

2 Research Design

In this chapter, we first provide a definition for the terms “semantic technologies” and “ontologies” to provide a basic understanding for the following chapters. After that, we define the research goals and research questions. This chapter concludes with the research methodology that has been applied to generate the answers to the research questions and achieve the research goals.

2.1 Semantic Technologies and Ontologies

Originally, the use of the term “semantics” as a noun or “semantic” as an attribute was limited to the academic fields of

- (1) semiotics, i.e. “the study of signs and symbols” (McComb, 2004, p. 9),
- (2) linguistics i.e. “the study of language” (McComb, 2004, p. 8).

In semiotics, semantics is the name for studying the relationships between signs and meaning (cf. Hoyningen-Huene, 1998, p. 251). In linguistics, it is “the study of meaning in language” (Riemer, 2010, p. i). In computer science, the term “semantics” has been used in the context of programming languages since the 1960s, with work by Floyd (Floyd, 1967) being the most prominent initial reference. In this context, “semantics” stood for the formal analysis of the execution of programs. With the advent of artificial intelligence as a field, the notion of “semantics” in computer science got broader, including the representation of terminological and factual knowledge by data structures (cf. Sowa, 2014).

In 2001, Berners-Lee et al. described the vision of a “Semantic Web” as an evolution of the World Wide Web into an ecosystem in which information would be represented and interlinked in ways accessible to computers and not just human consumers of a visual rendering (cf. Berners-Lee et al., 2001). This contribution has triggered a broad usage of the term “semantics” as study of representation, sharing, and processing of meaning in computer systems (cf. Hitzler, 2008, p. 13). Semantic technology is then the broad range of approaches for contributing to that end. Therefore, this thesis sees “semantic technologies” as technical approaches that facilitate or make use of the interpretation of meaning by machines. A prerequisite for machine interpretation of

knowledge is the collection and storage of relevant knowledge in a way that machines can understand. This can be achieved via knowledge representation languages such as the Resource Description Framework (RDF) (Manola & Miller, 2004) and the Web Ontology Language (OWL) (Bechhofer et al., 2004).

The term "ontology" is frequently used in the context of semantic technology, and there are many different options to define it (cf. Hepp, 2008b, pp. 3-6). It originates from philosophy and expresses the study of existence (cf. Gasevic et al., 2006, p. 45). In computer science, we can understand an ontology as "an explicit specification of a conceptualization" (Gruber, 1993, p. 199). "Conceptualization" can be seen as "an abstract model of some phenomenon in the world which identifies the relevant concepts of that phenomenon" (Alexiev et al., 2005, p. 16). "Explicit" means that these concepts and their restrictions are explicitly represented within an ontology (Alexiev et al., 2005, p. 16). Grimm et al. extend this definition by additional characteristics of ontologies in the context of knowledge representation and define it as "a formal explicit specification of a shared conceptualization of a domain of interest" (Grimm et al., 2007, p. 69). Based on these definitions, we understand ontologies as a formal and sharable means to explicitly model some real-world phenomenon for machine-readable knowledge representation. A detailed discussion about the characteristics of ontologies will be provided in section 4.1.

2.2 Research Goal

This thesis aims to investigate the usefulness of ontologies to support data quality management activities. Ontologies promise the concise representation of domain knowledge with its entities and relationships in a machine-readable way (cf. Grimm et al., 2007). In the context of data quality management, ontologies could provide the following benefits:

Knowledge reuse: The management of data quality requires capturing business knowledge in the form of logical rules that define the characteristics how to recognize incorrect data (cf. Loshin, 2001, p. 179). According to Loshin this knowledge "reflects the ongoing operations of a business" (Loshin, 2001, p. 185) and the same knowledge may also be relevant for other business areas (cf. Loshin, 2001, p. 286). For example, data requirements, such as the definition of credible values for a certain data element, could not only be used for data quality measurement, but also for the verification of

new data entries or imported data (cf. Loshin, 2001, p. 9). In many systems, such knowledge is often hidden within application logic. In order to make such knowledge reusable and transparent to business users, it is necessary to move it out of the application logic into an explicit representation (cf. Loshin, 2001, p. 279). One possible solution to preserve and publish data knowledge in a reusable way could be the structured representation of that knowledge via ontologies. E.g. data requirements could be represented with help of an ontology and linked to the accordant data element. Moreover, the data element could be linked to the data owner and the business tasks in which the data is being processed to support organizational tasks of data quality management.

Semantic reconciliation: Due to the expressivity of ontologies, it is possible to precisely define the semantics of data. When requesting information, we often ask ambiguous questions that may lead to completely different answers depending on the interpretation of an individual. With the use of ontologies, we are able to explicitly represent the concise semantics of data and annotate formal and informal definitions. This may lead to a reduction of misunderstandings and misinterpretations (cf. Madnick & Zhu, 2006).

Creation of a shared understanding: Explicit knowledge representation of a domain in form of an ontology facilitates communication about different viewpoints and thereby supports the creation of a shared understanding about a domain (cf. Fensel, 2001, p. 2; Hepp, 2008b, p. 5; Uschold & Gruninger, 1996, p. 8f.) Moreover, it is possible to enrich the elements of an ontology by textual definitions. If maintained precisely, such human-readable definitions may additionally reduce ambiguity and, therefore, support a common understanding (cf. Hepp, 2008b, p. 13).

Content integration: Several research approaches discuss the usefulness of ontologies for data and content integration within and across enterprises (cf. Alexiev et al., 2005; Fensel, 2002; Kokar et al., 2004; Niemi et al., 2007; Perez-Rey et al., 2006; Skoutas & Simitsis, 2007; Souza et al., 2008; Wache et al., 2001). The distribution of data and quality-relevant knowledge requires superior integration capabilities when managing data quality. Data quality management may, therefore, benefit from the integration capabilities of ontologies.

Deduction of implicit knowledge: Due to the explicit representation of concepts and relationships including their semantics within ontologies, it is possible to infer implicit

knowledge, e.g. through reasoning engines (Hepp, 2008b, p. 15). This novel feature of ontology-based information systems may open up additional capabilities for business cases, such as data quality management.

2.3 Research Questions

In order to evaluate the potential benefits of semantic technologies, we develop a prototype that utilizes ontologies to support data quality management tasks. We address the following research questions (RQ).

RQ1: What kind of data quality problems exist?

Data quality management aims to improve data quality. In order to investigate the usefulness of ontologies in this domain, we first need to know the types and causes of data quality problems that may occur in information systems. Hence, we initially examine the characteristics of data quality problems.

RQ2: Which activities have to be performed during data quality management?

In order to identify the required capabilities which may be supported by semantic technologies, we have to analyze the data quality management process for the tasks that have to be performed to manage data quality.

RQ3: Which knowledge has to be represented to support data quality management?

Based on the identification of activities which are part of data quality management and the types of data quality problems, we need to identify the knowledge required to perform these tasks.

RQ4: How can we represent knowledge relevant for data quality management to reduce manual work?

The identified knowledge shall be represented with modeling elements of an ontology language. The ontology shall thereby be processable by both humans and machines to reduce manual efforts for data quality management.

RQ5: How can we utilize knowledge for data quality management represented within ontological structures?

Once the data quality management knowledge is captured and represented in ontological structures, we need to find ways to use this knowledge for performing data quality management tasks. Thus, artifacts are needed to process the represented knowledge to serve data quality management tasks.

In order to satisfy the reusability of the findings, this thesis aims to provide domain independent solutions to the above research questions.

2.4 Research Methodology

According to Hevner et al. the information systems discipline is dominated by two research paradigms: behavioral science and design science. “The behavioral-science paradigm seeks to develop and verify theories that explain or predict human or organizational behavior. The design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts” (Hevner et al., 2004, p. 75). This thesis focuses on the design science paradigm to develop an innovative framework based on semantic technologies, called the Semantic Data Quality Management framework (SDQM), which aims to improve and extend the capabilities required for data quality management by providing efficient mechanisms to store and retrieve quality-relevant knowledge. Part of the framework is an ontology for sharing and utilizing quality-relevant knowledge, which we will refer to as the DQM Vocabulary in the following. The development procedure of SDQM is, therefore, based on two development methodologies: (1) the design science research methodology (DSRM) process by Peffers et al. (Peffers et al., 2008, p. 52ff.) for the development of the general framework of SDQM, and (2) the ontology engineering methodology by Uschold and Gruninger (Uschold & Gruninger, 1996) for the development of the DQM Vocabulary. Both methodologies will be explained in the following sections.

2.4.1 Design Science Research Methodology

The design science research methodology (DSRM) is based on an analysis of similarities between several different design methodologies to identify a consensual way to perform design science research (cf. Peffers et al., 2008, p. 52). In detail, DSRM has the following six processes (Peffers et al., 2008):

- (1) Problem identification and motivation
- (2) Define the objectives for a solution
- (3) Design and development
- (4) Demonstration
- (5) Evaluation
- (6) Communication

We chose to adjust the original DSRM by procedures and tools that have been proven to be pragmatic means during the development of the framework. For instance, we use a motivating scenario to illustrate the problem domain (cf. Uschold & Gruninger, 1996) and a requirements register to keep track of SDQM's requirements throughout its development. Figure 4 shows an adjusted version of the DSRM as chosen for this thesis including the generated outputs of the process steps.

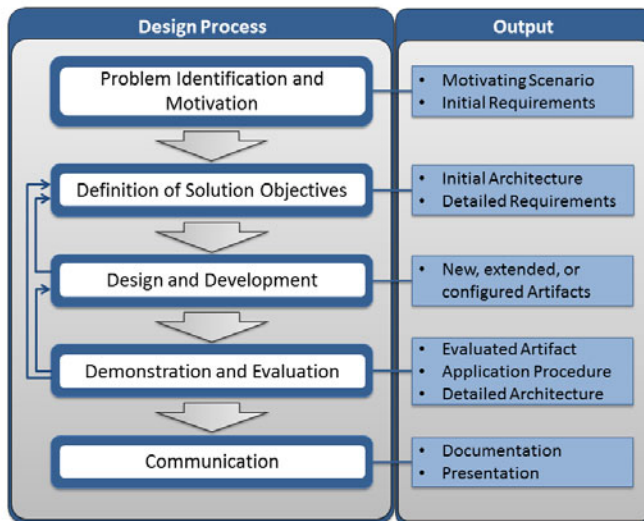


Figure 4: Design methodology as applied in this thesis (cf. Peffers et al., 2008)

The pure sequential execution of DSRM may not be possible in many cases due to incomplete knowledge (cf. Peffers et al., 2008, p. 56). For example, important technical requirements or defects in the developed artifacts may be initially discovered during the evaluation phase and, therefore, require to change the requirements register as part of the “Definition of solution objectives” phase and cause a change of the artifact in the development phase. Therefore, we added iteration paths that have occasionally been used during this thesis project to return to previous process steps. In the following, we will describe each process of the adjusted DSRM as applied in this thesis.

Problem identification and motivation: The design science research process typically starts with the identification of the research problem and the justification of its relevance (cf. Peffers et al., 2008, p. 52f.). In this thesis, we initially describe the general problem and its economic relevance in chapter 1. We further specify the problem by defining and motivating the research goals in section 2.2 and research questions in section 2.3. Since the research goals and research questions by themselves are not sufficient for the development of an artifact that shall be used in practical settings, we further specify the problem definition by deriving initial requirements from a motivating scenario in chapter 6. The motivating scenario is based on a practical problem setting in which the artifact shall be used (cf. Uschold & Gruninger, 1996, p. 29f.). Besides the practice-oriented requirements from the motivating scenario, the initial requirements also encompass research requirements derived from the research goals of this thesis.

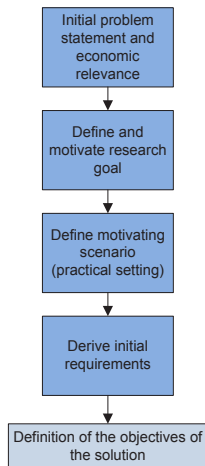


Figure 5: Problem identification and motivation process as applied in this thesis

Definition of solution objectives: Solution objectives are the objectives that the developed solution shall fulfill. Based on the initial requirements, we design a high level architecture with components that shall meet the requirements that were defined in the previous process. We then describe the purpose of each component and map the initial requirements to the accordant components of the solution architecture. At this point, new requirements may arise due to increasing knowledge about the problem domain. The new requirements should, therefore, be added to the initial requirements during the “review initial requirements” process step. The execution of this process differs from the original process as described in (Peffers et al., 2008, p. 55) as we already start to sketch a solution architecture and map requirements to define the objectives of the solution components. We argue that our procedure is more pragmatic and reduces complexity, since our objectives are defined as concrete deliverables based on the initial requirements which encompass the research requirements. Finally, we already start to analyze and collect related work to identify reusable artifacts.

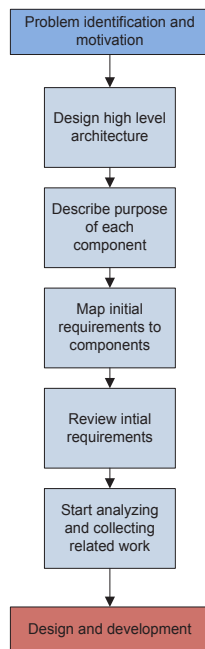


Figure 6: Process for the definition of solution objectives as applied in this thesis

Design and development: Before we start to actually develop the artifact, we first analyze whether existing artifacts can be reused for the components of our framework. The analysis is based on the description of components and its accordant requirements from the previous process. In cases of more than one reusable artifact, the most appropriate artifact has to be chosen. In cases where an existing artifact only partially fulfills the requirements, the artifact may be extended before its reuse. In cases where no suitable existing artifact can be found, a new artifact has to be developed from scratch according to the component's requirements. Moreover, the components of the architecture usually have to be integrated into a single framework and initially configured as part of the development process. Figure 7 illustrates the "Design and development" process as applied in this thesis.

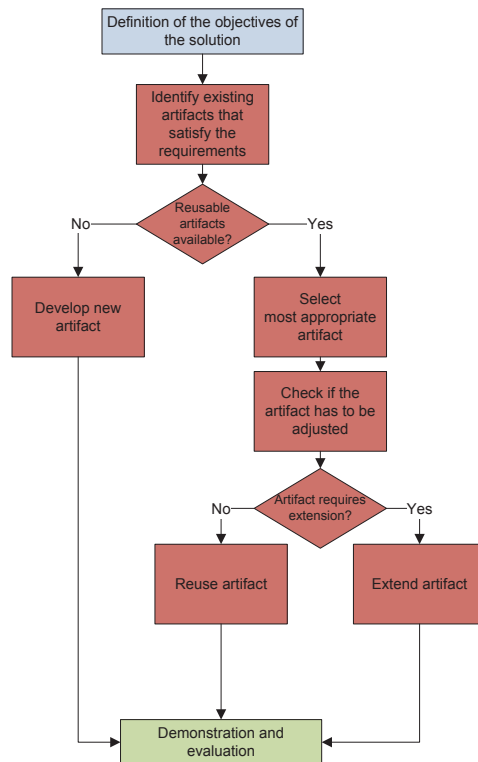


Figure 7: Design and development process as applied in this thesis

Demonstration and evaluation: We combined the activities “demonstration” and “evaluation” (which are originally separated in DSRM) to one process due to the tight interaction of demonstration and evaluation. Demonstration is the application of the developed artifact to the problem domain (cf. Peffers et al., 2008, p. 55). Evaluation identifies how well the developed artifact fulfills its intended use (cf. Peffers et al., 2008, p. 56). Therefore, it is typically performed based on information that has been collected during the demonstration (cf. Peffers et al., 2008, p. 56). In this thesis project, we perform the demonstration and evaluation process in two stages. After the development of the artifact has been finished, we initially demonstrate and evaluate the artifact as a prototype in a controlled environment. After the prototype has been evaluated successfully, we continue the demonstration and evaluation in a real-world environment as a practical use case. In cases where the evaluation identifies unacceptable limitations, we may need to return to the design and development process to enhance the artifact. For this project, we chose two major use cases: (1) data quality management of material master data (section 9.2) and (2) data quality management of Semantic Web data (section 9.3) to investigate the applicability of the artifact in both environments.

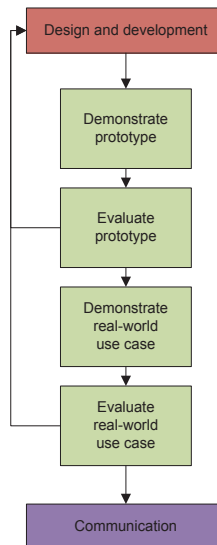


Figure 8: Demonstration and evaluation process as applied in this thesis

Communication: The DSRM ends with the communication of the research project which is performed by this thesis. Additionally, parts of this project have been published

Data Quality Management with Semantic Technologies

Fürber, C.

2016, XXVII, 205 p. 63 illus., Softcover

ISBN: 978-3-658-12224-9