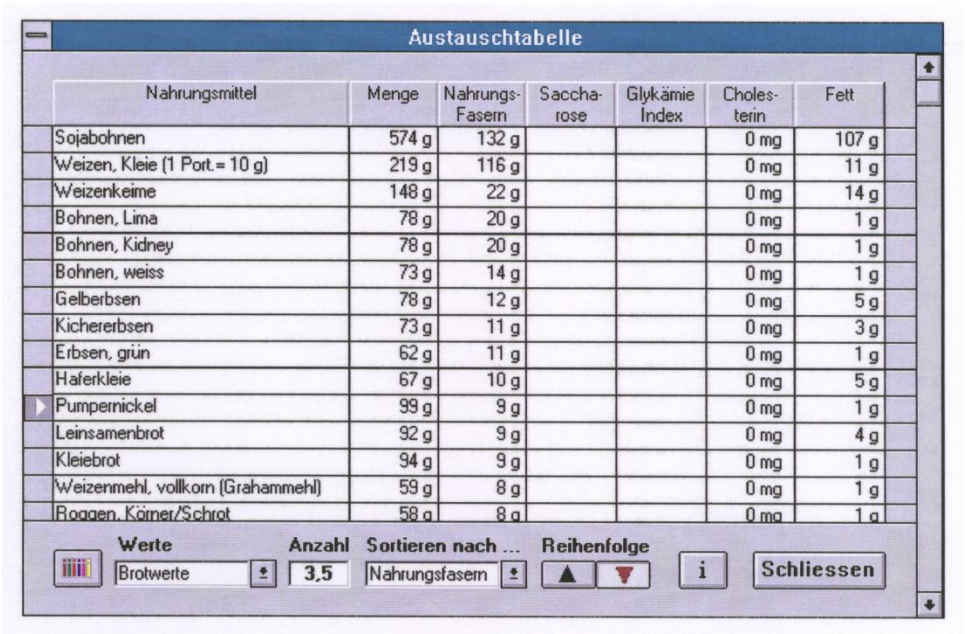


Figure A.6: Using the daily energy needs the mask "Diäterstellung" (diet creation) could make a proposal how the values for the different value groups might look like.



Austauschtabelle

Nahrungsmittel	Menge	Nahrungs-Fasern	Saccha-rose	Glykämie Index	Choles-terin	Fett
Sojabohnen	574 g	132 g			0 mg	107 g
Weizen, Kleie (1 Port.= 10 g)	219 g	116 g			0 mg	11 g
Weizenkeime	148 g	22 g			0 mg	14 g
Bohnen, Lima	78 g	20 g			0 mg	1 g
Bohnen, Kidney	78 g	20 g			0 mg	1 g
Bohnen, weiss	73 g	14 g			0 mg	1 g
Gelberbsen	78 g	12 g			0 mg	5 g
Kichererbsen	73 g	11 g			0 mg	3 g
Erbsen, grün	62 g	11 g			0 mg	1 g
Haferkleie	67 g	10 g			0 mg	5 g
Pumpernickel	99 g	9 g			0 mg	1 g
Leinsamenbrot	92 g	9 g			0 mg	4 g
Kleiebrot	94 g	9 g			0 mg	1 g
Weizenmehl, vollkorn (Grahammehl)	59 g	8 g			0 mg	1 g
Roggen, Körner/Schrot	58 g	8 g			0 mg	1 g

Werte: Brotwerte | Anzahl: 3,5 | Sortieren nach ...: Nahrungsfasern | Reihenfolge | i | Schliessen

Figure A.7: In the "Austauschtabelle" (exchange table), a user could enter group values at the bottom, for which he would have alternative food suggestions. The system listed foods and their quantities that match the search criteria as well as additional information, which are of interest to diabetics.

1994 A small project was launched called "Swiss Nutrient Database CH-NWDB" between the BAG and the Institute of Computational Science at ETH Zurich. The aim of the project was to carry out a situation analysis and to develop a concept how the Swiss food composition database could look like on the computer science site (contract number 316.94.0458 11.04. 1994). The preparatory work of the "Arbeitsgruppe Nährwerttabellen" was used as basis. The idea was to merge the various tables in Switzerland using source references and to use food classifications to highlight the data. The database should be used as an archive database, which means that end users cannot access directly. The idea was to have a reliable data collection that was completely documented. In succession, three informatics diploma theses were made, which belong into this conceptual phase.

1995 Two closely linked diploma theses of Dieter Richter and Ove Hiltwein were made at the Institute of Computational Science at ETH Zurich under the supervision of Prof. Dr. W. Gander, and under the supervision of Dr. H. Hinterberger and F. Schlotke [61, 84]. The goal was the design and implementation of an Information Server. An Information Server is an environment that will allow the user to read, copy, and explore data and information in a simple way [61]. The concept had foreseen that the information server should provide various communication interfaces for the users: phone, Internet and local network. The concept was based on the following communication interfaces:

- Over a phone system nutrition information should be provided, with the telephone keypad to be used as navigation

- Fax requests should be sent that are recognized via OCR and the answer will be returned by fax
- Via the Internet connection E-mails with requests should be answered where an E-mail must follow a certain syntax
- Through various Web services such as FTP, Telnet, and Gopher nutrition data and files should be delivered.
- An HTML website for retrieval of nutrition information

All user requests should be handled by so-called application processes in the Information Server. The application processes were familiar with the database schema, the storage strategy in the file system and data should be stored in a specified format. For each communication interface, the specified format should be transformed in the communication format of the interface.

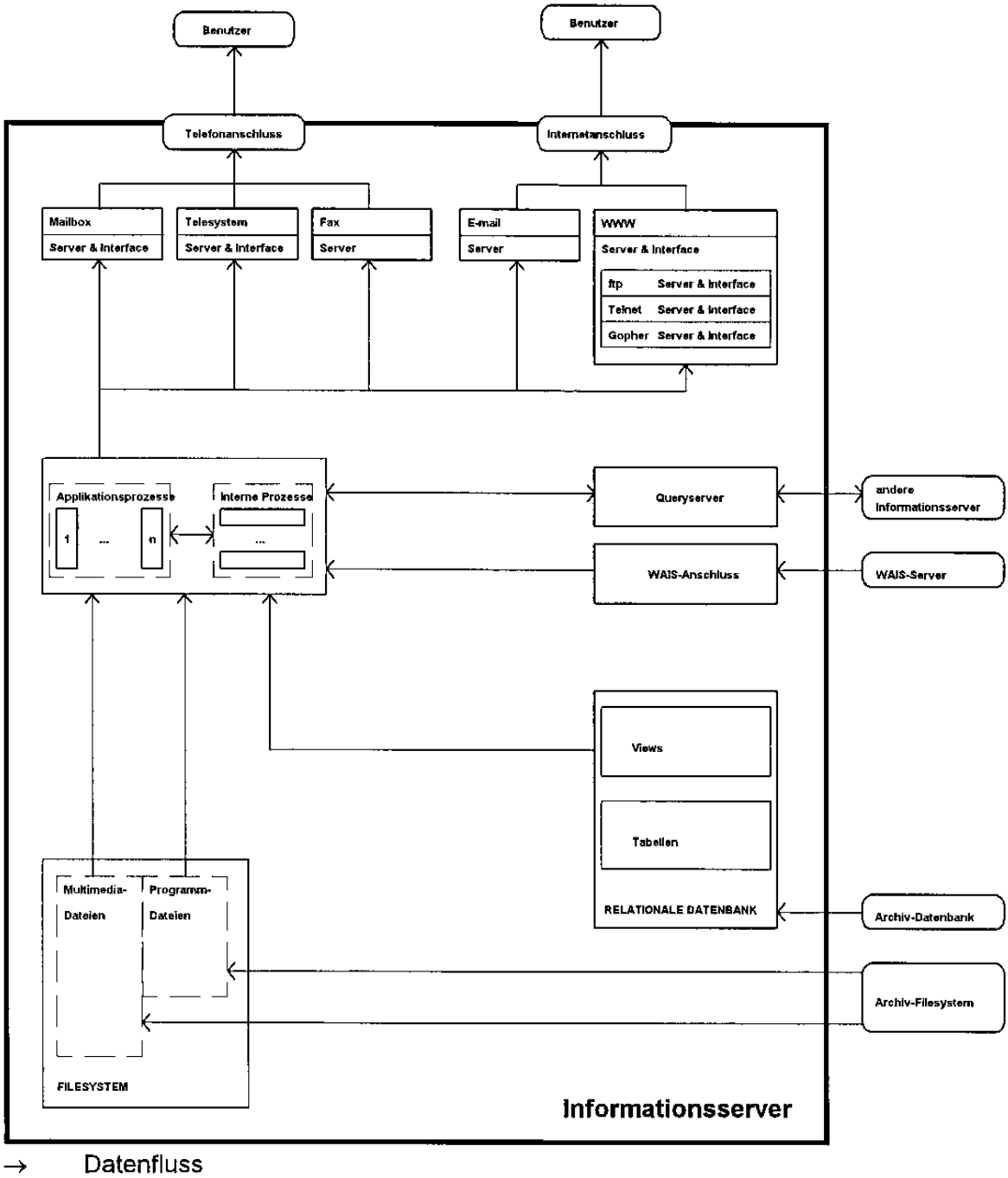


Figure A.8: Schematic representation of the concept of an information server. The modular concept pursued the purpose of generating independent components, which are extensible or replaceable. At the bottom of the figure you can find a module for persistent data storage, in the middle left are the process modules that perform the database queries, and on top are the modules that are responsible for the communication interfaces.

In this work, a database schema for food composition data in ACCESS and in Oracle has been implemented. From the above communication interfaces, the web interface using HTML and

CGI (written in C) were implemented and a FTP service was set up for retrieving images and ACCESS files from Paracelsus and DIAKON.

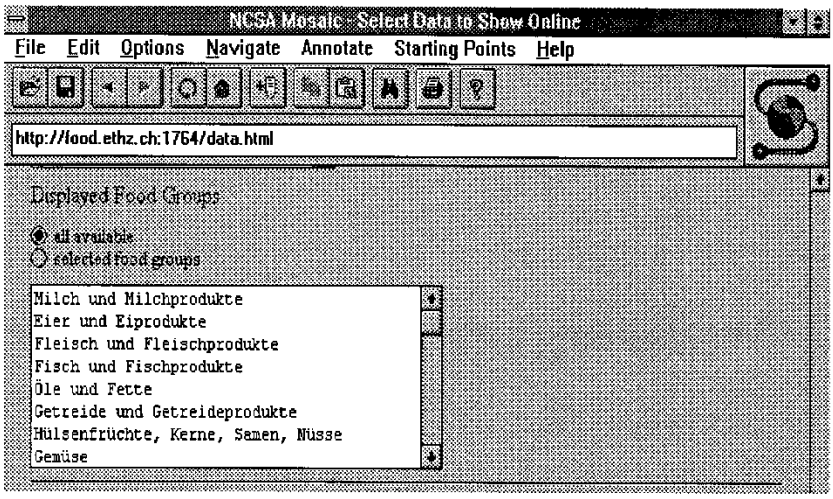


Figure A.9: Web browser client of the information server: Data were group in categories and required nutritional values were selectable by the user.

1995 Switzerland participated with Florian Schlotke as representative in the European research project COST Action 99, which lasted until 1999. COST stands for "Cooperation in Science and Technology" and is a research programme to facilitate scientific and technical co-operation at European level, complementing in particular the EU framework programmes and EUREKA. COST co-operation takes the form of concerted Actions, which involve the co-ordination of national research projects. In the field of Food Science and Technology, COST is mainly concerned with improving food safety, food quality and nutrition. COST Action 99 was specifically devoted to "Food Consumption and Composition Data" and the primary objective was to merge knowledge and expertise of experts in order to construct and establish a network of compatible food composition databases and data exchange with the high quality food consumption and food consumption data. 27 European countries participated in COST Action 99 and the project was financially supported by the European Commission, Directorate General for Research support. In this project, a working group dealt with the promotion and encourage of data exchange within Europe. Florian Schlotke et al [56] created recommendations for food composition data structure. The usage of thesauri was among these recommendations. The idea behind thesauri is that users have to make a choice from existing values and are not free to arbitrarily enter data. The publication was made in 2000.

1996 The first Swiss food composition database seminar was held on Monte Verita in Ascona. The seminar was organized by J. Lüthy (BAG), R. Hurrell (D-AGRL), H. Hinterberger (D-INFK), C. Wenk (SGE) and H. Ryser (SVE). The goals of the seminar were on one hand to inform about application areas in which food composition data were of increasing importance, and on the other hand to discuss and to find new ideas for the development and maintenance of a national food composition database that satisfies Swiss requirements.

1996 A software for the management of food classification was developed by David Schläpfer as part of his diploma thesis at the Institute of Computational Science at ETH Zurich under the responsibility of Prof. Dr. W. Gander and the care of Dr. H. Hinterberger and F. Schlotke [85]. A classification code was seen as a marker that designates a name, "Part of Plants" for instance, and was used for the classification of food. The software was the electronic help for the administration of such codes. The software was able to generate codes and display hierarchical structures in tree form. LanguaL was among many others, the most famous representative of possible applications. LanguaL is a unified language for describing foods and was defined in the COST Action 99 project [86] and later revised [87].

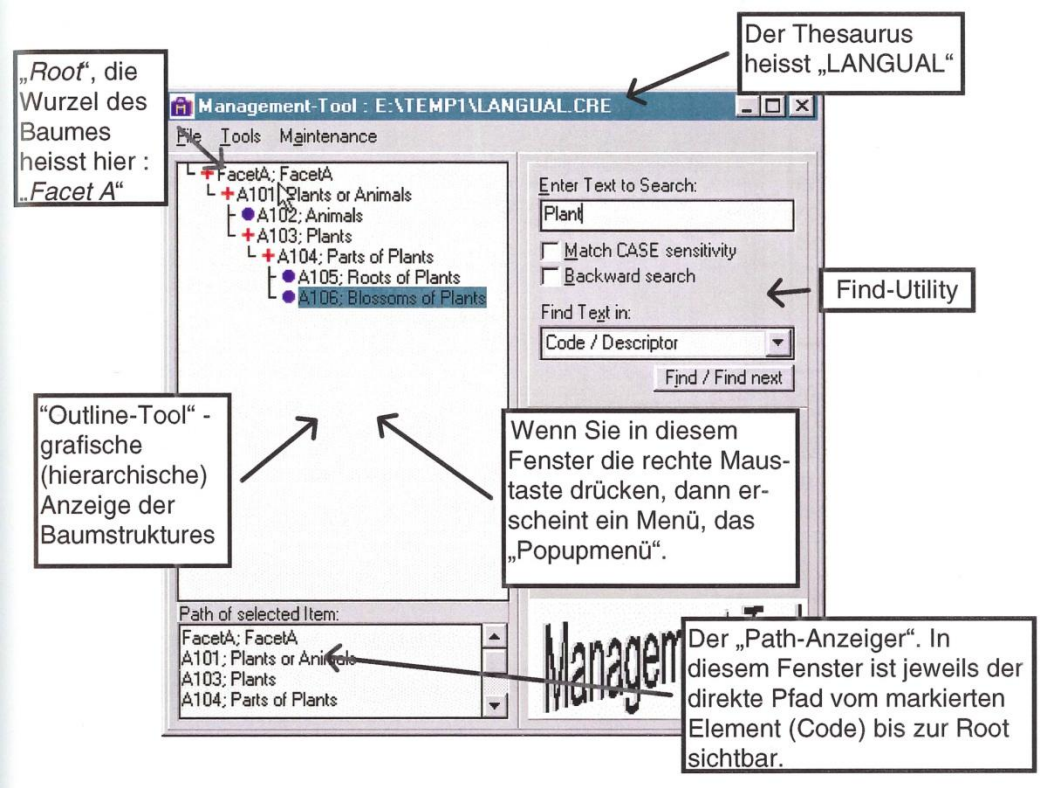


Figure A.10: Print Screen from the User Guide using LanguaL as an example. In the management tool you could define nodes and leaves and build up a hierarchical tree structure. The figure shows how a part of LanguaL was used as a test.

This software was later transferred to Denmark where the Food Product Indexer was developed based on it. The Food Product Indexer allows assigning LanguaL codes to foods.

1998 The project "Aufbau einer schweizerischen Nährwerttabelle" (Building a Swiss food composition database) started. The participants in the project were (financing means that the institution contributes with funding) [64]:

- Federal Office of Public Health (Dr. J. Lüthy), Financing
- Federal Office for Agriculture (Prof. J. Morel), Financing
- Institute of Food Science, ETH Zurich, Laboratory of Human Nutrition (Prof. Dr. R. Hurrell, Dr. S. Jacob)

- Institute of Food Science, ETH Zurich, Laboratory of Food Chemistry and Technology (Prof. Dr. R. Amadò, M. Risel, P. De Angelis, H., Kistler, L. Pellino)
- Institute of Computational Science, ETH Zurich (Dr. H. Hinterberger, F. Schlotke)
- Nestlé Suisse (Dr. B-M. Exl), financing
- F. Hoffmann La Roche (Dr. U. Moser), financing
- Migros (Dr. W. Steiner), financing
- Coop Switzerland (Dr. W. Stutz), financing

This project had the aim to gather all available food composition data in Switzerland, to classify them in terms of quality and to enter the data in a standardized way in a new user database system called iFoods. Not only data from the previously known tables were collected, but also data from a number of Swiss companies and institutions, which were previously contacted and asked for publishable data. Where no Swiss data were available, data from foreign sources were taken or, where possible, values were calculated. The user database contained only a selection of 39 nutrients, which were defined to be important.

At the beginning of the project a new data structure was created which corresponded to the recommendations of COST Action 99. The data structure has been implemented in Microsoft Access and dBASE IV and single foods could be viewed and printed in Microsoft Excel and as ASCII text. The data were edited directly in Access and dBASE IV. There was also a website on which information could be obtained on food composition data.

Schlotke developed in the course of the project the software iFoods, which was based on the recommendations of COST Action 99 and integrated some elements that were already developed at the Institute of Computational Science [62]. The goal was to create software with which the Swiss food composition data could be managed and which should offer functions and data structures for a standardized data collection and international data exchange. iFoods implemented the four-stage model based on Greenfield & Southgate [20]. Figure A.11 presents the workflow of iFoods with the four-stage model. Because of the four steps, iFoods contained several small applications, which made up the system. Excel was used as GUI for the data processing while FDX files and Access were used to store data. A scheme manager, with which one could add attributes to the table structure, was a special functionality of iFoods.

The iFoods Workflow: 10 Steps to Success

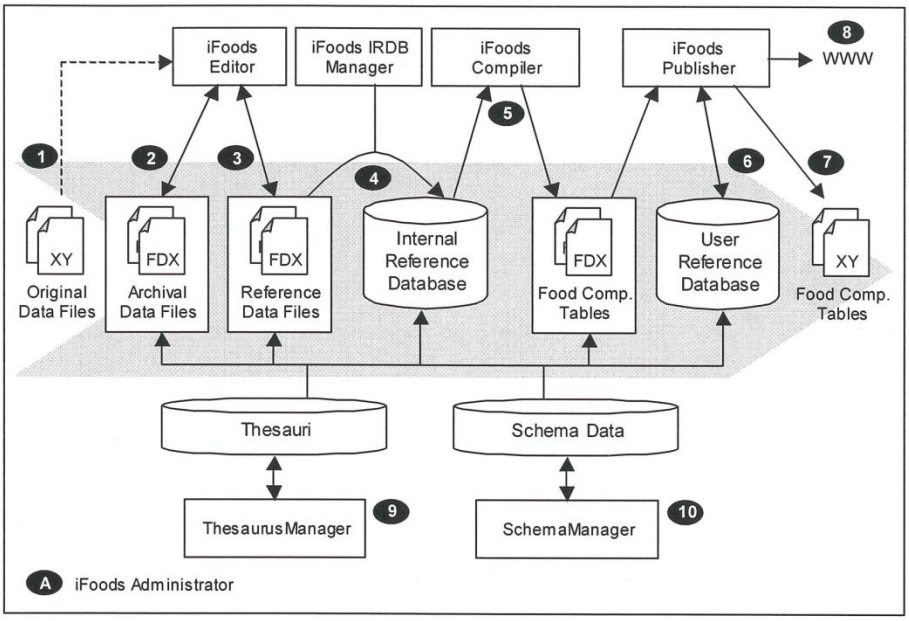


Figure 2: Workflow within the iFoods System

1. iFoods Editor assists you in editing and converting raw food composition data to standard food data x-change files (FDX files) according to the EUROFOODS recommendations for data management and interchange.. You can also edit FDX files from scratch.
2. FDX files are stored in several directories depending on the stage within the workflow. The first stage are the Archival Data Files. Archival data files hold all the data of the original files in a consistent and standardized data structure.
3. Using the iFoods Editor, archival data files can be enriched with additional data, e.g. additional food classifications or translations of food names. Units and component description are harmonized. Such files are now referred to as Reference Data Files.
4. Reference data files can be loaded into the Internal Reference Database using a loading process available within the FoodDataManager.
5. The iFoods Compiler application allows to analyze, aggregate, calculate and derive food composition data from the material available within the internal reference database. The result of this process is continuously stored in the User Reference Database.
6. Subsets of the user reference database can be published as FDX files, …
7. … as food composition tables in various text, database, or spreadsheet formats, or …
8. … directly via the internet for a selected audience.
9. Thesaurus Manager allows to create new or extend existing standardized vocabulary and hierarchical classification systems.
10. Schema Manager allows to extend the available set of properties used in food, component, data source and method description.

A. iFoods Administrator acts as a portal to the maintenance applications Internal Reference Database Manager, Thesaurus Manager, Thesaurus Loader, and Schema Manager.

Figure A.11: Process definition of the software iFoods. The four-stage model for food composition data of Greenfield and Southgate was implemented.

To collect more practical feedback, a test group was composed with representatives from nutrition advice, from a cantonal laboratory, from industry and the Swiss authorities, which tested and critically reviewed the food composition data.

At the Institute of Food Sciences, a quality evaluation system was developed, with which it was possible to determine food composition data quality by taking into account certain metadata. The so-called quality index represented the quality and was a letter between A and C, where A was the highest and C the lowest quality rating. The quality index for a nutrient value was determined by awarding points to various aspects such as representativeness, the number of samples tested, the validation of the analytical method used or the laboratory information. The sum of quality points determined the quality index. In figure x we can see these aspects, for which quality points are given. According to the sum of the quality points, one of the following three quality indices resulted:

- A = value is trustworthy (13-19 quality points)
- B = the value is determined trustworthy (6-12 quality points)
- C = the value is not very trustworthy (0-5 quality points)

Value Description	Value	Back to Table	Comments
Best Location	21.4		
Value Type	mean		
Date Of Analysis	1999-06-18		
Mean	21.4		
Median	20.3		
N	12		
Standard Deviation	1.23		
Standard Error			
Min	15.8		
Max	25.1		
Quality Index	C		
Probenzahl	keine Angaben, 1-2		
Sampling (repäsentativ?)	keine Angaben; nicht repäsentativ		
Probenbeschreibung	genaue Angaben		
Transport			
Lagerung	akzeptable Angaben, sachgemäss		
Vorbereitung	keine Angaben, unsachgemäss		
Standardmethode			
Methodendurchführung			
Wiederfindungsrate			
Bestimmungsgrenze			
Arbeitsbereich/Linearität			
Referenzmaterial			
Referenzmethode			
Interassay			
Intraassay			
Genauigkeit			
Präzision			
Labor Akkreditierung			

Figure A.12: Detail mask for meta-information about a nutrition value. The fields until Quality Index are standardized according to COST Action in 99 while the fields within the Quality Index were developed at the ETH Zurich. That is the reason for the linguistic difference.

The project did not end as planned in late 2000 but was extended for one year. The BAG, BLW and the industry companies Roche, Coop, Nestlé, Novartis Consumer Health, Biofamilia, Rivella and Migros took part in the financing.

2001 S. Jacob left the ETH Zurich and continued the project Swiss food composition database from outside. The food composition database was given to an external company. The reason was that iFoods in the stage had a prototype character and therefore the interaction of Excel and Access was not very stable running. The second reason was that the temporary appointment of F. Schlotke at ETH Zurich was expired.

2003 The Swiss food composition data were published and could be purchased as a CD-ROM or as printed table. The database contained about 700 foods. For the foods value, about 21'000 single food values have been used from 100 different sources [88].

2004 The software "Schweizer Nährwertdatenbank Online-Version" (Swiss food composition database online version) was created at the Institute of Computational Science at ETH Zurich by R. Lee under the direction of Prof. Dr. H. Hinterberger [63]. The software was a new edition of the Swiss food composition database and replaced the 2001 externally commissioned implementation. Microsoft Access was again used as database product. The software contained Access forms and a Web interface for data acquisition. The program was also able to generate PDF, XML and Excel reports for publications and to present nutrient data in graphical form.

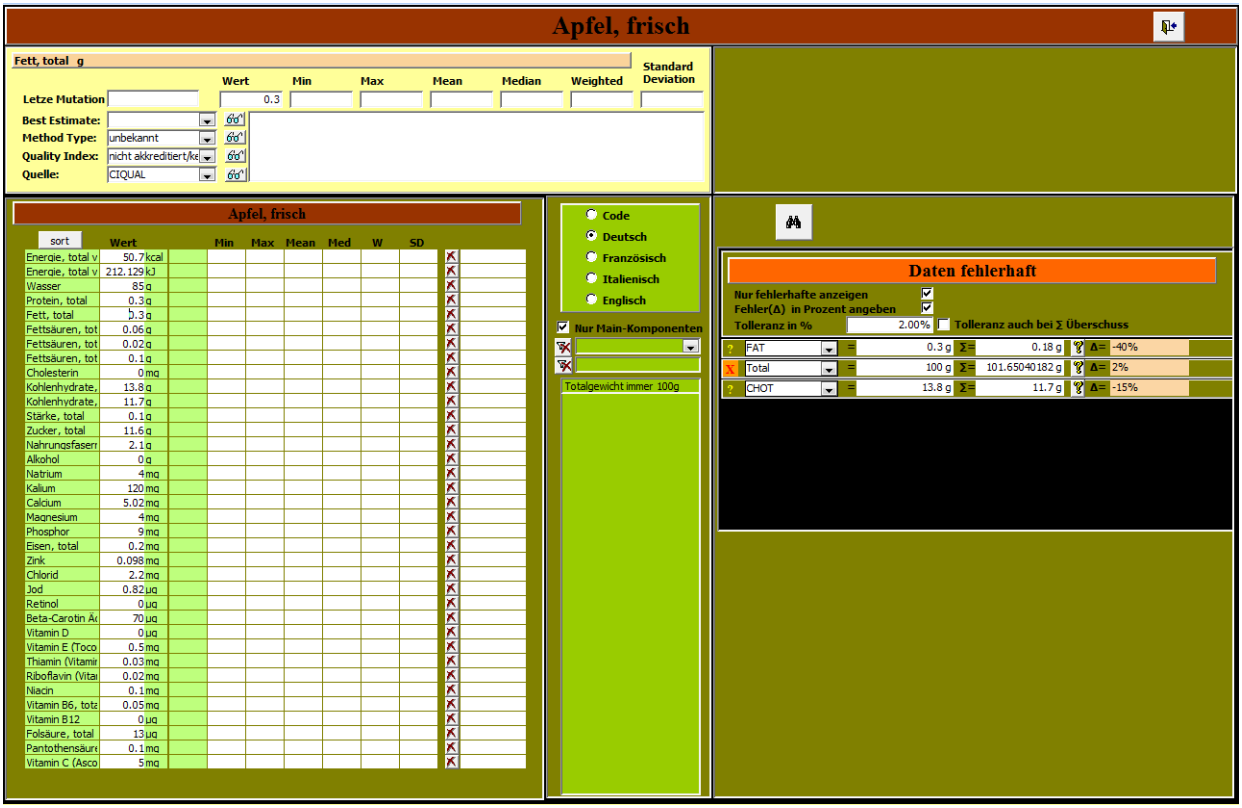


Figure A.13: Microsoft Access GUI of nutrition values for an apple. All components are listed in tabular form and statistical information as well as methods and source information can be stored. On the right side a user can see the quality analysis of a single food composition value: For values that can be calculated from other nutrition values, the system displayed a warning "Daten fehlerhaft" (data erroneous) if the entered food composition value differed more than a defined threshold from the calculated value.

Another functionality of the program was the recipe calculation. The software offered the possibility to choose different foods with quantifications, one method and for each food composition value one loss/gain factor that occurred during the cooking method. The automatic calculation then calculated the food composition values of the recipe.

2004 A European research project called EuroFIR was launched, which was financed by the European Commission within the Sixth Framework Programme, and lasted five years [58]. EuroFIR had the project type "Network of Excellence" (NoE) which is a multi-partner project aimed at strengthening scientific and technological excellence on a particular research topic by integrating at European level the critical mass of resources and expertise needed to provide European leadership and to be a world force in a given domain. The main result of an NoE should be a durable restructuring and reshaping of the way research is carried out in Europe in a given area [89]. In the project involved were 47 universities, research institutions and SMEs. EuroFIR's objectives were to link the various European food composition databases, make data comparable, which was achieved by introducing standardization and harmonization of food composition data, and to provide a single, authoritative source of food composition data in Europe. One result was the so-called eSearch facility. A website where a user is able to search and compare foods and nutrition values from all EuroFIR partners. Early 2009, the non-profit organization EuroFIR AISBL (EuroFIR Association Inernationale Sans But Lucratif) was launched to guarantee the long-term cooperation.

2006 A new project was announced by the BAG with the goal to maintain and update the Swiss food composition data. The Department of Agricultural and Food Science at the ETH Zurich received the order and could launch the project SwissFIR. ETH internally, the cooperation with the Department of Computer Science was again initiated. The task for SwissFIR was to collect new data and to enter them into the database. In order, industrial companies and research institutions have been contacted. The ETH Zurich participated in the project EuroFIR as partner in 2006. The project SwissFIR ended in early 2009.

2006 A website to access published data was created in the context of the master thesis of T. Hodapp at the Institute of Computational Science at ETH Zurich under the direction of Prof. Dr. H. Hinterberger and contextual support of Dr. P. Colombani [90]. Thus, the first official website was created on which published food composition data could be accessed for free. Data were taken from the Microsoft Access database of the "Schweizer Nährwertdatenbank Online-Version ". At that time EuroFIR had a first version of the Food Data Transport Package (FDTP) defined, an XML -based European standard format for an exchange of food composition data. A Java servlet fetched the data from the database and converted it into the FDTP format. In this way, the data was available in a machine readable form through a RESTful web service. Another component of the servlet transformed the XML into HTML and made it readable for human users.

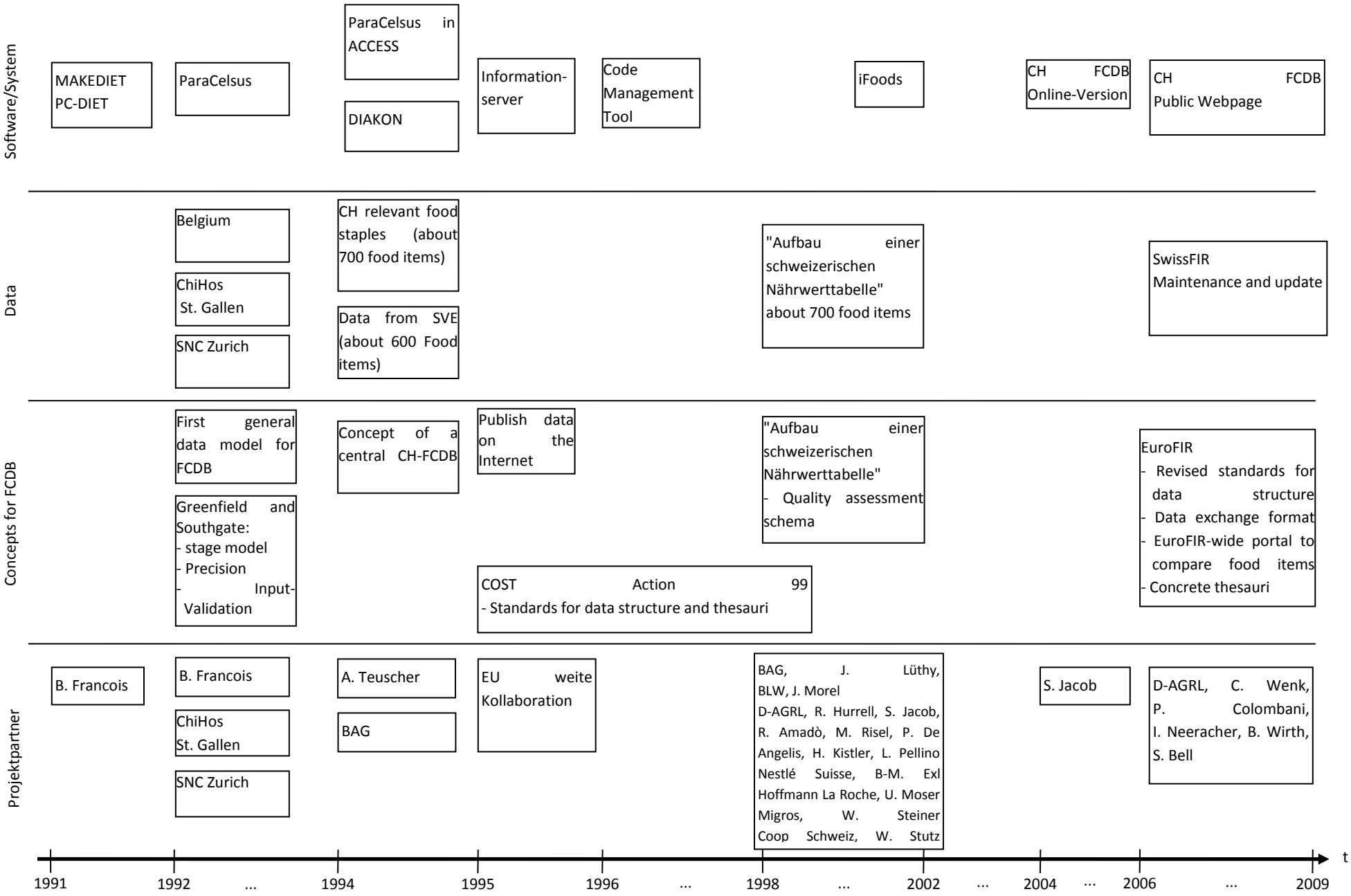
Another functionality was to generate an offline version. The servlet generated all the HTML

pages in a directory, and linked the pages together. This folder could be copied to CD-ROM or on a memory stick and were offline available for users.

2007 After the publication of the website, it was quickly realized that the whole management system of the Swiss food composition database must be renewed which ultimately led to the project FoodCASE. As part of a PhD thesis of Karl Presser at the Chair for Information Technology and Education at the ETH Zurich under the direction of Prof. Dr. H. Hinterberger and contextual support of Dr. P. Colombani, a new management software was created called FoodCASE. The software was also a research object for the modelling, evaluation and measurement of data quality in the PhD thesis. Involved in the technical development were Kajetan Abt, Sandra Kästner, Serkan Bozyigit, Raphael Tawil as well as Ines Neeracher und Birgit Wirth from the food science side.

The following diagram gives an overview of the tools, data, concepts and participants in the history of the Swiss food composition database system.

Overview


Software/System
MAKEDIET
PC-DIET
ParaCelsus
ParaCelsus in ACCESS
DIAKON
Information-server
Code Management Tool
iFoods
CH FCDB Online-Version
CH FCDB Public Webpage
Data
Belgium
ChiHos St. Gallen
SNC Zurich
CH relevant food staples (about 700 food items)
Data from SVE (about 600 Food items)
"Aufbau einer schweizerischen Nährwerttabelle" about 700 food items
SwissFIR Maintenance and update
Concepts for FCDB
First general data model for FCDB
Greenfield and Southgate:
- stage model
- Precision
- Input-Validation
Concept of a central CH-FCDB
Publish data on the Internet
COST Action 99
- Standards for data structure and thesauri
"Aufbau einer schweizerischen Nährwerttabelle"
- Quality assessment schema
EuroFIR
- Revised standards for data structure
- Data exchange format
- EuroFIR-wide portal to compare food items
- Concrete thesauri
Projektpartner
B. Francois
B. Francois
ChiHos St. Gallen
SNC Zurich
A. Teuscher
BAG
EU weite Kollaboration
BAG, J. Lüthy, BLW, J. Morel
D-AGRL, R. Hurrell, S. Jacob, R. Amadò, M. Risel, P. De Angelis, H. Kistler, L. Pellino
Nestlé Suisse, B-M. Exl
Hoffmann La Roche, U. Moser
Migros, W. Steiner
Coop Schweiz, W. Stutz
S. Jacob
D-AGRL, C. Wenk, P. Colombani, I. Neeracher, B. Wirth, S. Bell
t
1991
1992
...
1994
1995
1996
...
1998
...
2002
...
2004
...
2006
...
2009

Legend:

- CH = Swiss
- FCDB = Food Composition Database
- ChiHos = Children's Hospital
- SNC = School for Nutrition Counselling
- BAG = Federal Office for Public Health (Bundesamt für Gesundheit)
- BLW = Federal Office for Agriculture (Bundesamt für Landwirtschaft)
- D-AGRL = Department for Agricultural and Food Science

Appendix B: Data Quality Survey

At the concluding conference of EuroFIR in Vienna we made a workshop and a subsequently conducted a survey. The goal of the workshop was to clarify some aspects of food composition data that arise during the implementation of FoodCASE, because in previous discussions we heard completely different solutions to open questions. The intent of the survey was to find out what persons working in the area of food composition understand when we talk about data quality.

In the workshop 24 persons working in the area of food composition participated.

Method

We distributed a survey document with questions and possible answers. We went through every question, showing them on the beamer, explained the question to be sure that every participant was able to understand and ask the audience to discuss possible answer of every question. After the discussion the participants should answer the question on their document. The questionnaire can be found in Appendix B.

Method discussion

The open discussion before answering a question is of course a highly suggestive method. The discussions were very long because different opinions were argued. But the goal of the workshop was to find a majority solution to every question because these questions were blocking our implementation.

In the following all questions are listed with the number of answer that was given in bold. "All" means all persons participating in the workshop and in brackets is the number of food compiler, the experts in food composition. The questions are highly food composition specific but they should show the distribution of answer which is also data quality aspects.

Personal Information

Position: _____

Name (optional): _____

Single Values

---

1. Do we need to store information for more than one sample for a single value?

☐ Yes
All: 7 (Compiler: 1)
☐ No
All: 15 (Compiler: 12)

---

2. Which reference information do we want to store in FoodCASE?

☐ FoodCASE should provide only one multiline text field where we can input whatever we want
All: 1 (Compiler: 0)
☐ FoodCASE should provide fields for title, author, publication date and citation
All: 3 (Compiler: 3)
☐ FoodCASE should provide the set of fields that is used in the interchange package
All: 9 (Compiler: 6), 2 compilers proposed some additional fields
☐ FoodCASE should provide full reference managing functionality, including document storage
All: 4 (Compiler: 1)
☐ Like in technical annex
All: 1

Additional answers:
☐ Link to document/cite explorer
All: 9 (Compiler: 7)
☐ Link to reference management tool
All: 3 (Compiler: 2)
☐ Other links (whatever that means ☺)
All: 5 (Compiler: 5)
☐ Possibility to store documents
All: 1 (Compiler: 1)

---

3. How many references can a single value have?

☐ Only 1

All: 14.5 (Compiler: 9.5)

☐ 2 or more

All: 8.5 (Compiler: 3.5)

---

4. Do we need to store information for more than one method for a single value?

☐ Yes

All: 7 (Compiler: 4)

☐ No

All: 14 (Compiler: 9)

Aggregated Values

---

5. How do we round aggregated values?

General

☐ Do not round values

All: 6 (Compiler: 3)

Subquestion A (How should the number of digits be determined?)

☐ 2 significant digits

All: 1

☐ 3 significant digits

All: 1

☐ 4 significant digits

All: 1 (Compiler: 1)

☐ 5 significant digits

All: 0

☐ 6 significant digits

All: 0

☐ Relative error 1%

All: 0

☐ Relative error 2%

All: 0

☐ Relative error 3%

All: 1 (Compiler: 1)

☐ Relative error 4%

All: 0

☐ Relative error 5%

All: 0

Subquestion B (What unit should be used?)

☐ Use only target unit

All: 5 (Compiler: 1)

☐ Convert to other units
All: 3 (Compiler: 2)

Subquestion C (What value should be round to 0?)
☐ Values smaller than 0.001 -> 0
All: 0
☐ Values smaller than 0.0001 -> 0
All: 2 (Compiler: 1)
☐ Values smaller than 0.00001 -> 0
All: 0
☐ Values smaller than 0.000001 -> 0
All: 0

Other answers:
☐ Like in the book Food Composition data (Greenfield & Southgate)
All: 4 (Compiler: 4)
☐ We need official documentation from EuroFIR
All: 1
☐ Definable by users (for each component)
All: 3 (Compiler: 3)
☐ Ask your statisticians
All: 1
☐ Depends on component
All: 2 (Compiler: 1)

---

6. How should versioning work?

Subquestion A (How many versions should be possible to manage by the system?)
☐ No versioning needed
All: 1
☐ Only two versions, current and rest
All: 4 (Compiler: 3)
☐ Archive of versions
All: 12 (Compiler: 7)

Subquestion B (What data should be stored?)
☐ Store only values
All: 0
☐ Store all information
All: 11 (Compiler: 5)
☐ Store a subset of information: value, unit, method, reference and QI
All: 3 (Compiler: 3)

Subquestion C (What happens when a new version is created?)
☐ Creating a new version overtakes all data from the preceding version as a starting set
All: 13 (Compiler: 8)

☐ A new version is an empty set
All: 1

Subquestion D (What version is published on the website?)
☐ On the website publish always current version
All: 9 (Compiler: 7)
☐ In the software it should be definable which version is published
All: 6 (Compiler: 3)

Subquestion E (Are older versions editable?)
☐ Older versions are editable
All: 6 (Compiler: 4)
☐ Older versions are not editable
All: 10 (Compiler: 6)

---

7. How do we calculate the confidence code from the quality indexes?

☐ The confidence code is equal to the lowest quality index
All: 0
☐ The confidence code is the weighted average like the aggregated value
All: 4 (Compiler: 2)
☐ A new method must be found, any idea: _____
All: 1
☐ Wait until EuroFIR work is finished
All: 4 (Compiler: 1)
☐ According Cab Jacobs
All: 1
☐ See paper on EuroFIR website
All: 1
☐ Use statisticians and statistical programs
All: 2

Data Quality

---

8. In current research literature, there is a commonly accepted work that divides data quality into 4 categories and 16 dimensions. Please rate how important each quality dimension is to you using the following scale:

0	not important
1	little important

2 rather important
3 important
4 rather very important
5 very important

1. Accessibility and System

Quick and easy access	0	| 1	2	3	4	5
Access security	0	| 1	2	3	4	5
Performance of the system	0	| 1	2	3	4	5

2. Content

Believability	0	| 1	2	3	4	5
Accuracy	0	| 1	2	3	4	5
Objectivity	0	| 1	2	3	4	5
Reputation	0	| 1	2	3	4	5

3. Context

Data provide advantage for for their use	0	| 1	2	3	4	5
Relevancy	0	| 1	2	3	4	5
Timeliness	0	| 1	2	3	4	5
Completeness	0	| 1	2	3	4	5
Appropriate amount of data	0	| 1	2	3	4	5

4. Representation

Interpretability	0	| 1	2	3	4	5
Ease of understanding	0	| 1	2	3	4	5
Representational consistency	0	| 1	2	3	4	5
Compact representation but not overwhelmed	0	| 1	2	3	4	5

---

9. After having seen the 16 dimensions above, are there further attributes that come to your mind?

_____
_____
_____
_____
_____
_____

_____
_____
_____
_____

---

10. Please answer the following questions about a food composition system, e.g. FoodCASE, using the following scale:

0	not important
1	little important
2	rather important
3	important
4	rather very important
5	very important

How important is it to you …

… that the system offers functionality to analyse and improve data quality?

0 | 1 2 3 4 5

… that the system continuously gives data quality feedback and requests you to adjust data?

0 | 1 2 3 4 5

… each change of data is stored with personal information, e.g. username?

0 | 1 2 3 4 5

… that the system's functionality can be extended by users?

0 | 1 2 3 4 5

… that the system can manage user preferences and personal settings?

0 | 1 2 3 4 5

… that the system allows you to set country specific markers and groups?

0 | 1 2 3 4 5

… that the history of data entries can be recalled?

0 | 1 2 3 4 5

… that access to your data can be controlled and restricted?

0 | 1 2 3 4 5

Please answer the following additional questions.

Do you agree that system functionality has an influence on data quality?

no few partly mostly completely

Do you agree that the number of functionalities has influence on data quality?

no few partly mostly completely

---

11. Please answer the following questions about the characteristics of data using the following scale:

    0 not important
    1 little important
    2 rather important
    3 important
    4 rather very important
    5 very important

How important are the following information/aspects of data quality to you?

Sample Origin

The place where a sample was taken

0 | 1 2 3 4 5

The number of sample(s) that was taken

0 | 1 2 3 4 5

What was taken (what was the sample)

0 | 1 2 3 4 5

Date when sample(s) was taken

0 | 1 2 3 4 5

Who took the sample(s)

0 | 1 2 3 4 5

How the sample(s) was taken

0 | 1 2 3 4 5

Sample Transport

Who transported the sample(s) to the lab

0 | 1 2 3 4 5

How sample(s) was transported

0 | 1 2 3 4 5

How long took the transport

0 | 1 2 3 4 5

Laboratory

How the sample(s) was stored before analysis

0 | 1 2 3 4 5

How long the sample was stored before analysis

0 | 1 2 3 4 5

Laboratory/institution that made the analysis

0 | 1 2 3 4 5

Information about method used to analyse

0 | 1 2 3 4 5

When sample was analysed

0 | 1 2 3 4 5

Detailed description of what was analysed

0 | 1 2 3 4 5

How results were processed (e.g. average, conversion) by lab employee before publishing

0 | 1 2 3 4 5

Values in the database system, e.g. FoodCASE

Who entered data

0 | 1 2 3 4 5

The values and associated units were entered as in the original reference (without modification)

0 | 1 2 3 4 5

Date when data was entered

0 | 1 2 3 4 5

If data was validated/double checked by other person(s)

0 | 1 2 3 4 5

Data are free of contradictions

0 | 1 2 3 4 5

The system has duplication detection, e.g. information about data item that are similar or equal

0 | 1 2 3 4 5

The system has range detection, e.g. information that entered value is not in the range of other similar values

0 | 1 2 3 4 5

Aggregation and recipe calculation

Which values were processed to get the result

0 | 1 2 3 4 5

How data were aggregated and calculated

0 | 1 2 3 4 5

Person who processed data

0 | 1 2 3 4 5

When data was aggregated and calculated

0 | 1 2 3 4 5

The precision of values that is used in aggregation and calculation

0 | 1 2 3 4 5

The age of values used for aggregation and calculation

0 | 1 2 3 4 5

Are there other aspects that are important when you think about data quality?

_____

_____

_____

_____

Please answer the following additional questions.
Do you think that data which is not updated can lose quality over time?

no rather no indifferent rather yes yes

Do you think that missing component values degrade data quality?

no rather no indifferent rather yes yes

Do you think that the more fields in the database are filled the better is your data quality?

no rather no indifferent rather yes yes

Do you think that the number of accesses to your published data is an indicator for data quality?

no rather no indifferent rather yes yes

---

12. Please answer the following questions about presentation of data using the following scale:

0	not important
1	little important
2	rather important
3	important
4	rather very important
5	very important

How important is it to you that …

…operations can be carried out in only one way?

0 | 1 2 3 4 5

…that data quality can be displayed with diagrams or graphs?

0 | 1 2 3 4 5

Please answer the following additional question.
Do you agree that the representation of data has an influence on data quality?

no few partly mostly completely

---

13. Please answer the following questions about the specification of FoodCASE.

Did you understand the specification?

no rather no indifferent rather yes yes

Did the specification contain too many IT-technical details for you?

no rather no indifferent rather yes yes

Was the specification helpful to understand how the software functions?

no rather no indifferent rather yes yes

Is the current pilot of FoodCASE that what you expected from the specification?

no rather no indifferent rather yes yes

Please answer the following additional question.
Do you agree that a specification has an influence on data quality?

no few partly mostly completely

Recipe

---

14. How can a retention factor be determined?

Subquestion A (food group)

☐ Only if an ingredient has an appropriate EuroFIR food classification

☐ Only if an ingredient has an appropriate LanguaL code

☐ If an ingredient has an appropriate LanguaL code **AND** an appropriate EuroFIR food classification

☐ If an ingredient has an appropriate LanguaL code **OR** an appropriate EuroFIR food classification

Subquestion B (cooking method)

☐ FoodCASE should use only the defined cooking method in the EuroFIR report

☐ Users can add own cooking methods

Subquestion C (more than one cooking method)

☐ In FoodCASE it should not be possible to determine more than one cooking method

☐ In FoodCASE it should be possible to determine more than one cooking method

Subquestion D (calculation using multiple cooking methods)

☐ For every ingredient the smallest retention factor is taken

☐ For every ingredient every retention factor is taken

☐ Other: _____

---

15. What should FoodCASE do if yield factors are missing?

☐ Without yield factor recipe calculation is not possible
☐ The software should take 1 as default value

General

---

16. What information do we need from the codex alimentarius?

☐ No information
☐ Only the link to the homepage (http://www.codexalimentarius.net)
☐ The link to the appropriate site (e.g. http://www.codexalimentarius.net/web/more_info.jsp?id_sta=10743)

☐ Certain fields, which: _____

---

17. What information do we need from the E-Number and the INS-Code?

☐ No information
☐ E-Number: _____

☐ INS-Code: _____

---

18. What information should be deletable?

☐ Nowhere

Subquestion A	Subquestion B	Subquestion C
☐ Single value	☐ Aggregated value	☐ Recipe value
☐ Single food and value	☐ Aggregated food and value	☐ Recipe food and value

---

19. What happens with the data after the import through the web service?

☐ FoodCASE cannot import data, only export for e-Search facility
☐ Data is read-only, no storage
☐ Data can be stored with all information provided by the web service

---

20. There is a trade-off between information density and clarity on a GUI. What do you prefer?

☐ Information density
☐ Rather information density
☐ Balance of them
☐ Rather clarity
☐ Clarity

Appendix C: Usability Testing of FoodCASE

The management of food composition data is undoubtedly a complex task and the use of an electronic system can assist a compiler in several ways. Nevertheless, today no standardised food composition database management system (FCDBMS) seems to exist that is universally used.

FoodCASE

FoodCASE[1] is the FCDBMS currently under development within the doctoral thesis on quality of data on scientific database systems of Karl Presser at the Computer Science Department of the ETH Zurich. The specification of FoodCASE, based on the available standards related to food composition defined by EuroFIR, was written by Karl Presser in close collaboration with the Swiss compilers. Feedback on the specification was received – and considered whenever appropriate – from 28 compilers representing 28 institutions and 22 countries around the world.

FoodCASE is being built using Open Source software (Java, JBoss Application Server and PostgreSQL) and consists of the six modules (Figure C.1): Content management application, Database, Application server, Web page, Web service and Administrator application. On the server side, the Application server, the Database and the Web service are responsible that data are centrally stored and that the data can be retrieved from everywhere. The Content management application ("Compiler Client") and the Administrator application help the compilers to manage and to publish their data over the website. The Compiler Client is itself built in a modular way and consists of the modules Single Value, Aggregated Value, Recipe, and References (visible as separate registries, Figure C.2).

[1] FoodCASE means "Food Composition And System Environment".

FoodCASE represents a unique possibility for compilers around the world and in particular for EuroFIR Partners / EuroFIR AISBL Members as it is the intention of the ETH Zurich to make the system freely available to FCDB compiler institutions.

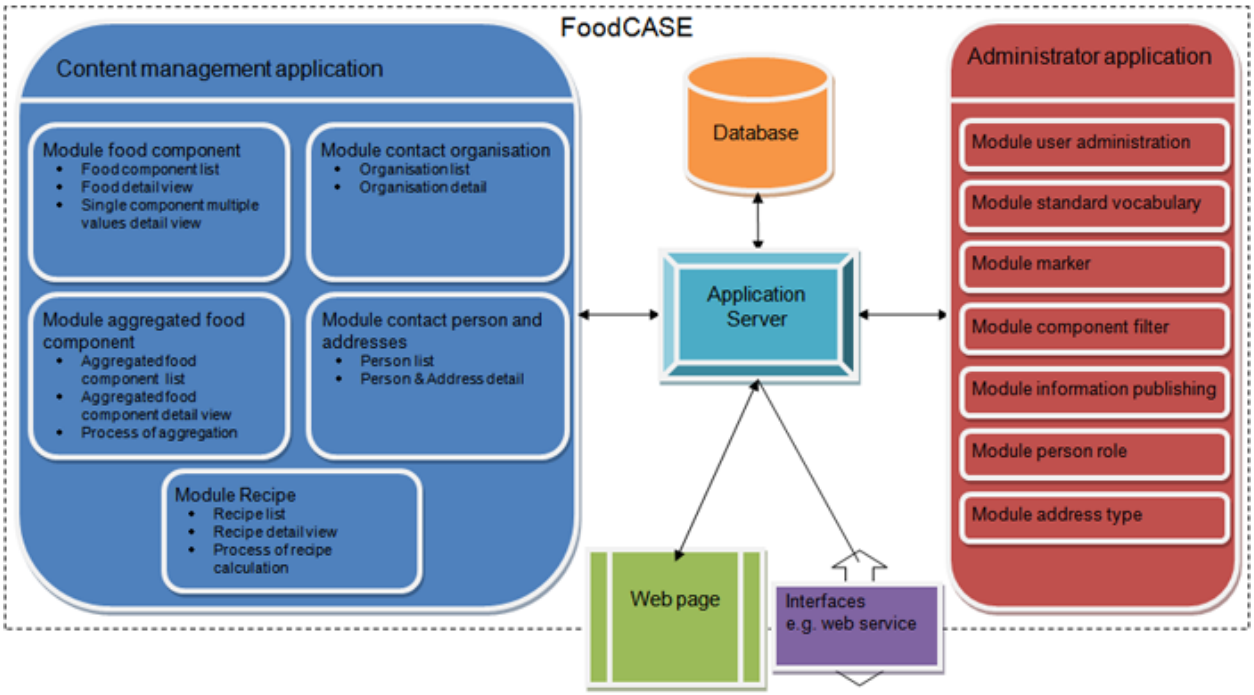


Figure C.1: Overview of FoodCASE.

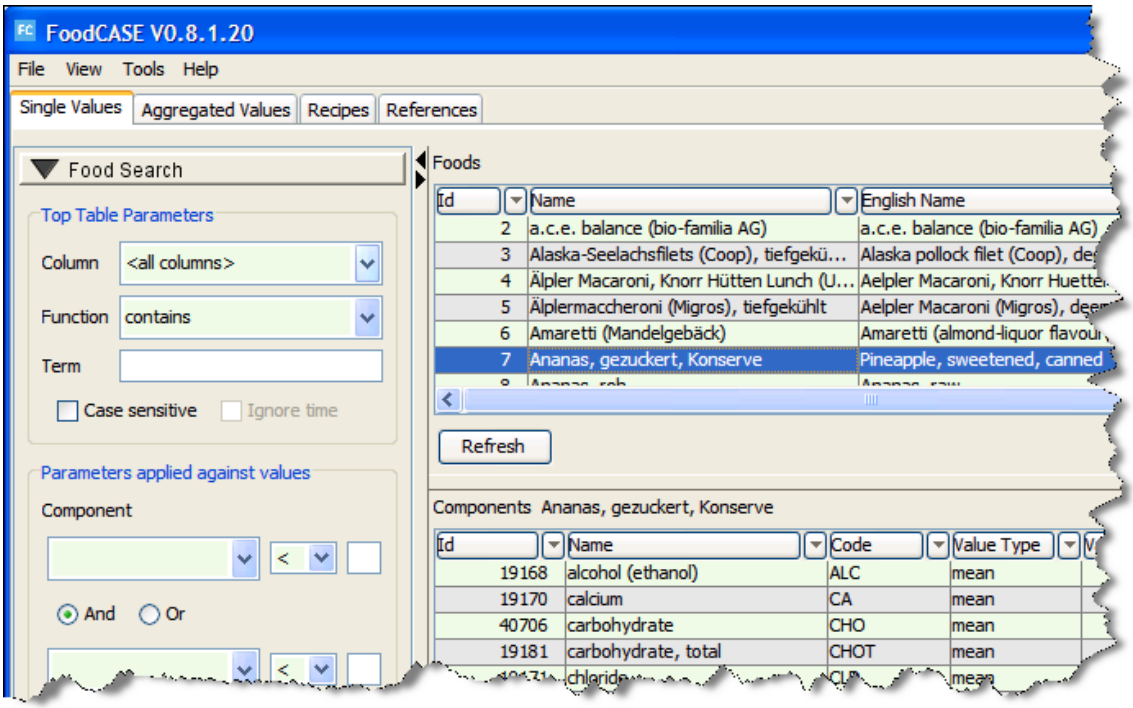


Figure C.2: The Compiler Client of FoodCASE and its four registries.

Usability testing of FoodCASE

The judgement of the two main modules of FoodCASE, the Single Value module and the Recipe module, combined with a general "look and play" of FoodCASE, offers a sufficiently deep insight into the system in order to evaluate its potential as a standardised FCDBMS. In order to obtain a first feedback, the pilot version V0.8 of FoodCASE was usability tested as a task of EuroFIR WP1.8.

Methodology

The approach used for the usability testing of FoodCASE V0.8 relied on the experience obtained with the usability testing of EuroFIR's eSearch prototype [77], the usability testing carried out on the Online-Version of the Swiss Food Composition Database (the predecessor of FoodCASE) [78] and considering the recommendations laid down by Hornbæk [79].

The Usability Testing and Process

The usability testing designed for FoodCASE V0.8 consisted of two specific, but everyday tasks of a compiler's routine work: addition of a component/value pair with meta information to a food existing in the database and creation of a recipe. Additionally, the participants were invited to have a general look and play and to give feedback about their experience whenever possible, both positive and negative, and suggestions for improvements, using a structured online feedback form on Surveymonkey.com. All in all, the testing consisted of 40 questions.

Participants

In total 53 persons involved in compilation of food composition databases representing 29 institutions and 22 countries (with two non-European countries) were invited to participate in the usability testing. The invitation was accepted by 36 persons and all of them started the testing. However, only 25 completed the online feedback form. The Swiss compilers did not participate in the testing, because they were already very good accustomed to FoodCASE due to its use in everyday's work and preparation of the testing.

Results

The results are given separately for each of the 40 questions, but only the results of the 25 full responders are displayed. The difference between the 25 full and 36 partial responders for the first eight questions, for which all 36 partial responders gave feedback, was however negligibly small.

General questions

1. How do you estimate your general computer skills?

Answer Options	absolute beginner	beginner	average user	power user	absolute professional	Rating Average	Response Count
I am a...	0	0	16	5	4	3.52	25
	0 %	0 %	64 %	20 %	16 %	100 %	25
						answered question	**25**
						skipped question	**0**

2. How do you estimate your skills as food composition data compiler?

Answer Options	absolute beginner	beginner	average user	power user	absolute professional	Rating Average	Response Count
I am a...	1	8	6	8	2	3.08	25
	4 %	32 %	24 %	32 %	8 %	100 %	25
						answered question	**25**
						skipped question	**0**

3. Do you manage your food composition data electronically?

Answer Options	Response Percent	Response Count
Yes	80 %	20
No	20 %	5
	answered question	**25**
	skipped question	**0**

4. Which software are you using to manage your food composition data?

Answer Options	Response Percent	Response Count
Excel or software based on Excel	10 %	2
Specific software based on relational database management systems (e.g. Access, SQL Server, Oracle, MySQL, PostgreSQL)	60 %	12
Other software (specify below)	30 %	6
	answered question	**20**
	skipped question	**5**

Other software: Food Table Input software written in VB6 using Access databases; MS Access, Alimenta; DB2 Slovak DMS version 1.2 & DB3 - Excel & DB4 Slovak Alimenta 4.2; specific software: Axiant 4GL Client; DBMS & Excel

5. Are you using FoodCASE for the first time?

Answer Options	Response Percent	Response Count
Yes	68 %	17
No, I already used a test version	32 %	8
	answered question	**25**
	skipped question	**0**

6. What operating system runs on your computer you use now for this testing?

Answer Options	Response Percent	Response Count
Windows XP	68 %	17
Windows Vista	8 %	2
Windows 7	8 %	2
Linux	4 %	1
Mac	12 %	3

No idea	0 %	0
	answered question	**25**
	skipped question	**0**

First impression of FoodCASE

7. Select the choice which most closely matches your feelings.

Answer Options	terrible	bad	neither bad nor good	good	excellent	Rating Average	Response Count
The first impression of the look-and-feel of FoodCASE V0.8 is...	0	1	7	13	4	3.80	25
	0 %	4 %	28 %	52 %	16 %	100 %	25
						answered question	**25**
						skipped question	**0**

8. Do you think that the basic navigation and structure of FoodCASE is intuitive in respect of performing some simple tasks?

Answer Options	not intuitive at all	somewhat intuitive	neither / nor	rather intuitive	completely intuitive	Rating Average	Response Count
FoodCASE is...	1	3	3	17	1	3.56	25
	4 %	12 %	12 %	68 %	4 %	100 %	25
						answered question	**25**
						skipped question	**0**

Task 1: Adding component/value data to an existing food

Question 9 was about identifying and inserting the Food Id of the food to be used by the participant to complete question 10, so the corresponding results are not relevant and were not included in this report.

10. Use the following fictional report to insert an additional magnesium value and its corresponding meta information to your food in FoodCASE.

Date of Report: March 4, 2010 Author: Mr PerfectLab Organization: PerfectScience Lab, University of Zurich, Switzerland Description: 10 samples of the food were purchased in the supermarket PerfectFoods in the city of Zurich on March 3, 2010. The samples were transported to the university laboratory according to the information given on the package and then stored in the laboratory as recommended on the package. The magnesium analyses were carried out the day after the purchase. Magnesium was analysed according to a correct AOAC method (atomic absorption spectroscopy) and using all relevant certified reference materials and other standards. The analyses were carried out in triplicate for each of the 10 samples; the mean and standard deviation for the 10 samples was 35 ± 15 mg / 100 g edible portion. Were you able to successfully complete the task, i.e. do you see the value just entered in the component list?

Answer Options	Response Percent	Response Count
Yes	88 %	22
No	12 %	3
	answered question	**25**
	skipped question	**0**

11. Can you tell why you did not succeed in completing the task? *(=question with skip logic, only appearing for those who responded No to previous question 10)*	
Answer Options	**Response Count**
	3
answered question	**3**
skipped question	**22**

Comments:

1. I suppose I performed my task too quickly, I should have had a wider knowledge of the information contained in the software
2. Could not figure out how to do it and time delay on downloading very frustrating - been trying for 4 hours to complete task Sorry :(even with sons help I am not intuitive enough.
3. I don't know! I entered the information but was not able to save and close the data entry window

12. Is FoodCASE built logically to complete this task?							
Answer Options	**completely illogical**	**somewhat illogical**	**neither illogical nor logical**	**rather logical**	**completely logical**	**Rating Average**	**Response Count**
FoodCASE is built...	0	1	2	18	4	4.00	25
	0 %	4 %	8 %	72 %	16 %	100 %	25
						answered question	**25**
						skipped question	**0**

13. How simple/difficult was it to complete this task?							
Answer Options	**absolutely simple**	**somewhat simple**	**neither simple nor difficult**	**rather difficult**	**absolutely difficult**	**Rating Average**	**Response Count**
It was...	1	10	13	1	0	2.56	25
	4 %	40 %	52 %	4 %	0 %	100 %	25
						answered question	**25**
						skipped question	**0**

14. How long did it approximately take to complete this task?							
Answer Options	**< 3 min**	**3 to 5 min**	**>5 to 10 min**	**>10 to 15 min**	**> 15 min**	**Rating Average**	**Response Count**
It took me...	0	13	3	4	5	3.04	25
	0 %	52 %	12 %	16 %	20 %	100 %	25
						answered question	**25**
						skipped question	**0**

15. Do you think that the time you needed is appropriate for this task?		
Answer Options	**Response Percent**	**Response Count**
Yes, and if I would use FoodCASE regularly I think I would be even faster	44 %	11
Yes, but I don't think I would be faster if I would use FoodCASE regularly	12 %	3
No, but if I would use FoodCASE regularly I think I would be faster	40 %	10
No, and I don't think I would be faster if I would use FoodCASE regularly	4 %	1
	answered question	**25**
	skipped question	**0**

16. Are there any changes or improvements you would like to suggest in order to complete this task in a better way? If so, please describe (optional).	
Answer Options	**Response Count**
	16
answered question	**16**
skipped question	**9**

Comments:

1. In Single Value Detail / Samples the field Date of Sampling could have a calender.
2. Difficult to say. I suppose that many real life taks are more complex like e.g. the information available does not follow the FoodCase order. This was the ideal case. Maybe the user could give preferences for the order of the tasks to match the indivisual needs - not something for the version 1.0 :)
3. the indication of stored information and storaging of the entered infromation is hardly noticable
4. "About Highlighting/selecting a food item for showing content (open it): It is not intuitive that the chosen food should be clicked on (marked with darker background colour) to show content.
   Information text, " mouse-over"", (yellow background): The text box with info on thesaurus details (for every info word) is closed too soon. The info cannot be read in such a short time.
   The text box should be able to close manually.
   Use of the ""Duplicate""s in contrast to ""New"": There could be some ""mouse-over"" text with info of the use of Duplicate, in contrast to using the function New when adding a new component (why not include duplicate for food item? Sometimes only cooking method is the difference)."
5. Some of the fields might be filled in by default, e.g. mg for MG (!) and Matrix Unit by food. A comparison of error rates using each approach (full input and defaults) would be interesting. Error rates for full input would be reasonably easily accessed automatically by a validation routine.
6. "The list of components available for this specific food should appear in the field ""components"". There is a big white space below ""Components"", but none components are showed. Therefore, it was not clear if the component, which I had to include, was already available or not. So, it was not directly clear that I have to create a new component, because the list does not appear."
7. "There are no keyboard shortcuts for commands. In regular everyday use, everything should work also completely without mouse.
   Search may be too limited. Usually there are more than one column used as search condition (like e.g. name begins with Ce AND edible portion < 100)"
8. It would be helpful to receive instant feedback to be reminded on missing relevant parameters.
9. to have the values added to the system electronically. So you can complete a file, for example Excell, somewhere on your computer with many different values and have them read in to the system all at once
10. "Entry of method indicator - in my practice I use directly codes method indicators (e.g. MI0123), it is more convenient than to make a choice from a long list of methods.
    I have entered two values for MG (testing duplicate and new entry option) - but I do not know how to delete one of the inserted values.
    a minor comment: In the sample form a calendar for entering date in the date of sampling is missing (in other boxes for entering date is present)
    According to the standard (version available on the technical web) the reference entity shoud contain as mandatory title, authors, publication date and original language, an according to Anders a CitexploreID and translation of the title into English - this does not infuence performing the task, but I would like to only mention it"
11. I would suggest to add indications about the meanings of the buttons and the steps to follow to achieve the goal
12. I thought first that I would add a new value and still see the old value. But I probably deleted the old value.
13. The time taken is related to value documentation and deciding what information is applicable and what should be entered. FoodCASE is logical and all information can be entered easily once it has been interpreted.

14. It would be useful to include explanatory information about use
15. I suggest training for me as a medium kind of internet user (propably not normal as no researcher I know is know can be described as normal). We hve some common sense but propablly not enough. Even with help from my child wjhois much more computer literate we gave up - just took too much time. however, he thought it a well written proramme and I though i had great potential if only I know how to use it.
16. "I think I might have missed some points. My scree is quite small, maybe I do not see all parts of the panels"

Task 2: Creating a recipe with help of clipboard

17. For the purpose of the testing we use the following shortened list of ingredients: 2 raw eggs 1 tea spoon white sugar 3.75 dL whole milk pasteurized 125 g butter of choice 750 g wheat flour bakery type 550 Everything has to be mixed in a specific order (sorry, can't name the order, it is a secrete). The dough is left to levitate for one hour at room temperature, the dough is woven in a traditional way, and then baked in the oven at 200 °C for 45 min. Name the recipe "Bauernzopf_yourInstitution", i.e. for ETHZ it is Bauernzopf_ETHZ. The yield factor for water is 0.9, the other yield factors are 1.0. Were you able to successfully complete the task, i.e. do you see the food you just have created in the Recipes list?

Answer Options	Response Percent	Response Count
Yes	72 %	18
No	28 %	7
	answered question	**25**
	skipped question	**0**

18. Did you consider re-ordering the recipes? *(=question with skip logic, only appearing for those who responded No to previous question 17)*

Answer Options	Response Percent	Response Count
Yes	57 %	4
No	43 %	3
	answered question	**7**
	skipped question	**18**

19. Re-order the recipes by clicking on the column head of the recipes. Is your recipe now visible? *(=question with skip logic, only appearing for those who responded No to previous question 18)*

Answer Options	Response Percent	Response Count
Yes	0 %	0
No	100 %	3
	answered question	**3**
	skipped question	**22**

20. What problems did you encounter carrying out this task (optional)?

Answer Options	Response Count
	22
answered question	**22**
skipped question	**3**

Comments:

1. "The first time I tried to create the recipe, that i named as _INRAN_CA, I encountered the following problems. After adding the ingredients and selecting ""calculate recipe"", the list of ingredients was empty. I tried to cancel the empty recipe (named _INRAN_CA),with no success. So once opened the recipe I selected ""add ingredients"" and I added the ingredients from the list. I only added 3 ingredients, I selected skimmed milk instead of whole milk and I tried to change the selected items, but with no success.

So after many attempts I created the recipe with a different name ""INRAN_1"", this time I managed to perform the task, but the ingredients butter is repeated twice and there was no way to cancel one of them."

2. "I didn't understand how to activate the Food Indexer to add the LanguaL codes.
   I couldn't enter Weight after cooking, which is a useful information when yield factors are unknown."
3. it was difficult to understand where to write quantities of the single ingredients, and other information. When I finally loaded this information, and I was ready to calculate the recipe, the software informed me that a quantity was missing, even if I saw all values on the screen. I re-create the recipe again and I finally complete the task.
4. "After I added all the ingedients, I tried to calculate. Then FoodCase complained that some of the amounts were missing (Please fill in all values for all ingedients. Then later on I got the message that yield factors are invalid (in the screen 0,9 1 and 1). BTW. The latter happened when the recipe name was empty :)
   Then I even tried to give preparation method. No difference. BTW: How could I the prepararion method if I wanted?"
5. the recipe list was not saved even I repeated the recipe creation, I did not expect that there would be possible to use metric units, I recalculated milk volume and eggs into grams instead
6. Did not complete it because of difficulties saving the added ingredients. The Add Ingredient screen only provides the options to "Calculate" or "Cancel". Although closing this, leaves a dialog with a "Save" button, this does not seem to work.
7. "- Selection list; Add a ""Close"" function It is not possible to get back to the last screen without having chosen a food item.
   - Save ingredients before calculation. Make it possible to save added ingredients before calculation, in case of interruption."
8. "It was not clear how to edit the ingredients' amount for the recipe. I had to restart the system."
9. "Using the clipboard to create a recipe was rather tedious and non-obvious job. Maybe there could be other ways to do it also.
   Calculation of recipe does not give any feedback of progress to user. Progress bar of some kind would be nice.
   Recipe procedure description is not asked during recipe creation, only afterward if one edits the recipe."
10. General yield factors would be useful.
11. at first it was not clear to me that I had to press the refresh button to see my recipe on the list. once you know this this will not be an issue anymore
12. I do not have details about weight relations of tablespoons, egg weight and volume of milk - I mean how they are recalculated to weight. I made two entries one ingredients entered in g and one in household measures - results are slightly different. In different countries household measures could be use differently.
13. It took some time to realize the required procedure.
14. I tried other recipes and I succeded, in this specific case I have used both the clipboard (create a recipe) and adding the ingredients in the creating recipes duction, but at the end no record was stored. However, I thought about the order, but it took too much time to try all the permutations of elements...
15. No problems. Had to know to right click foods and change to the aggregated values.
16. Some problems with the search of aggregated foods
17. Not very intuitive at times. Some problems with calculate button not working and added ingredients not being saved (possibly due to slow internet response).
18. "At first I did not see the unit drop off list and so I began converting all quantities to g on my own.
    I did not manage to see the created recipe in the recipe list in the end. I tried for approximately an hour but could not spare any more time to try all possible combinations if it is a matter of ingredient order."
19. "I did not find an ingredient in the lis (whole milk, pasteurised).
    I could not find where to insert the preparation procedure"
20. Time delay on line too long - maybe becauce of my remote location in Africa vs Europe

21. "I started too late working on my tasks!!!!
The problem with recipe is that I get the list of ingredients in german..... and I am not so good at that so I do not find the needed ingredients.....I did not find eggs, whole milk (onlu skimmed), etc...and now it is past midnight and time is over, correct?"

22. "Some metadata (e.g. English name and preparation method) can't be entered when creating a new recipe. To add that information I needed to open the details of the completed recipe and edit them.
I had trouble finding all the buttons/commands at first because I don't know the application well. I clicked 'New' under recipes at first instead of 'Create Recipe' on the clipboard and then I was confused for awhile. This would probably not be a problem once users gained some experience."

21. Is FoodCASE built logically to complete this task?

Answer Options	completely illogical	somewhat illogical	neither illogical nor logical	rather logical	completely logical	Rating Average	Response Count
FoodCASE is built...	0	6	6	9	4	3.44	25
	0 %	24 %	24 %	36 %	16 %	100 %	25
						answered question	**25**
						skipped question	**0**

22. How simple/difficult was it to complete this task?

Answer Options	absolutely simple	somewhat simple	neither simple nor difficult	rather difficult	absolutely difficult	Rating Average	Response Count
It was...	3	5	8	7	2	3.00	25
	12 %	20 %	32 %	28 %	8 %	100 %	25
						answered question	**25**
						skipped question	**0**

23. How long did it approximately take to complete this task?

Answer Options	< 3 min	3 to 5 min	>5 to 10 min	>10 to 15 min	> 15 min	Rating Average	Response Count
It took me...	1	1	4	5	14	4.20	25
	4 %	4 %	16 %	20 %	56 %	100 %	25
						answered question	**25**
						skipped question	**0**

24. Do you think that the time you needed is appropriate for this task?

Answer Options	Response Percent	Response Count
Yes, and if I would use FoodCASE regularly I think I would be even faster	24 %	6
Yes, but I don't think I would be faster if I would use FoodCASE regularly	8 %	2
No, but if I would use FoodCASE regularly I think I would be faster	68 %	17
No, and I don't think I would be faster if I would use FoodCASE regularly	0 %	0
	answered question	**25**
	skipped question	**0**

25. Are there any changes or improvements you would like to suggest in order to complete this task in a better way? If so, please describe (optional).

Answer Options	Response Count
	14
answered question	**14**
skipped question	**11**

Comments:

1. There should be an easy way to cancel/add/modify the ingredients and the recipes.
2. "I would suggest to implement searching by more independent keywords, e.g. raw eggs (not egg, raw).
   As in some cases the order of ingredients is important, I would suggest to enable the ordering of ingredients - I started by entering eggs, which are now on the second place.
   Is the preparation method linked to an ingredient or the whole recipe? To me, this is not clear from the UI."
3. I was not sure when the FoodCase Saves the recipe. In the recipe window there is only Calculate and Cancel. How should I have saved the recipe? What is the connection of the recipe and the food? That was not clear. e.g. If I have the food, how I connect it with a certain recipe?
4. to assure a user that the recipe is stored before the calculation is carried out, it is useful to store preliminary recipe, otherwise you have to create recipe again and it is very tedious
5. The instructions confuse Aggregated foods and Recipes and it was difficult to ensure the correct screens were being used. When an ingredient was added, it seemed difficult to enter data in fields other than the amount. The hierarchy of dialogs does not seem intuitive (although it may be a desadvantage that I have implemented recipe management using a somewhat different approach).
6. "- Info about highlighting food item/nutrient for selection in main screen.
   - Selection list: with English optional as first language. When selecting an ingredient, nutrient or other item, search alternatives should be clearer in the first stage (more like in the left column). Why not be able to write in the ""grey line"", otherwise hide it. The search field is too high up in the second screen."
7. "Drag-and-drop of lines from food item panel to recipe component panel.
   Finding food items by typing first letters of values of selected column."
8. "To add a recipe reference field into the recipe documentation table
   An information about retention factors in not accessible, firstly to see them and secondly to add new ones in the case of neccessity"
9. Yes, I would like to have some statements explaining the most suitable stepwise procedure
10. A whritten description within the system would be good.
11. Some improvements in recipe calculation. More parameters added, like time and tempeture.
12. Searching ingredients based on English food name. This would be much clearer with the help of a user guide - easy when you know how!
13. I think a desk top version may be simpler with appropriate training.
14. "Drag and drop to clipboard might make things easier for new users, though I am aware that this is an early testing version and you wouldn't want to waste time yet implementing those kinds of features.
    For advanced users, shortcut keys might be helpful to speed things up and reduce mouse movements/RSI."

Second impression of FoodCASE

26. Considering the two tasks performed and the few minutes you looked at FoodCASE at the beginning, what do you think about the understandability of

Answer Options	in need of explanations	somewhat in need of explanations	neither / nor	rather self-explanatory	self-explanatory	Rating Average	Response Count
FoodCASE is...	3	13	0	8	1	2.64	25
	12 %	52 %	0 %	32 %	4 %	100 %	25
						answered question	**25**
						skipped question	**0**

27. Are the terms used in FoodCASE generally understandable (e.g. in the menus, buttons, descriptions)?							
Answer Options	**not understandable at all**	**somewhat not understandable**	**neither / nor**	**rather understandable**	**fully understandable**	**Rating Average**	**Response Count**
The terms are...	0	2	3	15	5	3.92	25
	0 %	8 %	12 %	60 %	20 %	100 %	25
If you did not rate "fully understandable", why did you so?							**17**
						answered question	**25**
						skipped question	**0**

Comments:

1. Because I encountered several problems to understand how to proceed to create a recipe.
2. "It is not clear how and where write some information.
   In my case, the internet connection was very slow, so it didn't help me in load some info."
3. "I suppose that frases like ""parameter applied against values"" or using in general the term ""parameter"" may be confusing for the compilers. The combination column-function-term is fully correct but in a sense perhaps too ""official"" for an average user. A normal user IMHO does not think in columns and functions. So the result is in one hand very ""over correct"" but not intuitive. Sorry, do not have a good solution either :(Something should be done but do not get exactly what. Yes, the user could learn to use this but there is still a big BUT....
   Maybe how the things connect to each other (like the food and the recipe). Then the search perhaps should be somehow separated from the data (like colours or some other element). Then the user could keep the two things more distinct: when I am searching something and when I am changing the data.
   Some small things: When I added the food, it has a mysteriously whole list of values. However, they disappeared when I saved it. Then, I would like to see some of the metainformation in the normal single value screen."
4. the principal of structuring the data, e.g. two magnesium values (sinlge entries) in one component profile, it is unusual approach, I expected separated files with single data entries per one sorce
5. When, for example, there is a "Save" button, it is not clear what is being saved (e.g. the ingredient list or the whole recipe record).
6. """Mouse over"" can be more in use, and with ""close"" function. See earlier comments.
   I could see that the recipe was created with help of the clipboard, as in the instruction."
7. We were not able to define analysis replicates of the samples.
8. "i did not completely understand which fields you have to complete. for instance when you made a recipe no extra information is asked for, but if you click on the recipe after it is made you get to the details and it is required to fill in the Englisch name. With adding the single value this is also an issue.
   Method field: there are three names required: method name, original name and method indicator, I do not understand this difference"
9. "Function of markers,the aggregation procedure and the operations in the clipboard remain unclear to me. Details collected for the single (food detail) and aggregated food (e.g. recipe 114/130 for biber) are different - I do not understand why - does it mean that it is necessary to fill the forms twicely for one food? (e.g. for biber?)
   ID for foods in the single value and aggregated value modules are different - what is the Original Food ID for e.g. biber?
   In the food detail (for single food) in the other properties form, there are three boxes for food preparation - why?
   I do not understand how to create outputs from the system (if they are possible), I mean e.g. matrix tables"
10. A users manual is essential to avoid confussion and errors and save time.
11. Some buttons can be interpreted in an arbitrary way. As an example, what does it mean "aggregation"? Is that the array of the set of nutrients? Please, explain more especially for non English people

12. People might try to start from Single values.
13. The relationship among single food, agreggated food, values, samples. The hierarchic structure is not completely clear due to the free style of foodCASE
14. It is not entirely intuitive - probably because of the amount of information that could be added and the comp;ecity of tasks that could be performed.
15. eg an entry in column restrict access to data but you do not understand that unless you delete entry
16. I was not able to fulfil the task...it means tghat it was not fully understandable....
17. "There are too many truncated columns which are cut off, so it's hard to see what's going on. I tried reducing the DPI on my computer to see more text, which helped a bit, but it was still problematic. The user needs a convenient way of turning these on and off because the screen is too cluttered otherwise. My monitor's resolution is 1440x900, on a smaller monitor it would be worse.
It took a little bit of time to work out how to do the exercises because I wasn't familiar with the interface. If I were to repeat them now though, I think it would be easier and faster.
I was a little bit confused when I logged in because in some places I couldn't see any English. That may not be a problem for your purposes."

28. After having performed the two tasks, which choice most closely matches your feelings?

Answer Options	terrible	bad	neither bad nor good	good	excellent	Rating Average	Response Count
My second impression of FoodCASE is...	0	0	6	17	2	3.84	25
	0 %	0 %	24 %	68 %	8 %	100 %	25
						answered question	**25**
						skipped question	**0**

29. After having performed the two tasks, do you think that the basic navigation and structure of FoodCASE is intuitive in respect of performing these tasks?

Answer Options	not intuitive at all	somewhat intuitive	neither / nor	rather intuitive	completely intuitive	Rating Average	Response Count
FoodCASE is ...	1	5	3	15	1	3.40	25
	4 %	20 %	12 %	60 %	4 %	100 %	25
						answered question	**25**
						skipped question	**0**

30. Is FoodCASE, as far as you have used it, in your opinion well-arranged?

Answer Options	not well arranged at all	somewhat not well arranged	neither / nor	rather well arranged	completely well arranged	Rating Average	Response Count
FoodCASE is ...	0	2	7	14	2	3.64	25
	0 %	8 %	28 %	56 %	8 %	100 %	25
						answered question	**25**
						skipped question	**0**

31. How do you like the design of FoodCASE?

Answer Options	terrible	bad	neither bad nor good	good	excellent	Rating Average	Response Count
The design is...	0	1	9	13	2	3.64	25
	0 %	4 %	36 %	52 %	8 %	100 %	25
						answered question	**25**
						skipped question	**0**

32. What do you think about the information density/number of fields on the two modules single value and recipe calculation?

Answer Options	too little	somewhat little	as it should be	rather high	too high	Rating Average	Response Count
The information density/number of fields is...	0	1	11	12	1	3.52	25
	0 %	4 %	44 %	48 %	4 %	100 %	25
						answered question	**25**
						skipped question	**0**

33. FoodCASE was designed as a basic system for the everyday tasks of a compiler. How do you rate FoodCASE in this respect?

Answer Options	less than a basic system	somewhat less than a basic system	a basic system	rather more than a basic system	more than a basic system	Rating Average	Response Count
FoodCASE is...	0	1	9	10	5	3.76	25
	0 %	4 %	36 %	40 %	20 %	100 %	25
						answered question	**25**
						skipped question	**0**

34. According to your personal opinion, does FoodCASE in its V0.8 contain all features you need to manage your food composition database?

Answer Options	only few features I need	some features I need	about half the features I need	most features I need	all features I need	Rating Average	Answer Options
FoodCASE contains...	0	4	0	18	3	3.80	25
	0 %	16 %	0 %	72 %	12 %	100 %	25
What is missing according to your opinion (optional)?							**13**
						answered question	**25**
						skipped question	**0**

Comments:

1. importing / exporting data
2. "Our database is a compiled database and data came from other references, mainly food compostion databases.
   In FoodCASE maybe is missing a section where to store exactly the origin of these data: food code and food name of the item. At the moment it could be used the remarks section, but it could be better to use specific records."
3. "Component grouping - Yes, it is coming :)
   Own food aggregations - perhaps markers could do the task. I.e. making free aggregations of the foods.
   Still, we constantly use one own ingredient classification and one food classification (near to those of EFSA). For us, they are must.
   Then we have some components that do not have the EuroFIR code. (like GIs) There will be always some such components. That is why there is the EuroFIR component code and the other national code. Without this we could not add the components until they have the EuroFIR code - and this may take years."
4. "e.g. missing reference for recipes, possibility to store portion size for meals (recipes), possibility to add own retention factors, information on number of values entering recipes comparing the number of ingredients (i.e. whether all ingredients keeps all nutrients or what values are missing, number of entries would be acceptable), could be indicated what reference was used of recalculating household measures (spoons, dcL, etc.)
   Labelling at Single Food record: ingredient list of food product should be possible to be recorded with

amounts and also to have possiblity in future to connect ingredients with the food coding, there might be needed that each ingredient is described separately as a unique food; it would be useful to have field for citexplore code in reference; I noticed duplicity in reference list of the record Mr. Perfect Laboratory, there should be some monitoring of the existing reference list records and comparing with new entries, or at least notify user to check the existence of the reference before entering a new one; in standard ARTICLE NUMBER stands for GS1, you have further filed with BAR CODE, it should be clarified what code is expected"

5. Management of nutrient retention factors.
6. "Difficult to consider at this stage.
   It should be possible to look in many databases at one time to find identical food items for possible comparison. The search should be able to export to Excel og Access.
   Information about ""Refresh"" button. What happens? I cannot see any change."
7. "possibility to read in information electronically
   some automatic procedures (for instance when a value changes at one product and you 'take over' this value for an other product that this value is changed aswell): more examples have been given to you before
   at this test working with the aggregated values is not tested, so I would have to go into that at an other moment"
8. "The component entity is missing - how could be entered original names of components?
   A box to enter recipe reference
   I haven´t find the edible portion field
   An option to view retention factors (and enter own)
   An option to create a database of portions (is not included in the standard) but it is very useful (US FCDB uses it)
   I haven ´t understand how to enter a list of food groups
   Are there any control mechanisms of the system that could warn the complier that something is wrong with the values (e.g. sum of mineral does not exceed the value for ash)?
   How could be provided saving of data uprades?"
9. maybe some felds that were not in the present usability test, like food description and coding. Moreover the possibility to create an average array (if I have understood well how to use the aggregation functions, I can say up to now not very clear)
10. "Keep old values together with the new.
    Are the Eurofir thesauri fully applied? E.g. AR, AV, BE etc (Value type). Mean is often selected. It looks like MN is used too often. According to scope notes MN is not always fullfilled in our database."
11. I would like to be ableto compare multiple data sheets of info eg raw vs cooked of different countries etc
12. well, I need more time to check it out
13. "As I mentioned, I think drag and drop and shortcut keys might enhance some features.
    Auto-completion of fields/suggestions might speed things up?"

35. Can you describe things you especially liked about FoodCASE (optional)?	
Answer Options	**Response Count**
	16
answered question	**16**
skipped question	**9**

Comments:

1. The design is clear, the structure well defined, a user can work intuitively. Great done!
2. The lay-out it’s very similar to that of LanguaL, with the 3 main sections or panels; for ones familiarized with LanguaL it’s easier to get into the system, I guess. Also, if I would use FoodCASE regularly I think I would be faster managing the tasks and into knowing the system.
3. It seems compact and well structured.

4. "Structurized and heavily organized. The missing field were clearly marked - except once it was in an other tab - a tricky thing to solve. Mostly gives good feedback,"
5. good and easy searching functions, flexibility to modify user interface (e.g. moving columns, changing ordering, changin proportion of panels),
6. Adherence to the EuroFIR standard.
7. A lot of possibilities, but not finished yet...
8. "FoodCASE (at this stage) seems to be plain and simple basic implementation of management program. Everything works, but no new innovations, no fancy features to impress users. If the program aims to be accepted by users (and not to be used only because users must or have no other choices) developers should concentrate on usability and user experience.
   A good start, many things to be done still..."
9. "extensive search possiblities
   the possiblility to use household measures in the ingredients of the recipes
   that the eurofir requirements are implented"
10. "In general it is a convenient system for data management
11. The system is based on the EuroFIR standard
    Highlighting the mandatory fields in red helps not to omit mandatory fields
    Entering the recipe is convenient"
12. Very detailed system but time will tell if it is practical too.
13. the system in itself
14. The possibility to fill in all the information for the values.
15. "Standard thesauri fully integrated
    The amount of information that foodCASE can store (references, methods, sample information, etc.)
    Cooking and agreggation clipboard panel"
16. "I did not have time to check it with attention
    so for example I did not look at the way it presents the various options for food documentation...."
17. "The Quality Panel is a great idea and works well. It helped me to complete my tasks when I was still learning the interface and was a bit confused.
    The copy to clipboard concept works well and I liked it."

36. Can you describe things you especially disliked about FoodCASE (optional)?

Answer Options	Response Count
	10
answered question	**10**
skipped question	**15**

Comments:

1. "Structurized and heavily organized - if that is not your structure :). The search system is not for me :)
   Use some additional elements to separate different aspects of the interface. Too much battleship grey for me :)
   Sometimes I would need more feedback from the system.
   You can open the same food several times but they seem to be rather independent screens than several simultaneous windows to the same data (i.e. when you change one, it woud make the same change in every copy). This is just a detail."
2. Difficulty in knowing what command buttons do and if they have worked. For example, component search does not appear to give an indication why nothing happens, although relates to whether a food is selected. I did not find this intuitve.

3. "The info boxes closed too early, so that you cannot finish reading.
   No info on ""Duplicate"", ""Clipboard"", ""Refresh""."
4. "I was lost in the aggregation procedure and using marks. Use of the clipboard is not clear to me at this stage of testing.
   Very small letters in the box reason for sampling (and other with this format)"
5. Not sure on the use of items such as cooking / aggregation / food markers / value markers, I would like to see a documentation of use.
6. the lack of explanations
7. No.
8. "The refresh should be automatic when searching
   More control on calculation (not like a black box). Some logs about calculations."
9. No explanation of functions for me being rather medium food data base illiterate compared to normal users
10. "There are some bugs that need to be fixed:
    After opening the 'Single Value Detail' form for Tofu - all-trans retinol I was unable to close it and had to just move it to the background.
    I was switching often between Firefox and the form I was using for the exercise. Often after successfully selecting an input from a list (e.g. acquisition type in the Mg exercise) the entire form would lose focus and it would transfer to Firefox.
    The use of the word method for 'Method type' as well as the 'Method' tab confused me for a moment and I went to the wrong place to enter the method. The former is perhaps more of a source, data source, source type or derivation?"

Likeliness of using FoodCASE

37. How do you rate FoodCASE, as it is now in V0.8, compared to what you use now to manage your food composition database?

Answer Options	much worse	worse	the same	better	much better	Rating Average	Response Count
FoodCASE V0.8 is...	0	4	2	8	11	4.04	25
	0 %	16 %	8 %	32 %	44 %	100 %	25
						answered question	**25**
						skipped question	**0**

38. If the first full release of FoodCASE V1.0 will contain the features you need (including any you mentioned previously that are missing), how will it compare to

Answer Options	much worse	worse	the same	better	much better	Rating Average	Response Count
FoodCASE V1.0 will be...	0	1	3	11	10	4.20	25
	0 %	4 %	12 %	44 %	40 %	100 %	25
						answered question	**25**
						skipped question	**0**

39. How likely would you want to use the first full release of FoodCASE V1.0, i.e. including the features you need, to manage your food composition database?

Answer Options	not likely at all	somewhat not likely	neither / nor	rather likely	very likely	N/A	Rating Average	Response Count
If FoodCASE was freely available, including granted free support (no matter who will deliver the support), I would want to use it...								
	0	0	2	7	15	1	4.54	25
	0 %	0 %	8 %	28 %	60 %	4 %	100 %	25
If FoodCASE was freely available, including granted free support, and support delivered by EuroFIR AISBL, I would want to use it...								
	0	0	2	5	17	1	4.63	25
	0 %	0 %	8 %	20 %	68 %	4 %	100 %	25
If FoodCASE was freely available, except support which I/my institution would have to cover, I would want to use it...								
	0	6	2	6	9	2	3.78	25
	0 %	24 %	8 %	24 %	36 %	8 %	100 %	25
If FoodCASE was not freely available, and no free support was available, I would want to use it...								
	3	7	3	4	4	4	2.95	25
	12 %	28 %	12 %	8 %	16 %	8 %	100 %	25
If you did not rate "rather likely" or "very likely" in one or more of the above statements, what are the reasons for you rating?								12
							answered question	**25**
							skipped question	**0**

Comments:

1. "The ideal situation is:
   - use a software without paying any expences linked to the software support;
   - use a software with the full access to data, in particular exporting them without any problem (For es. we have the data published on the web)"
2. It depends on the price.
3. It might be used (1) to manage standard nutrient retention factors and (2) for the input of data for the EuroFIR Data Repository, but further investigation and discussion will be needed in these areas.
4. In my opinion, there would be no need to maintain a national food management software if there is a good alternative available for everyone. But this alternative should be suitable for our needs, freely extendable, have an open API, preferable open source code, support available (free or non-free) and bugs fixed at regular interval. In other words, individual institutions should have enough control of FoodCASE in order to start using it.
5. It depends on the functions of the final version, possibility of transfer of our data and of course of the price for services.
6. It will depend on the cost, the terms of use, and the feasibility to cover technical support in our organization.
7. In the case the software was not freely available I can arrange the system as I like using other tools
8. Maintenance is a big issue.
9. Depending on the price of the product, if it is not free
10. Funding for this realy limited in developing world
11. I do not know if I would fine any funding to purchase the software if not freely avaiable
12. FSANZ are currently running their own project trying to combine their risk assessment (e.g. exposure assessment) database with their food composition database. FoodCASE would only fulfill one of these aims. FSANZ's position is slightly different to many other institutions as both of the tasks mentioned above are carried out within the one institution (in fact the one section of FSANZ) so we have different business requirements to many others.

40. If you have any additional comments that you would like to make, please record them here (optional).	
Answer Options	**Response Count**
	13
answered question	**13**
skipped question	**12**

Comments:

1. I think the system does not work properly yet.
2. It would useful if the copy/paste function between FoodCASE and other applications would be implemented.
3. "The system has a lot of potential and will help my work as a compiler, for sure.
   It is in general quite auto-intuitive and self-explained although it would be nice to have a (simple) instructions manual prior to its use. Something with a very ""straightforward structure""...
   Interesting the Look&Feel View possibility; however, it's not always working.
   Keep up with the good work!"
4. It would be nice to see more functions, e.g export, LanguaL coding etc.
5. FoodCASE is a good basis for data management, but I have never been convinced that a purely tabular interface allows the data displays to be fully intuitive.
6. Well done! The system looks quite promissing and includes what the food composition data world has been waiting for for long!
7. "You have performed a very good and useful job.
   I wish you good luck in finalising your mission."
8. At the moment, we do not use a software to manage our FC data. The system is quite advanced and detailed and could thus be very helpfull. However, guidelines on the use have to be provided and technical assistance (or training our IT experts instead) might be essential.
9. Developing the system is rather imporant. Please, continue to improve FoodCASE
10. It was very interesting to try the software. Thank you.
11. I think the concept is great but to be widely apllicable and used it needs to be a tool in the Food Comp world, for data intwrchange etc. if need to pay for it participation will be limited to a few countries with limited requirements and adoption.
12. sorry I did not have time to complete this .....my fault
13. Thank you for involving us in testing. I enjoyed seeing your progress so far. Keep up the good work!

Discussion

The usability testing of FoodCASE was quite a time consuming task. The average time spent on the online feedback form was about 120 min and ranged for most of the testers between 45 and 270 min. The reason why 30 % of the testers who started giving feedback on the online form did not finish it is not obvious. It can be guessed that they stopped the testing because of time issues, as all stopped just after the first set of general questions and before commencing to use FoodCASE. The alternative is that there were problems with starting / using the software or the online feedback reporting tool. But with one exception ("online reporting tool did not proceed to next question"), nobody gave an according feedback, which makes the first guess of time issues the most likely one.

The extensive optional comments given by the participants are highly remarkable and indicate that they had a particularly high interest in the testing. Although the tested version of FoodCASE was still the pilot V0.8 and not all features were implemented yet, the general feedback of the testers was positive. This result is of interest for EuroFIR/EuroFIR AISBL as 20 of the 22 represented countries were of European origin. Also to remark is that 5 out of the 25 testers did not use an electronic system to manage their food composition data.

Along with the general positive feedback of the testing several comments indicated that some features were still missing and necessary to make the system complete. Nevertheless, 68 % of the testers rated their impression of FoodCASE as good and further 8 % as excellent. Most of the testers also thought that the system in its present state is rather intuitive, though several comments indicate that training and/or a user guide would be helpful.

84 % of the testers thought further that FoodCASE in its pilot state already contained most or all features they needed, and 76 % of the testers stated it was better or much better than their system. The ratio increased to 84 % of the testers feeling that FoodCASE in its first final V1.0 would be better or much better than their current system. This is reflected in the feedback given for the question aiming at discovering if FoodCASE might potentially be a FCDBMS to be adopted and supported by EuroFIR AISBL. 60 % of the testers would "very likely" use FoodCASE if it was freely available and an undefined entity would deliver free support, additional 28 % would use it "rather likely" under the same circumstance. If the free support was delivered by EuroFIR ASIBL, then the ratio of the "very likely" user even increased somewhat to 68 %, while it decreased correspondingly with the "rather likely" users to 20 %. Main reasons for not selecting either "rather likely" or "very likely" were related to financial issues.

Both authors of the present report are strongly bound to FoodCASE and therefore their opinion on its potential and usefulness is biased. Nevertheless, the results of the entire testing – which was approved by the EuroFIR coordinator – and in particular of the many optional comments given by the testers are positive and a wish to use FoodCASE apparent (provided it was free and support organised). The recommendation to EuroFIR AISBL is certainly to investigate in depth its real potential for the association and its members. Since FoodCASE adopts current standards and can integrate also future standards, it not only could be of real benefit for the Full and Associate members of ASIBL, but it could also represent a good economic opportunity to sell the system to the food industry (because of its standardised recipe calculation module). The intention of the ETH Zurich, the owner of FoodCASE, is in any case to make FoodCASE freely available for food compiler institutions, possibly managed through EuroFIR AISBL.

Appendix D: Quality Function Survey in Evento

General Description

In a pilot project for quality assurance in the BA&PM Team at ZHAW, some quality test functions are implemented for helping to find missing or incorrect event registrations. In this usability testing, we want to collect feedback from users if the quality features correspond to what the user needs and whether these features are user-friendly. The goal of this investigation is to test the functionality, not the user. This means in other words that you cannot do anything wrong.

1. General questions

1. How do you estimate your general computer skills?						
Answer options	Beginner		Average user		Professional	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	1	2	1	1

How many times per week do you use Evento?						
Answer options	Never		2-3 times a week		5 times a week and more	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	1	0	4	0

2. Task 1: Find a missing module registration

Find a missing module registration in your program of study using the provided instructions and check whether the registration is really missing. If the report shows no missing registration in your program of study, please take another program of study.

2.1 Please indicate the ID of the person and the course number.	
ID-Person	Event number
153561	g.PT501-NP.2009 HS.001
141138	g.PT304-P.2009 HS.002
100458	L.AL-TMG-K, L.AL-KMT-K, L.AL-TRT-K
148401	g.ID113-2-V.2009 HS.017
148401	g.ID113-2_2-V.2009 HS.017

2.2 Were you able to successfully complete the task?			
Answer options	No	Yes	No answer
Answer number	1	2	
Number of answers	1	4	0

2.3 What problems did you encounter?

Answers:

1. Problem with multiple used modules
2. Event status “mA.Erlassen” -> “dispensiert” -> “StdAnl.Anm” must be deleted (status: erlassen)
3. No module registration; because student was freed from module g.ID133_2 (ID event=316593)! Eventually check if a registration on the module exists with status “mA.Erlassen”.

2.4 Did you find the data quality report at the place where you expected the report to be?						
Answer options	No				Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	2	3	0

If not 100% Yes, where is it more suitable to place the report?

Answers:

1. The placement is ok, I would have instinctively search in the tab „Alle Anmeldungen“

2.5 Does the report contain enough information?						
Answer options	No				Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	1	0	0	2	1

If not 100% Yes, what information are you missing?

Answers:

1. Column "NummerHM" should contain information for which module the event and the registration is missing.
2. Degree programme information if a module exists in multiple degree programmes.
3. If possible, indicate which module is missing.

2.6 How simple/difficult was it to complete the task?						
Answer options	Simple				Difficult	No answer
Answer number	1	2	3	4	5	
Number of answers	3	0	2	0	0	0

2.7 What do you think about the waiting time for the report to open?						
Answer options	Very long		Adequate		Very short	No answer
Answer number	1	2	3	4	5	
Number of answers	0	1	4	0	0	0

2.8 How long did it take to complete the task?						
Answer options	< 1 min	1-2 min	2-3 min	3-4 min	> 4 min	No answer
Answer number	1	2	3	4	5	
Number of answers	0	2	0	1	1	1

2.9 Do you think that the time you needed to complete the task is reasonable?						
Answer options	No		50-50		Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	1	4	0

2.10 (Optional) Are there any changes or improvements you would like to suggest?

Answers:

1. See comments above plus it would be useful if a selection can be made for a certain issue, e.g. department, so that not all data will be loaded and presented.
2. Indicate that the report can be called from the tab „Alle Personen“

3. Task 2: Find missing sub-module registration

Find a missing sub-module registration in your program of study using the provided instructions and check whether the registration is really missing. If the report shows no missing registration in your program of study, please take another program of study.

3.1 Were you able to successfully complete the task?			
Answer options	No	Yes	No answer
Answer number	1	2	
Number of answers	1	4	0

3.2 What problems did you encounter?

Answers:

1. If a student is only registered for a repetition exam, then the report shows a missing registration for the sub-module. Consequence: The list is too long, it is not possible to check all entries. -> Most of the error messages are in fact no errors.
2. maw.WOM-V.2009 HS.001 -> Registration on the degree programme exists. Student xyz 100808
3. a.MB3 IDPerson 132552 a.MB3.2009 HS 001 has a sub-module registration on a.MB3-V.2009 HS 001. List seems to be incorrect.

3.3 Did you find the data quality report at the place where you expected the report to be?						
Answer options	No				Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	1	4	0

If not 100% Yes, where is it more suitable to place the report?

Answers:

1. (could also be under the tab „Anmeldungen“ but the current place is simpler)

3.4 Does the report contain enough information?						
Answer options	No				Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	1	3	1

If not 100% Yes, what information are you missing?

Answers:

1. Could contain information on which other sub-modules of the same module a student is registered (if the student is registered on others)

3.5 How simple/difficult was it to complete the task?						
Answer options	Simple				Difficult	No answer
Answer number	1	2	3	4	5	
Number of answers	3	0	2	0	0	0

3.6 What do you think about the waiting time for the report to open?						
Answer options	Very long		Adequate		Very short	No answer
Answer number	1	2	3	4	5	
Number of answers	0	1	3	1	0	0

3.7 How long did it take to complete the task?						
Answer options	< 1 min	1-2 min	2-3 min	3-4 min	> 4 min	No answer
Answer number	1	2	3	4	5	
Number of answers	0	1	3	1	0	0

3.8 Do you think that the time you needed to complete the task is reasonable?						
Answer options	No		50-50		Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	2	3	0

3.9 (Optional) Are there any changes or improvements you would like to suggest?

Answers:

1. Selection criteria should be possible before loading the list
2. Data seems to be incorrect because sub-module registration where found

4. Post task impression

4.1 Considering the two tasks performed, what do you think about the understandability of the new data quality functionalities? The data quality functionalities are …						
Answer options	Self-explanatory				In need of explanations	No answer
Answer number	1	2	3	4	5	
Number of answers	2	2	0	0	0	1

4.2 Are the used terms in the report understandable?						
Answer options	100% No				100% Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	1	3	1

If not 100% Yes, where are the problems?

Answers:

-

4.3 What is you general impression of the report that should help to assure data quality?						
Answer options	--	-	-/+	+	++	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	2	2	1

4.4 Do you think that the data quality functionalities will help you to find wrong or missing registrations more efficient?						
Answer options	No		50-50		Yes	No answer
Answer number	1	2	3	4	5	
Number of answers	0	0	0	0	3	2

4.5 (Optional) If you have any additional comments that you would like to make, please record them here.

1. We are all very skilled person using Evento, probably persons should be ask if the terms used in the reports are understandable, who are not Evento-Insiders. The idea to assure quality is super! Thank you :-)
2. For persons with adequate education, the reports are quite usable
3. Would it be possible to have a link from EXCEL back to Evento? The data in the reports are only controllable if the degree programme is well known. -> Who uses the report & who corrects the data? -> Are only data analysed from education or also further education? Grouping by semester

References

1. Crosby, P.B., *Quality Is Free: The Art of Making Quality Certain*1979: New York Mentor.
2. English, L.P., *Improving Data Warehouse and Business Information Quality*1999, New York, USA: Wiley Computer Publishing.
3. Redman, T.C., *Data: An unfolding Quality Disaster*, in *Information Management Magazine*2004.
4. (TDWI), T.D.W.I. *TDWI's Data Quality Report*. 2002; Available from: http://tdwi.org/research/2002/02/tdwis-data-quality-report.aspx?sc_lang=en.
5. Becker, W., et al., *Proposal for structure and detail of a EuroFIR Standard on food composition data*, 2008, EuroFIR Consortium.
6. Salvini, S., et al., *Guidelines for Quality Index Attribution to Original Data from Scientific Literature or Reports for EuroFIR Data Interchange, draft document*, 2010, EuroFIR.
7. Juran, J.M., *Juran's Quality Handbook*. Fifth Edition ed1951, 1999: McGraw-Hill.
8. Quality, A.S.o. *Basic Concepts*. [cited 2010; Available from: http://www.asq.org/glossary/q.html.
9. Garvin, D.A., *What Does 'Product Quality' Really Mean?* MIT Sloan Management Review, 1984. **26**(1).
10. Standardization, I.O.f., *ISO 9000:2005*, in *Standards for quality management systems*.
11. Deming, W.E., *Out of the Crisis*1986: MIT Press.
12. Ishikawa, K., *What is total quality control? The Japanese way* 1985, Englewood Cliffs, N.J.: Prentice-Hall.
13. *Merriam-Webster*. Available from: http://www.merriam-webster.com.
14. Drucker, P.F., *Managing in a Time of Great Change*1995, New York: Truman Talley Books/Dutton.
15. Bourdreau, A. and G. Couillard, *System Integration and Knowledge Management.* Journal of Information Systems Management, 1999. **16**(4): p. 1-9.
16. Eppler, M.J., *Managing Information Quality*. Second Edition ed2006: Springer Verlag.
17. Redman, T.C., *Data Quality for the Information Age*1996: ARTECH HOUSE, INC.
18. Olson, J.E., *Data Quality: The Accuracy Dimension*2003: Morgan Kaufmann Publishers.
19. Huang, K.-T., Y.W. Lee, and R.Y. Wang, *Quality information and knowledge*1998, New Jersey: Prentice Hall.
20. Greenfield, H. and D. Southgate, *Food Composition Data: Production, Management and Use*2003: FAO Rome.
21. Tayi, G.K. and D.P. Ballou, *Examining data quality.* Communications of the ACM, 1998. **41**(2): p. 54-57.
22. Madhavan, N.K., *A Framework for Achieving Information Integrity.* IS Audit&Control Journal, 1996. **II**: p. 30.
23. Kahn, B.K. and D.M. Strong, *Product and Service Performance Model for Information Quality: An Update*, in *Third Conference on Information Quality*, I.N. Chengalur-Smith and L. Pipino, Editors. 1998, MIT.
24. Eppler, M.J., *Qualitätsstandards – Ein Instrument zur Sicherung der Informationsqualität in Multimedia-Produktionen*, in *Qualitätssicherung in Multimedia-Projekten*1999, Springer Verlag.
25. Porter, M.E., *Towards a Dynamic Theory of Strategy.* Strategic Management Journal, 1991. **12**: p. 95-117.
26. Batini, C. and M. Scannapieco, *Data Quality, Concepts, Methodologies and Techniques*2006: Springer.

27. Wang, R.Y. and D.M. Strong, *Beyond Accuracy: Whata Data Quality Means to Data Consumers.* Journal of Management Information Systems, 1996. **12**(4): p. 5-33.
28. Wand, Y. and R. Wang, *Anchoring Data Quality Dimensions in Ontological Foundations.* Communications of the ACM, 1996. **39**(11): p. 86-95.
29. Rahm, E. and H.H. Do, *Data Cleaning: Problems and Current Approaches.* IEEE Data Engineering Bulletin, 2000. **23**(4): p. 3-13.
30. Storey, V. and R. Wang. *Modeling Quality Requirements in Conceptual Database Design*. in *Proceedings of the 1998 Conference on Information Quality*. 1998. MIT Boston.
31. Storey, V. and R. Wang, *Data Quality, Chapter 3: Extending the ER Model to Represent Data Quality Requirements*, ed. K.I. Series2001: Springer.
32. Hauser, J.R. and D. Clausing, *The House of Quality*, in *Hardvard Business Review*1988. p. 63-73.
33. Juran, J.M., *Juran on Quality by Design: The new Steps for Planning Quality into Goods and Services*1992: The Free Press.
34. Wang, R.Y., M.P. Reddy, and H.B. Kon, *Toward quality data: An attribute-based approach.* Decision Support Systems, 1995. **13**(3-4): p. 349-372.
35. Wang, R.Y., M. Ziad, and Y.W. Lee, *Data Quality, Chapter 3: Extending the ER Model to Represent Data Quality Requirements*, ed. K.I. Series2001: Springer.
36. Buneman, P., S. Khanna, and W.-C. Tan. *Why and Where: A characterization of Data Provenance*. in *8. International Conference on Data Theory (ICDT)*. 2001. London, UK.
37. Wang, R.Y. and S.E. Madnick. *A Polygen Model for Heterogeneous Database Systems: The Source Tagging Perspective*. in *16th International Conference on Very Large Databases (VLDB)*. 1990.
38. Wang, R.Y., M. Ziad, and Y.W. Lee, *Data Quality, Chapter 2: Extending the Relational Model to Capture Data Quality Attributes*, in *Data Quality*2001, Springer.
39. Scannapieco, M., et al., *The DaQuinCIS architecture: a platform for exchanging and improvingdata quality in cooperative information systems.* Information Systems, 2004. **29**(7): p. 551-582.
40. Buneman, P., A. Deutsch, and W.-C. Tan, *A deterministic model for semi-structured data*, 1998, University of Pennsylvania: Philadelphia.
41. Shankaranarayanan, G., R.Y. Wang, and M. Ziad. *IP-MAP: Representing the Manufacture of an Information Product*. in *5th International Conference on Information Quality*. 2000. Cambridge, MA, USA.
42. Scannapieco, M., B. Pernici, and E. Pierce, *IP-UML: TOWARDS A METHODOLOGY FOR QUALITY, IMPROVEMENT BASED ON THE IP-MAP FRAMEWORK*, in *Information Quality, Advances in Management Information Systems*, E.M.P. Richard Y. Wang, Stuart E. Madnick, Craig W. Fisher Editor 2005.
43. Würthele, V.G., *Dissertation: Datenqualitätmetrik für Informationsprozesse*, in *Department of Computer Science, ETH Zurich, Switzerland*2003.
44. Mihaila, G.A., L. Raschid, and M.-E. VIdal. *Using Quality of Data Metadata for Source Selection and Ranking*. in *3rd international WebDB Workshop*. 2000. Dallas, Texas, USA.
45. Rothenberg, J. *Metadata to Support Data Quality and Longevity*. in *First IEEE Metadata conference*. 1996. Silver Spring, Maryland, USA.
46. Naumann, F., U. Leser, and J.C. Freytag. *Quality-driven Integration of Heterogeneous Information Systems*. in *25th International Conference on Very Large Data Bases*. 1999. Edinburgh, Scotland: Morgan Kaufmann.

47. Pipino, L., L. Yang, and R. Wang, *Data Quality Assessment.* Communications of the ACM, 2002. **45**: p. 211.
48. Naumann, F., *Datenqualität*, in *Informatik Spektrum*2007, Springer: Heidelberg. p. 27-31.
49. Ballou, D.P., et al., *Modeling Information Manufacturing Systems to Determine Information Product Quality.* Management Science, 1998. **44**(4): p. 462-484.
50. Hinrichs, H., *Datenqualitätsmanagement in Data Warehouse-Systemen*, in *Informatik*2002, University of Oldenburg: Oldenburg.
51. Klier, M., *Metriken zur Bewertung der Datenqualität - Konzeption und praktischer Nutzen*, in *Informatik Spektrum*2007, Springer: Heidelberg. p. 223-236.
52. Even, A. and G. Shankaranarayanan. *Value-Driven Data Quality Assessment*. in *International Conference on Information Quality*. 2005.
53. Heinrich, B., M. Kaiser, and M. Klier. *Metrics for measuring data quality – Foundations for an economic data quality management*. in *ICSOFT 2007*. 2007. Barcelona.
54. Wang, R.Y., V.C. Storey, and C.P. Firth, *A Framework for Analysis of Data Quality Research.* IEEE Transactions on Knowledge and Data Engineering, 1995. **7**(4): p. 623-640.
55. Madnick, S.E. and R. Wang, *Overview and Framework for Data and Information Quality Research.* ACM Journal of Data and Information Quality, 2009. **1**(1): p. 1-22.
56. Schlotke, F., et al., *EUROFOODS Recommendations for Food Composition Database Management and Data Interchange*, 2000.
57. Holden, J., S. Bhagwat, and K. Patterson, *Development of a Multi-Nutrient Data Quality Evaluation System.* Journal of Food Composition and Analysis, 2002. **15**(4): p. 339-348.
58. EuroFIR. 2010; Available from: www.eurofir.org.
59. Schenker, R., *Diplomarbeit: Verwaltung von Nahrungsmittel-Analysedaten*, in *Institute of Computational Science*1992, ETH Zurich: Zurich.
60. Vonäsch, C., *Semesterarbeit: Portierung von ParaCelsus von dBASE auf ACCESS*, 1994, ETH Zurich: Zurich.
61. Miltwein, O., *Verwaltung von Nährwertdaten II: Entwurf und Implementierung eines Informationsserver*, in *Institute of Computational Science*1995, ETH Zurich: Zurich.
62. Schlotke, F. and L. Jacob, *User Manual iFoods*, 2001, ETH Zurich: Zurich.
63. Lee, R., *Benutzerhandbuch Schweizer Nährwertdatenbank Online-Version*, 2004, ETH Zurich: Zurich.
64. Jacob, S., M. Risel, and F. Schlotke, *Projekt Schweizer Nährwertdatenbank, Projektdokument: Dokumentation zur Testversion*, 1999, ETH Zurich: Zurich.
65. Management, A.X.S.S.o.D.B. and Systems, *Interim Report, 75-02-08.* FDT - Bulletin of ACM SIGMOD, 1975. **7**(2): p. 1-140.
66. Brodie, M.L. and J.W. Schmidt, *FInal Report of the ANSI/X3/SPARC DBS-SG Relational Database Task Group*, 1981.
67. Chen, P., *The entity-relationship model—toward a unified view of data.* ACM Transactions on Database Systems (TODS), 1976: p. 9-36.
68. Booch, G., J. Rumbaugh, and I. Jacobson, *Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series)*2005: Addison-Wesley Professional.
69. Kay, A.C., *The early history of Smalltalk*, in *The second ACM SIGPLAN conference on History of programming languages*1993, ACM: Cambridge, Massachusetts, United States. p. 69-95.

70. Hinterberger, H., *Data Quality Management in Collection-Based Research*, 2007, Institute of Computational Science, ETH Zurich.
71. Shewhart, W.A., *Statistical Method from the Viewpoint of Quality Control*1939: Dover Publications Inc.
72. Bacon, F., *Novum Organum*, 1620.
73. Imai, M., *Kaizen: The Key to Japan's Competitive Success*1986, New York: Mcgraw-Hill Higher Education.
74. Møller, A. and T. Christensen, *EuroFIR Web Services - EuroFIR Food Data Transport Package*, 2008, EuroFIR Consortium.
75. Mock, R., *Data Quality Analysis for Food Composition Databases*, in *Department of Computer Science*2011, ETH Zurich: Zurich.
76. Colombani, P. and K. Presser, *Usability testing of the single value module and recipe module of FoodCASE and recommendations for future use by EuroFIR AISBL*, in *EuroFIR European Food Information Resource Network of Excellence*2010, EuroFIR.
77. Bell, S., P. Colombani, and A. Fragodt, *Usability testing of EuroFIR eSearch facility – cycle 1*, 2009.
78. Kobel, V., *Planungsphase eines Labor-Usability-Tests für die erste Online-Version der Schweizerischen Nährwertdatenbank*, 2005, ETH Zürich/Zürcher Hochschule Winterthur: Zurich.
79. Hornbæk, K., *Current practice in measuring usability: Challenges to usability studies and research.* International Journal of Human-Computer Studies, 2006. **64**(2): p. 79-102.
80. Eidgenössisches Kriegs-Ernährungs-Amt, *Tabelle der Nährwerte der Lebensmittel*, 1944. p. 378-384.
81. Högl, O. and E. Lauber, *Nährwert der Lebensmittel*, S.L.E. Gesundheitsamt, Editor 1964: Bern. p. 713-753.
82. Gander, M., *Semesterarbeit: Entwicklung von PC-Diet*, 1991, ETH Zurich: Zurich.
83. Schlotke, F., *Diplomarbeit: Entwurf und Implementation eines interaktiven Diätverwaltungsprogramms für Diabetiker*, in *Institute of Computational Science*1994, ETH Zurich: Zurich.
84. Richter, D., *Verwaltung von Nährwertdaten I: Entwurf und Implementation von Datenstrukturen und Datenbankschnittstellen für den Datenaustausch*, in *Instiute of Computational Science*1995, ETH Zurich: Zurich.
85. Schläpfer, D., *Entwurf und Implementation von 3 sich ergänzenden Tools zur Generierung und Verwaltung von streng hierarchischen Codestrukturen und DAGS zur hauptsächlichen Anwendung im Lebensmittelbereich*, in *Institute of Computational Science*1996, ETH Zurich: Zurich.
86. Møller, A. and J. Ireland, *LanguaL 2000 The LanguaL Thesaurus*, in *COST Action 99*2000: Brussel.
87. Møller, A. and J. Ireland, *LanguaL 2008 The LanguaL Thesaurus*, in *EuroFIR*2007: Brussel.
88. Jacob, S. *Präsentation der Schweizer Nährwertdatenbank*. in *Wissenschaftlichen Symposium*. 2002. Kinderspital, Bern.
89. *The Sixth Framework Programme in brief*. European Commission 2002.
90. Hodapp, T., *Masterarbeit: Design and Implementation of the Swiss Online Food Composition Database*, in *Institute of Computational Science*2006, ETH Zurich: Zurich.

Curriculum Vitae

Particulars

Karl Presser
Steinackerstr. 15
CH-8450 Andelfingen

Phone	+44 44 301 2 301
Mobile	+44 79 456 57 78
E-mail	karl.presser@premotec.ch
Date of Birth	03.12.1976
Native place	Zurich
Civil status	married, 2 kids

Postgraduate Activities

15.03.07 - now

ETH Zurich, Institute of Information Systems:
Doctoral student
A Requirement-Oriented Data Quality Model and Framework of a Food Composition Database System (FoodCASE)

01.03.07 - now

Premotec GmbH:
Partner
IT Services for the information systems Evento and FoodCASE

08.03.04 - 28.02.07

Balzano Informatik AG:
Software engineer (T-SQL, C#.NET und ASPX),
Chief developer 10 employees,
Product manager,
Project leader,
Member of the extended executive board
Product development and commercial launch of an application for time table optimization using evolutionary algorithms (AI)

Activities concurrent with the Studies

22.07.02 - 05.03.04

Balzano Informatik AG:
Software engineer (T-SQL, C#.NET und ASPX),
Product development of an application for time table optimization using evolutionary algorithms (AI)

01.05.02 - 31.07.02	ETH Zurich, Institute of Information Systems: **Research assistant** Efficient transaction Management for XML documents
01.10.01 - 28.02.02	ETH Zurich, Institute of Information Systems: **Teaching assistant** Advanced Informatics course: SML, Prolog, Eiffel und Java
01.03.01 – 30.04.01	yodoba AG: **Software developer** Java implementation, CGI/Perl programming, Macromedia Flash development
18.04.00 - 03.08.01	Citrin, Feisthammel und Partner: **Software developer** Different projects, mainly implementation in Java using Servlets, JDBC and Sybase
02.07.99 - 10.09.99	H. Weidmann AG: **Supporter und script developer** Setup of the mailing system and the mail server, Changeover from Mailworks to MS Exchange 5.5, Windows scripting
01.03.98 - 31.07.98	EKZ Zürich: **PC-Supporter**
22.04.97 - 10.10.97	Zürich Insurance: **Assistant** Application development with Lotus Approach, data input, computer installation

---

Education

Sep 2007 – Dec 2007	Language course: Writing English for Science Level B2, 26 lessons, 2 Credits
Oct 1997 – Mar 2004	Study in computer science at ETH Zurich, Degree: Dipl. Informatik-Ing. ETH,
Oct 2001 – Feb 2002	Course in didactics with practical monitoring
Aug 1990 – Jan 1997	Kantonsschule Zürich-Freudenberg Degree: Matura Typus B

Languages

German	native Level C2 in European Language Portfolio
English	5 years of education, study and diploma thesis Level C1 in European Language Portfolio
French	6 years of education Level B1 in European Language Portfolio
Latin	6 years of education Level A2 in European Language Portfolio