Quality Aspects and Decision Points:

how is quality in the modeling language ORM guaranteed when identifying fact types

Author:
Eddy Klomp
s0413348
Information science

PREFACE

The last phase before receiving my bachelor-diploma, consist of writing a bachelor thesis. After receiving my diploma, I hope to specialize myself in the field of information modeling, where this paper is all about. Information modeling is according to me an essential part when building information systems. My opinion is that a good information systems forms the basis of reacting to the internal and external environment of a company both efficiently and effectively. Therefore the process of building an information model must be taken seriously, although its a difficult thing to do. In my opinion, this process is not always taken seriously and the outcome can be dramatic. Although this paper doesn't encompass the whole subject of information modeling, I hope it can be a small step in creating efficient en effective information systems.

My bachelor thesis couldn't be established without the help of Dr. P. (Patrick) van Bommel. He helped me with choosing the subject of my bachelor thesis and guided me in realizing this bachelor thesis. He helped me with tips about structuring this paper and placing this paper to an intellectual higher level, for instance, by suggesting to write something about the impact of certain decisions. He also made suggestions about 'broadening' or 'deepening' my research. The last suggestion was the outcome. I also would like to thank Dr. S.J.B.A. (Stijn) Hoppenbrouwers, for recommending literature and giving tips about doing scientific text analysis. I also liked to thank Prof.Dr. E. (Erik) Proper for recommending Xfig as application for drawing (ORM-)schema's.

Eddy Klomp

CONTENTS STATEMENT

CHAPTER 1

INTRODUCTION

Problem Statement

Research Question
How is the quality in the modeling language ORM guaranteed when identifying the fact types?

The steps for identifying the fact types are presented in the book of T. Halpin: *Information Modeling and Relational Databases: from conceptual analyses to logical* design. Identification of the fact types are the first three steps of the procedure for designing a conceptual schema, known as the conceptual schema design procedure (CSDP).[1] The whole procedure consist of seven steps. We take a closer look at the first three steps of this procedure and analyze why certain decisions are made in these steps, what the quality aspects are of these decisions and what the impact is. This is done by looking at diverse quality aspects described in several articles (van Bommel et al.. 2007, Krogsty et al.. 2006, Krogstie and Jørgenson 2003, Proper 2006, Stamper 1973).

Objective

A personal reason to answer the above stated research question is my personal interest in information modeling. An other reason is that there isn't an overview concerning the quality of choices in the design procedures in ORM. With this paper the quality aspects of the first three design procedures in ORM are made clear.

This paper has both scientific- and social relevance. It has scientific relevance because it describes why certain decisions are made in a specific modeling language. It can be a tool to create new modeling languages or to optimize existing modeling languages. This paper has social value, because it can be useful when making choices about applying a modeling language in a certain situation and it helps to make decisions when designing models, for instance by looking at the impact of a certain decision.

Theoretical framework

The research question is analyzed from the perspective of information science. The field of information modeling is the central theme. In the design procedures all kinds of quality aspects are taken into account (syntax, semantics, pragmatics, etc.), which are also studied in the field of communication science.
Knowledge territory:

- Information science
  - Information modeling
    - Quality aspects of information modeling
      - Quality aspects in the ORM design procedure
        - Guarantee quality when identifying fact types in ORM

1 Halpin, page 59

CHAPTER 2

METHOD

To answer the research question one must make the domain and its variables clear. The domains are the first three steps in the modeling language ORM. The variables are the decisions in these steps and the quality aspects of these decisions. This paper is considered to be descriptive as it makes clear the quality aspects in the first three steps of the conceptual schema design procedure of the modeling language ORM.

In this paper four sub-questions are answered, these are:

1. What kind of decisions are made when identifying fact types?
2. Why certain decisions are made when identifying fact types?
3. What are the quality aspects of these decisions?
4. What is the impact of these decisions?

The research question how the quality of procedures in the modeling language ORM can be guaranteed can only be answered if one knows what and why certain decisions are made in these procedures. If one knows these questions one can look at the quality aspects of these decisions. Eventually one would like to know the impact of a certain decision, based on its quality aspects. If these questions are answered, one knows how quality in the modeling language ORM is guaranteed in the first three steps of the conceptual schema design procedure, which is the research question to be answered.

To answer the first two question, one has to take a closer look in the book of Halpin and discover what and why certain decisions are made in its first three steps. To answer the third question one has to analyze quality measures, by studying several articles, and apply them on the results of the first two questions. During these analyses we use ORM to clarify some examples. We also use ORM to build a framework regarding these choices and its quality aspects. The impact of these decisions are measured by looking at the outcomes when decisions aren't applied.

Before deriving conclusions from the answers of the sub-questions, one has to make clear what the results are derived from former research. This is done by doing a quantitative text analyses (Roberts 2000) The results are presented in Appendix A. These results give an indication how many times the variables are being described, by looking at the sum off all words being present per document. The relevance of the article is derived from it and the articles are being studied by looking at the abstract, the conclusion and in most cases the paragraph with the related word in the document.

In chapter four, five and six we will make clear all decisions described in the first three steps of the conceptual schema design procedure. Per decision, we describe:

- an introduction;
- Halpin's view based on this decision;
- the linkage between these decisions and their quality aspects;
- an ORM-schema of this linkage;
- a clarification of this linkage;
- the impact of this decision.

CHAPTER 3

FRAMEWORK

When designing or studying a modeling language, several 'decision points' (or choices) are made. The modeler would like to know what the quality aspects are of these decision points. To help him or her we present a framework (figure 3.1). This framework consist of 'decision points', and 'quality aspects'. There are linkages between these aspects. These linkages are presented in the conclusion, where the overall framework is described.

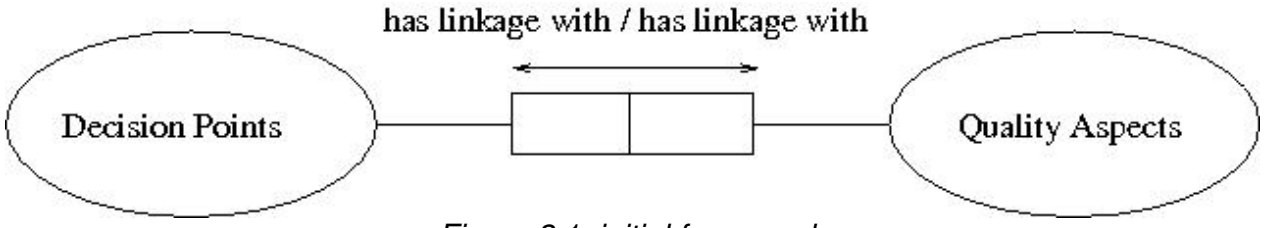


Figure 3.1: initial framework

As we would like to present a framework, one would like to know what these 'decision points' and 'quality aspects' are. Therefore we must describe the instances of these 'decision points' and 'quality aspects'. We also zoom in on the modeling language ORM, as we would like to know what the quality aspects are of the first three steps of the Conceptual Scheme Design Procedure (CSDP) in ORM, which is our research question.

The instances of the 'decision points' in ORM are described in Halpins book: *Information Modeling and Relational Databases: from conceptual analyses to logical design.* He describes these 'decision points' within steps of a design procedure (CSDP). Before applying this procedure, one has to divide the Universe of Discourse into manageable subsections and after applying this procedure, one has to integrate the sub-schema's into a global conceptual schema. The CSDP is realized in seven steps, which can be realized sequentially[2]:

1. Transform familiar information examples into elementary facts, and apply quality checks.
2. Draw the fact types and apply a population check
3. Check for entity types that should be combined, and note any arithmetic derivation
4. Add uniqueness constraints, and check for logical derivations.
5. Add mandatory role constraints, and check for logical derivations.
6. Add value, set comparison, and sub-typing constraints
7. Add other constraints and perform final checks.

Our main focus is on the first three steps of this procedure, because these three steps describe the identification of fact types.[3]

It is interesting to ‘zoom’ in these first three steps and to find out why certain decisions are made. These decisions will be presented here as 'decision points':

- Step 1:
  - Elementary facts
  - Clarify entities
  - Predicates

2 Halpin, page 59

3 Halpin, page 59

    - Using inverse predicates if predicates are the same
    - Textual Language Representations
    - Splitting multiple facts which leads to loss of information
    - Feedback
    - Synonyms
    - Modeling the current situation
- Step 2:
    - Vague predicates
    - Drawing reference modes
    - Populate
    - The purpose of formalizing
    - Objectification
- Step 3:
    - Value subtyping
    - Combine entities
    - Derivation rules
    - Include drawing of derivation rules
    - Eager evolution
    - Different entities with same kind of information

The instances of 'quality aspects' are derived from several articles describing quality of modeling (van Bommel et al.. 2007, Krogsty et al.. 2006, Krogstie and Jørgenson 2003, Proper 2006, Stamper 1973). Each article describes the quality aspects different and some are even inconsistent with each other. Some articles are more vague about quality aspects then other articles, which make it all difficult to combine. I tried to pick the neatest and most sensible description of quality aspects and tried to combine them, which resulted in these instances.

- Physical quality: the physical representation of artifacts of the model. It also include the use of fonts, graphics and representation medium (van Bommel et al.. 2007, Proper 2006, Stamper 1973)
- Empirical quality: comprehensibility of the model in terms of size, complexity, the number of symbols, the layout for graphs and readability (van Bommel et al.. 2007, Proper 2006).
- Syntactic quality: conformity to the syntax of the modeling language (van Bommel et al.. 2007, Proper 2006, Krogsty and Jørgenson 2003).
- Semantic quality: how well the model represents the domain in its completeness and validness (Krogsty and Jørgenson 2003, Krogsty et al.. 2006).
- Domain quality: how well the domain fits some desired situation (van Bommel et al.. 2007).
- Pragmatic quality: The interpretation of a model by humans and machines. (Krogsty and Jørgenson 2003, Krogsty et al.. 2006, Stamper 1973). This can be split into:
    - Quality of socio-cognitive interpretation: how an individual or group interprets the model, in view of how the model was intended to be interpreted by one or more of its modelers (van Bommel et al.. 2007).
    - Quality of technical interpretation: how a tool or group of tools interpret the model , in view of how the model was intended to be interpreted by one or more of its modelers (van Bommel et al.. 2007).
- Organizational quality: how the model, can track back earlier organizational goals (Krogsty and Jørgenson 2003, Krogsy et al.. 2006).
- Knowledge quality: how well actual knowledge matches knowledge need (van Bommel

et al.. 2007).
- Social quality: the level of agreement about the model among stakeholders (individuals or groups) about the statements of the model (van Bommel et al. 2007, Krogsty and Jørgenson 2003, Krogsy et al.. 2006).
- Ethical quality: the conformance of the model and its creation process to government/societal laws (Proper 2006, Stamper 1973).

A framework with quality aspects in modeling, where first identified by Lindland et al., who introduced three quality aspects: syntactic, semantic and pragmatic (Lindland et al. 1994). Moody et al. have done an empirical study to prove the validness of this framework, with a positive outcome (Moody et al. 2002).

I followed more or less the QoMo framework (van Bommel et al.. 2007) as it provides clear descriptions of quality aspects.[4] I didn't follow the description of the pragmatic quality, because the description of Krogsty (Krogsty et al.. 2006) is according to me, closer to the intended meaning of pragmatics, described by Stamper (Stamper 1974). According to me the quality of socio-cognitive and technical interpretation are part of pragmatic quality. When pragmatic quality in this article is used, it is the combination of both quality of socio-cognitive and quality of technical interpretation.

Proper identifies social quality as being conform social / governmental laws (Proper 2006) and Krogsty identifies social quality as the agreement of knowledge and interpretation (Krogsty 2003), where the latter is closely linked to pragmatic quality. Propers pragmatic quality is almost the same as Krogsty's social quality. Propers identification of quality conform social / governmental laws is also important. Therefore I follow Krogsty, concerning social quality and choose a new quality aspect called ethical quality, which describes Propers social quality.

Moody gives a review of research in conceptual model quality and identifies some theoretical and practical issues, which need to be addressed. They conclude that researchers and practitioners should work together to establish a common standard (or standards) for conceptual model quality. Therefore quality standards need to be defined. It is highly likely there will be multiple quality standards. For example, there is a difference between enterprise-level modeling and application-level modeling (Moody 2005). It is very true quality standards need to be defined, but it is very unlikely there will be total consensus among researches. Maybe the quality definitions presented in this paper will be a step towards establishing quality standards for a conceptual model. However, the purpose of this paper is different.

The purpose of this paper is to link the instances of the 'decision points', presented in the first three steps of the Conceptual Schema Design Procedure (CSDP), with the 'quality aspects' described in this chapter. An example of a linkage could be the following: the manner in which predicates are presented is important, because there has to be agreement among stakeholders about the statements of the model. So there is a link between 'predicates' and 'social quality' (which is the case and will be described in chapter 4). An overall linkage will be presented in the conclusion.

4 This framework is also based on the article described by Krogsty et al.. 2006, which is based on the SEQUAL framework described by Lindland (Lindland 1994)

CHAPTER 4

STEP 1: TRANSFORMATION OF INFORMATION INTO ELEMENTARY FACTS

Elementary facts

If we desire to transform information into elementary facts, one has to derive all kinds of information from the Universe of Discourse and transform them into elementary facts. One might ask himself why the transformation process is important. To answer this question it is important to know what facts really are and what the word *elementary* really means.

Halpin gives us some indication of what the answer might be. About facts Halpin says: "the system is to treat the assertion as being true of the Universe of Discourse".[5] This indicates that the system actually doesn't care about whether this is the case or not. It is like a soldier who takes orders from his commander, whether he likes it or not. About elementary facts Halpin says: "an elementary fact asserts that a particular object has a property, or that one or more objects participate together in a relationship. The adjective 'elementary' indicates that the fact cannot be 'split' into smaller units of information that collectively provide the same information as the original. Elementary facts typically do not use logical connectives (e.g. not, and, or, if) or logical quantifiers (e.g. all, some)".[6] The following sentences are not elementary facts:

- *The Employees with first name 'Ann' and 'Bob' work at Department with name 'Finance'.*
- *The Employees with first name 'Ann' or 'Bob' work at Department with name 'Finance'.*
- *The Employee with first name 'Ann' does not work at Department with name 'Finance'*
- *If the Employee with first name 'Ann' works for Department with name 'Finance' then the Employee with first name 'Bob' works for Department with name 'Finance'.*
- *All Employees are working for Company with name 'Phoenix'.*
- *If some Employees are working for Company with name 'Phoenix' then that Employee is a 'non-smoker'.*

A correct elementary fact could be the splitting of the first sentence, which are described in figure 4.1:

- *The Employee with first name 'Ann' works at Department with name 'Finance'.*
- *The Employee with first name 'Bob' works at Department with name 'Finance'.*

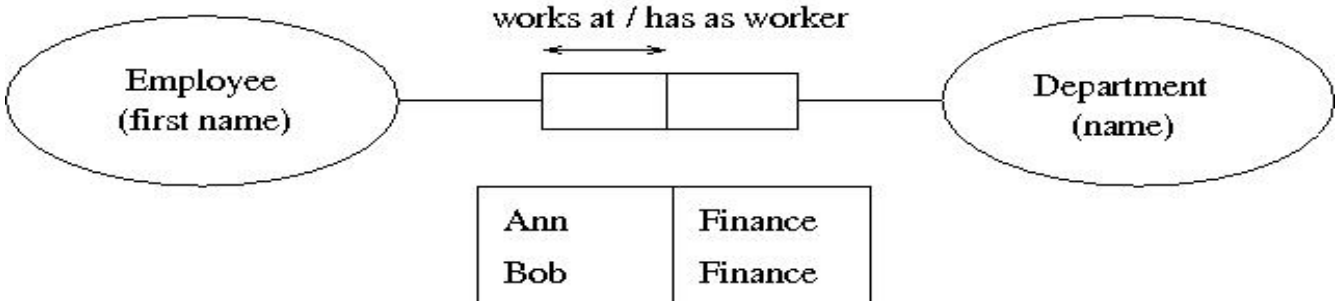


Figure 4.1: Bob and Ann working at the Finance department

5 Halpin, page 60

6 Halpin, page 61

The fact that systems don't care about whether the fact is true or false indicates that there is a distinction between *syntactic* and other quality checks: a system will only look at the *syntactic quality,* which means for example: all facts must be described in a syntax which a system can understand. This doesn't necessarily mean only elementary facts must be used, however most databases only allow elementary facts. Prolog (programming in logic), for example, can use logical connectives. The reason for using only elementary facts when modeling is, according to Halpin, because of practical reasons: "most databases don't allow non-elementary facts to be stored conveniently, are incapable of making relevant interferences and most commercial applications have no need to store such information".[7] To guarantee the *quality of technical interpretation* (most databases only interpret elementary facts), one must make the decision that only elementary facts are allowed, so this decision is based on the *quality of technical interpretation*. As you will see with most decisions, these decisions also influences the syntax of a modeling language. In this case one can say the decision that only elementary facts are allowed influences the conformity of a sentence to the syntax, which is a *syntactic quality* aspect (figure 4.2).

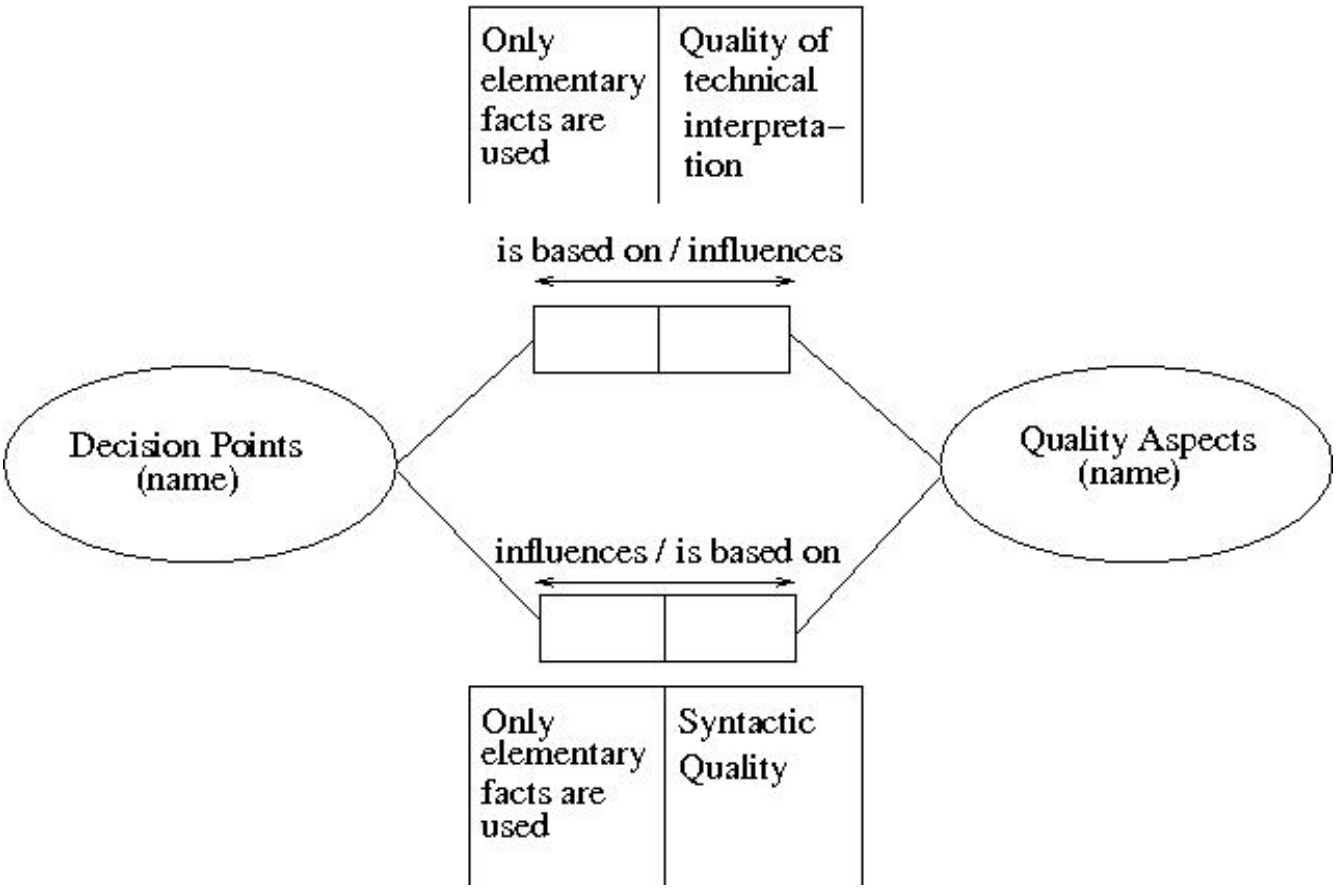


Figure 4.2: Only elementary facts are used is based on quality of technical interpretation. This decision influences the syntactic quality

7 Halpin, page 61

The usage of elementary facts, can say something about whether the quality of technical interpretation and the syntactical quality are good or bad (or in between). It is dependent on the size of the domain and the presence or absence of elementary facts, to make any judgment about the syntactical quality and the quality of technical interpretation.

The consequence to use facts, which aren't elementary is that databases can't interpret facts and the process of modeling can be difficult, with an uncertain outcome. If one has to use tools like Prolog, the usage of elementary facts is not necessary. The impact of using elementary facts is therefore dependent on whether a technical tool is used or not, and which technical tool this is. The impact can be high if all sentences must be changed, because the technical tool demands it.

Clarify entities

If we look closer at the previously described sentence (figure 4.1): *"The Employee with first name 'Ann' works at Department with name 'Finance'"*, we can see that the entity, "*The Employee with first name 'Ann'"*, is clearly identified. If one says: *"'Ann' works for 'Finance'"* you might conclude that Ann is working for the Finance department, which is true, but you can also conclude that Ann is working for the ministry of Finance, which isn't the case. Therefore one has to clarify entities, to diminish ambiguity.

Halpin says: "entities must be clearly identified by special kinds of definite descriptions."[8] He suggests three definite descriptions:

1. Entity type;
2. Reference mode
3. The value.

The entity type is the specification of the kind of entity being referred to. In our example *"The Employee"* is the entity type. Consider the sentence: "*'Ann' works for 'Finance'"*. Without the entity type, *"Ann"* could also be a robot. By adding "*The Employee 'Ann'*", it suggests that Ann is an employee. The reference mode is the manner in which the value refers to the entity. In our example *"first name"* is the reference mode. The sentence *"The Employee 'Ann' works for Department 'Finance'"* is pretty clear. However, there might be some confusion about who *"Ann"* might be. It can be a first name, but it can also be a surname. By adding *"The Employee with first name 'Ann'"* it makes things more clear.

Halpin gives us some reasons why entities must be clarified. This is done because humans may misinterpret. Also information systems are not able to add context and may misinterpret. So the decision to clarify entities is based on both interpretation aspects (technical and social-cognitive), which can be described as *pragmatic quality*. This decision also influences the *syntactical quality*, because, like the previous example, it influences the conformity of a sentence to the syntax (figure 4.3).

8 Halpin, page 62

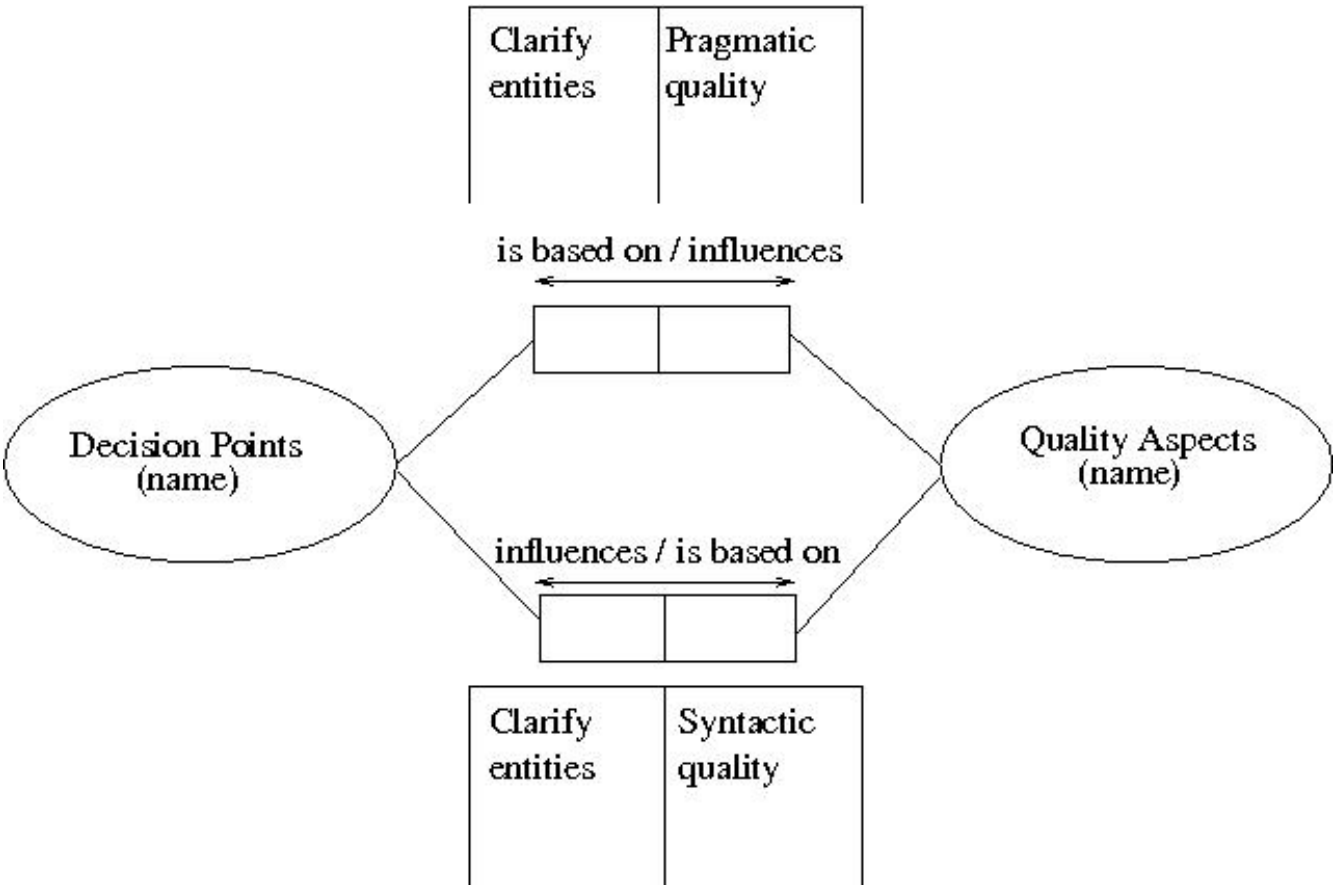


Figure 4.3: The clarification of entities is based on the pragmatic quality. This decision influences the syntactic quality

The clarification of entities can say something about whether the syntactical quality and the pragmatic quality are good or bad (or in between). As with the previous example, it depends on the size of the domain, the presence or absence of good and bad clarifications, to make any judgment about the pragmatic and syntactic quality.

If one doesn't clarify its entities, humans and machines may misinterpret the information, which can lead to confusion, disagreement, delay or even quarrels. Machines may even demand clarification of entities to work, so you will be forced to clarify your entities. It can also influence the modeling process negatively, as for example it may be unclear what the entities are. If the value itself clarifies the entity and the reference mode, it still is dangerous if this action isn't done, although it might be possible. This is the case only if the model won't be used in the future and the model isn't too complex. The impact not to clarify entities could be great, so clarification of entities are of high recommendation.

Predicates

In the above sentences we only used binary predicates. Predicates are used to describe the role entities play. It's clear that in the sentence: *"The Employee with first name 'Ann' works at Department with name 'Finance'",* the role the entity "*The Employee with first name 'Ann'"* plays that it *is working* for the Finance department. This is a binary predicate, as Ann is working for the Finance department, and the Finance department has as a worker Ann (figure 4.5). This last predicate, *has as worker*, is not present in the sentence. Although lots of predicates are binary there can be a unary predicate or an n-ary predicate. The following sentence has a unary predicate: *"The Employee with first name 'Ann'* works"(figure 4.4) and the following sentence has a n-ary predicate: *"The Employee with first name 'Ann' works for the Department with name 'Finance' during the Year 1999.*" (figure 4.6) In this last sentence n = 3. The sentence can also be read from right to left: *"During the year 1999, the Department with name 'Finance' has as a worker the Employee with first name 'Ann'".*

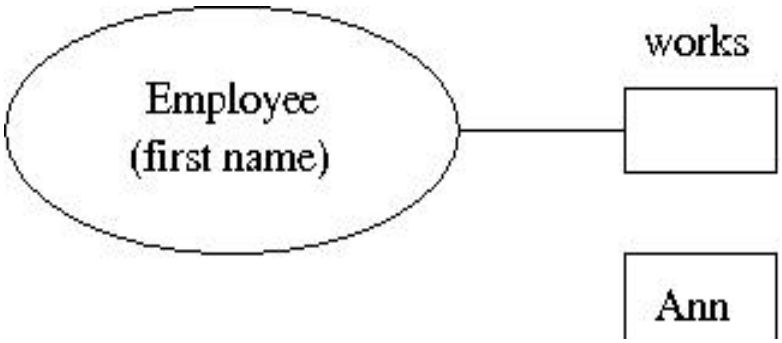


Figure 4.4 Unary predicate

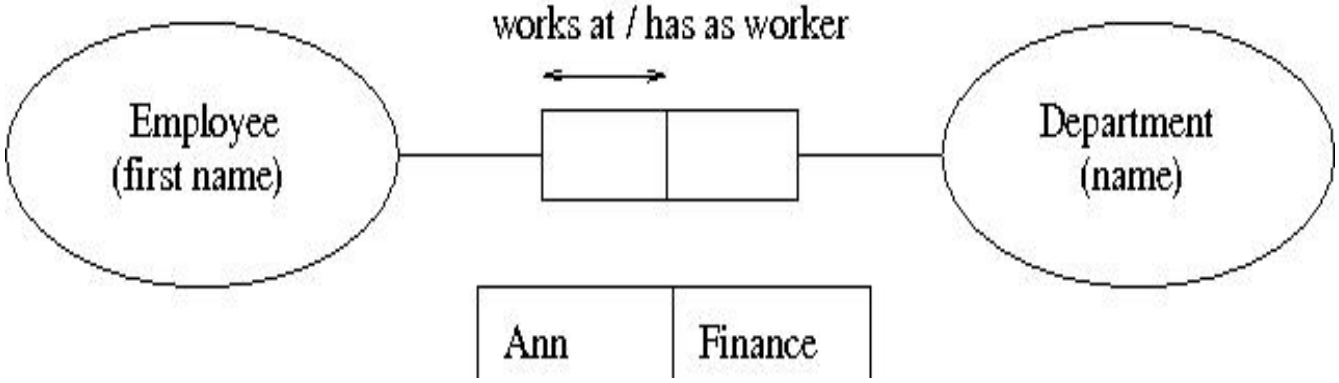


Figure 4.5: Binary predicate

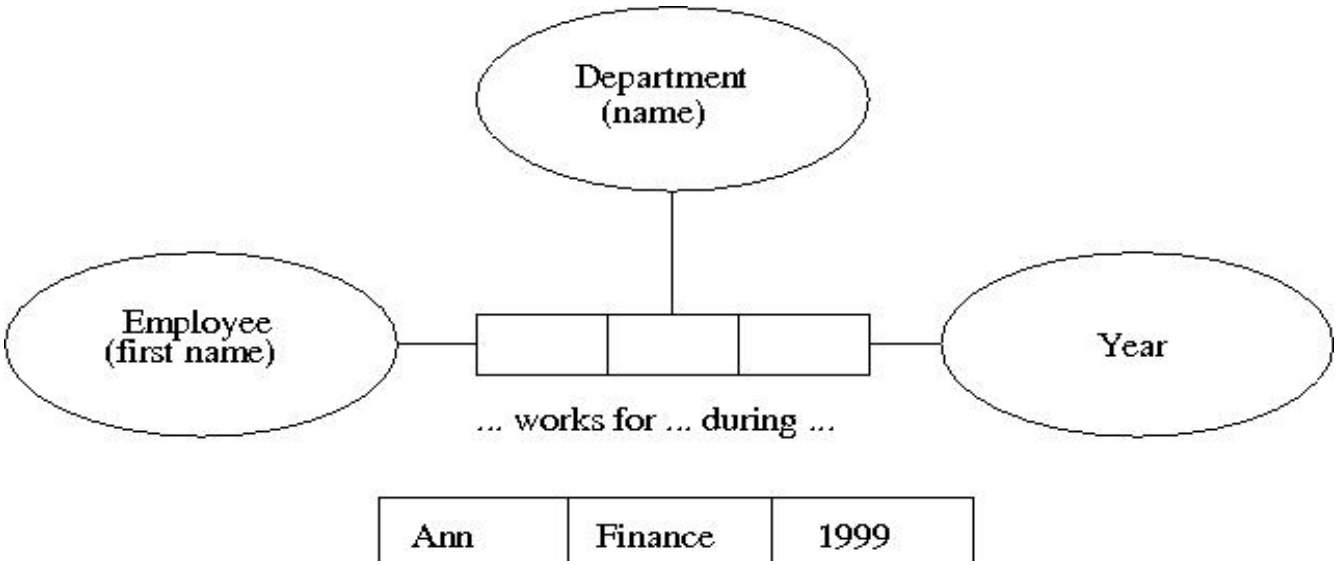


Figure 4.6: N-ary predicate (N = 3)

Halpin describes that semantically the above stated binary and n-ary sentences are the same as their inverse, but syntactically they are different. Therefore Halpin suggest allowing the inverse predicate in binary sentences.[9] The following sentence is syntactically correct: *"The Employee with first name 'Ann' is working for / has as worker the Department with name 'Finance'"*. The reason for doing this is according to Halpin to help communication and simplify constraint specification. About n-ary sentences Halpin says: "There are many possible orderings, but only one is displayed at a time".[10]

The improvement of communication can lead to an improvement of interpretation and agreement. Here, the decision to include the inverse predicate is based on both *the quality of socio-cognitive interpretation,* and the *social quality* (based on the level of agreement). Halpin describes that n-ary predicates are presented in one possible ordering.[11] Why this is the case is not described, but it is clear that this might affect the complexity of a sentence, which is an *empirical quality* aspect. The decision to include the inverse predicate influences the *semantic quality* of the sentence, because it influences the conformity of a sentence to the syntax (figure 4.7)

9 Halpin, page 65

10 Halpin, page 66

11 Halpin, page 67

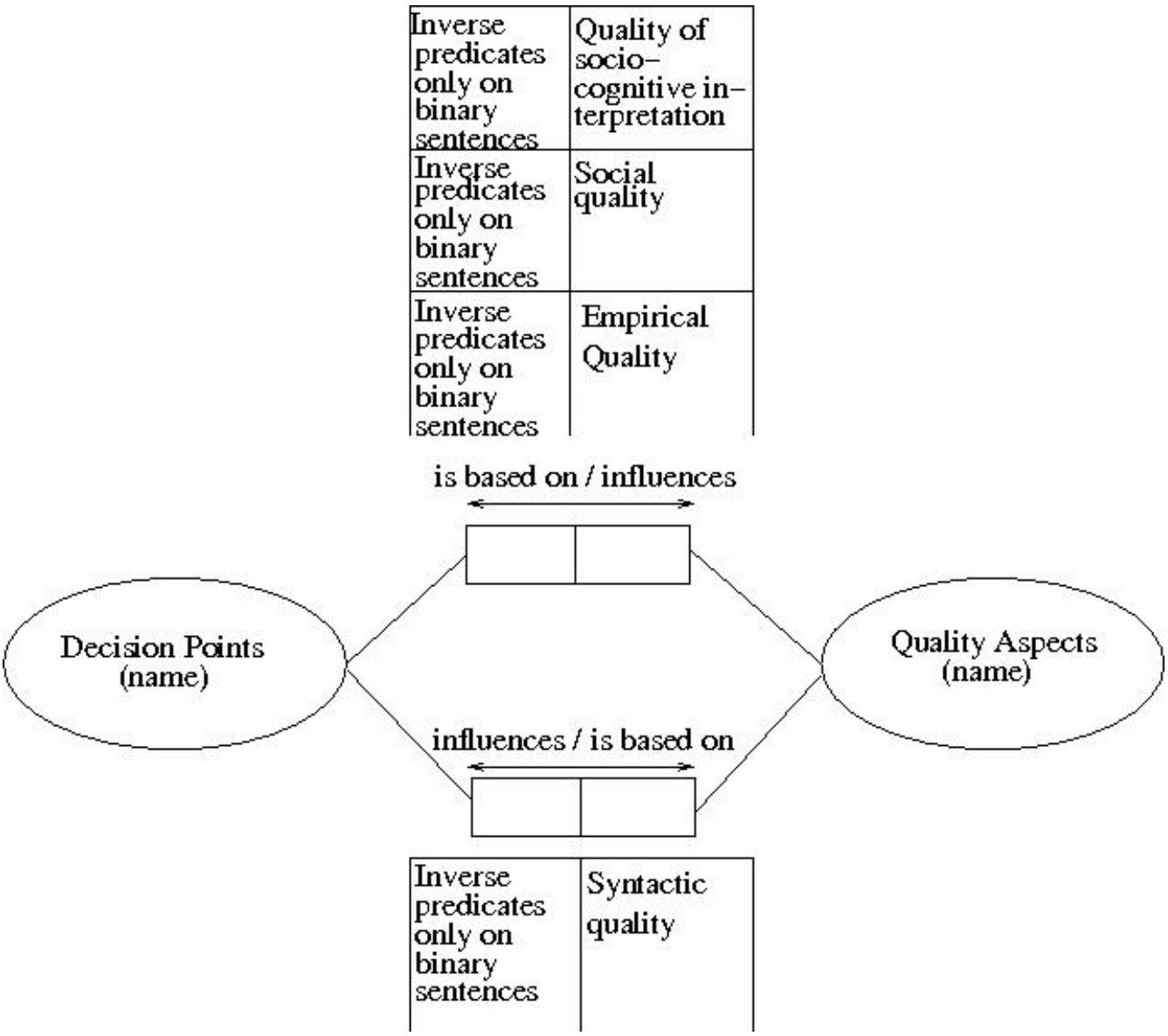


Figure 4.7: The use of inverse predicates only on binary sentences, is based on the quality of socio-cognitive interpretation, social- and empirical quality. This decision influences the syntactic quality.

The use of inverse predicates only on binary sentences can say something whether the quality of socio-cognitive interpretation, the social quality, the empirical quality and the syntactic quality are good, bad, or in between. It depends on the size of the domain, the presence or absence of inverse predicates on binary sentences, to make any judgment about these quality aspects.

If one doesn't use inverse predicates, it can lead to misinterpretation and disagreement. If one does use inverse predicates on n-ary predicates, it might lead to complex sentences, which are hard to understand. The impact of not doing these actions, is not so very hard. Most sentences, without inverse predicates, are still understandable and can be modeled easily, if these sentences aren't too complex.

Using inverse predicates if predicates are the same

A symmetric relation might be the fact that one person likes another person, where this other person automatically likes the first person. So if Ann likes Bob, then Bob likes Ann. This can lead to problems when using the above stated predicates. Suppose Ann and Bob are getting married. One might suggest this relation is symmetric as well. If Ann is married to Bob, then Bob is married to Ann. This relation will then be described as: “The Person with first name 'Ann' is married to / is married to the Person with first name 'Bob'”.

Halpin describes that this relation can lead to problems.[12] Marriage can be described as being mutually exclusive (figure 4.8), as no entry may occur in both columns. This implies this relation is not symmetric, but asymmetric, since we cannot have two rows of the form (a,b) and (b,a) Asymmetry also implies irreflexivity, since we cannot have a row of the form (a,a).[13]
To help avoid such problems Halpin concludes: “at the conceptual level no base predicate should be the same as its inverse.”[14] A better sentence is described in figure 4.8:

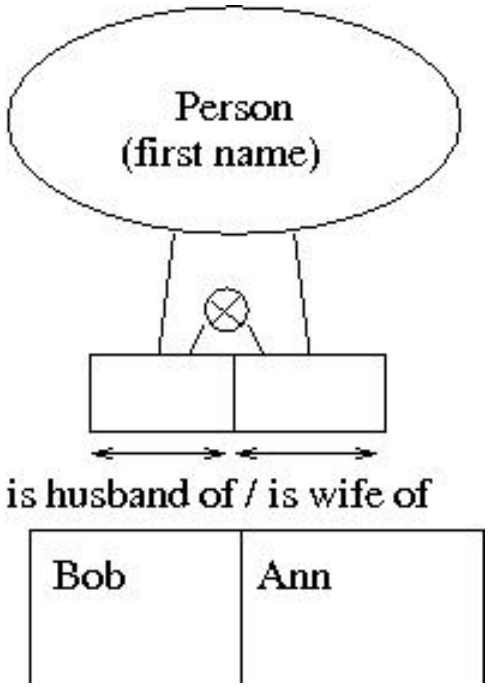


Figure 4.8: The Person with name 'Bob' is husband of / is wife of the Person with name 'Ann'.

When applying quality to this design principle, one might conclude the use of inverse predicates is forbidden when predicates are the same, because it might indicate symmetry, which may be not the case. As a way, the purpose of this decision is to protect the modeler, because the model must represent the domain. Therefore this decision is based on *semantic quality:* the model must be a correct reflexion of the Universe of Discourse. This decision also influences the *syntactic quality*, because it influences the conformity of a sentence to the syntax (figure 4.9).

12 Halpin, page 66

13 Halpin, page 289

14 Halpin, page 66

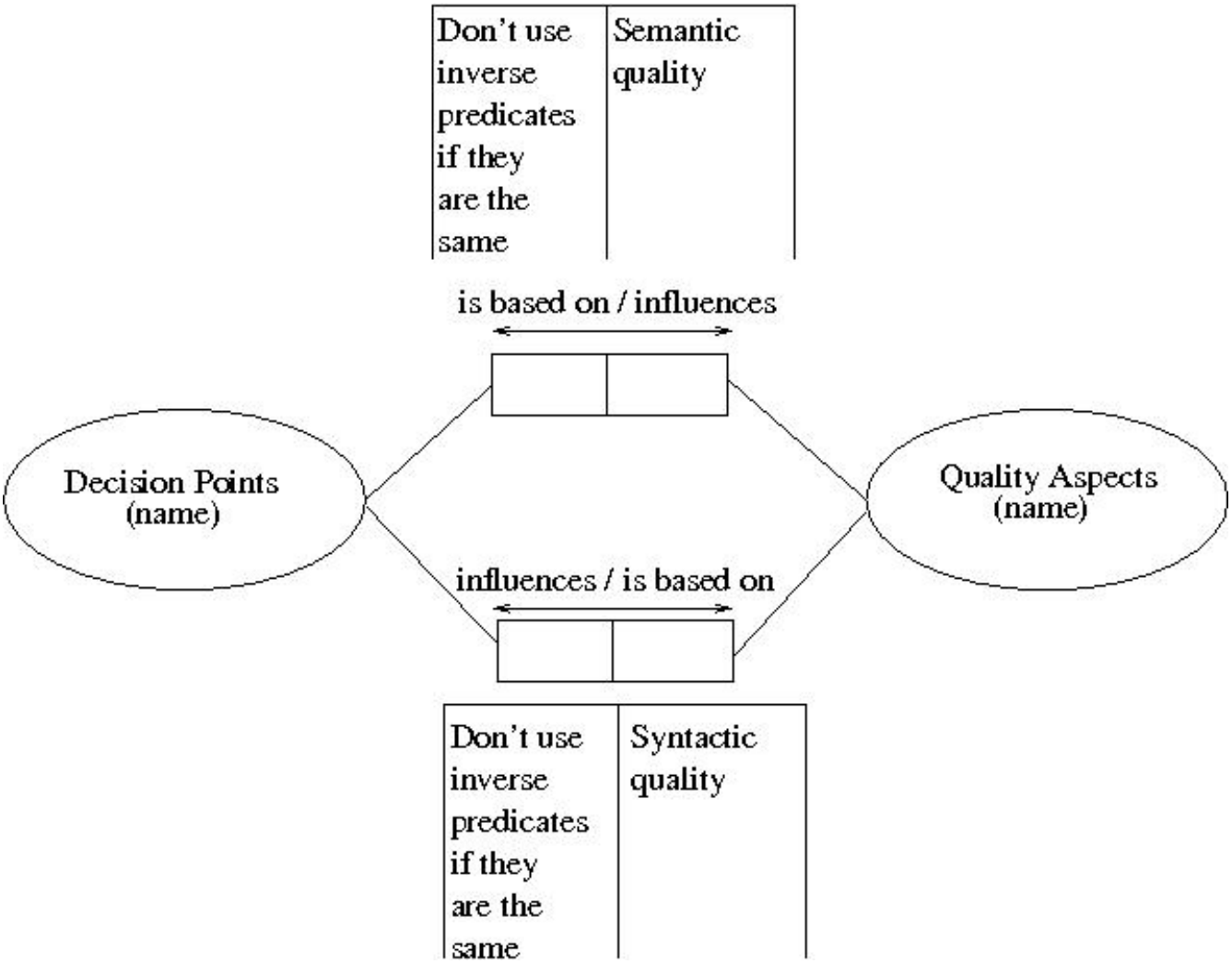


Figure 4.9: Prohibiting the use of inverse predicates if predicates are the same is based on semantic quality. This decision influences the syntactic quality.

Prohibiting the use of inverse predicates even if predicates are the same can say something about the semantic- and the syntactic quality. It depends on the size of the domain and how many times inverse predicates are used even if predicates are the same, to make any judgment about the semantic- and syntactic quality.

If one decides to use inverse predicates even if predicates are the same, it can have significant consequences, as it might not be a good representation of the domain, which might lead to an incorrect implementation if a new system is to be created. It is very important to be cautious with the use of inverse predicates even if predicates are the same.

Textual Language Representations

ORM is a conceptual modeling tool for information modeling and for creating relational database systems. A characteristic of ORM is that it is easy to understand for humans and easy to implement on computers. The use of sentences, to get things clear when communicating with domain experts or other stakeholders, or to help implementing a database, can be useful. We are using elementary facts in a defined form. There are several ways to write down these sentences in a syntactic correct way.

For example the sentence (figure 4.1): *"The Employee with first name 'Ann' is working for / has as worker the Department with name 'Finance'",* can also be written in a more abbreviated form: *Employee (first name) 'Ann' is working for / has as worker Department (name) 'Finance'.* In this example the words 'the' and 'with' can be dropped and one can place reference modes in parentheses after the object. One could also display an elementary fact in a diagram form. More formal are the textual language representations like RIDL or ConQuer. In Halpins book there are all kinds language representations. He doesn't really suggest using a special form, but he mentions ConQuer as being a general language, which can be used to define business rules and can be mapped automatically in SQL. According to Halpin, it could be used as a very high level language for capturing business ruses in general, both derivation rules and constraints.[15]

Using ConQuer for example is based on the *quality of technical interpretation*: it can be automatically mapped in SQL. LISA-D for example is very expressive[16] (page 670), but lacks tool support according to Hofstede (ter Hofstede 1996), in the case of LISA-D the *semantic quality* has high value, because it represents the domain very well. Using a textual language representation depends on what you want to do with it, what is important for you and what you like to use. It can be based on different quality aspects, in the case of ConQuer the quality of technical interpretation and in the case of LISA-D the semantic quality. Using a formal representation has consequences for the syntactic quality, because it influences the conformity of a sentence to the syntax (figure 4.10)

15 Halpin, page 676

16 Halpin, page 670

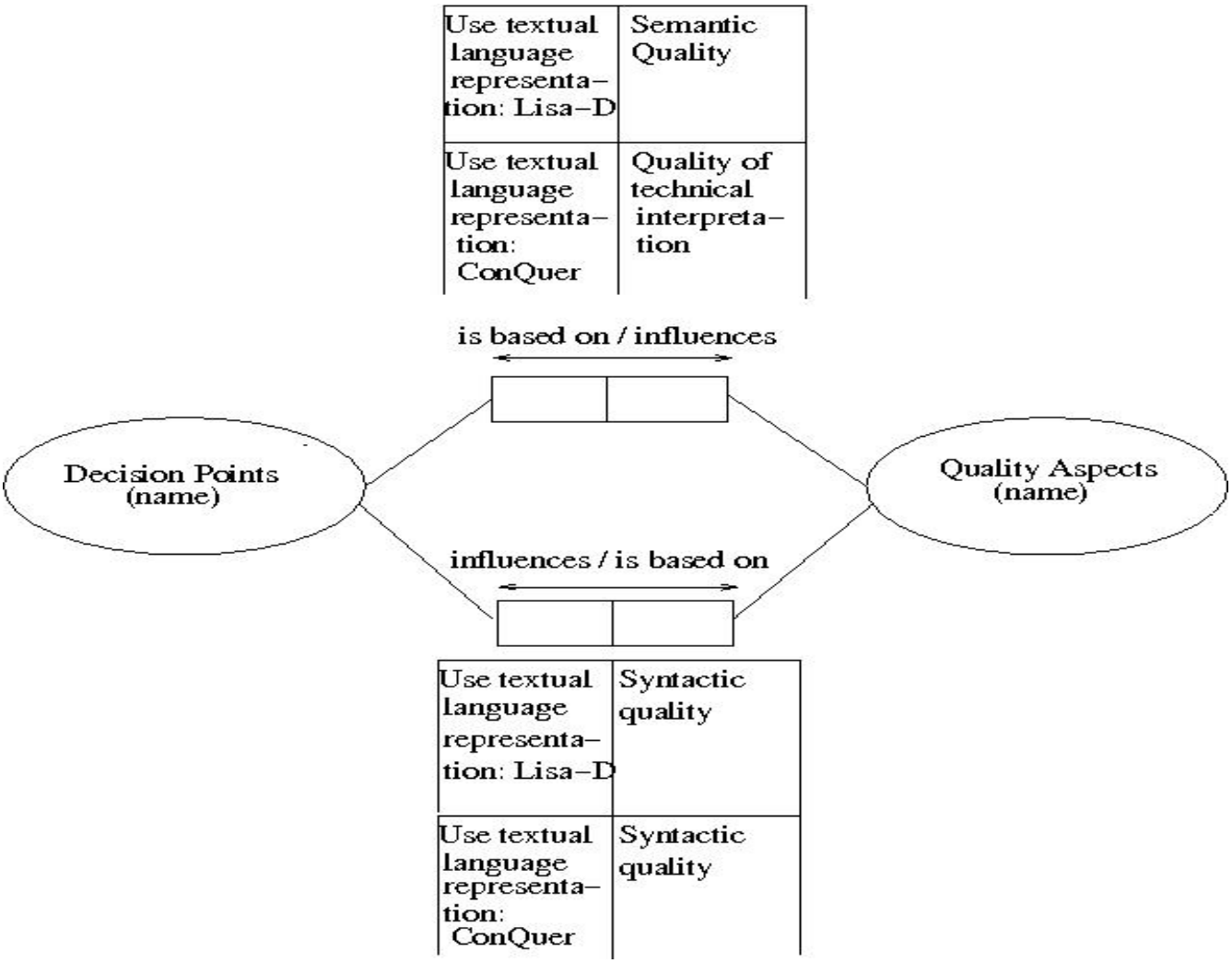


Figure 4.10: Choosing a textual language representation based on a quality aspect (Lisa-D: semantic quality, ConQuer: Quality of technical interpretation). This decision influences the syntactic quality.

The use of a textual representation languages gives an indication whether one quality aspect will be higher than another quality aspect. For example if one uses ConQuer, the quality of technical interpretation will be indicated higher than the semantic quality. It also gives us an indication about whether the syntactic quality is good or bad. It depends on the conformity to the syntax to make any judgment about the syntactic quality.

If one doesn't follow the syntax of a textual language representation, it might lead to misinterpretation and disagreement. It is therefore necessary to use a textual language representation which is conform its syntax or using an informal language representation which won't lead to misinterpretation and disagreement.

Splitting multiple facts which leads to loss of information

Splitting sentences can lead to a loss of information, which is against the rule of how elementary facts should be defined: the adjective 'elementary' indicates that the fact cannot be 'split' into smaller units of information that collectively provide the same information as the original.[17]

Halpin warns about splitting sentences in multiple facts, which might result in a loss of information.[18] For example: "*The Employee with first name 'Ann' studied a Course with name 'Knowledge Management' received a Grade with rating '7'".* This sentence can't be split into the following sentences: *The Student with first name 'Ann' studied the Course with name 'Knowledge Management'* and *The Student with first name 'Ann' received a Grade with rating '7'* (figure 4.11).

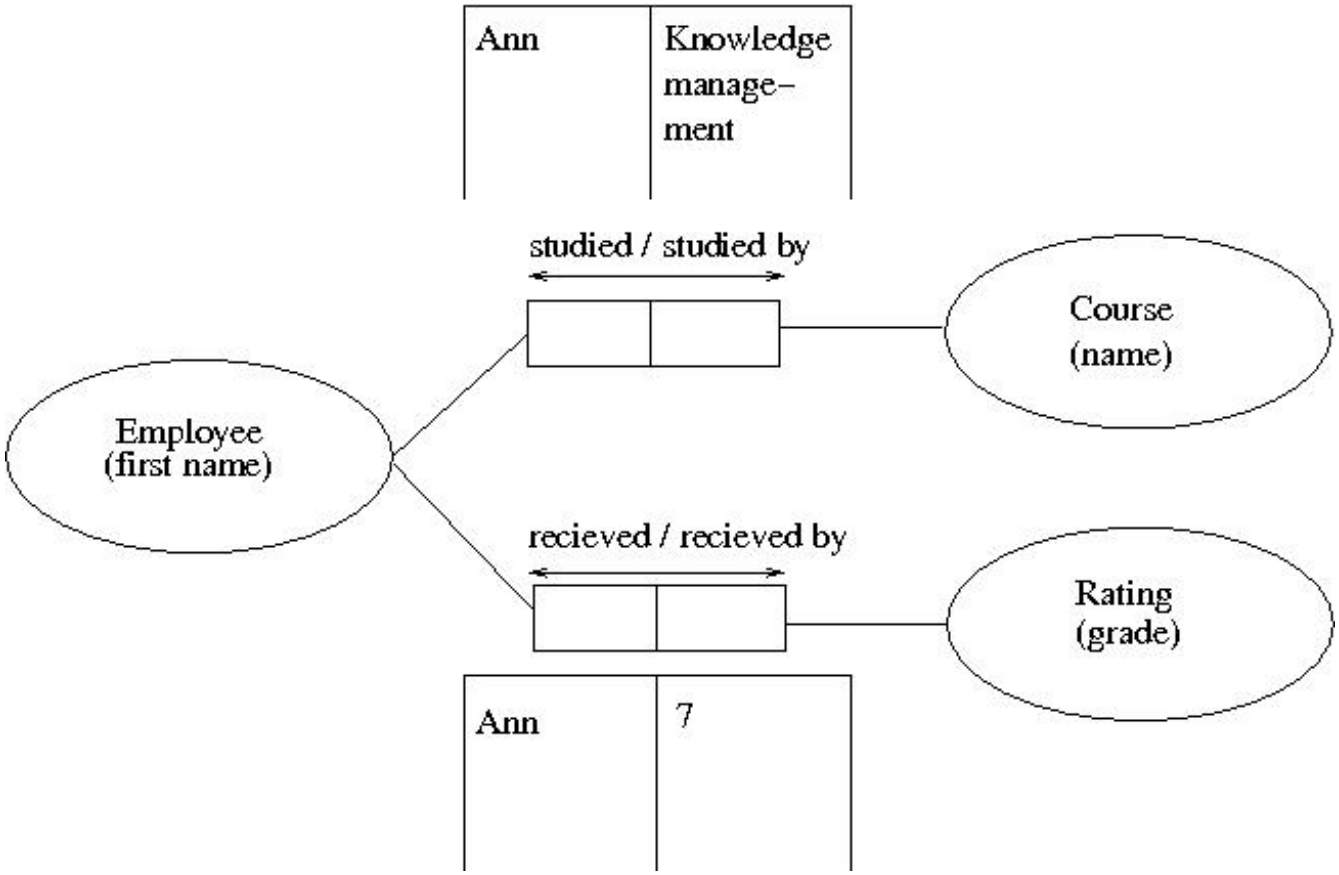


Figure 4.11: Wrong use of splitting multiple facts, which leads to loss of information.

17 Halpin, page 61

18 Halpin, page 69

In the last sentence we don't know if this grade was actually for the course Knowledge Management. The splitting of multiple facts which result in the loss of information is therefore forbidden. This decision is based on the *semantic quality*, because the model might not represent the domain (because of the loss of information). This decision also influences the *syntactic quality*, because it influences the conformity of a sentence to the syntax (figure 4.12).

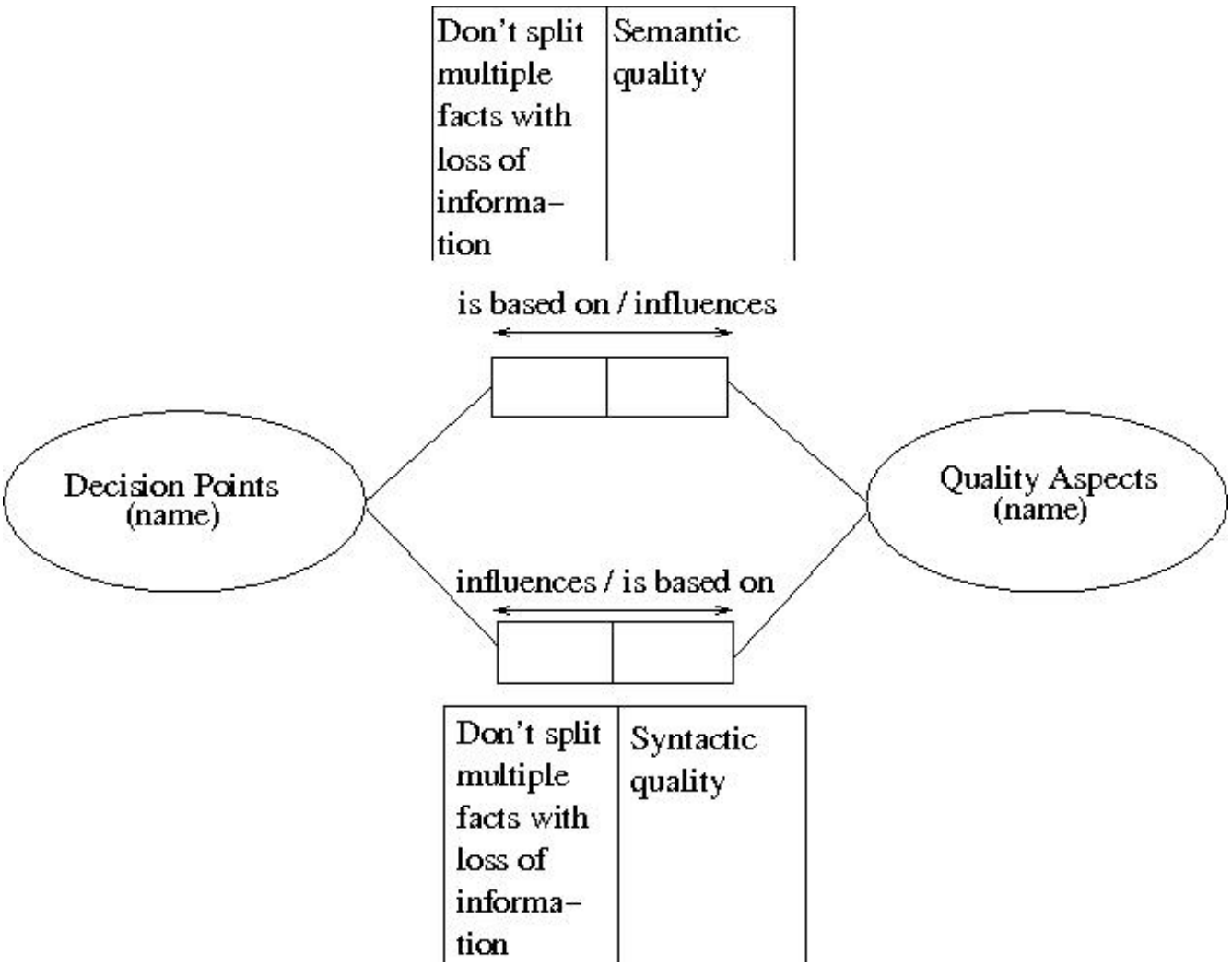


Figure 4.12: Prohibiting the splitting of multiple facts with loss of information is based on semantic quality. This decision influences the syntactic quality.

When prohibiting splitting multiple facts which leads to loss of information, one can say something about the semantic- and the syntactic quality. It depends on the size of the domain, the presence and absence of splitting multiple facts incorrectly, to make any judgment about the semantic and syntactic quality.

If one decides to split multiple facts although it results in the loss of information, it can have significant consequences, as it might not be a good representation of the domain, which might lead to an incorrect implementation if a new system is to be created. It is very important to be cautious with splitting multiple facts, because it might lead to loss of information.

Feedback

When describing a sentence, it has to be done in collaboration with another person, which is frequently the domain expert. This person must be able to understand these sentences, for communication purposes and communicating with a domain expert guarantees quality when implementing an information system.

Halpin suggest the first step of the CSDP (transform familiar information examples into elementary facts, and apply quality checks) can actually be split in two steps. The first, which is done by the domain expert is to verbalize the information. The second which is done by the modelers is to refine their verbalization by ensuring the facts are elementary and the objects are well identified.[19] For example a domain expert might describe the following sentence: "*Employee Ann goes to London from Amsterdam with KL001."* This can be described into the following sentence by a modeler: "*The Employee with first name 'Ann' goes from the City with name 'Amsterdam' to the City with name 'London' with Flight number with code 'KL001'."* (figure 4.13). Here it is assumed that KL001 is a flight number and Amsterdam and London are cities. But are they really? KL001 might be a boat and flight number is incorrect in that case.

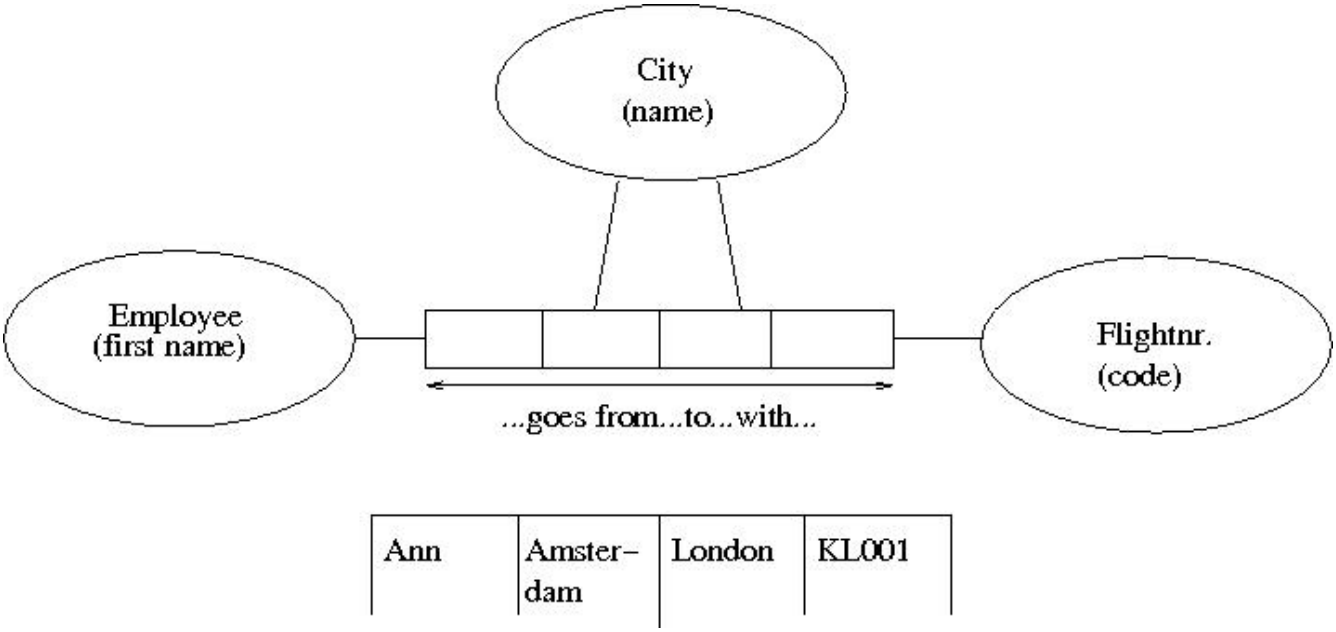


Figure 4.13: The Employee with first name 'Ann' goes from the City with name 'Amsterdam' to the City with name 'London' with Flight number with code 'KL001

All these features rely on interpretation, therefore communication with the domain expert is very important. The model might come out to be semantically incorrect or interpretation might be different. Therefore feedback is essential for the *semantic quality* and the *quality of socio-cognitive interpretation* (figure 4.14).

19 Halpin, page 70, 71

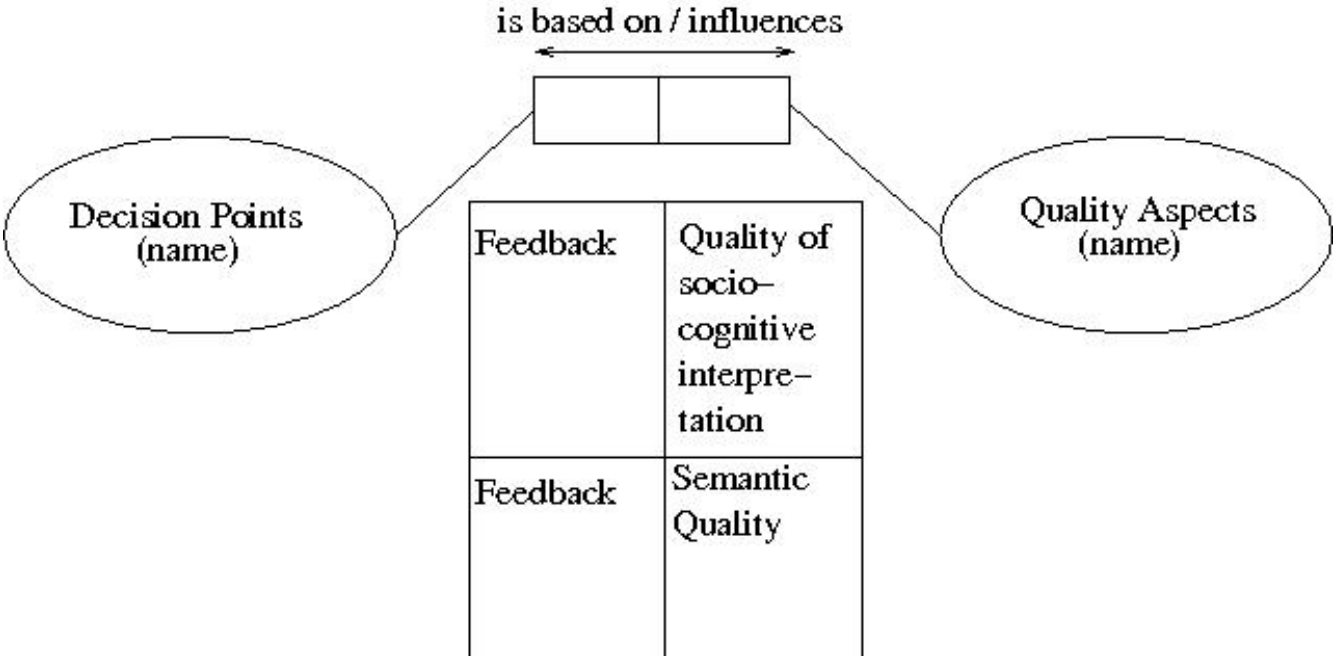


Figure 4.14: Including feedback is based on the quality of socio-cognitive interpretation and semantic quality.

One can say something about the semantic quality and the quality of socio-cognitive interpretation. It depends on the domain, the presence or absence of the feedback and the quality of the feedback by the domain expert, to make any judgment about the the semantic quality and the quality of socio-cognitive interpretation.

If one doesn't do something with feedback of stakeholders, it might lead to misinterpretation, disagreement, delay and even worse, a wrong model. Therefore feedback is necessary, even if it is time consuming and hard to get. The impact of not doing this can lead to unpleasant circumstances, even if the model is not very complex.

Synonyms

People can use different names, or different sentences, but say the same things. This might lead to confusion. For example an 'Employee' might also be a 'Worker' or 'Human capital'.

For these issues Halpin suggest the following: “if the domain experts all prefer the same term, you should use that. In large projects, different people might use different terms for the same concept. In that case, you should get them to agree upon a standard term, and also note any synonyms that they might still want to use”.[20] This way there is no confusion.

The decision to use standard terms and / or a list of synonyms is based on *social quality,* as it improves agreement among stakeholders. Stakeholders should agree which term to use. This decision doesn't influence the syntax of the model, because using different terms is not forbidden and still conform the syntax (figure 4.15)

20 Halpin, page 74

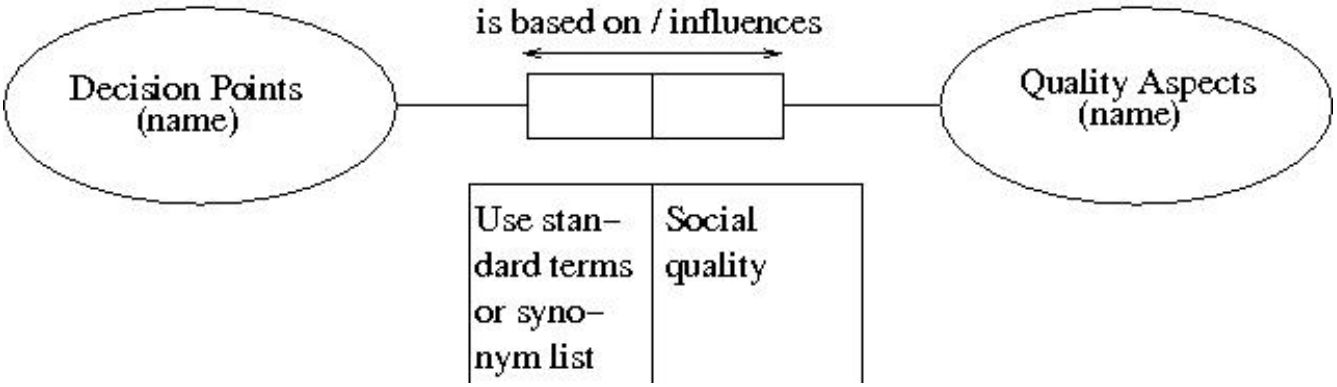


Figure 4.15: Using standard terms or a synonym list is based on social quality

When using standard terms and / or a list of synonyms one can say something about the social quality. It depend on the size of the domain, the presence or absence of standard terms and a list of synonyms, to make any judgment about the social quality.

If one decides not to use standard terms and / or a synonym list, it might lead to disagreement about the term which is being used. A stakeholder might prefer another term. A consequence is that there is disagreement about the presented model. If the system isn't too complex and there are not a lot off stakeholders the impact on the social quality is less, and one can decide not to use standard terms and / or a synonym list.

Modeling the current situation

When a company want to improve its information system, a company has two choices:

- Building a new information system
- Improving its information system

Before doing this a company might want to build a model of the current situation, to find bugs, learn about its information system and find ways to improve its information system.

Halpin describes it is better first to build a model of the current situation. From here one can look for a better way to improve business. A proper understanding of the current situation is according to Halpin a great assistance in designing the future model.[21]

The decision to build a model of the current situation before building a model of the desired situation is based on *domain quality* (figure 4.16) Does the current domain fits into the desired situation. What has to be done to create this desired situation? According to Halpin it should always know its current situation when one wants to make improvement.

21 Halpin, page 75

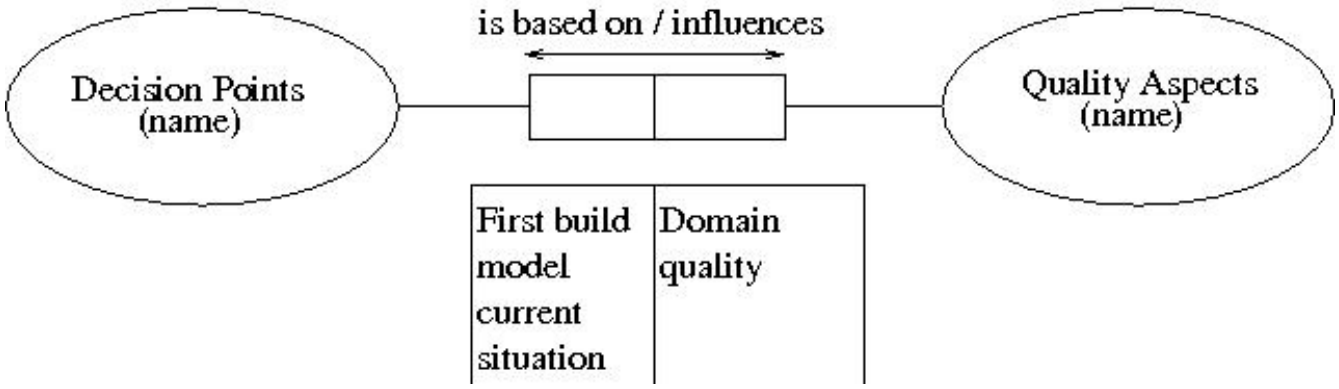


Figure 4.16: Building a model of the current situation before building a model of the desired situation is based on the domain quality.

When building a model of the current and the desired situation one can say something about the domain quality. It depend on the quality of the current- and the future model, to make any judgment about the domain quality.

If one decides not to model the current situation, it is not easy to see where improvements in the domain can be made. The impact of not doing this can be great, especially if the domain is big and complex. The impact is less in small and not too complex domains. Of course one cannot build the current situation, if there is no current domain, for example when beginning new businesses.

CHAPTER 5

STEP 2: DRAW FACT TYPES, AND POPULATE

Vague predicates

Predicates are displayed as 'fact types' in a conceptual schema diagram. This fact type is drawn and is displayed with the name of this fact type and its inverse. For example: *"The Employee with first name 'Ann' has a / is of Company-car with registration number 'YJ-KF-29'".* The problem with this kind of sentence is that the verb 'have' can mean different kinds of things (figure 5.1).

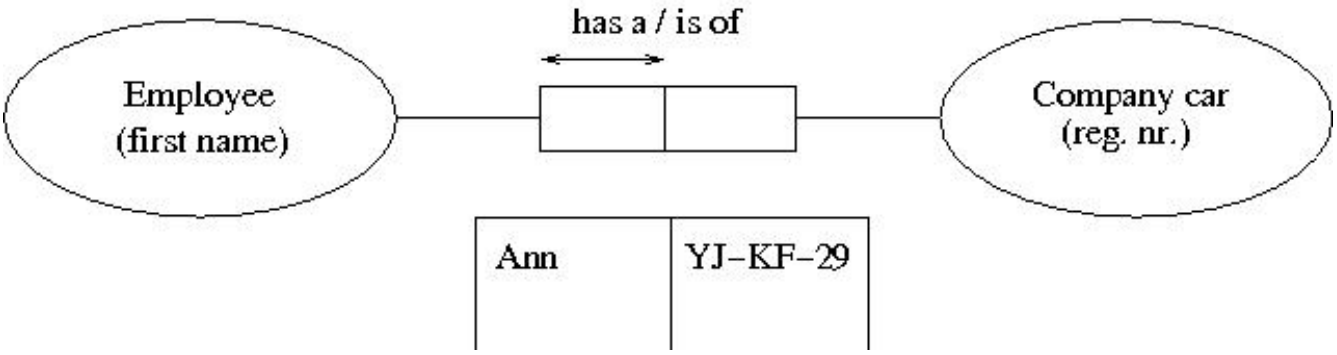


Figure 5.1: Example vague predicate: the Employee with first name 'Ann' has a / is of Company-car with registration number 'YJ-KF-29'

This is also acknowledged by Halpin. He says: "it is best to be avoided if there is a more descriptive, natural alternative."[22] In the previous example 'has' can mean 'drives', 'owns' or even 'stole'. 'Has' may be used in some fact types, but if is best to avoid such vagueness.

The decision of avoiding the word 'have' or other vague words, is based on some quality issues, as one would like to avoid vagueness. The quality aspects are based on the *quality of socio-cognitive interpretation*, because some stakeholders may interpret the verb 'have' differently. It is also based on the *semantic quality,* because the word 'having' may not really be representative for the domain. It is also based on *social quality,* because stakeholders should agree on using a level of abstraction, where 'have' is a high level, and using 'owns' is a low level of abstraction. The decision doesn't influence the syntactic quality. Although a wrong level of abstraction is used and people may interpret differently, it doesn't influence the syntax. Using vague predicates might be still conform the syntax of a modeling language.

22 Halpin, page 81

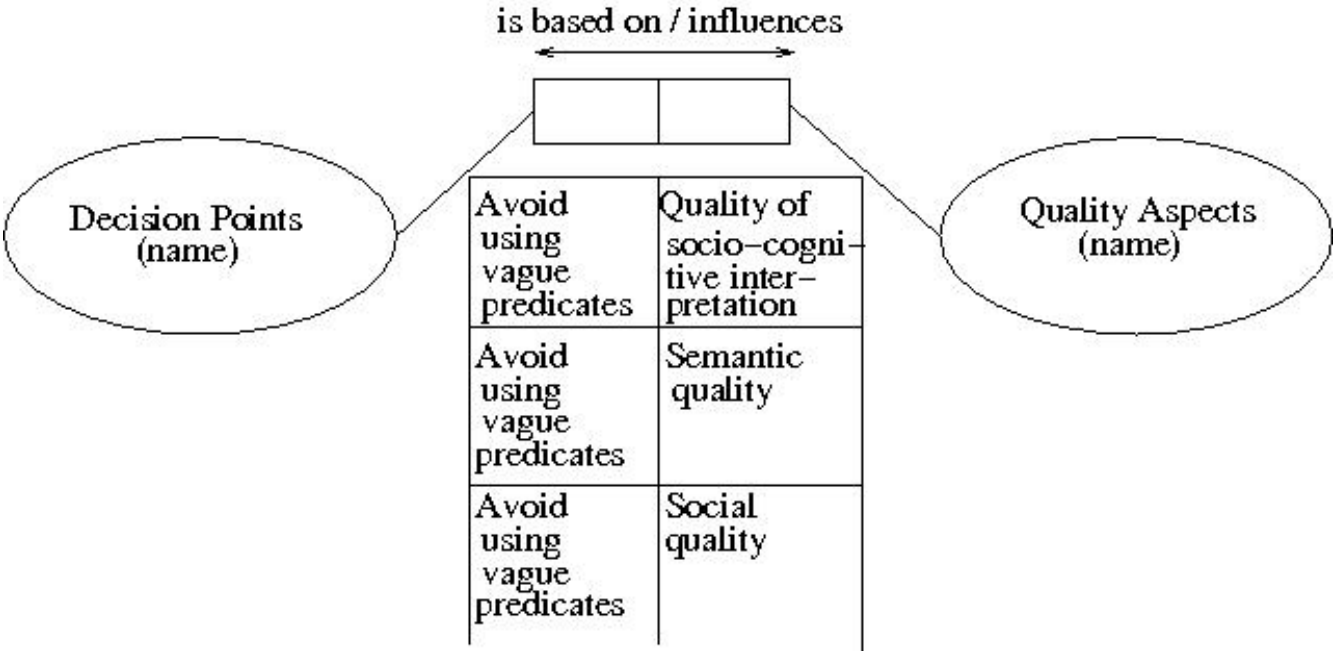


Figure 5.2: Avoiding the use of vague predicates, like 'have', is based on the quality of socio-cognitive interpretation, the semantic quality and the social quality.

When using or avoiding vague predicates like 'have', one say something about whether the quality of socio-cognitive interpretation, semantic quality, social quality and syntactic quality is good, bad or in between. It depend on the size of your domain, the chosen level of abstraction and the number of usages of these vague terms, to make any judgment about these quality issues.

If one does use vague terms like 'have', it might lead to a non-representative model, disagreement, different interpretations, etcetera. One has to clearly state it's level of abstraction. Sometimes using vague words can't be avoided in my opinion, because they can represent a large population. If one does use vague 'terms' and all stakeholders do agree about using it, it is possible to use. Be careful though, in large models, vagueness can have a great impact, because of the consequences described above.

Drawing reference modes

As already described in the 'clarify entity' section (step 1), the reference mode is described as the manner in which the value refers to the entity. We may want to draw these reference modes as well. In the popular example (Figure 4.1): *"The Employee with first name 'Ann' works at Department with name 'Finance'"*, the reference mode is 'first name' (Employee) and 'name' (Department), as it is the manner in which the value refers to the entity.

Halpin suggest to use these reference modes in the conceptual scheme diagrams. Halpin describes two ways of doing this.

1. placing the name of the reference mode in parentheses next to the name of the entity type (Figure 4.1).
2. placing the relation explicitly by using fact types, and 'open entity-types' (Figure 5.3)

Halpin says the first figure is preferred, unless we want to illustrate the reference schemes explicitly. This is done because the first figure is closer to the way we verbalize facts and it simplifies the diagram.[23]

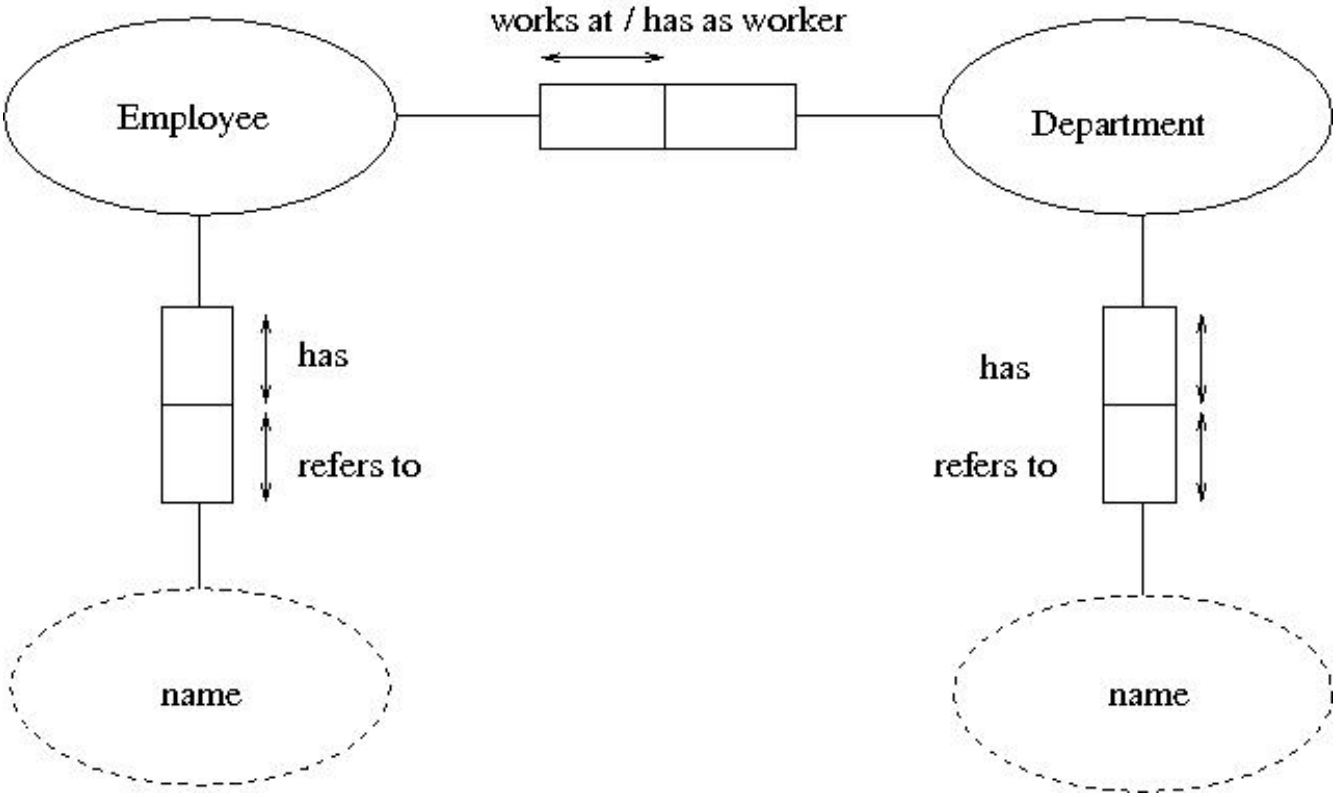


Figure 5.3: placing the relation explicitly by using fact types, and 'open entity-types'. In this schema the name of a person is unique.

The decision to prefer parentheses, like in figure 4.1, is based on the fact that it is closer to the way we verbalize facts and it simplifies the diagram. This verbalization and simplification is a matter of *empirical quality,* as it is closer to one's understanding and reduces complexity of the model. One can say the lay-out of the first model is closer to how humans conceive a model. Of course, this must be done unless we want to illustrate reference schemes explicitly. This decision has influence on the *physical quality*, as it influences the way artifacts are presented and *syntactic quality*, because the presentation of the reference mode must be conform the syntax (figure 5.4).

23 Halpin, page 81

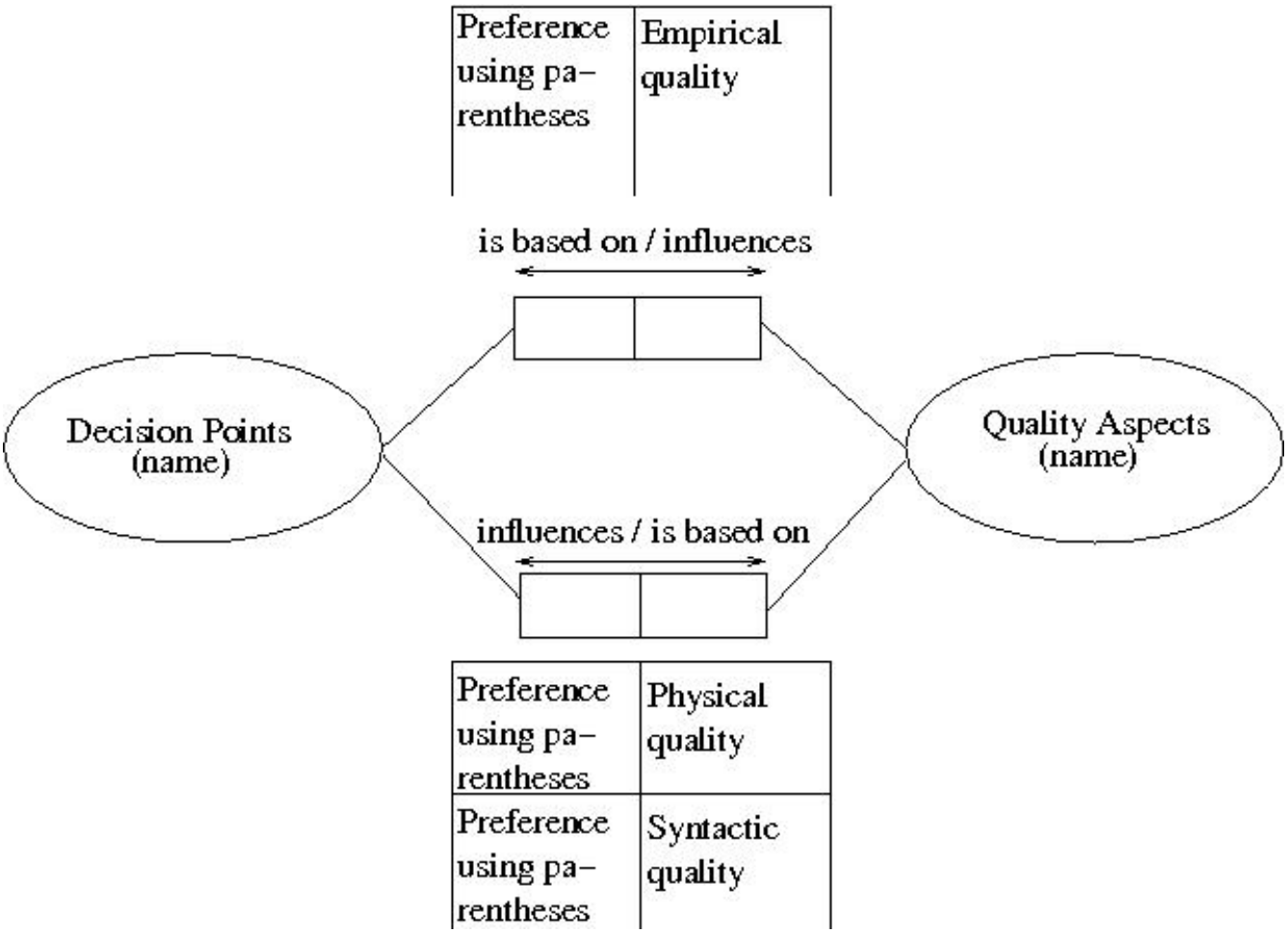


Figure 5.4: the preference of using parentheses, unless one wants to illustrate reference schemes explicitly, is based on the empirical quality. This decision influences the physical quality and the syntactic quality.

When using reference schemes like the way described above, one can say something about whether the physical, empirical and syntactical quality is bad, good or in between. It depends on the usages of these drawings and the size of the model, to make any judgment about these quality issues.

When one uses reference schemes the wrong way, this might lead to complex models, which are hard to verbalize. The model may not be easy to understand by humans. The impact is not really great, because the model may not be wrong, unless it is not conform the syntax. A consequence can be that the modeler must explain its model verbally, which may not be necessary if he had done it right.

Populate

How can the modeler be sure the conceptual scheme diagram is correct? This can be performed if instances are described in a model. Populating is about describing these instances in a model.

About populating Halpin says: “In ORM, a fact type is simply a table for displaying instances of an elementary fact type. The term “fact table” is used in a different sense in data warehousing. A diagram that includes both a schema and a sample database is called a knowledge base diagram.[24] He also says: “Populating the schema diagram is useful not only for detecting schema diagrams that are nonsensical, but also for clarifying constraints.”[25]

You may want to populate the conceptual scheme diagram. The outcome is a knowledge base diagram. The decision to populate the scheme is based on *syntactic quality*: as one would like to know if all constraints are correct and eventually if the model is conform its syntax. It is also based on *semantic quality*, as one would like to know if all instances are correct, so the model is representing the domain. It also useful to populate, because one doesn't want to have different interpretations, which is based on *quality of socio-cognitive interpretation*. It can also influence the level of agreement positively, which is based on *social quality*. It is obligatory to use populations, so the decision doesn't influences the syntactic quality (figure 5.5)

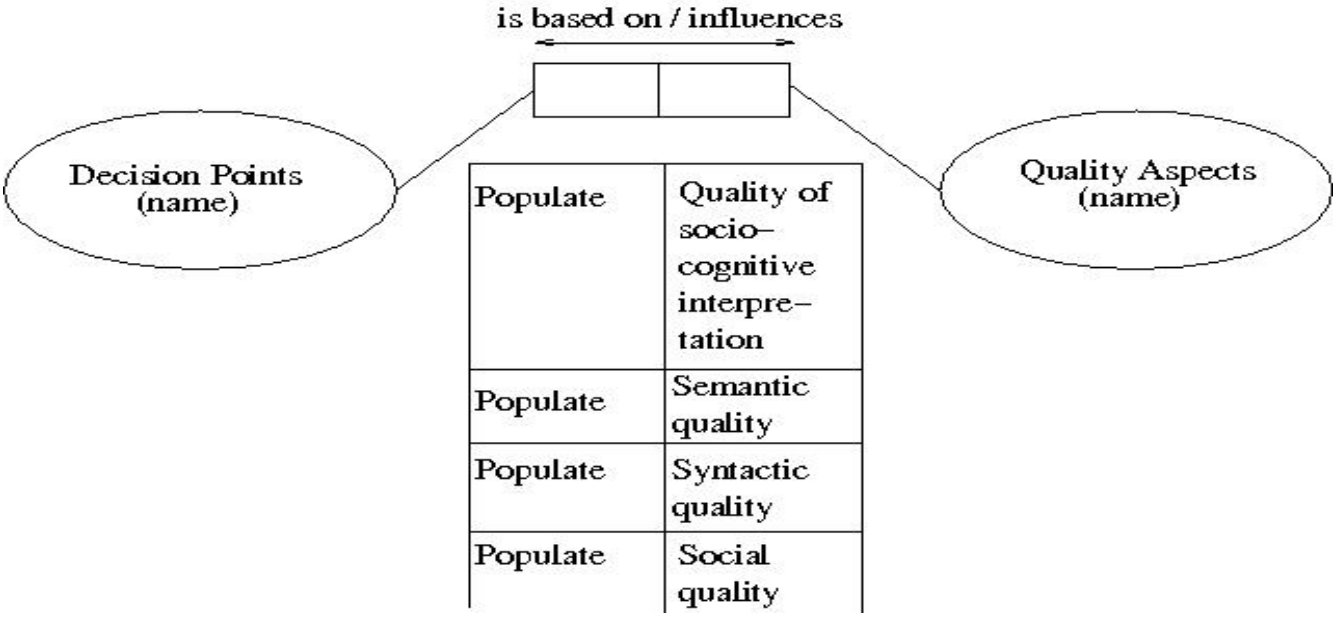


Figure 5.5: the decision to populate is based on the quality of socio-cognitive interpretation, semantic quality and syntactic quality.

When populating fact types, one say something about the syntactic, semantic, socio-cognitive and social quality. It is bases on the size of the model, the number of usages and the need to populate, to make any judgment about these quality issues.

24 Halpin, page 82

25 Halpin, page 82

When one doesn't populate, this might lead to an incorrect model, or a model which is not representative, interpreted differently or might lead to disagreement. The impact depends on its usefulness. Small models, which are easy to understand, doesn't really need to be populated. Complex models should be populated, if one doesn't want to have any negative consequences.

The purpose of formalizing

Why should an ORM-scheme be formally specified? Why shouldn't we use our own modeling language to describe entities, predicates etc. This may be necessary because more informal, non-standardized modeling languages are better to understand for all stakeholders.

Halpin says: “apart from communication with humans, conceptual schemes provide a formal specification of the structure of the Universe of Discourse, so that the model may be processed by a computer system. Hence the schema diagrams we draw must conform to the formation rules for legal schemes. They are not just cartoons.”[26]

Halpin describes that the decision to formalize is based on technical reasons (model may be processed by a computer system), which suggest it is based on the quality of technical interpretation. However this is true, I think it a formalization or standardization involves less interpretation differences. So it is based on the *pragmatic quality,* because it is including both quality of socio-cognitive and technical interpretation. It is also based on *social quality* as it can improve the level of agreement among stakeholders. The decision to formalize has influence on the *syntactic quality* of the model, because formalization gives the syntax its shape (figure 5.6).

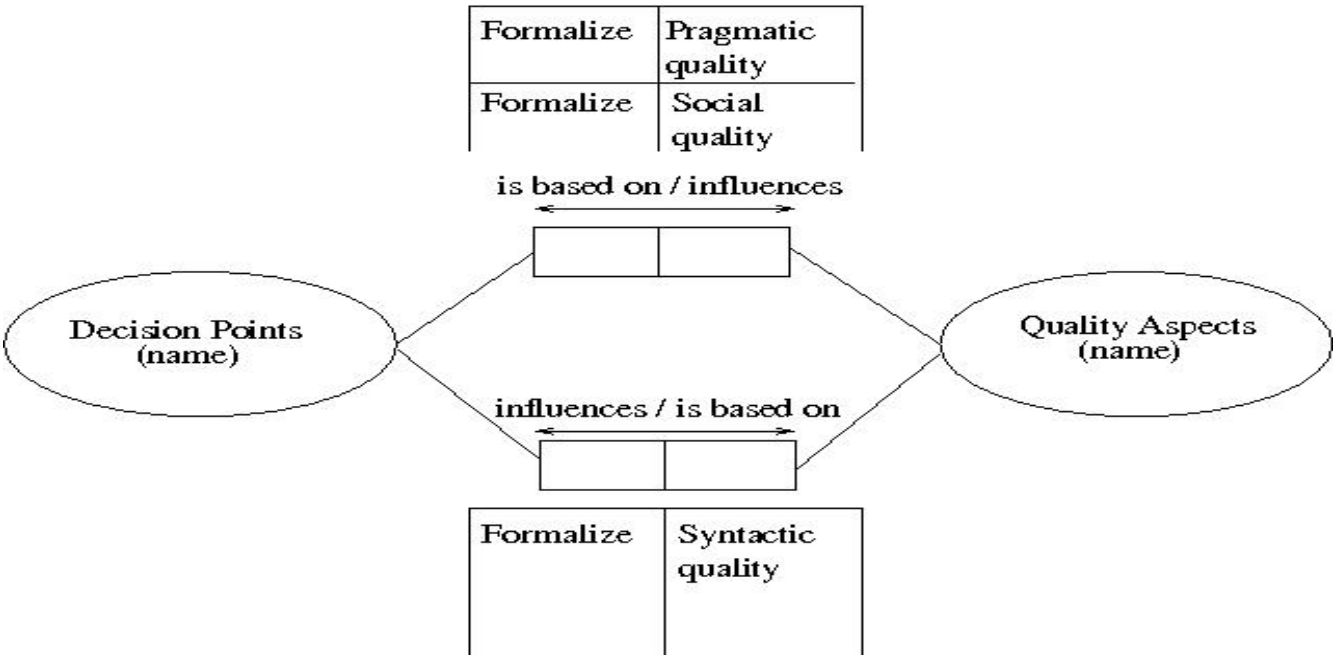


Figure 5.6: the decision to formalize is based on the pragmatic quality and the social quality. This decision influences the syntactic quality.

26 Halpin, page 84

When formalizing certain rules, one say something about the syntactic, pragmatic and social quality. It depend on the frequent use of formalization rules, and the size of the model, to make any judgment about these quality issues.

Because one doesn't formalize certain rules, the syntax is of no importance. When presenting this 'informal' model, it can lead to major disagreement and misinterpretation among stakeholders and computers. The only way 'informal' models can be used, is when it is used as a utility (or draft) for the modeler to formalize the informal model. The impact can be very dramatic, especially for the modeler, because his job is to model according to a certain syntax. If he doesn't do this, it is bad for his reputation.

Objectification

Objectification is described as making an object out of a relationship. For example the following sentence can be objectified: "*The Employee with first name 'Ann' studied a Course with name 'Knowledge Management' received a Grade with rating '7*'" (figure 5.7). We now use the relationship between the person and the course as an object itself which results in the following sentences: *"The Employee with first name 'Ann' studied a Course with name 'Knowledge Management'"* and *"This resulted in receiving a Grade with rating '7*'" (figure 5.8) . 'This' is being referred to the first sentence. The first sentence is the objectified association, which is depicted by a rectangle around the predicate being objectified.

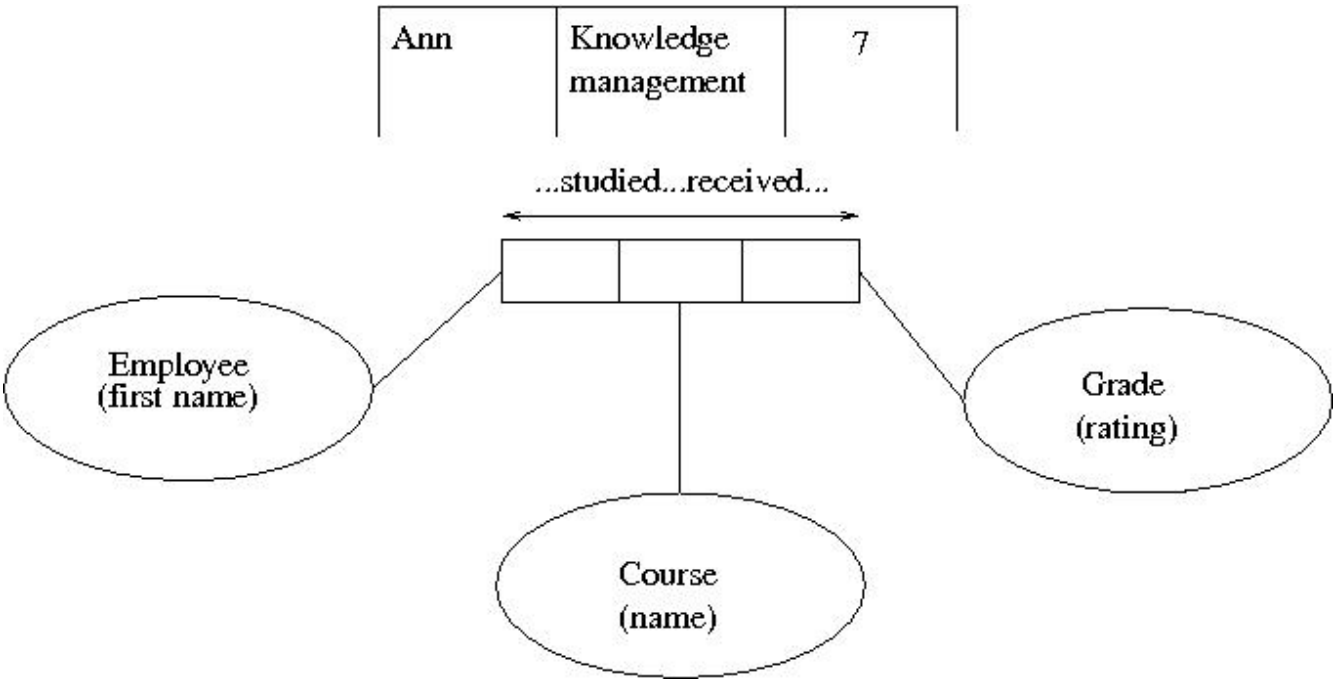


Figure 5.7:The Employee with first name 'Ann' studied a Course with name 'Knowledge Management' received a Grade with rating '7'.

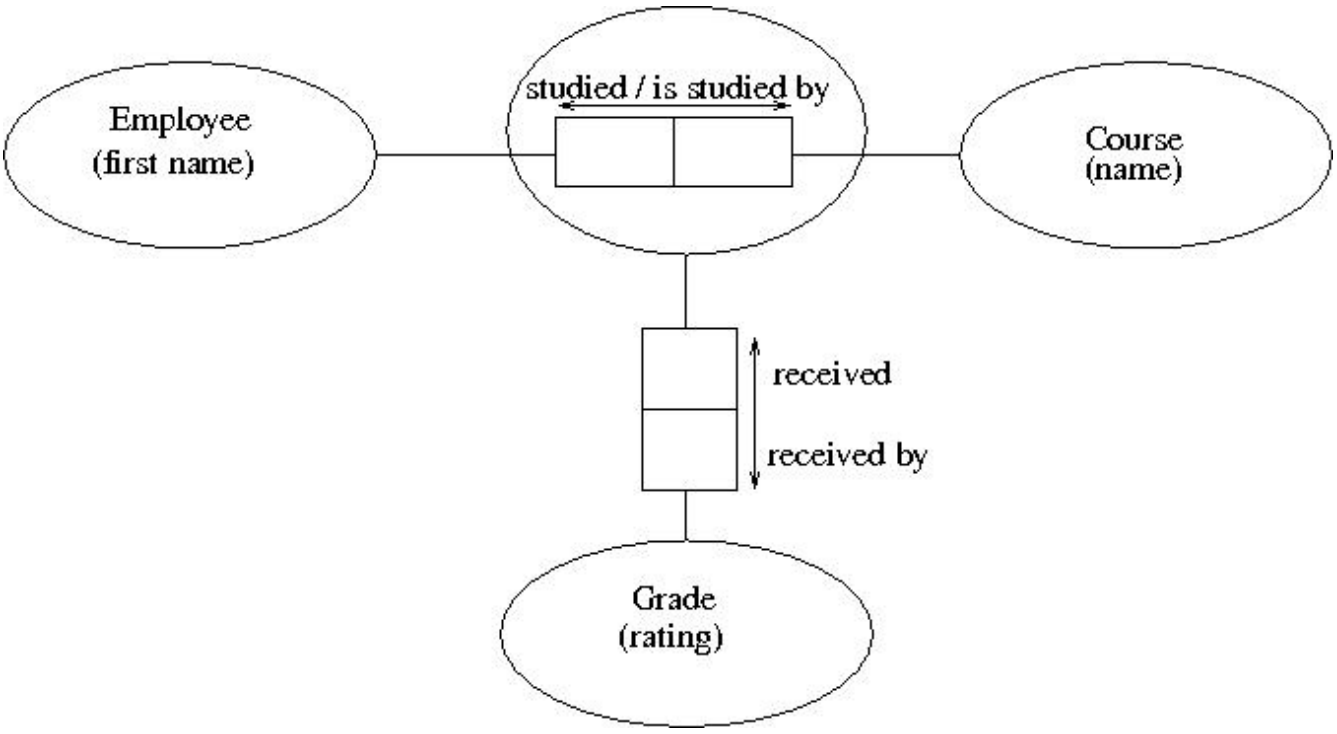


Figure 5.8: "The Employee with first name 'Ann' studied a Course with name 'Knowledge Management'" and *"This resulted in receiving a Grade with rating '7'".*

Halpin describes there is a difference in preference between using ternary fact types and using objectification. Using ternary fact types is preferred, since it is simpler to diagram and populate. Objectification is preferred if the objectified association has an optional role or more than one role to play. If for example one wants to add the date using ternary fact types one must add another ternary fact type: "*The Employee with first name 'Ann' studied a Course with name 'Knowledge Management' received a Grade with rating '7*'" and "*The Employee with first name 'Ann' studied a Course with name 'Knowledge Management' starts at date '01-10-2007*'" (figure 5.9). By using objectification one can simply add the binary: "This started at date *'01-10-2007*'" (figure 5.10). The objectification also simplifies constraint specification.[27] Objectification is also preferred since the role played by the objectified association is optional.

27 Halpin, page 86

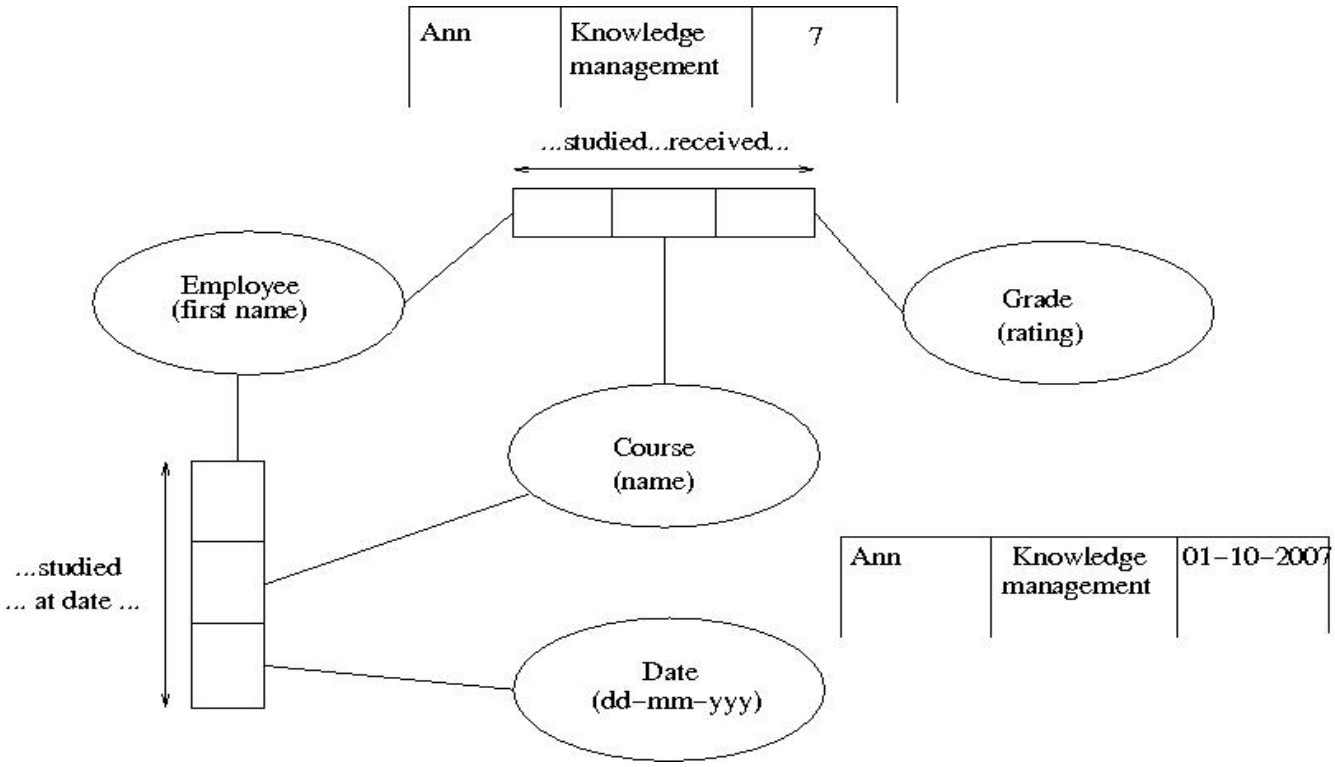


Figure 5.9: "The Employee with first name 'Ann' studied a Course with name 'Knowledge Management' received a Grade with rating '7'" and *"The Employee with first name 'Ann' studied a Course with name 'Knowledge Management' starts at date '01-10-2007'"*

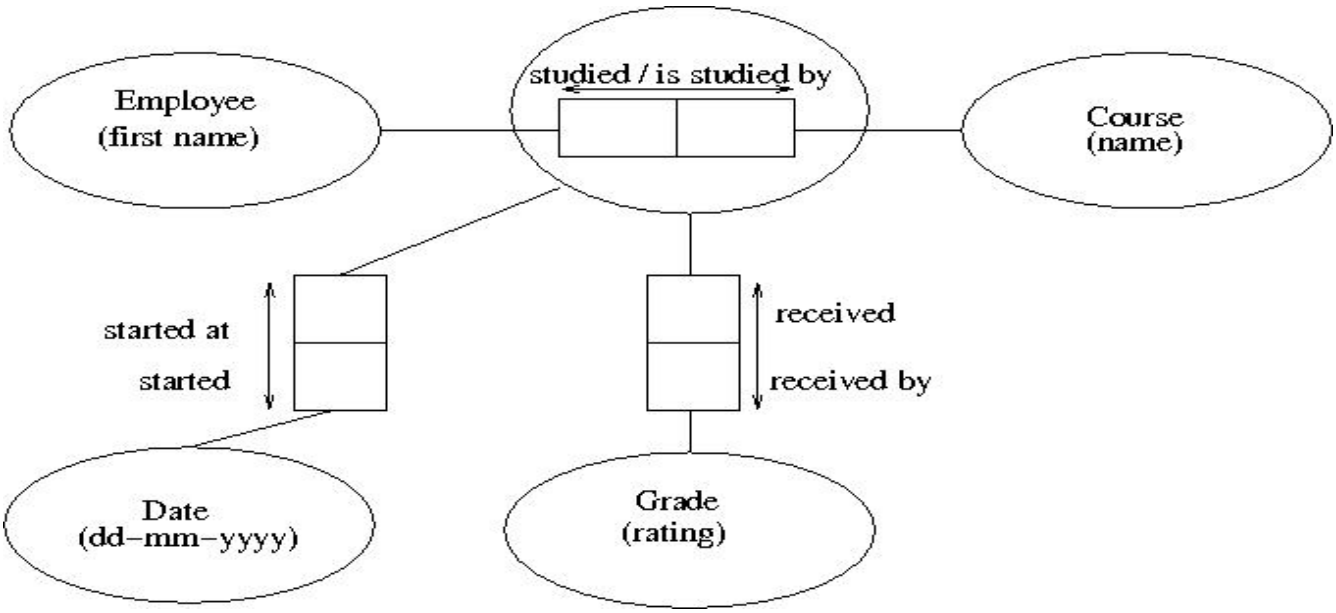


Figure 5.10: Including: "This started at date '01-10-2007'"

If objectification can be used (the objectified association is mandatory), it is up to the modeler which way he wants to draw these relationships. The preference of using n-ary fact types or using objectification indicates the syntax is based on the *empirical quality,* as it reduces complexity of the diagram. This decision has impact on both physical quality and syntactic quality. The consequence for the *physical quality* is the way the artifact is presented. In this case, with an n-ary fact type (square), or with objectification (round). The consequence for the *syntactic quality* is that the presentation of the n-ary fact type or the objectification must be conform the syntax (figure 5.11).

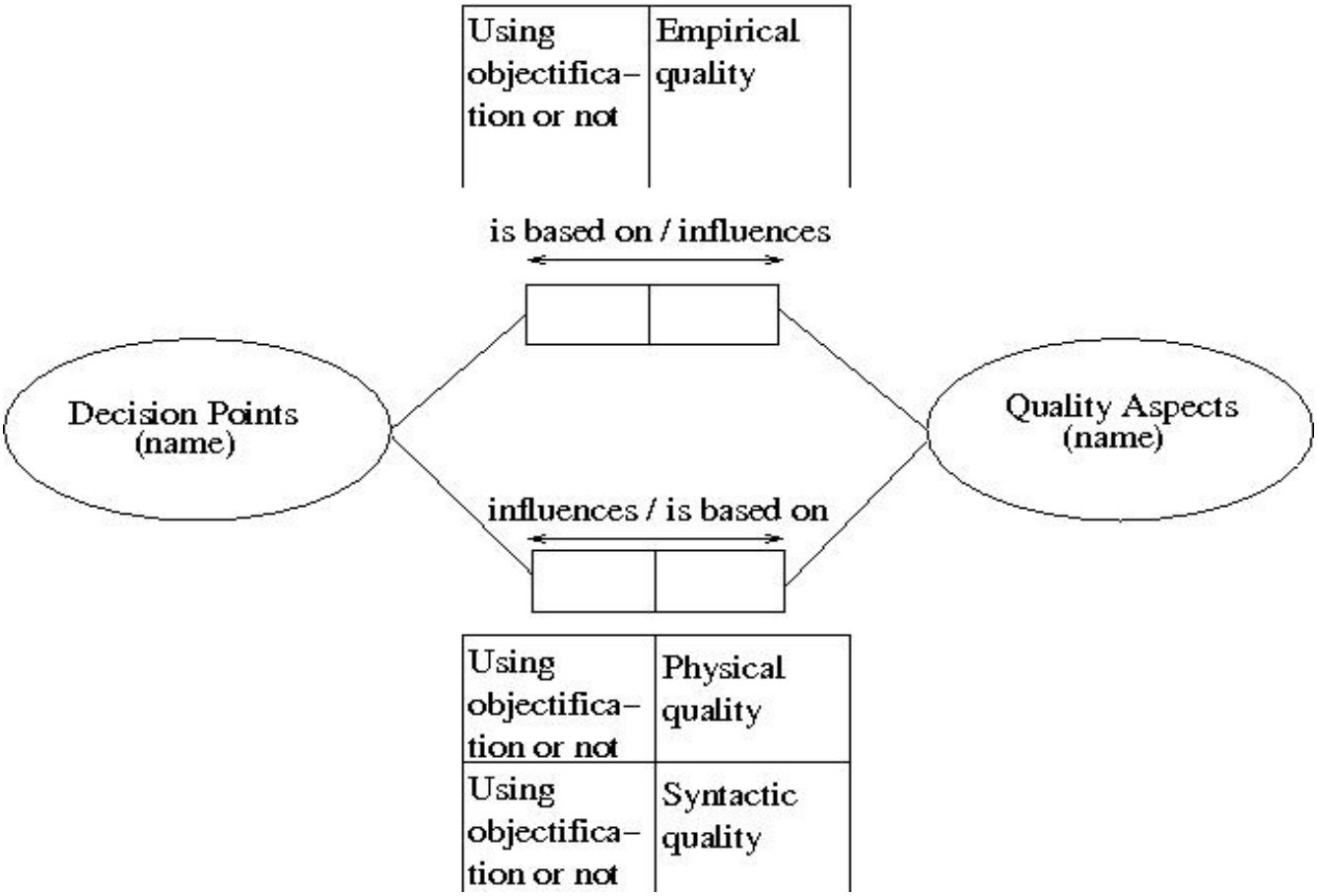


Figure 5.11: Using objectification or not, depends on the empirical quality. This decision influences the physical quality and the syntactic quality.

When a choice is made between using n-ary fact types or objectification, one can say something about the physical, empirical and syntactical quality. If one makes the wrong choice, these quality issues can be bad. If one does make a good choice, these quality issues can be good. It depend on the choice, to make any judgment about these quality issues.

When one makes the choice between using n-ary fact types or objectification the wrong way, it might lead to more complex models, although in my opinion the consequences aren't really dramatic. A consequence can be that the modeler must explain its model verbally, which may not be necessary if he had made a good choice.

CHAPTER 6

TRIM SCHEMA; NOTE BASIC DERIVATIONS

Value subtyping

Entities can play a certain role in the Universe of Discourse. When this is the case, one has to classify this entity into specialized types. For example: *"The Employee with first name 'Ann' is working for / has as worker Department with name 'Finance'"* (figure 4.1) is the sentence we use in this paper to indicate that there are Employees working for the Finance Department. In the Universe of Discourse it may be necessary to describe that only managers have a company car: *"The Manager with first name 'Ann' leases / is leased by a Company-car with license number 'YJ-37-OP'"*. It may also be necessary to describe females must wear long skirts: *"The Female with first name 'Ann' must wear / is worn by Clothes of type 'long skirt'"*. The result is described in figure 6.1. Employee is considered to be a supertype. Manager and Female is considered to be a subtype, since Managers and Females also are Employees. In this case subtypes may overlap: they are not mutually exclusive.

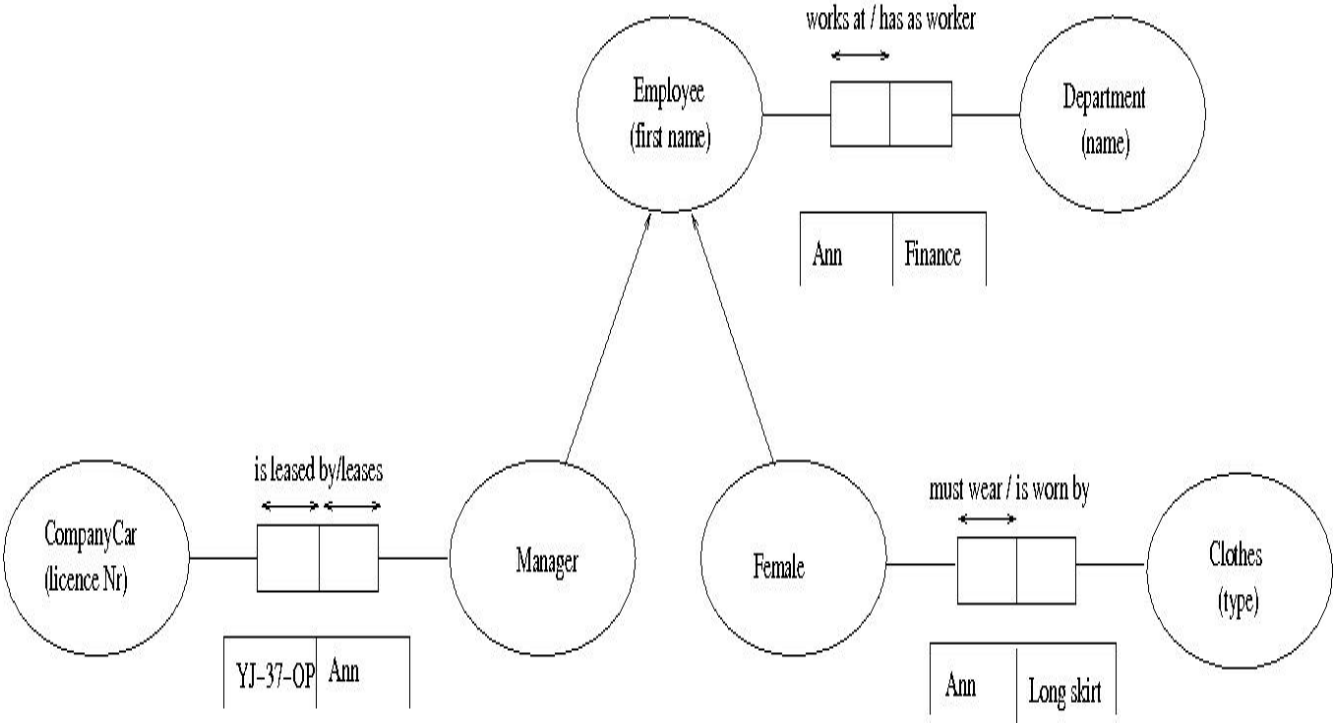


Figure 6.1: "The Employee with first name 'Ann' is working for / has as worker Department with name 'Finance'" and "The Manager with first name 'Ann' leases / is leased by a Company-car with license number 'YJ-37-OP'" and "The Female with first name 'Ann' must wear / is weared by Clothes of type 'long skirt'"

Like subtypes, value types can also overlap. For example: *"The Person with surname 'Paris' is living in the city with name 'Paris'"*. We can draw value types like subtypes (figure 6.2), but this isn't done in ORM.

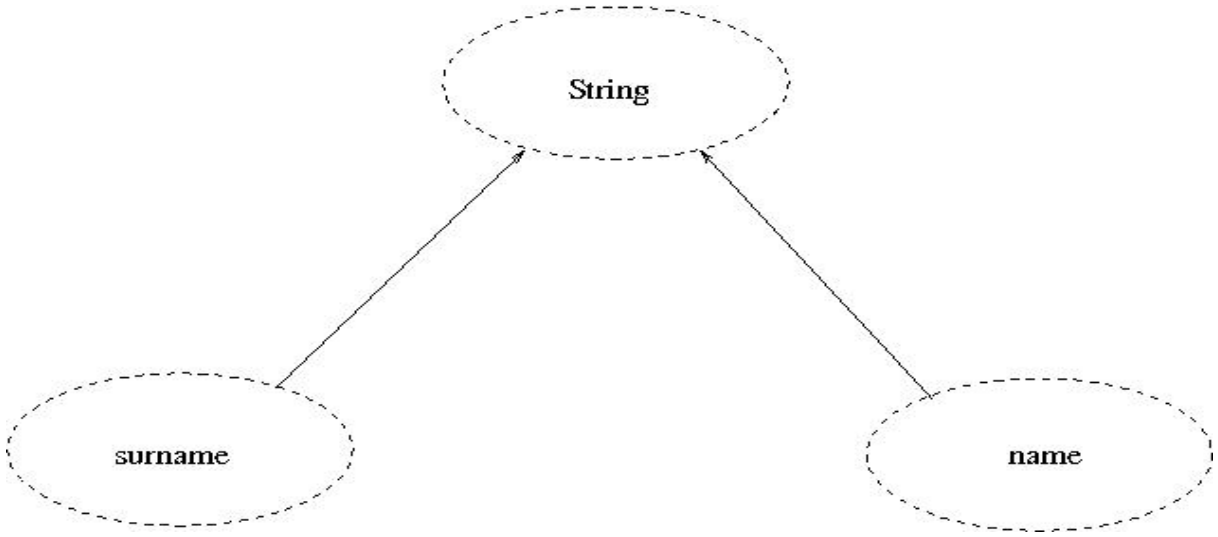


Figure 6.2: explicit subtype drawing

Halpin says: “Primitive entity types never overlap”[28], which means they are always mutually exclusive, which is in our example (figure 6.1) the entity 'Employee'. According to Halpin subtypes are not mutually exclusive.[29] This is also the same with value types. The difference is the following according to Halpin: “Value types often overlap, but are still shown separately on a schema diagram, since they are implicitly assumed to be a subtype of String(figure 6.2).”[30] In our example surname and name may overlap because they have common instances. Halpin concludes with saying: “Although the explicit depiction of value subtyping or value type overlap may clarify the situation, for compactness we leave this implicit.”[31]

The reason for leaving value subtyping implicit is according to Halpin to make the model more compact, which is true: too much information might lead to complexity. This indicates the decision to leave subtyping implicit is based on *empirical quality* as complexity is reduces. This decision influences the *physical quality*, as it influences the way artifacts are presented and *syntactic quality*, because the presentation of the reference mode must be conform the syntax (figure 6.3).

28 Halpin, page 94

29 Halpin, page 94

30 Halpin, page 94

31 Halpin, page 94-95

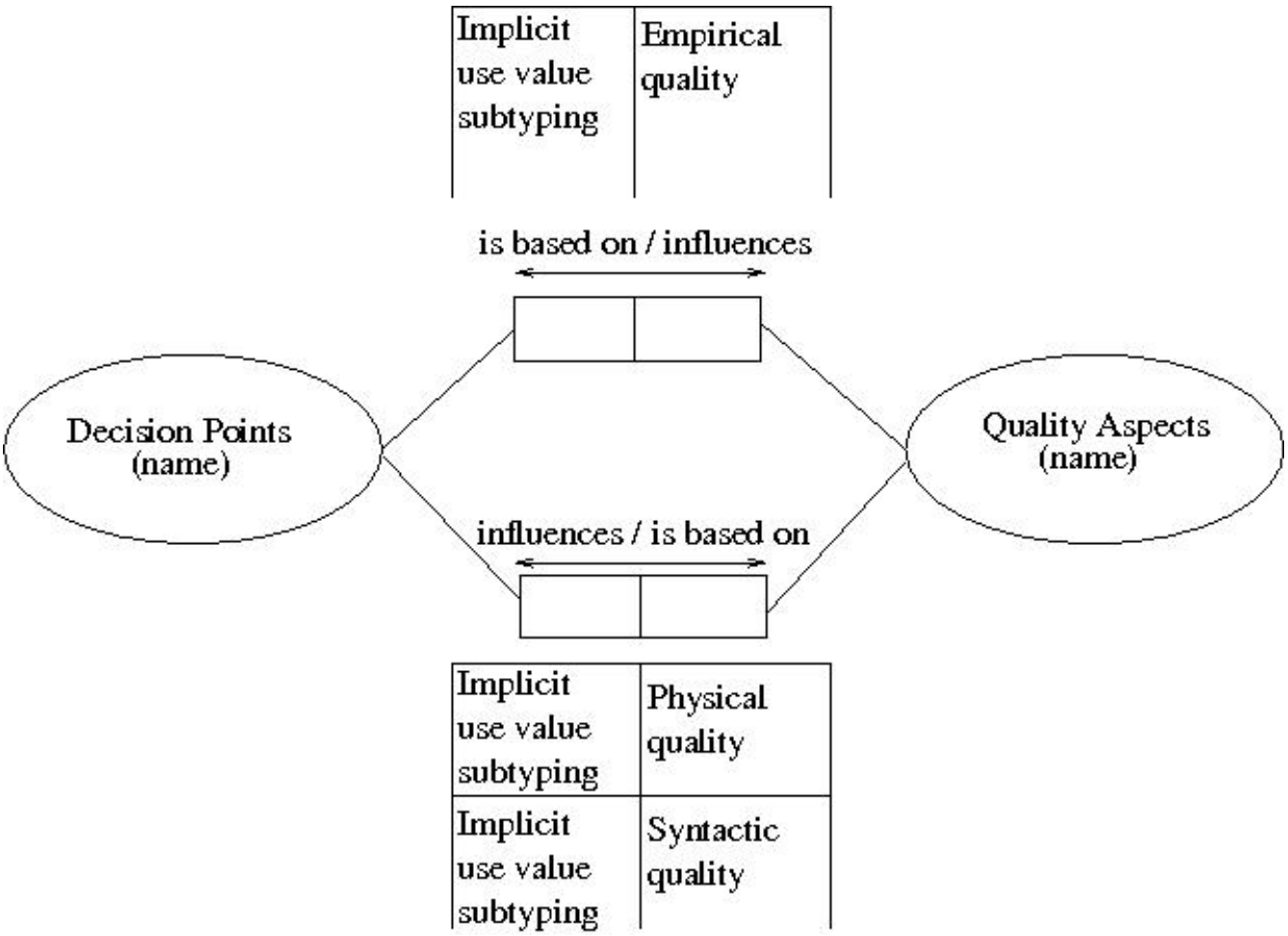


Figure 6.3: The implicit use of value subtyping, is based on the empirical quality. This decision influences the physical quality and the syntactic quality.

When the implicit use of value subtyping is performed, one say something about the empirical, physical and syntactic quality. It depends on the the size of the model, the amount of times explicit value subtyping is used, to make any judgment about these quality issues.

One can decide to draw the supertype of the implicit value subtypes (explicit use of value subtyping). This isn't really damaging, because it can clarify the type of value subtypes (string, character, integer, etc.) in less complex models or a small Universe of Discourse. In large Universes of Discourse or if a model might expand in the future, the use of explicit value types is not advised. It might lead to unnecessary complex models. The impact is not so very large when drawing explicit value subtypes, but it might lead to unnecessary complex models.

Combine entities

There is a possibility that some entity values are presented in more then one entity type. For example the sentence: *"The Employee with first name 'Ann' is working for / has as worker the Company with name '"Fiktyf"* and *"The Shareholder with first name 'Ann' has stocks in / is owned by the Company with name "Fiktyf".* Both shareholder and Employee refer to the same person (Ann). A possible schema could be figure 6.4. Problem with this figure is that one instance is involved in different entities.

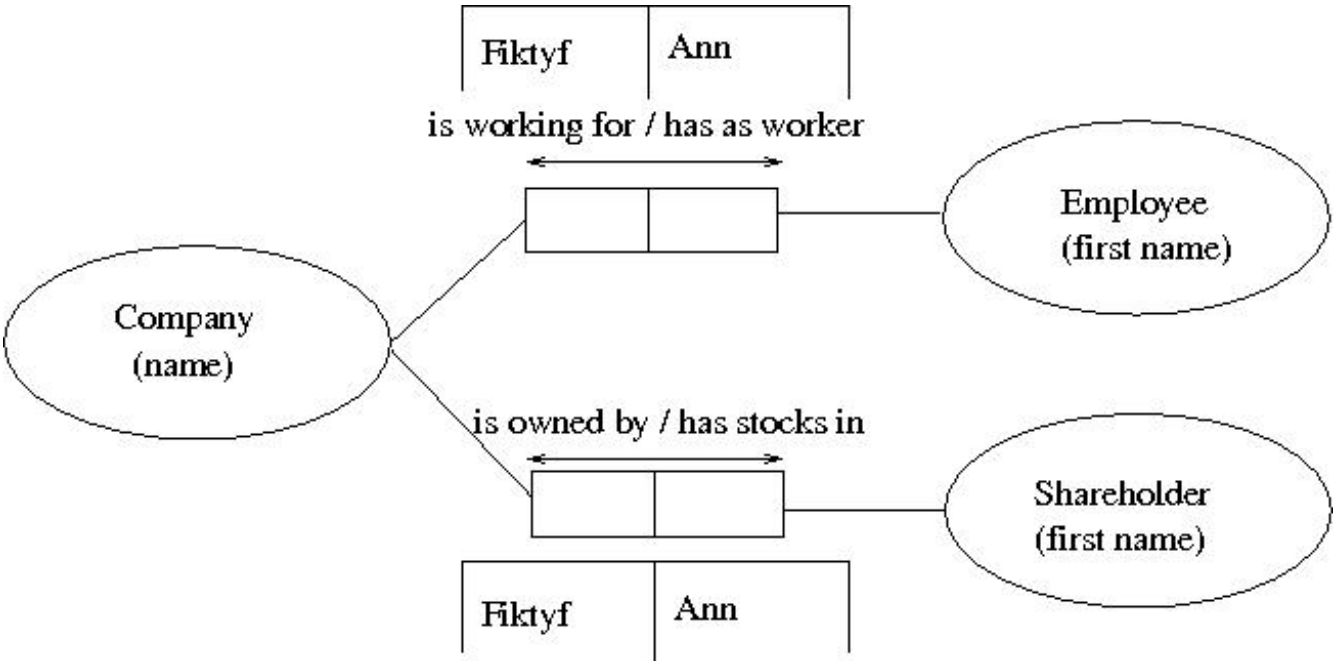


Table 6.4: *"The Employee with first name 'Ann' is working for / has as worker the Company with name '"Fiktyf"* and *"The Shareholder with first name 'Ann' has stocks in / is owned by the Company with name "Fiktyf" (incorrectly drawn)*

Halpin says: "Primitive types never overlap".[32] This means that these types must be mutual exclusive. Halpin suggest to ask a domain experts if your in doubt whether instances in entity types refer to the same object in the Universe of Discourse.[33] An exception occurs when using subtyping: if a fact holds only for shareholders, an additional subtype shareholders is created. This is not the case here. Halpin also says: "one reason for suspecting that two entity types should be combined is if they both have the same unit-based reference mode". In our example both shareholder and employee have the same reference mode (first name). A syntactical correct schema is presented in figure 6.5.

32 Halpin, page 94

33 Halpin, page 95

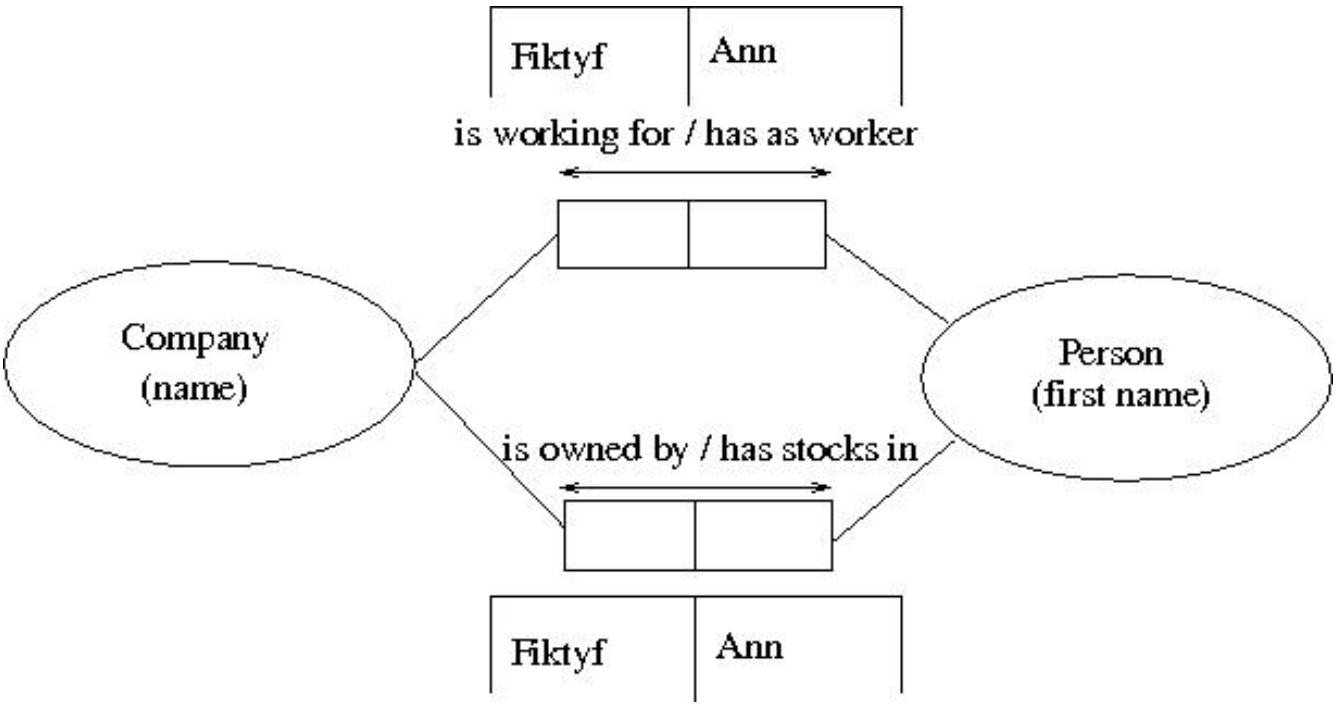


Table 6.5: *"The Employee with first name 'Ann' is working for / has as worker the Company with name '"Fiktyf"* and *"The Shareholder with first name 'Ann' has stocks in / is owned by the Company with name "Fiktyf" (correctly drawn)*

The reason to combine entities if instances are the same is based on syntactical and empirical reasons. Halpin clearly says primitive types must not overlap. What he doesn't say is that it reduces the complexity of the model. If a fact is based on more entities, when one entity is enough, the complexity of the model is reduces. It is based on both syntactic and empirical quality. It also influences syntactic quality, because instances cannot be present in two entities.

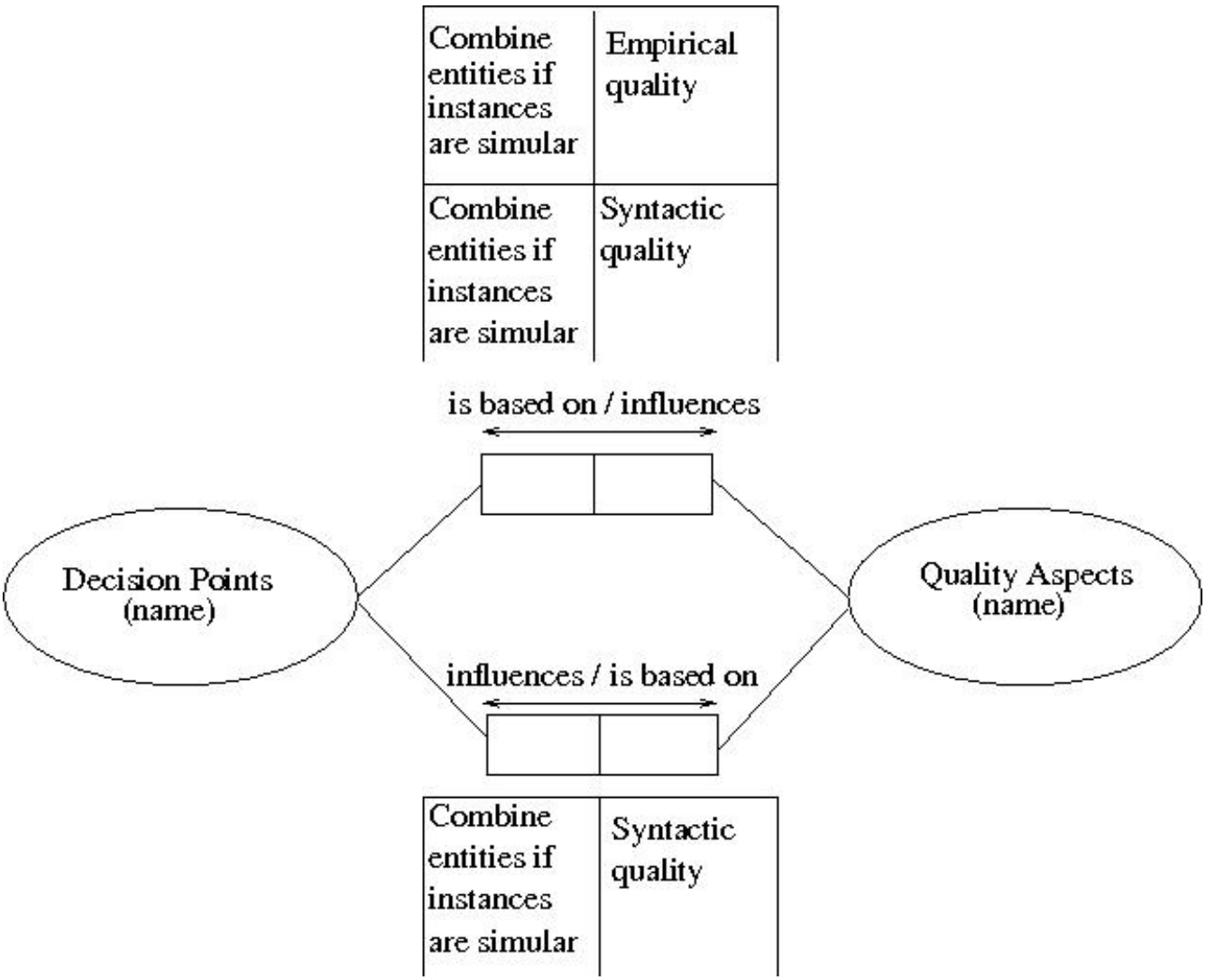


Figure 6.6: The decision to combine entities if instances are the same is based on the syntactic quality. This decision also has influence on the same syntactic quality.

When entities are combined, because instances are the same, one say something about the syntactic and empirical quality. It depends on the the size of the model, the complexity of the model and the amount of times combining is used compared with the amount of times combining should be used, to make any judgment about the syntactic and empirical quality.

When combining is not used, when it should be used, the complexity can increase. The model is syntactically not correct if this is the case. The impact of a syntactically wrong and complex model is not really great. There is no damage when interpreting such a model. Like in 'the purpose of formalizing' (step 2), there is some damage for the modeler, because his job is to model according to a certain syntax.

Derivation rules

A value can be derived by other values. Consider the following three sentences (figure 6.7):

1. *"The Employee with first name 'Ann' has the Age of years '40'".*
2. *"The Employee with first name 'Ann' must work for a Working period of years '25'".*
3. *"The Employee with first name 'Ann' is retired at the Retirement age of years '65'".*

One can see clearly that one entity can be derived from another entity. The age can be derived from working period and retirement age. The working period can be derived from the age and the retirement age. The retirement age can be derived from the age and the working period.

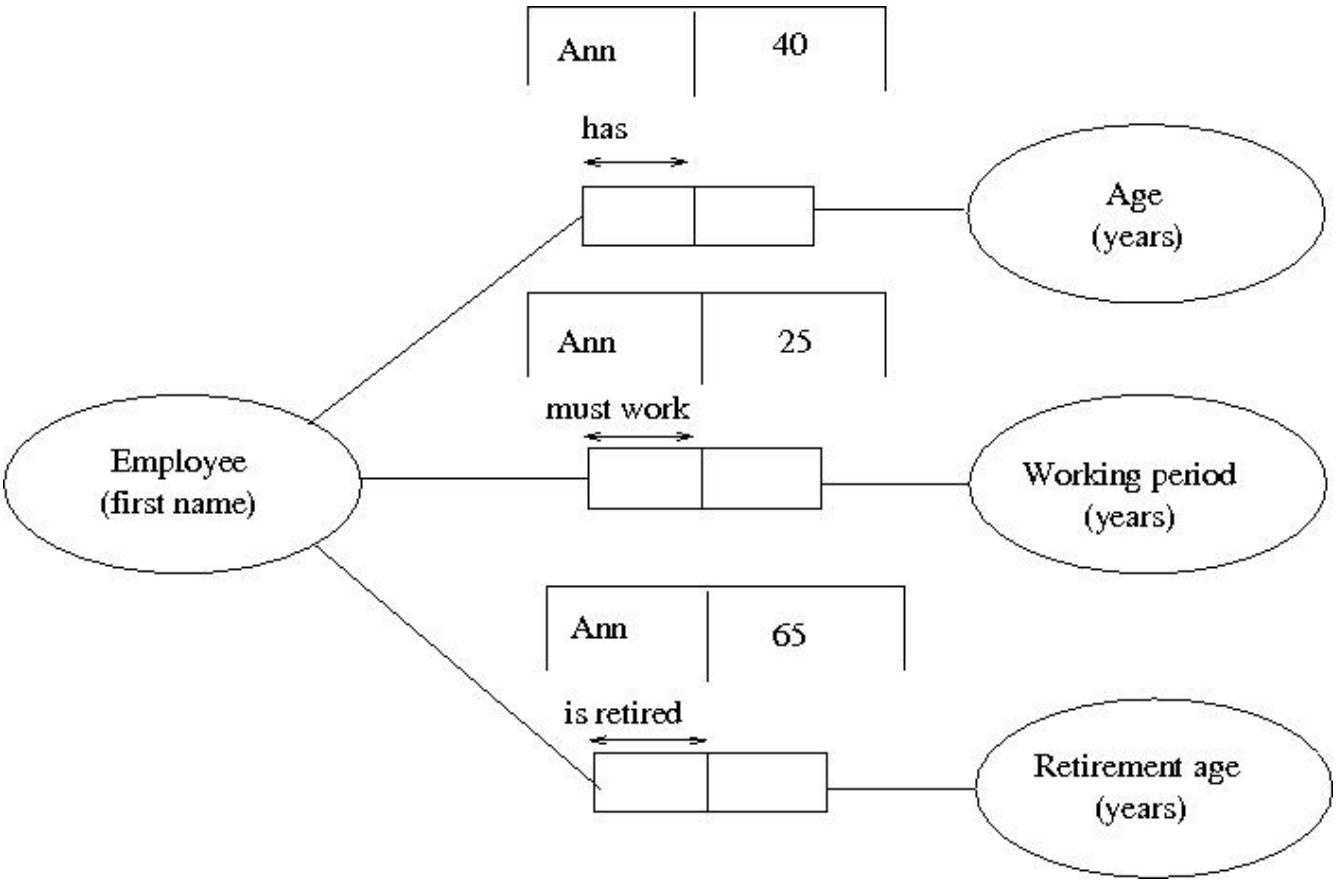
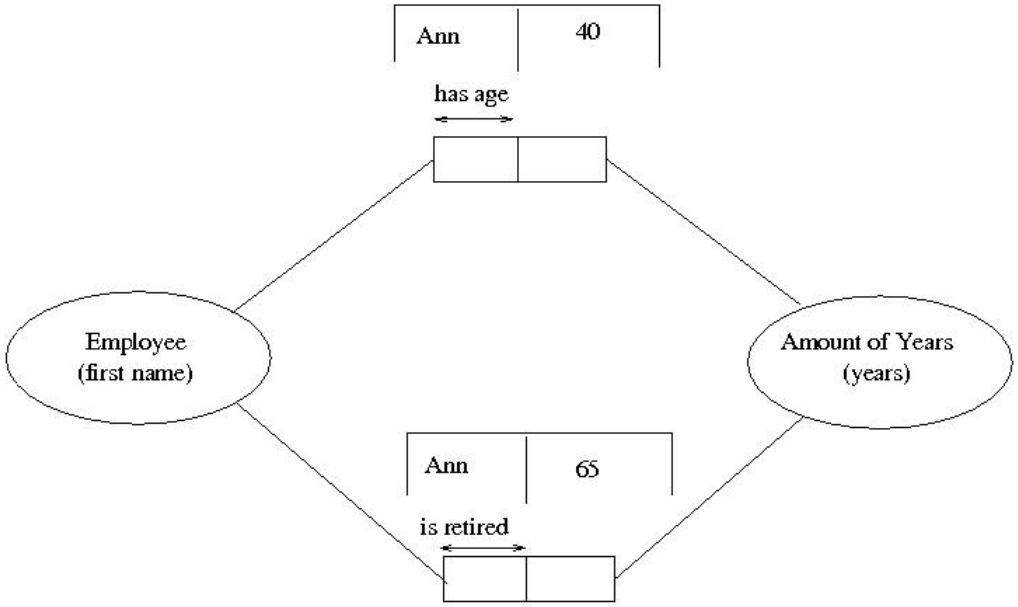


Figure 6.7: "The Employee with first name 'Ann' has the Age of years '40'" and "The Employee with first name 'Ann' must work for a Working period of years '25'" and "The Employee with first name 'Ann' is retired at the Retirement age of years '65'" (incorrectly drawn)

Halpin says: “in most cases, we usually decide beforehand that one specific fact type will always be the derived one.”[34] So it doesn't matter which is the derived fact type (definiendum) He also writes: “To minimize the chance of human error, we have the system derive this value rather than have humans compute and store it”.[35] About drawing these derivation rules Halpin argues that these rules can be written formally, for example by using ConQuer, and informally, for example by writing comment in branches.”[36] An example is given in figure 6.8.



1. {Working period = Retirement age −/− age}

2. Employee works for Working period of Amount of Years iff Employee has Amount of Years1 and
Employee is retired at Amount of years2 and
Amount of years = Amount of Years1 −/− Amount of Years2

Figure 6.8: Example with derivation rule in informal text (1) and using ConQuer (2).

The decision not to include to derived facts is according to Halpin because it minimizes human error. There are systems that can do this for you. Not doing this, might lead to different interpretation. Some derived facts may be hard to compute by the human brain, so some humans may have a different interpretation. One can say this decision is based on the *quality of socio-cognitive interpretation.* This decision influences both physical and syntactical quality. The *physical quality* because artifacts in the model are presented differently and *syntactic quality* because derived facts are not allowed (figure 6.9). We will see later that in some special cases they are allowed.

34 Halpin, page 98

35 Halpin, page 97

36 Halpin, page 97

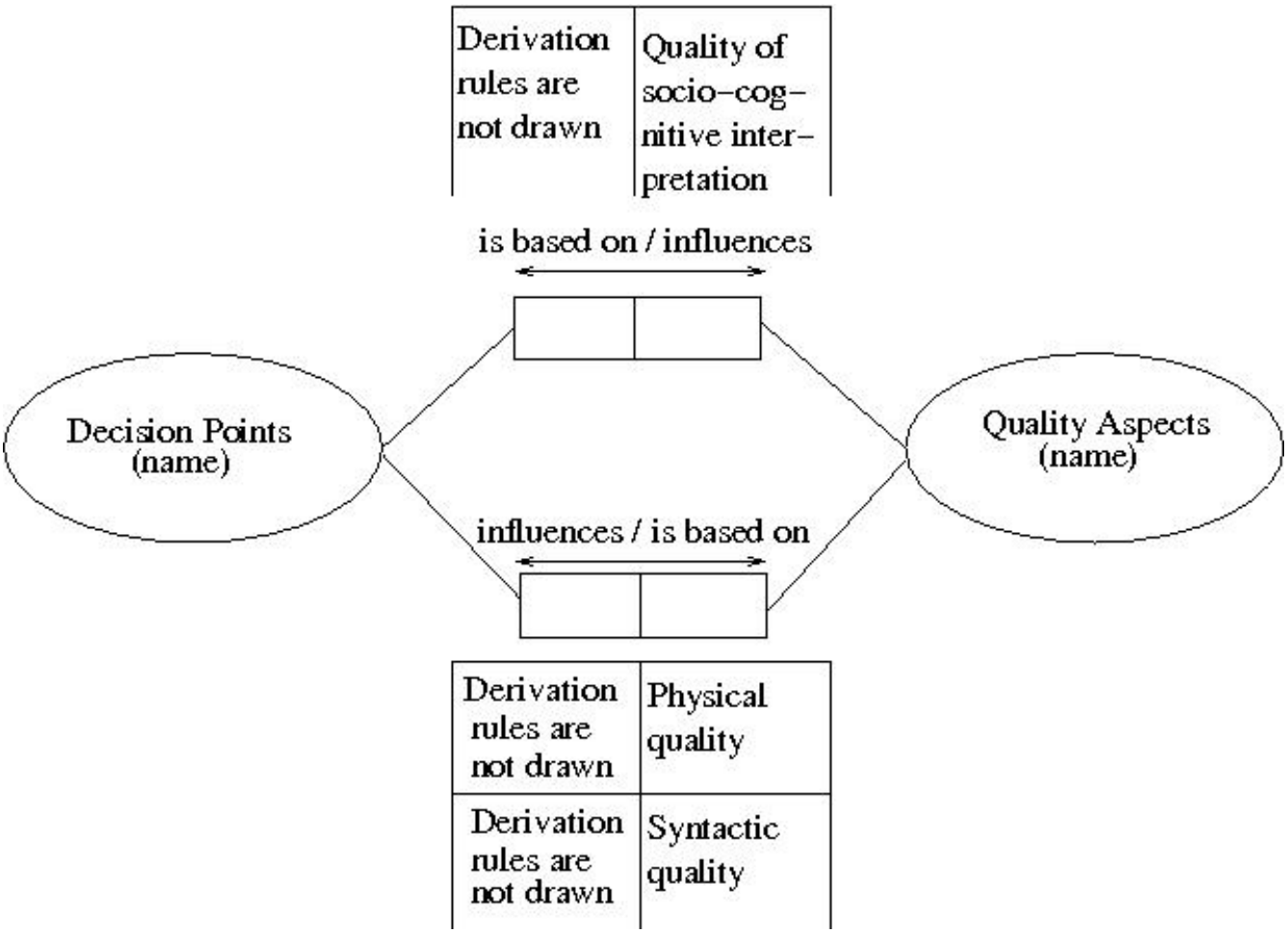


Figure 6.9: The decision not to draw derivation rules, is based on the quality of socio-cognitive interpretation. This decision influences the physical quality and the syntactic quality.

When derivation rules are not drawn, but are in textual form (figure 6.8), one can say something about the quality of socio-cognitive interpretation, the physical quality and the syntactic quality. It depends on the the size of the model, the amount of times derivation rules are not drawn and the amount of times derivation rules are present in textual form, to make any judgment about these quality issues.

If one draws these derivation rules or leave any textual form out, it can have a significant impact. A rule may not be interpreted correctly by stakeholders. A misinterpretation might lead to a wrong implementation of the system. Computers are better in deriving facts, so there is no misinterpretation. To clarify the model it can be important to write the derivation rule in textual form.

Include drawing of derivation rules

Most of times derived fact types aren't drawn but are presented in a textual form. There are also examples where derived fact types must be drawn. Consider the hight, width and area of a piece of land. The length and width of this piece of land are 20 by 30 meters. Which makes it an area of 600 square meters. The area can be derived by its length and width. If a piece of land could have a length of 600 meters, it is a different measure than 600 square meters, although the area is derived from its length and width.

Halpin acknowledge these two measures can't be compared, because 'area' is a different type of quantity. Derived fact types with different types of quantity must be kept separate. It can be drawn, but it must be distinguished from a base fact type. To do this in ORM, an asterisk is placed beside any derived fact type that is included in the diagram.[37] Figure 6.10 presents an ORM scheme including these derived fact types.

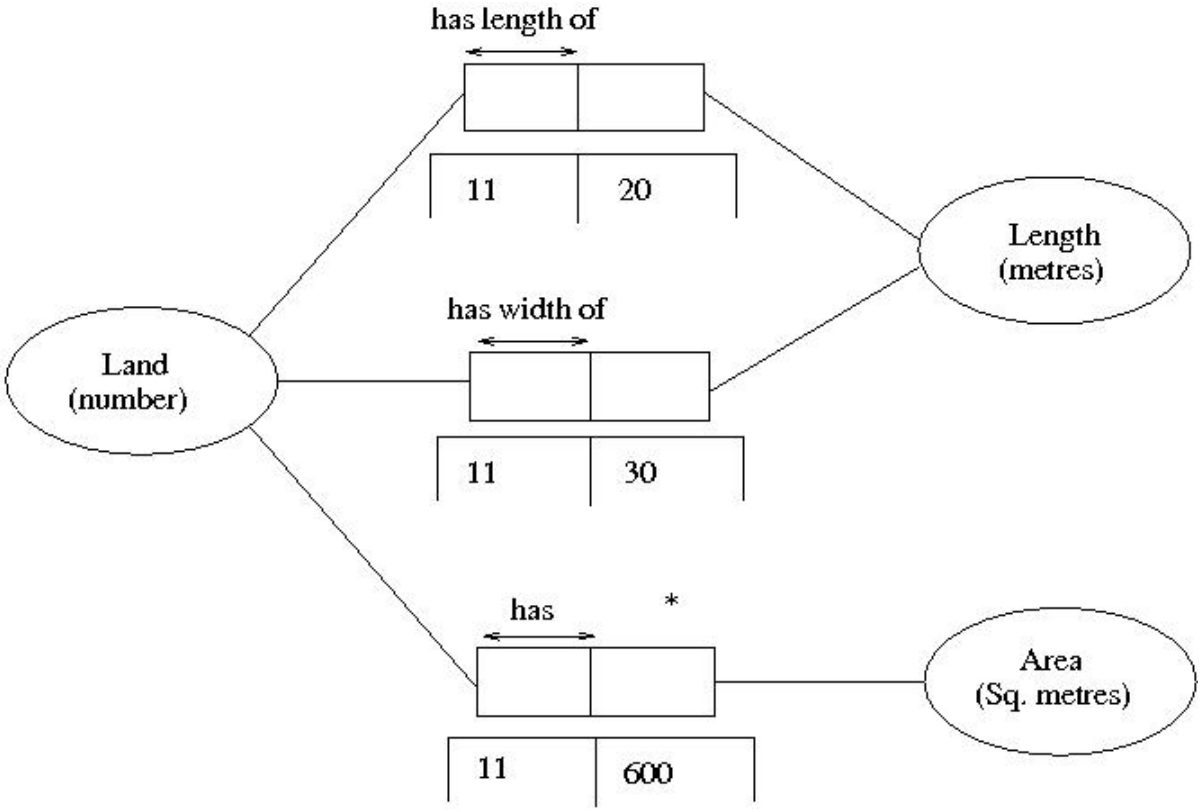


Figure 6.10: "Land with number '11' has length of 20 meters" and "Land with number 11 has width of 30 meters" and "Land with number 11 has area of 600 square meters".

37 Halpin, page 99-100

According to Halpin the decision to include the drawing of derivation rules in some special cases is based on *syntactic quality*, as one must present rules if they represent a different type of quantity (meters compared with square meters). It is also based on *pragmatic quality*, which isn't described by Halpin. It might lead to wrong interpretation. In this example, area could be interpreted as being 'metric', which isn't the case. This decision influences the syntax of a model, as it is obligatory to include derivation rules in a model, when it represents a different type of quality. The *physical quality* is also influenced because artifacts are absent or present in the model (figure 6.11).

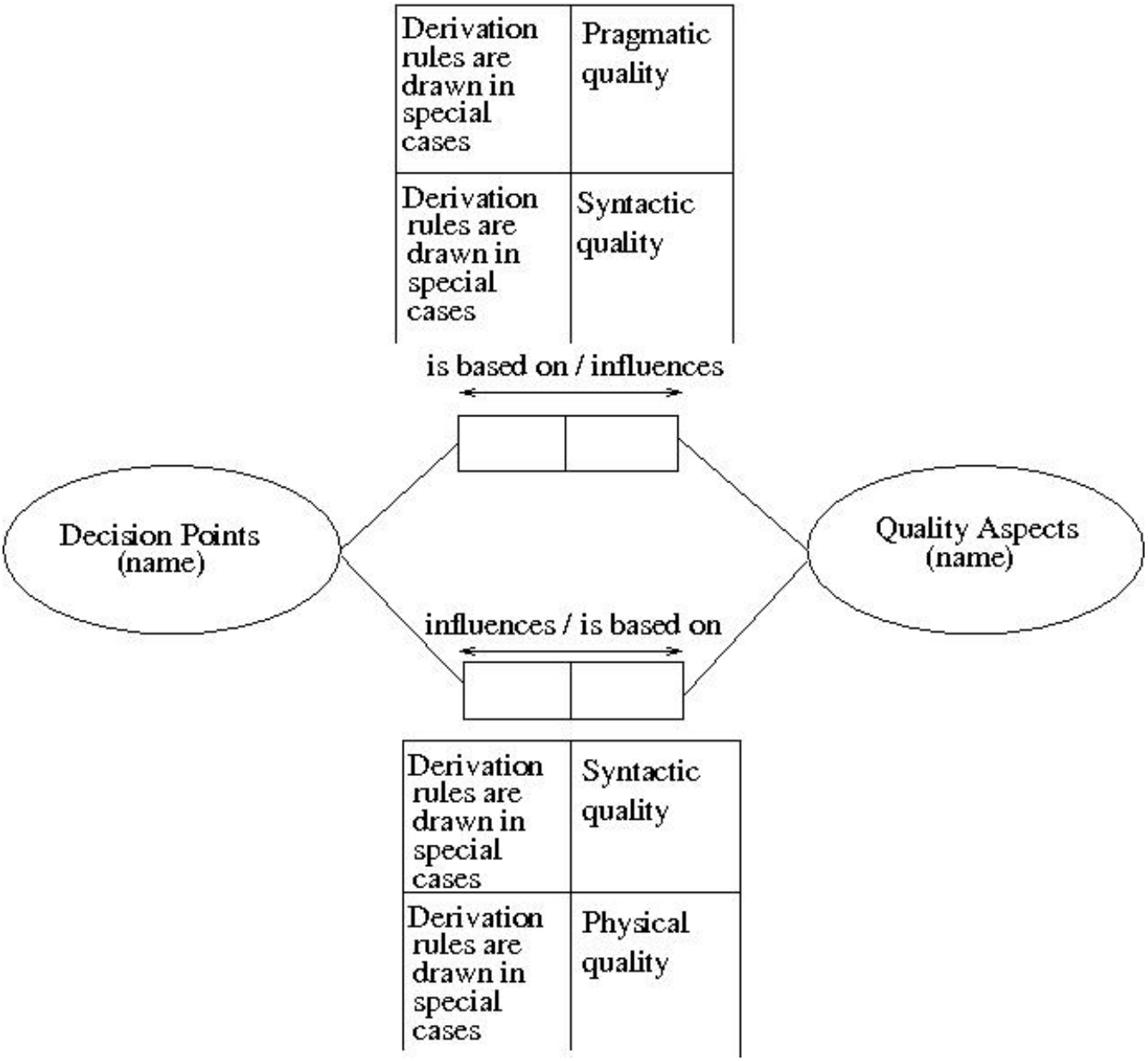


Figure 6.11: The decision to draw derivation rules in some cases, is based on the syntactic quality and pragmatic quality. This decision influences the physical quality and the syntactic quality.

When derivation rules are drawn, because they deal with different types of quantity, one can say something about the syntactic quality, pragmatic quality and the physical quality. It depends on the the size of the model and the amount of times derivation rules are drawn in these special cases, to make any judgment about these quality issues.

When these derivation rules are not drawn, this could have severe consequences, as your model can be incorrect. A stakeholder or a computer could interpret some entities as being of the same type. A computer could make wrong calculation, because it might compare apples with oranges. The impact is high, so one must apply this rule correctly.

Eager evaluation

Most of times computers don't store derived facts, because it can be computed one information is requested. It cost space, to store these facts. This is called 'lazy evaluation'. However, sometimes it is better to store derived facts. This is called 'eager evaluation'.

Halpin says in most cases lazy evaluation is preferred, but sometimes eager evaluation is chosen because it offers significantly better performance. Halpin suggest using a double asterisk to indicate this choice (in stead of a single asterisk)[38]

The decision to draw derived fact types, when eager evaluation is applied, is based on *quality of technical interpretation,* as it stimulates the performance of a machine. This decision influences the *physical quality*, because the artifacts in the model are presented differently (a double asterisk). It doesn't influence the syntactic quality, because both lazy evaluation and eager evaluation are syntactically correct (figure 6.12).

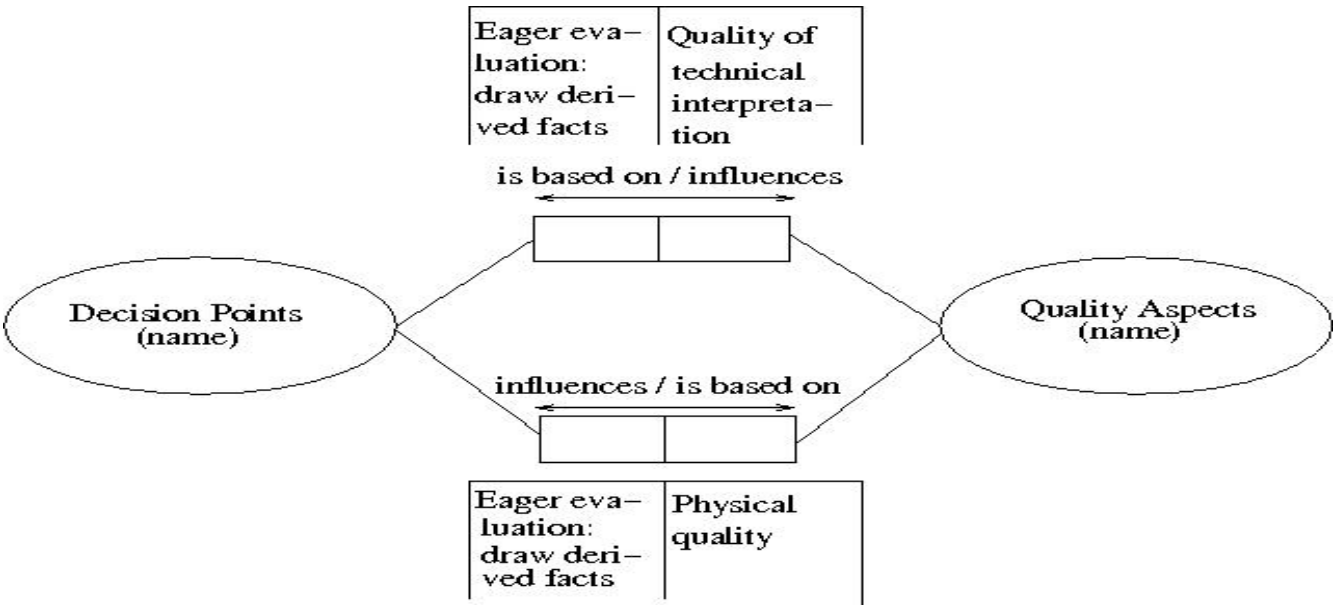


Figure 6.12: The decision to draw derivation rules when applying eager evaluation, is based on the quality of technical interpretation. This decision influences the physical quality.

38 Halpin, page 100-101ties could

When derivation rules are drawn with a double asterisk, because eager evaluation is applied, one can say something about the quality of technical interpretation and the physical quality. It depends on the the size of the model and the amount of times derivation rules are drawn in these special cases, to make any judgment about the quality of technical interpretation and the physical quality.

When these derivation rules are not drawn, this could have consequences for the performance of a computer. When performance is crucial it has high impact. When performance is not an issue, it has low impact. The choice between eager evaluation and lazy evaluation depends on your priorities (size or performance).

Different entities with same kind of information

There is a choice to combine entity types if these types hold the same kind of information. Consider the following three sentences (figure 6.13):

- *"The Finance Department Employee with name 'Ann' is of Sex with code 'Female'"*
- *"The Production Department Employee with name 'Bob' is of Sex with code 'Male'"*
- *"The Facility Department Employee with name 'Claire' is of Sex with cod 'Female'"*

All these different entities (... Department Employee) have the same kind of information (their sex). One can also combine these sentences which result in figure 6.14, which include value constraints (Finance Department Employee, Production Department Employee, Facility Department Employee).

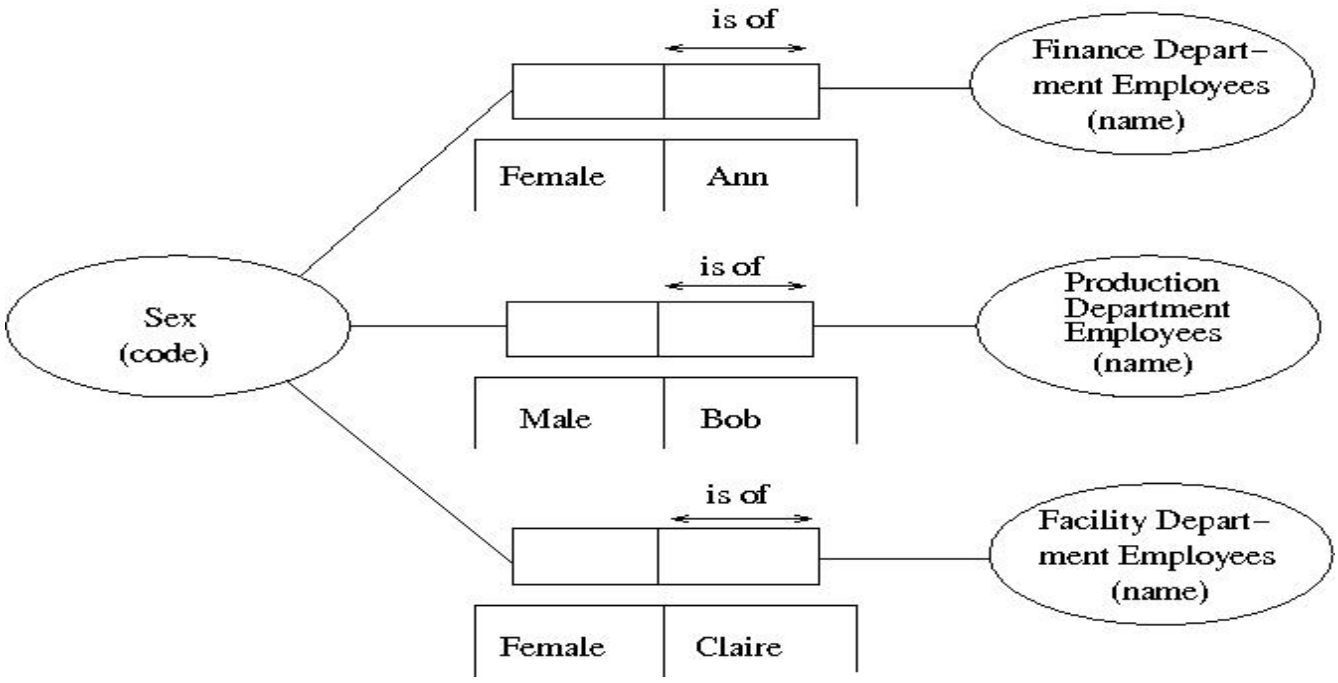


Figure 6.13: "The Finance Department Employee with name 'Ann' is of Sex with code 'Female'" and "The Production Department Employee with name 'Bob' is of Sex with code 'Male'" and "The Facility Department Employee with name 'Claire' is of Sex with cod 'Female'"

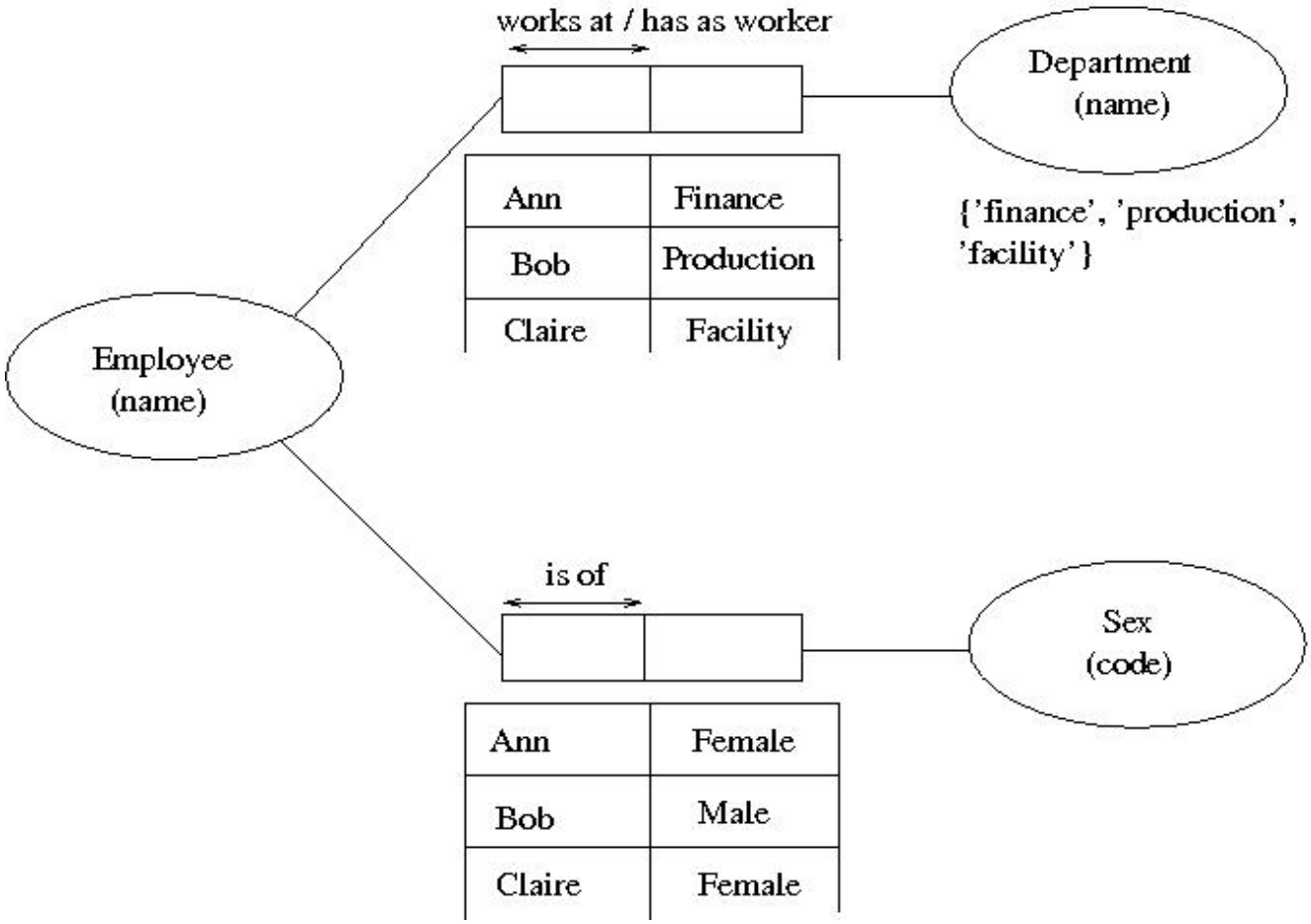


Figure 6.14: Combining figure 6.13

About the choice to combine entity types Halpin says: “In such cases, we ask ourselves the following question: Do we ever want to list the same kind of information for the different entity types in the same query?”[39] In this case: do we want to make the request “List all the Employees in the same query?” If so, we should combine the entity types (figure 6.14), if not, we shouldn't combine (figure 6.13). Halpin also says subtyping can also be used, because additional information is required for specific kinds of employees.[40]

The decision which path to follow (combining or not) is according to Halpin based on *organizational quality,* as it links back to organizational goals. For example: an organizational goal could be to increase the control of its departments, by centralizing the organization. Therefore overall human performance reports must be generated. This is done by listing all the employees in the same query, which is a decision to combine the entities. It doesn't have syntactic consequences, because both choices (combining or not combining) are syntactically correct. This decision influences the *physical quality*, because the presentation of the artifacts depend on the choice being made (figure 6.15).

39 Halpin, page 101

40 Halpin, page 101

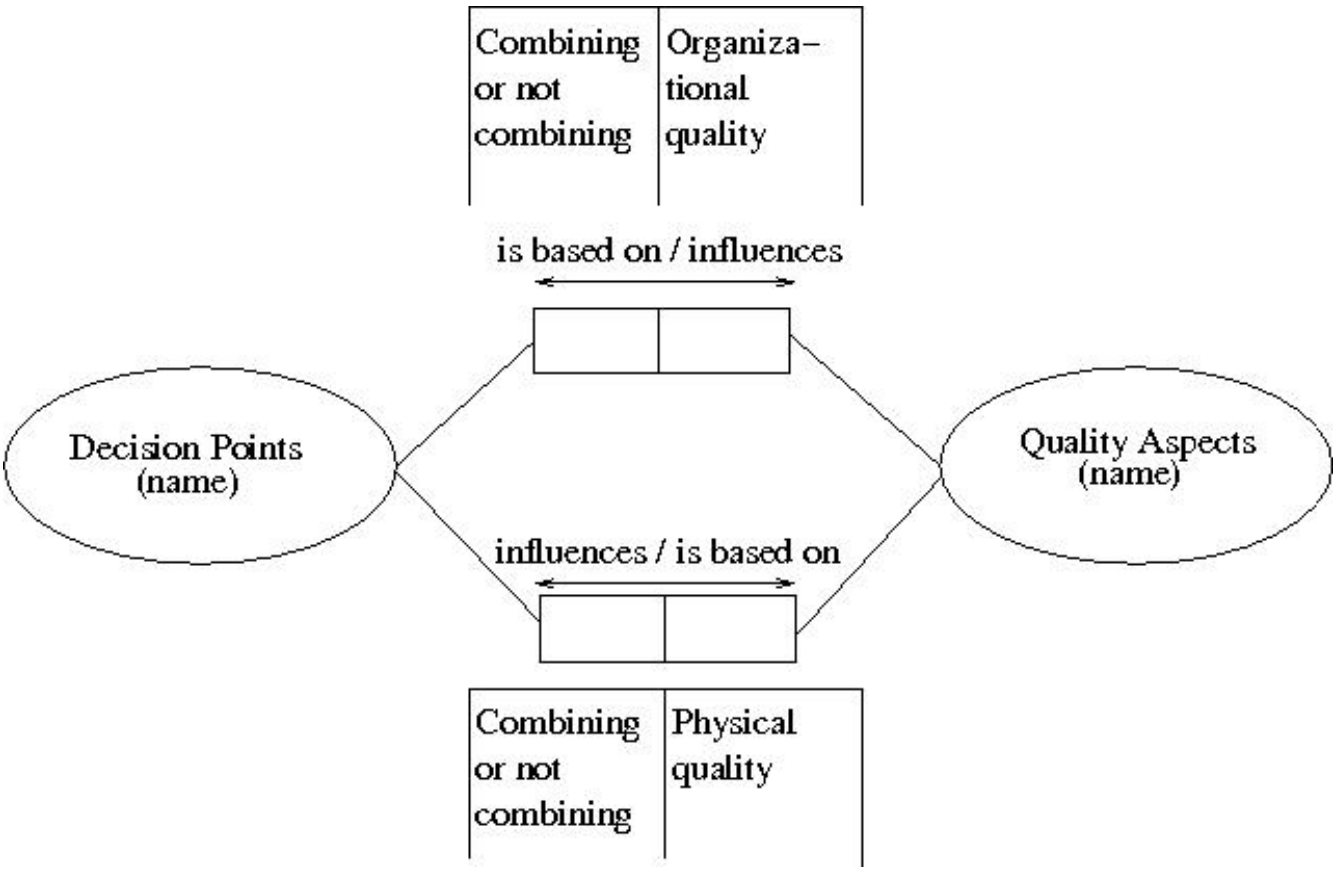


Figure 6.15: The decision to combine entities or not, is based on the organizational quality. This decision influences the physical quality.

When the choice is made to combine entities (or not to combine entities), because different entities have the same kind of information, one can say something about the organizational quality and the physical quality. It depends on the match between the organizational goals and the choice being made, to make any judgment about the organizational quality and the physical quality.

When a choice is made to combine entities, because different entities have the same kind of information and this choice is not matching the organizational goals, it might lead to lots of organizational difficulties. For instance a centralized information system is being made, when a decentralized information system is required. The centralized information system represents figure 6.14 as it needs a list of all employees. A decentralized information system represents figure 6.13 as it doesn't need a list of all employees, because the departments are highly autonomous. It is therefore to make a good choice matching these organizational goals.

CHAPTER 7

DISCUSSION

In the first step of the CSDP most decisions are based on pragmatic (interpretation) issues and semantic (completeness and validness) issues. It is a very important step, as a modeler must obtain the correct input, via verbalizations, in collaboration with the domain expert. Sadly, modelers frequently jump to the next step (Proper 2006).This implies its consequences for the quality of the outcome, especially the pragmatic and semantic quality. Proper says: “Achieving conceptual clarity and consensus among stakeholders is an important yet often neglected part of system development, and requirements engineering in particular” (Proper 2006). One can say skipping the first step can lead to a model, which is not representing the domain and a model, which can lead to many misinterpretations by humans and computers.

Is it also true that one quality aspect has influence on other quality aspects. For instance a model which is not representing the domain can lead to misinterpretation by stakeholders. Also misinterpretation by stakeholders can lead to disagreement of stakeholders. One can say an impact on semantic quality can lead to an impact on the quality of socio-cognitive interpretation. An impact on the quality of socio-cognitive interpretation can lead to an impact on the social quality (Appendix B).

In most cases wrong semantic quality, can lead to a wrong implementation of a system. It is therefore very dangerous to jump to the next step, because in step 1, lots of decisions[41] are based semantic quality. Lots of decisions in step 1 are also based on the quality of socio-cognitive interpretation or the quality of technical interpretation (pragmatic quality)[42]. Not making the decisions correctly can lead to misinterpretation problems, which can lead to disagreement (social quality). It is therefore necessary not to skip step 1.

Lots of decisions are taken to be seriously, because making some decisions incorrectly can have severe consequences. On the other hand, when the Universe of Discourse is not very large and not too complex, making some 'bad' decisions doesn't necessarily lead to severe consequences, like in the case of clarifying entities. Be careful though, almost all decisions based on semantic quality, doesn't have this property. Therefore in these cases it is wise to follow the described path, because it might lead to models, not representing the domain.

In most cases the domain size does matter to make any judgment about the quality based on a certain decision. For instance, if a modeler forgets to populate a certain fact type in a very large model, one might still conclude the quality of socio-cognitive interpretation, syntactic-, semantic and social quality as being good. If the domain is very small, the quality can decrease.

41 Using inverse predicates if predicates are the same, splitting multiple facts which lead to loss of information, feedback.

42 Elementary facts, clarify entities, predicates, feedback

In all steps, most decisions influences the syntactic quality. If a decision is made it implies consequences for the syntax of a model. Some decisions don't have any consequences.[43] These decisions are more or less optional, in stead of obligatory. Most decisions in step 2 and step 3 influences the physical quality. This is no surprise, because step 2 and 3 is about modeling and step 1 is about transforming information in elementary facts. In step 2 and 3 most decisions imply artifacts being drawn differently, so a decisions has its impact on the physical quality. In these steps one can also notice lots of decisions are based on empirical issues.[44] The purpose of most of these decisions is not to create complex models.

There are almost no decisions based on knowledge quality, organizational quality, domain quality and ethical quality . These quality aspects are relevant, but not in our research domain (identifying fact types). For example a modeler can think about whether he should be working for a dictatorial government (ethical quality). This doesn't have anything to do with identifying fact types.

43 Feedback, synonyms, modeling the current situation, vague predicates, populate, eager evolution, different entities with same kind of information.

44 Drawing reference mode, objectification, value subtyping, combine entities.

CHAPTER 8

CONCLUSION

As described in chapter 3, we now will present the populated framework (figure 8.1), with several linkages between decision points and quality aspects. As described in chapter 7, there are also quality aspects influencing each other. Knowledge- and ethical quality are not relevant for this domain. Lots of decisions in step 1 are based on semantic- and pragmatic quality aspects. The latter includes quality of socio-cognitive interpretation and the quality of technical interpretation. Semantic quality can have severe consequences, therefore it is recommended not to skip step 1. Lots of decisions in step 2 and 3 are based on empirical quality. Therefore these steps are necessary for creating models which aren't too complex. Pragmatic quality, and especially the quality of socio-cognitive interpretation are highly relevant quality aspects in all steps. Not applying decisions based on these quality aspects, might lead to misinterpretation and disagreement, because there is a strong connection between the quality of socio-cognitive interpretation and social quality. The population of the framework is shown in Appendix B. In most cases there is a linkage between decision points influencing quality aspects, because a decision can influence the syntax (or physical aspects). Optional decisions don't have this linkage. Obligatory decisions do have this linkage.

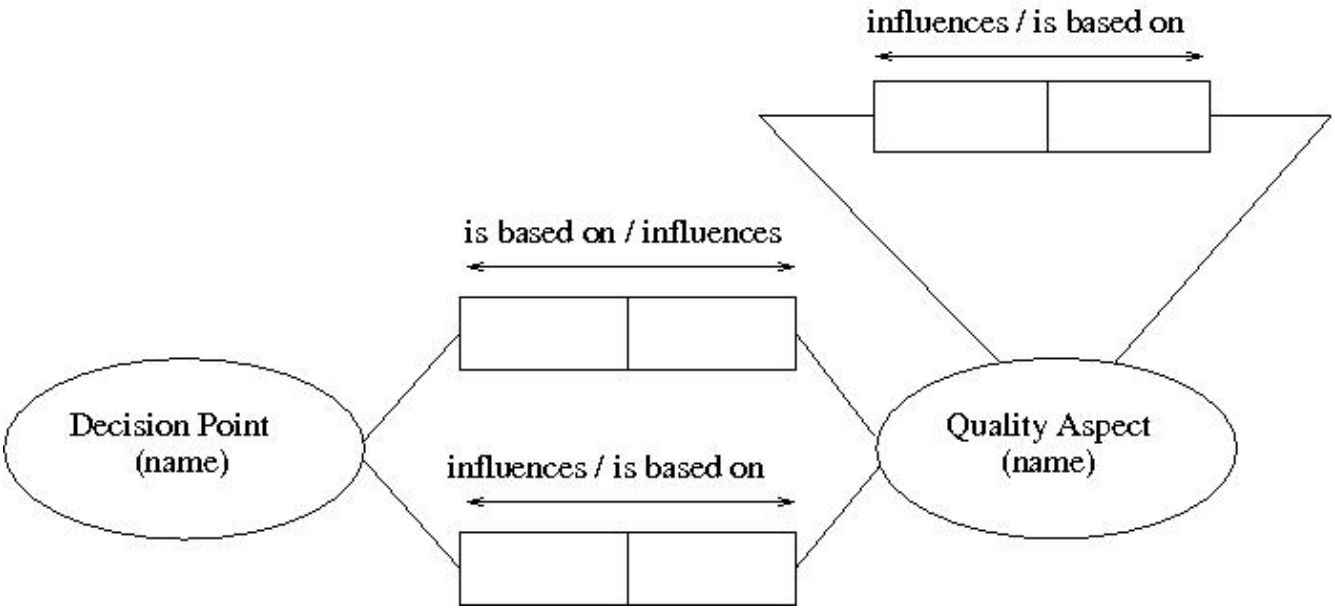


Figure 8.1: Quality aspect (name) influences / is based on Decision point (name) , Decision point (name) influences / is based on Quality aspect (name) , Quality aspect (name) influences / is based on Quality aspect (name).

CHAPTER 9

REFERENCES

- van Bommel P., Hoppenbrouwers S.J.B.A. and Proper H.A., *QoMo: A Modelling Process Quality Framework based on SEQUAL,* Workshop of conference in advanced information systems engeneering, Trondheim, Norway, 2007.
- Halpin T., Information Modeling and Relational Databases: from conceptual analysis to logical design, *Morgan Kaufmann Publishers,* San Francisco, USA, 2001.
- ter Hofstede A.H.M., Proper H.A. and van der Weide T.P., *Fact Orientation in Complex Object Role Modeling Techniques,* Proc. 1st Int. Conf. on Object-Role Modeling (ORM-1), page 45-59, Magnetic Island, Australia, 1994.
- Krogstie J. and Jørgenson D., Quality of interactive models, *Lecture Notes in Computer Science 2784,* page 351-363, Springer, Berlin, Germany, 2003.
- Krogstie J., Sindre G. and Jørgenson H., Process models representing knowledge for action: a revised quality framework, *European Journal of information systems,* Vol 15, page 91-102, 2006.
- Lindland O.I., Sindre G. and Sølvberg A., Understanding quality in conceptual modeling, *IEEE Software*, Vol.: 11 (2), page 42 – 49, 1994.
- Moody D.L., Sindre G., Brasethvik T. and Sølvberg A., Evaluating the Quality of Process Models: Empirical Testing of a Quality Framework, *Lecture Notes in Computer Science 2503*, page 380-396, Springer, Berlin, Germany, 2002.
- Moody D.L., Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions, *Data & Knowledge Engineering,* Vol.: 55, page 243-276, 2005.
- Proper E., Architecture driven work systems engineering, *Lecture Notes,* RU Nijmegen, 2006.
- Roberts C.W., A Conceptual Framework for Quantitative Text Analysis, *Quality and Quantity,* Vol 34, page 259 – 274, 2000.
- Stamper R.K., Information in Business and Administrative Systems, *John Wiley & Sons*, New York, USA, 1973.

APPENDIX A

QUANTITATIVE TEXT ANALYSIS

These are the results of the quantitative text analysis. The rows represent the articles, the columns represent the variables. The numbers indicate the quantity of usages of a variable per document.

	Quality	Physical quality	Empirical quality	Syntactic quality	Semantic quality	Domain quality	Pragmatic quality
Stam & Proper	4	0	0	0	0	0	0
Frederiks & van der Weide	14	0	0	0	0	0	0
Proper et al.	3	0	0	0	0	0	0
van Bommel et al. 1	2	0	0	0	0	0	0
van Bommel et al. 2	3	0	0	0	0	0	0
Hindriks et al.	3	0	0	0	0	0	0
Proper	133	1	1	1	4	0	1
Moody et al.	142	0	0	10	14	0	12
Moody	28	0	0	0	0	0	0
Krogsty et al.	124	5	2	3	22	0	15
Krogsty & Jorgenson	80	5	2	4	11	0	6
Moody2	392	0	0	0	0	0	0
van Bommel et al. 3	66	1	1	2	7	1	4
Lindland et al.	35	0	0	1	2	0	2
Moody & Schanks	360	0	0	0	0	0	0

Stam & Proper	Quality of socio-cognitive interpretation	Quality of technical interpretation	Organizational quality	Knowledge quality	Social quality	Ethical quality	CSDP
Frederiks & van der Weide	0	0	0	0	0	0	0
Proper et al.	0	0	0	0	0	0	0
van Bommel et al. 1	0	0	0	0	0	0	5
van Bommel et al. 2	0	0	0	0	0	0	0
Hindriks et al.	0	0	0	0	0	0	0
Proper	0	0	0	0	0	0	0
Moody et al.	0	0	0	0	2	0	0
Moody	0	0	0	0	0	0	0
Krogsty et al.	0	0	0	0	0	0	0
Krogsty & Jorgenson	0	0	3	1	4	0	0
Moody2	0	0	3	0	5	0	0
van Bommel et al. 3	0	0	0	0	0	0	0
Lindland et al.	2	1	0	2	6	0	0
Moody & Schanks	0	0	0	0	0	0	0
	0	0	0	0	0	0	0

APPENDIX B

POPULATION FRAMEWORK

As the number of instances is too large to present in the framework presented in chapter 9, we present you the instances separately in this appendix.

Quality aspect (name) influences / is based on Decision point (name) [45]

Quality of technical interpretation	<-> Only elementary facts are used
Pragmatic quality	<-> Clarify entities
Quality of socio-cognitive interpretation	<-> Inverse predicates only on binary sentences
Social quality	<-> Inverse predicates only on binary sentences
Empirical quality	<-> Inverse predicates only on binary sentences
Semantic quality	<-> Don't use inverse predicates if they are the same
Semantic quality	<-> Don't split multiple facts with loss off information
Quality of socio-cognitive interpretation	<-> Feedback
Semantic quality	<-> Feedback
Social quality	<-> Use standard terms or synonym list
Domain quality	<-> First, build model of the current situation
Quality of socio-cognitive interpretation	<-> Avoid using vague predicates
Semantic quality	<-> Avoid using vague predicates
Social quality	<-> Avoid using vague predicates
Empirical quality	<-> Preference using parentheses
Quality of socio-cognitive interpretation	<-> Populate
Semantic quality	<-> Populate
Syntactic quality	<-> Populate
Social quality	<-> Populate
Pragmatic quality	<-> Formalize
Social quality	<-> Formalize
Empirical quality	<-> Using objectification or not
Empirical quality	<-> Implicit use value subtyping
Empirical quality	<-> Combine entities if instances are similar
Syntactic quality	<-> Combine entities if instances are similar
Quality of socio-cognitive interpretation	<-> Derivation rules are not drawn
Pragmatic quality	<-> Derivation rules are drawn in some special cases
Syntactic quality	<-> Derivation rules are drawn in some special cases
Quality of technical interpretation	<-> Eager evaluation: draw derived facts
Organizational quality	<-> Combining or not combining entities

45 Decisions about the textual language representation are not presented, because it depends on the language being used.

Decision point (name) influences / is based on Quality aspect (name)

Only elementary facts are used	<-> Syntactic quality
Clarify entities	<-> Syntactic quality
Inverse predicates only on binary sentences	<-> Syntactic quality
Don't use inverse predicates if they are the same	<-> Syntactic quality
Use a technical language representation	<-> Syntactic quality
Don't split multiple facts with loss off information	<-> Syntactic quality
Preference using parentheses	<-> Physical quality
Preference using parentheses	<-> Syntactic quality
Formalize	<-> Syntactic quality
Using objectification or not	<-> Physical quality
Using objectification or not	<-> Syntactic quality
Implicit use value subtyping	<-> Physical quality
Implicit use value subtyping	<-> Syntactic quality
Combine entities if instances are similar	<-> Syntactic quality
Derivation rules are not drawn	<-> Physical quality
Derivation rules are not drawn	<-> Syntactic quality
Derivation rules are drawn in some special cases	<-> Physical quality
Derivation rules are drawn in some special cases	<-> Syntactic quality
Eager evaluation: draw derived facts	<-> Physical quality
Combining or not combining entities	<-> Physical quality

Quality aspect (name) influences / is based on Quality aspect (name)

Semantic quality	<-> Quality of socio-cognitive interpretation
Quality of socio-cognitive interpretation	<-> Social quality