Measuring Quality of Database Schemas by Reviewing – Concept, Criteria and Tool

Olaf Herden

Oldenburg Research and Development Institute
for Computer Science Tools and Systems (OFFIS),
Escherweg 2, 26121 Oldenburg, Germany
olaf.herden@offis.de

Abstract

This paper deals with measuring the quality of conceptual database schemas. We specify a meta model for defining and measuring quality aspects. We also define a process for configuring reviews. Furthermore, we present a couple of criteria for reviewing object oriented conceptual schemas and suggest how to measure them. Then we list some criteria not suited for reviews. We have realized a small prototype for documenting and administrating reviews which is sketched briefly.

1 Introduction

For building high quality information systems the three level approach has been established in the community (see e. g. [CBS98]). It starts with conceptual modeling, defined as the process of constructing a model of the information needed, independent of *all* physical considerations. After conceptual modeling follows logical database design by constructing a model of the information based on a specific data model, but independent of a particular database management system or any other physical considerations. The last step is physical database design, the task of producing a description of the implementation of the database on secondary storage.
Although relational databases are most widespread these days, ER (entity relationship) modeling seems not to be the best approach for conceptual modeling because the postulation to be independent of *all* physical considerations is violated, conceptual and logical database design are often mixed in practice. As improvement different kinds of extensions to pure ER models have been developed ([Teo90, Tha00]) as well as object oriented notations are used to be really independent at the conceptual stage of developing databases. We are assuming object oriented modeling at the conceptual level for the rest of the paper.
According to ISO 8402 quality assurance is defined as "all planned and systematic activities implemented within the quality system and demonstrated as needed,

to provide adequate confidence that an entity will fulfill requirements for quality". Quality assurance can be subdivided into constructive (all technical, organisational and psychological actions to ensure quality features) and analytical (all actions for inspecting and evaluating products during software development) quality assurance. Constructive quality assurance is extensively given by using the three level approach as design methodology and its application leads to a database implementation that meets the technical requirements well. But to really meet the users' requirements and to build a system being maintainable analytical quality assurance in terms of an explicit review of the conceptual database schema is necessary.
In this paper we present a meta model for defining criterions and reviews for measuring the quality of database schemas and we describe the process of configuring and performing reviews. Furthermore, we present a documentation tool implementing this meta model and process. Moreover, we list some concrete criteria for object oriented database schemas and motivate why other criteria are not relevant in this context.
The remainder of the paper is organized as follows: in section 2 we describe the meta model for measuring database schemas. Section 3 lists concrete criteria for measuring. The implementation for administrating measures and reviews is sketched in section 4. Section 5 reflects related work. The paper ends with a conclusion and an outlook to future work in section 6.

2 Concept for Measuring Schema Quality

Our meta model for reviews of database schemas is depicted in figure 1.

The rating of a schema is done by criteria (objects of class *Criterion*). A criterion is defined by its name, a description and a metric. Each criterion is measured by a method (objects of class *MeasuringMethod*). A measuring method has a degree of automation (*yes*, *no*, *partly*) and in case of the value *yes* the attribute *autoDescription* contains the kind of automatic measuring. For every measuring method the *objectivity* can be determined by one of the values *objective*, *quite objective*, *subjective* and *quite subjective*.
The criteria can be used as basis for configuring a *ReviewType*, defined as a weighted set of criteria. By setting the *minimum* attribute, an optionally threshold can be defined for a criterion within a review type. If the value of the criterion falls below the threshold value then the whole review fails, i.e. this criterion is a K.O. criterion. These values can be found in the association class *ReviewTypeCriterion*. Furthermore, this meta class contains the attribute *metrics* which puts the metrics in concrete terms for this review type.
One *ReviewType* can serve as basis for many *Reviews*. The values determined during such a review are stored as objects of the class *MeasuredValue*. The self–association of meta class *Review* allows to define a sequence of reviews.
Subject of a review is a conceptual database schema (instances of meta class *DatabaseSchema*).
The process of working with this meta model is depicted in figure 2: the reviewer or project manager can define a new review type by using existing or defining new

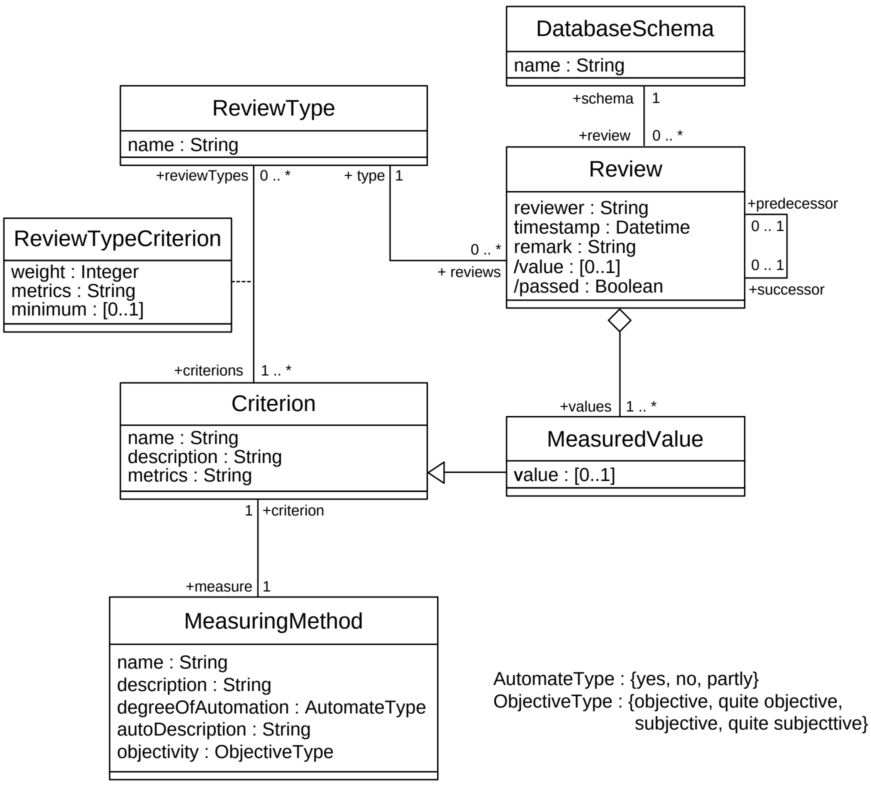

Figure 1: Meta model of reviews

criteria and working in his knowledge about the domain and the project (step I). This template of reviews can be used for a sequence of reviews (step II). In between times the quality of the schema can be improved iterativly (step III).

3 Criteria of Schema Quality

In this section we present some quality criteria for conceptual object oriented database schemas. These criteria are possible instances of class *Criterion* from the meta model in figure 1.
They are described by their properties:

- *Description* gives a brief explanation of the criterion
- *Relevance* describes the criterion's significance in the concrete context
- *MeasuringMethod* suggests a method for determinating the value
- *Metric* is the measure for the criterion
- *Degree of automation* estimates the ability of automating the measuring
- *Objectivity* rates the criterion's objectivity

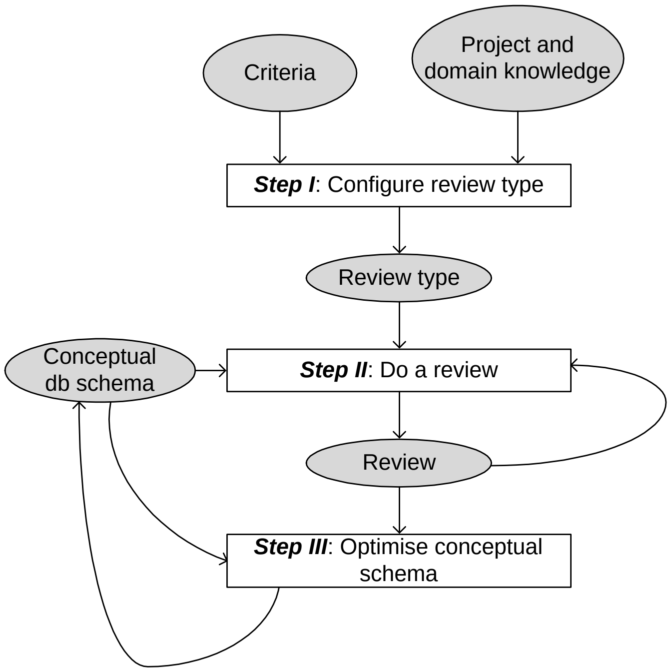

Figure 2: Process of defining and doing reviews

Table 1 shows the criteria; in the row *metric* the abstract term *aspect* is used. An aspect can be any schema element (classes, attributes, connections), its concrete granularity is determined when configuring a review type (by setting attribute *metric* of meta class *ReviewTypeCriterion*). It depends on the concrete demand within the project or the organisation the review type is configured for.

Criterion	(Technical) correctness
Description	Are all modelled aspects correct with respect to the requirements and terms of the domain?
Relevance	Essential for any schema.
Measuring	By review of an expert with deep knowledge in the domain.
Metric	Correct technical aspects in proportion to all technical aspects.
Degree of automation	No.
Objectivity	Objective.

Criterion	(Technical) Consistency
Description	Are all modelled aspects free of contradictions?
Relevance	Together with correctness this criterion is very important for the acceptance of the schema.
Measuring	By review of a technical expert. He has to analyse every modelled aspects and to compare these with respect of technical inconsistencies subsequently.
Metric	Consistent technical aspects in proportion to all technical aspects.
Degree of automation	No.
Objectivity	Objective.

Criterion	(Technical) Relevance
Description	Are technically relevant all aspects of the schema?
Relevance	Significant with respect to the users' acceptance.
Measuring	By review of a technical expert. He has to determine all relevant aspects and to compare them with those modelled in the schema.
Metric	Relevant technical aspects in proportion to all technical aspects.
Degree of automation	No.
Objectivity	Objective.

Criterion	Scope
Description	Is the scope of the schema appropriate?
Relevance	Criterion is significant with respect to the users' acceptance. The scope is relative with regard to the goal of the project. E.g., if the goal of a project is a solution for only one department, then the requirements are not as high as in the case of defining a reference schema.
Measuring	By review of a technical expert. He has to determine all aspects required with regard to the projects' goal and to compare them with those modelled in the schema.
Metric	Modelled aspects in proportion to all aspects required.
Degree of automation	No.
Objectivity	Objective.

Criterion	Level of detail
Description	Is the level of detail of the schema appropriate?
Relevance	Criterion is relevant with regard to users' acceptance.
Measuring	By review of a technical expert. He has to determine the desired level of detail for every aspect and to compare them with the schema.
Metric	Aspects modelled in the right level of detail in proportion aspects.
Degree of automation	No.
Objectivity	Objective.

Criterion	Completeness
Description	Is the schema complete (with regard to the requirements)?
Relevance	Criterion is relevant with regard to users' acceptance.
Measuring	Can be derived from the two former criteria scope and level of detail.
Metric	Minimum of the values for scope and level of detail.
Degree of automation	No.
Objectivity	Objective.

Criterion	Minimality
Description	Is the schema modeled compactly and without redundancies?
Relevance	Important because conceptual schemas should be precise.
Measuring	By review of a technical expert. He has to check whether there are aspects modelled more than once.
Metric	Redundant aspects in proportion to all modelled aspects.
Degree of automation	No.
Objectivity	Objective.

Criterion	Ability of integration
Description	Is the schema suited for organisation comprehensive standardizations or modelings?
Relevance	Depends on the context of the project or organisation. The more an enterprise is involved in an economical association or the more its international orientation, the more relevant is the ability of integration.
Measuring	Determining, whether the names of aspects are common terms in the branch or wether they are internationally accepted.
Metric	Common aspects in proportion to all modelled aspects.
Degree of automation	Partly, e.g. by using a thesaurus.
Objectivity	Objective.

Criterion	Readability
Description	Are explained all terms used in the schema?
Relevance	Very important, especially for the dialogue with technical experts and for people coming along to the project later.
Measuring	Review of documentation. Determining, whether for every aspect exists a meaningful annotation.
Metric	Rate of annotated aspects in proportion to all modelled aspects.
Degree of automation	Partly, with regard to existence of annotations the measuring can be done automated, with regard to quality of annotations human interaction is necessary.
Objectivity	Quite objective.

Table 1: Criteria for reviews of object oriented schemas

As a conclusion, we can emphasise that measuring of most criteria can not be automated. On the other hand all the mentioned criteria are at least quite objective. At this point we assume that the reviewer has deep knowledge and he is not prejudiced. As a consequence constructive methods *how* the expert has to do his review can be given.
Table 2 turns the attention to some qualitity aspects being relevant to the database development process but not suited for reviews. The reason is given in the right column of the table.

Criterion	Why not suited for a review?
Extensibility (Can new requirements be integrated into the schema easily?)	Very relevant for every schema, because 80% of the cost of an information system are maintenance cost. Nevertheless extensibility can rarely be determined for a conceptual schema. We only can state that object orientation as data model for the conceptual level is a good basis for extensibility.
Reusability (Can (parts of) the schema be used for new schemas?)	Although reusability is very important to probably save effort in later projects it is not appropriate for reviews. Similar to extensibility, object orientation as data model for the conceptual level is a good basis for reusability.
Schema management (Are well ordered administration, use and update of the schema possible?)	Schema management is very relevant, especially with regard to comprehensibility of development. It should be provided by the modeling tool or its repository. During a review only the correct use of version numbers can be checked.
Transformability (Is it possible to transform the schema to the logical layer?)	Transformability is important because conceptual schemas only make sense when being transformable. A generic transformation algorithm should be part of the underlying methodology.
Normalisation (Does the schema fulfill normalisation rules?)	This criterion is very relevant in the relational data model. So when using the ER model or an extension as conceptual model checking of normalisation rules should be part of a review. But it is not relevant to object oriented schemas because there do not exist any commonly accepted normal forms for object oriented database schemas.

Table 2: Criteria not suited for reviews

4 Implementation

The architecture of the implementation is shown in figure 3 whereat the grey coloured components are already realised. The meta model from figure 1 is implemented as repository database. A persistence layer encapsulates the methods for creating and manipulating objects. On top of this there is a graphical user interface providing dialogues for the reviewer to define criterions, review types and document reviews (see figure 4 for an example). Moreover, documented reviews can be exported as HTML documents (see figure 5 for an example).
The architecture is open: the middle layer offers an interface providing the manipulation of objects to external systems. These external systems can realize (semi–)automated measurings.

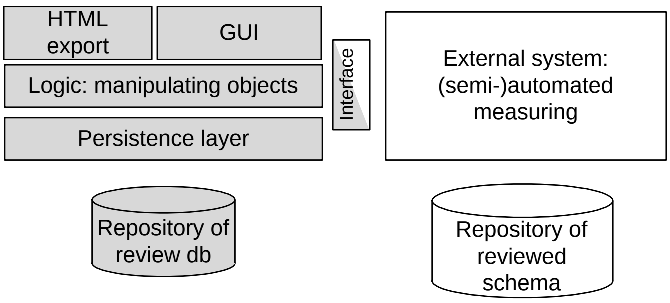

Figure 3: Architecture of implementation

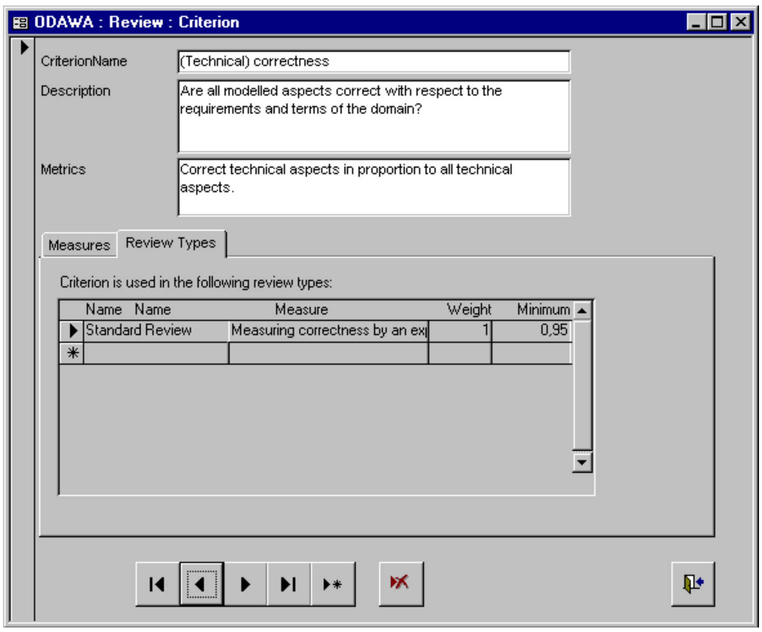

Figure 4: GUI of the tool

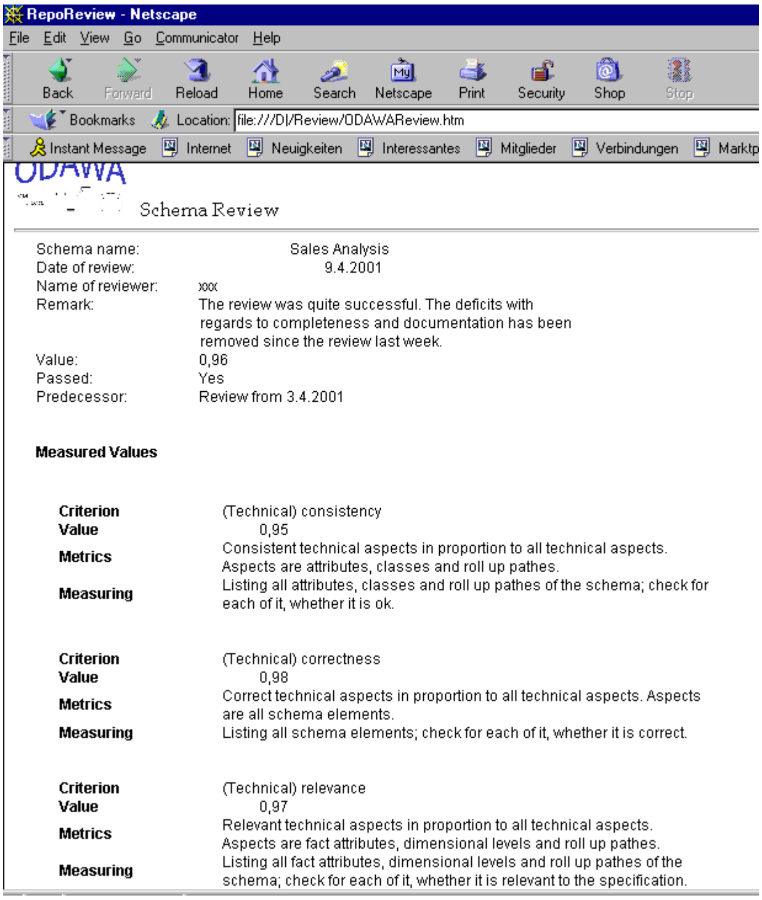

Figure 5: HTML output of the tool

We have used the method and the tool within the scope of our research project ODAWA (OFFIS Tools for Data Warehousing) [Her00, Her01]. For conceptual modeling we are using MML (Multidimensional Modeling Language) [HH99, HH00], a language providing object oriented as well as multidimensional modeling aspects. We have composed a special kind of review type by selecting some criteria from table 1. To put *aspects* in concrete forms as granularity we have selected the schema elements *fact attributes*, *dimensional levels* and *roll up pathes* which are all very important in the context of data warehouses. The metric for technical completeness is calculated by

$$\frac{cnt(fact\ attributes)+cnt(dimensional\ levels)+cnt(roll\ up\ pathes)}{cnt(relevant\ fact\ attributes)+cnt(relevant\ dimensional\ levels)+cnt(relevant\ roll\ up\ pathes)}$$

where *cnt* is a function counting the number of named aspects.

As practical procedure the expert did a brainstorming of all relevant fact attributes, dimensional levels and roll up pathes with regard to the scope of the project and his domain knowledge.

By doing such reviews some inaccuracies of the conceptual design have been detected in three iterations of step II and step III in the process model in figure 2. Especially the design of roll up pathes being relevant for data analysis in the future

have been refined during this process.
We think that finding these errors at a very early stage of information system development can improve the conceptual schema and save much development cost.

5 Related Work

There have been some approaches concerning with quality of database schemas in the past, an overview is given in [Bur98].
One famous work is [BCN92] (adopted by many others e.g. [Ris92, Ris93]). Some criteria that must be fulfilled to call a database schema being of high quality are listed in the publications. Each of these criteria (completeness, correctness, minimality, expressiveness, readability, self explanation, extensibility and normality) is described by prosa and some examples are given.
[Bru91, RG94] enumerate specific rules for developing high–quality relational data models. Focus of these works is the practice, some procedural methods for developing data models are included.
In [MS94] quality aspects are defined that should be measured by metrics. To compare different database schemas the measured values can be weighted. The aim is to find the best solution by evaluating alternative solutions for a task. The aspects of quality are simplicity, completeness, flexibility, integration, understandability and implementability.
In software engineering, reviews are known as task for checking documents produced during the software development process (see e.g. [GG93, Som00]). The goal of a review is to detect strengths and weaknesses of the reviewed document.
[CM00] lists some quality criteria for database schemas and postulates an explicit review, but it is restricted to the ER model.
All the listed approaches do not allow to define new review types by putting together criteria or adding new ones. Moreover, most of them are not tool–supported. The cited contributions are summarized in table 3.

	Quality criteria	Measuring/ Metrics	Reviews	Tool	Database model
Bruce, Reingruber and Gregory [Bru91, RG94]	●	○	○	○	ER
Batani et. al. [BCN92]	●	○	○	○	ER
Rishe [Ris92, Ris93]	●	○	○	●	ER
Moody/Shanks [MS94]	●	●	(●)	○	ER
Claxton and McDougall [CM00]	●	○	●	○	ER
This paper's approach	●	●	●	(●)	ER/OO

Table 3: Comparison of related work

6 Conclusion and Future Work

In this paper we have motivated why an explicit review of the conceptual schema is useful during database development. We have shown a meta model for criteria, review types and reviews. Then we have listed criteria being relevant for object oriented database schemas and we have shown how users can build review types by using these criteria. After sketching the implementation, we briefly reflected related work.

As future work, we want to design and implement more algorithms for automated measuring. Furthermore, the tool should be improved to support the reviewer when measuring (quite) objective values.

References

[BCN92] Carlo Batani, Stefano Ceri, and Shamkant Navathe. *Conceptual Database Design. An Entity–Relationship–Approach*. Redwood City, 1992.

[Bru91] Thomas A. Bruce. *Designing Quality Databases With Idef1X Information Models*. Dorset House Publishing, 1991.

[Bur98] William C. Burkett. Database schema design quality principles. *http://www-scf.usc.edu/ wcb/dmq/dmqmain.html*, 1998.

[CBS98] Thomas Connolly, Carolyn Begg, and Anne Strachan. *Database Systems*. Addison Wesley, 2. edition, 1998.

[CM00] John C. Claxton and Peter A. McDougall. Measuring the quality of models. *TDAN – www.tdan.com*, 3(14), 2000.

[GG93] Tom Gilb and Dorothy Graham. *Software Inspection*. Addison–Wesley, 1993.

[Her00] Olaf Herden. A Design Methodology for Data Warehouses. In Annika Hinze and Bart-Jan Hommes, editor, *Proceedings of the 7th Doctoral Consortium on Advanced Information Systems Engineering*, pages 13–23, Stockholm (Sweden), June 2000.

[Her01] Olaf Herden. *Homepage ODAWA*. http://odawa.offis.uni-oldenburg.de, 2001.

[HH99] Arne Harren and Olaf Herden. Conceptual Modeling of Data Warehouses. In *Proc. of Demonstration and Posters E/R99*, Paris (France), November 1999.

[HH00] Olaf Herden and Arne Harren. An Extensible Language for Conceptual Modeling of Data Warehouses. In *Poster CAiSE'00*, Stockholm (Sweden), June 2000.

[MS94] Daniel L. Moody and Graeme G. Shanks. What Makes a Good Data Model? Evaluating the Quality of Entity Relationship Models. In Pericles Loucopoulos, editor, *Proceedings 13th Intl. Conference on Entity Relationship Approach (ER94)*, pages 94–111, Manchester (Great Britain), December 1994. Springer.

[RG94] Michael Reingruber and William W. Gregory. *The Data Modeling Handbook: A Best–Practice Approach to Building Quality Data Models*. John Wiley and Sons, 1994.

[Ris92] Naphtali Rishe. *Database Design: The Semantic Modeling Approach*. Mc Graw–Hill, 1992.

[Ris93] Naphtali Rishe. A Methodology and Tool for Top–Down Relational Database Design. *Data and Knowledge Engineering*, 10(1):259–291, 1993.

[Som00] Ian Sommerville. *Software Engineering*. Addison–Wesley, 2000.

[Teo90] Toby J. Teorey. *Database Modeling and Design: The Entity-Relationship Approach*. Morgan Kaufmann, 1990.

[Tha00] Bernhard Thalheim. *Entity–Relationship Modeling*. Springer, 2000.