An Architectural Viewpoint for Conceptualization

Author: Angenita Heijmans
Date: August 06, 2002
Thesis number: 509

Supervisor: drs. Stijn Hoppenbrouwers
Second supervisor: prof. dr. Erik Proper

Summary

Organizations, in particular, ICT-oriented organizations, are regularly involved in 'conceptualization': explicit construction and or and / or description of concepts (for example terminological meaning and data structure). However organizations do not always acknowledge that they are performing such conceptualization processes. This is unfortunate, because especially in ICT-intensive organizations, conceptualization is an important activity that may have serious consequences when performed inadequately. Before organizations can be made more aware of conceptualization processes (assuming this is needed at all), such processes should first be understood; eventually, they might be evaluated and perhaps they could even be managed.

Being part of an IT hype, architecture is often seen as 'the solution to every problem'. Even though this is not always true, architecture is important in reducing complexity, something that is very useful in the complex IT environment architecture takes place in.
In this thesis, first architecture is explored as an approach to charting conceptualization processes and their environment, after which the IEEE 1471 Recommended Practice for Architectural Description of Software Intensive Systems is followed in order to construct an architectural viewpoint for conceptualization.

Foreword

To complete my study of computer science at the University of Nijmegen, I participated in the research about conceptualization processes in ICT supported organizations, performed by Stijn Hoppenbrouwers connected to the research group Information Retrieval and Information Systems.

My assignment was to use architecture to chart conceptualization processes and the environment they take place in. In this assignment Stijn Hoppenbrouwers not acted as my supervisor, but also played the role of domain expert and 'customer' at the same time and did all three with equal enthusiasm. Stijn Hoppenbrouwers also supervised the papers I wrote for Information Retrieval 1 and 2, which I finished before starting on my thesis. Therefore first of all I would like to thank Stijn Hoppenbrouwers for working with me for over one year. I experienced our cooperation as very pleasant. Especially because I was the first student who's thesis Stijn supervised, I'd like to say he did a great job. I'm sure it has not always been easy, because I had to write a thesis for computer science and one for my management study in quite a short time-span. I hope the results presented in this thesis are useful in his research.

Erik Proper was my second supervisor and also served as advisor (about architecture). I'd like to thank him for taking the time to act as second supervisor for both my theses.

I would like to thank my fellow students and the teachers of the department computer science, who made my stay at the University of Nijmegen both fun and interesting.

Next would like to thank my friends, for distracting me from my study when necessary. Special thanks go to Judith for coming up with the name of my fictional company: Eridani Insurance and to the other people who brainstormed with me on a name.

Finally the most important persons I would like to thank are my parents and my sister, who supported me during my entire study and the writing of my final thesis.

Angenita Heijmans
August 06, 2002

Contents

SUMMARY **3**

FOREWORD **5**

1 INTRODUCTION **9**

1.1 CONCEPTUALIZATION IN AN ICT CONTEXT 9
1.2 PROBLEM DEFINITION 10
1.3 ARCHITECTURE: REDUCTION OF COMPLEXITY 10
1.4 RESEARCH QUESTIONS 10
1.5 STRUCTURE OF THE THESIS 10

2 CONCEPTUALIZATION **11**

2.1 LANGUAGE AND MEANING 11
2.2 COMMUNICATION 12
2.3 CONCEPTUALIZATION AND CONCEPT MANAGEMENT 14

3 ARCHITECTURE **17**

3.1 INTERPRETATIONS OF ARCHITECTURE 17
3.2 CONTEXT OF ARCHITECTURE 17
3.3 ARCHITECTURE AND ALIGNMENT 17
3.4 WHAT IS ARCHITECTURE? 18
3.5 KINDS OF ARCHITECTURE 19
3.6 ARCHITECTURAL METHODS 19

4 METHODOLOGY AND APPROACH **23**

4.1 AN ARCHITECTURAL VIEWPOINT FOR CONCEPTUALIZATION 23
4.2 METHODOLOGICAL QUESTIONS 24
4.3 MODELS AND DESCRIPTIONS 25
4.4 MODELING METHODS 26
4.5 APPROACH: MODELING EXERCISE 27
4.6 CASE STUDY 27

5 ARCHITECTURAL VIEWPOINT **29**

5.1 STAKEHOLDERS 29
5.2 MODELS AND DESCRIPTIONS 33
5.3 CONCEPTUAL FRAMEWORK 34
5.4 FUNCTIONS 43
5.5 LANGUAGE USE TYPOLOGY 45
5.6 ARTIFACTS 53
5.7 I/O MODEL 55
5.8 CONCEPTUALIZATION PROCESS MODEL 57

6 CONCLUSION **67**

APPENDIXES **69**

A NIAM 69
B CASE: ERIDANI INSURANCE 74
C EXAMPLE CLAUSES 76
D QUESTIONS FOR UNIQUENESS CONSTRAINTS 79
E QUESTIONS FOR MANDATORY ROLE CONSTRAINTS 80
F QUESTIONS FOR EQUALITY, EXCLUSION, AND SUBSET CONSTRAINTS 81

BIBLIOGRAPHY **89**

1 Introduction

This chapter describes the foundation of the research that was done to obtain the degree Master of Science (doctorandus) in Computer Science. First the context of the problem that was researched is defined, the next paragraph explains why architecture was chosen as a solution to this problem. The third paragraph defines the goal of the research and the questions that need to be answered to reach that goal. Finally paragraph 1.4 gives the structure of this thesis.

1.1 Conceptualization in an ICT Context

In daily life, people hardly ever think about concepts, because communication is often about tangible things. Everyone knows what is meant with 'a bee'. But things get more difficult when one is asked to explain the meaning of a concept. In that case a simple word as 'bee' will not be that simple anymore, let alone a more difficult concept as 'virtue' (Plato, 380BC), because a virtue is not something that exists in the real world to which you can point and say: "That is a virtue."

Webster (2002) defines 'concept' as: "s*omething conceived in the mind" or "an abstract or generic idea generalized from particular instances"*. For now, it will be enough to note that a concept is an abstract notion, an image connected to something that only exists in people's mind, and can differ from person to person.

When a simple concept as 'bee' can give problems, companies might be expected to have more problems with concepts than they can cope with, because they use all kinds of difficult concepts. This is illustrated by the following example, by Miles Kingston (1992):

"Roger … has a speach problem. One which prevents him from speaking intelligently in the presence of his business peers. Here is something Roger said recently to a colleague: 'The client has an overly high expectation, market penetration wise.'

What does it mean? Nothing, I'm afraid. Roger is talking nonsense. But did his college turn round and say: 'What is that meant to mean, for heaven's sake?' No, I'm afraid he did not. What he said was: 'Well, when you're in a customer-focused situation, you need more product awareness than their sort of low-profile image is going to generate.'

Yes, when Roger talks nonsense, his colleague talks nonsense back to him."

Of course this citation overstates the problem, because when Roger and his colleague talk about 'market penetration' and 'product awareness' they will certainly have a notion what these concepts mean. The fact that they can talk about these concepts without asking, "what is that meant to mean, for heaven's sake?" illustrates that they at least have some common idea about the meaning of the concepts they use. But a situation in which Roger's colleague does not know what Roger means, is also thinkable. In that case his colleague can ask Roger the question "what do you mean?"

And this leads to a problem in ICT supported communities. When Roger is asked to fill in the 'degree of market penetration' in his database, there is no guarantee that Roger will be able to pick up the phone and call someone in his company who can inform him what 'market penetration' means. Perhaps the person who entered the question about market penetration into the database is on vacation, or even worse, has left the company a couple of years ago, without sharing the meaning of the concept 'market penetration' with anyone.

This is where conceptualization comes in. Conceptualization is the creation and adaptation of concepts. An explicit conceptualization process could be used to create a source where Roger and his colleagues can look up the meaning of 'market penetration'. But starting to think about conceptualization is a very important step too. When you look at organizations more closely, you will see that the environments in which conceptualization takes place are very complex, because they contain many different agents, communicating with each other through many different channels. Conceptualization processes may be hard to identify, because they are performed implicitly and often nobody realizes that what they are doing is conceptualization. Clearly this complex situation could use some clarification.

1.2 Problem Definition

The research described in this thesis contributes to a solution to the following problem:

Conceptualization processes take place in a complex environment and are often performed implicitly. Before conceptualization processes can be recognized and possibly even managed, all elements of conceptualization processes and the complex environment they take place in, need to be charted and that is exactly what the research described in this thesis aims to make possible.

1.3 Architecture: Reduction of Complexity

Architecture is one of the biggest hypes in ICT. There isn't a problem or it supposedly can be solved under architecture. But even though architecture is over-hyped, the notion behind architecture is a serious one. Or rather the problems that architecture is supposed to solve are serious ones. Even though everyone claims to either be an architect or to 'work under architecture', there is no agreement about what architecture is, and what it is supposed to do. Without going deeper into the meaning and purpose of architecture right now, we can already say that one of the goals of architecture is to reduce complexity. Many definitions of architecture agree on the fact that architectures describe parts of complex systems and their relations. These individual parts are less complex than the system as a whole. In this thesis we assume that conceptualization and the complex environment in which conceptualization takes place are parts of a system that can be usefully viewed from an architectural perspective.

1.4 Research Questions

Paragraph 1.1 made clear that conceptualization processes take place in a complex environment and pointed out that a reduction of this complexity was needed before conceptualization processes can be recognized or even managed. This is the main question underlying the larger research effort the research described in this thesis is part of (also see Hoppenbrouwers, forthcoming).

Q0: How can we understand / evaluate / manage conceptualization processes and their environment?

Paragraph 1.3 tentatively suggested that architecture could be used to reduce complexity of a system. Therefore, the main research question of this thesis is the following:

Q1: How can architecture be used to chart conceptualization processes and the complex environment they take place in?

In order to answer this question, we first need to explore the topic of conceptualization and the suggested direction in which we expect a solution to be found: architecture. We will return to the research questions and add more detailed questions in chapter four.

1.5 Structure of the Thesis

In this paragraph the structure of the remainder of the thesis is described. Before architecture can be used to chart conceptualization processes and their environment, it is needed to learn more about conceptualization and about architecture. Therefore chapter two will explore conceptualization and chapter three will explain what architecture is and how it could be useful in this research.
Chapter four will explain the methodology used in this research and will give the questions that need to be answered in chapter five. Chapter five will answer the questions raised in chapter four. And chapter six will analyze the results of chapter five to answer the research question and relate it to the research question of the bigger research effort.

2 Conceptualization

This chapter will explore the topic conceptualization. To explain what conceptualization is, and what it is used for, it is necessary to first discuss language and meaning. Next it is needed to give some information about communication. Finally this chapter introduces conceptualization and concept management.

2.1 Language and Meaning

According to Ferdinand de Saussure, one of the founders of linguistics, language involves two perspectives: the abstract language itself (*langue*) and what people do with it (*parôle*). Most linguists are only interested in *langue*, the system of rules and knowledge, shared by a group of people, or what we call language. To most linguists, language is an abstract system of rules and not what individual people produce and what people hear in conversations. But even though it is not the main focus of linguists, the study of language meaning (semantics) is important too.

When you look at a simple sentence like "Harry sees Jim", you can describe the structure of the language (syntax) as follows:

Harry sees Jim
noun verb noun

And you know that 'sees' is a verb with third person singular present tense (grammar / morphology). Of course this tells us something about the meaning of the sentence (grammatical meaning), but when we want to know the meaning of the word 'see' (the stem of sees), we have to go into lexical meaning. The online version of the Webster dictionary (2002) describes 'to see' as follows:

Main Entry: ***see***
Function: verb
1 a : to perceive by the eye b : to perceive or detect as if by sight
2 a : to have experience of : UNDERGO <see army service> b : to come to know : DISCOVER c : to be the setting or time of <the last fifty years have seen a sweeping revolution in science -- Barry Commoner>

This description continues for some more lines. It is interesting that when you look into the lexical meaning of a word, the description will be longer than the word itself. So with lexical meaning, it is not possible to break down a large chunk into smaller pieces and analyze them. Additionally, part of the meaning of a sentence depends on the situation or *context* of the sentence. This is illustrated by the following example by Harvey Sacks (in Redeker, 1999):

A: I have a 14 year old son,
B: That's fine.
A: and I also have a dog.
B: Oh, I'm sorry.

When you don't know the context of this conversation, it may not a make a lot of sense. But when you know that the conversation is between a tenant (A) and a landlord (B), you suddenly understand the meaning of the conversation, assuming you know the meaning of the words 'son', 'dog', etc. The kind of meaning that depends on contextual information is called pragmatic meaning.

Now we are talking about 'the meaning of words', the question can be asked whether words truly mean. Of course, meaning is not something that is contained in a word, meaning is formed in a person's head, and may differ from person to person. This special quality of meaning, is illustrated well by the following quotes of David Berlo (1960):

- Meanings are in people
- Communication does not consist of the transmission of meanings, but of the transmission of messages
- Meanings are not in the message; they are in the message-users
- Words do not mean at all; only people mean

- People can have similar meanings only to the extent that they have had, or can anticipate having, similar experiences
- Meanings are never fixed; as experience changes, so meanings change
- No two people can have exactly the same meaning for anything

These quotes about meaning will also be interesting later on, when we start talking about (problems with) concepts and conceptualization. For now it will be enough to say that what a word does when it means is the following: it triggers an association (concept) in the hearers mind. Given that the language of speaker and hearer are sufficiently the same, and also their knowledge of the context is sufficiently the same, then the concepts triggered in the mind of the speaker are sufficiently the same as the ones that the speaker means to trigger and communication succeeds (Hoppenbrouwers, 2001).

2.2 Communication

Not every way of communication involves language. Gestures, images and sounds are also considered communication. But the communication that is considered in this thesis is linguistic communication. In the rest of this thesis by 'communication' 'linguistic communication' is meant, unless stated differently. This paragraph will give a model of communication.

Communication theory has two important definitions of communication, a technical and a semantical one. The latter definition will be most useful in this thesis. The semantical definition of communication is: "A dynamic process of exchanging meaningful messages." Important in this definition are the words 'dynamic' and 'meaningful', although meaningful may not be chosen very well, the meaning of the words has to be interpreted. A model for communication that fits this definition well is shown in figure 2.2.1.

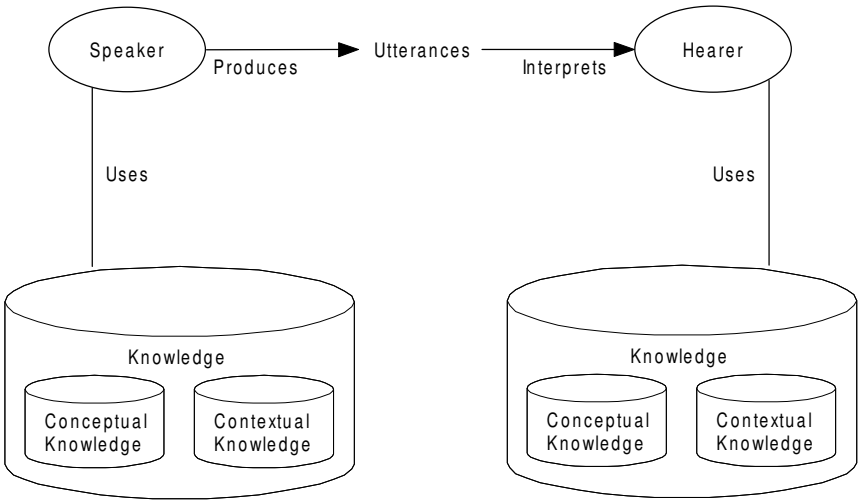


Figure 2.2.1. Model for communication

Hoppenbrouwers (2001) explains the model by the following statements:
People actively make sense of language by combining all knowledge they have (interpretation)
Langauge triggers certain limited associations; the rest is filled in (interpretation)
Picking the right "triggers" involves knowing both the context and the language (both production and interpretation)

The two forms of knowledge in the model are conceptual and contextual knowledge. *Conceptual knowledge* means all typically linguistic knowledge, word forms, word meanings, syntactic knowledge. In short everything people need to know to work with and understand language. *Contextual knowledge* is needed to interpret what is said in context. The speaker uses contextual knowledge to produce an utterance. Some standard things he needs to know are:

- When is the utterance going to be received?
- Where is the utterance going to be received?
- Who will be the hearer?
  - What does the hearer know?
  - What language does the hearer speak?

The hearer uses contextual knowledge to interpret an utterance. Some standard things the hearer needs to know are:

- When was the utterance sent?
- Where was the utterance sent?
- Who was the speaker?
  - What does the speaker know?
  - What language does the speaker speak?

Sometimes this knowledge is not sufficient to interpret a given utterance. After you tried to derive the meaning of an utterance from the knowledge already available, you can find out the meaning by talking about it:

1. You can ask what is meant ("Do you mean …?") or make clear you are not sure you understand the other person
2. You can ask questions about language ("What do you mean with …?")

2.2.1 Communication in an ICT Supported Community

What the model in figure 2.2.1 does not show, is that often communication is not performed directly. Often (especially in ICT supported communities) a medium is used. The use of an information system as a medium causes various abstractions in communication.

An utterance can after it is performed be transformed into a signal, for example in a phone call. This makes it possible for two communicating partners not to be at the same place (*abstraction of place*). This should not be a big problem, because the hearer exactly hears what the speaker has said, and can ask questions to derive the meaning of the utterance.

The second abstraction, *abstraction of time*, is more difficult, an utterance can be written down, or saved in a database. The speaker now not only has to choose a certain message, but also the format in which the utterance is saved. The hearer has to retrieve the original utterance from the database. In this case, the hearer can only use his knowledge to find out what the speaker meant, because he cannot ask questions about meaning.

The last problem is the fact that data in a database can be manipulated, for example by transferring the contents of a database to a new database. The utterances stored in the original database may change form. This is what is known as *abstraction of form*. The new database may restrict the possible words used in the utterance, so the person who is transferring the database has to choose different terms.

The section about abstraction of form already hinted at the possibility that a database would not accept just every utterance. Often databases have certain fields, so you can not fill in just anything. This is what makes a database a medium only supporting *closed language*. In an ICT context an example of *open language* use would be an e-mail, because you can write anything you like in the message-field. The type of language use that is the most interesting in this thesis is closed language use, because when creating a closed language systems (a process called 'freezing language' by Hoppenbrouwers (forthcoming)), (explicit) conceptualization processes are very important.

2.2.2 Discourse Environments

Communication always takes place in an environment or domain. Domains are restricted areas of focus. A way of looking at communication is looking at terminological domains. In a terminological domain the focus lies on sets of terms that have been clearly defined, called terminologies. But the way of looking to environments in this thesis is based on Discourse Environments. A Discourse Environment is a communicative environment in which sets of people enter into organized communicational activity (Hoppenbrouwers, 2001). An organization may contain a number of Discourse Environments. A Discourse Environment may have a specific terminology, but often contains multiple terminologies from which sets of terms are used alongside each other.

According to Hoppebrouwers (forthcoming) Discourse Environments contain terminologies, from which certain concepts may be described by artifacts. Discourse Environments also contain a set of functions that are performed by certain agents, who all have a vocabulary of their own. An agent may be part of multiple Discourse Environments.

2.3 Conceptualization and Concept Management

This paragraph first describes meta-communication, followed by the need for conceptualization. Next it is explained what a concept is. The next paragraph describes conceptualization. Finally something will be said about the managing of the entire process: concept management.

2.3.1 Meta-communication

In paragraph 2.2. we already discussed a couple of ways to find the meaning of an utterance. One of these ways was discussing language with your communication partner. This is also the subject of this paragraph: communication about communication. The form of meta-communication in paragraph 2.2 ("Do you mean …?" and "What do you mean with …?") is one of two forms of meta-communication.
It is used to fill in a gap in message understanding. The second form, and most important form in this paragraph is *linguistic meta-communication*. Linguistic meta-communication is about the language that is used to communicate with. In an information systems context, the meta-communication process will concern a closed set of concepts that are allowed in the medium system. This set of concepts is selected from the language that is used in the domain. These meta-communication processes are called 'conceptualization processes'.

2.3.2 The Need for Conceptualization

Often conceptual problems do not lead to big issues. People accept problems with concepts, because they are part of working life, because they can cope with these problems easily and often the people involved in ICT do not have the expertise to deal with conceptual problems. Then why conceptualize? Hoppenbrouwers (2001) gives some important reasons. The first reason why conceptualization would be useful is that people like to reduce the things that do not belong to their most important activities to a minimum. And communication and language use are part of these supporting activities. In short people try to be effective. The second reason is that people want certainty. Communication can help create operational certainty (for example by sharing information) and meta-communication in its turn creates communicational certainty.

2.3.3 Concepts

The definition of a concept used in this thesis is based on a theory of the Swiss linguist Ferdinand de Saussure. De Saussure distinguished between '*signe*' (sign), '*signifiant*' (signifier) and '*signifé*' (signified). In short a *signe* is a combination of 'something thing in reality' (signifé) and something that refers to it (*signifiant*) (Underwood, 2001).

But this explanation of *signe* leads to a problem, because not every word refers to something in reality. When I say 'horse' it is easy to understand it refers to something in reality (or at least some kind of mental image), but there may be many different ways in which 'horse' can be interpreted, involving different aspects (or views on) reality. And it gets even more difficult when I use a word like 'concept management', because in that case there is no such thing as a relation to reality.

This problem is solved by De Saussure's *signifé*. According to De Saussure a *signe* consists of a combination of *signifé* (best translated to 'mental image') and a *signifiant*. De Saussure only considered an '*image acoustique*' as a *signifiant*, because language is mainly based on sound. Because ICT also concerns written expressions, this thesis will consider written expressions as *signifiants* too, which makes it possible for a *signe* to be a combination of a mental image and a written language expression. Figure 2.3.3.1 visualizes De Saussure's signe.

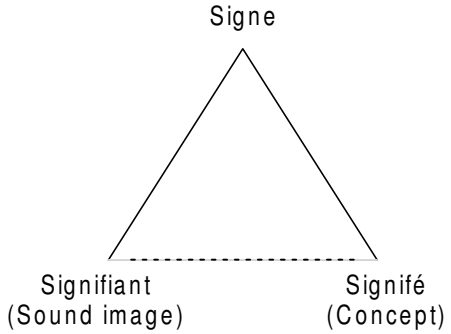


Figure 2.3.3.1. De Saussure's preferred terminology

This thesis will not use the word *signe*, because it contains more than just language expressions, but it will use the word 'concept' instead. From this point we will assume 'concept' to refer to meaning as understood through its form. This means we will use 'concept' rather than 'signe', assuming that the intricacies of the signe – concept relation are clear.

2.3.4 Conceptualization

As you may have already guessed, conceptualization is simply the process of creating concepts. This paragraph will provide some more details about conceptualization processes.

It is important to make a distinction between explicit conceptualization and implicit conceptualization. Often organizations do think about conceptualization, but are not aware they are doing so, or do not put their results to paper. This is what is known as an implicit conceptualization process. Explicit conceptualization is performed knowingly. It is also important to note that even though no actual conceptualizations are created, an organization may still have performed a conceptualization process. The organization may have concluded there was no need to make certain concepts explicit.

Paragraph 5.8 will explain the steps in an explicit conceptualization process and will give an example how these steps are performed in a sample organization.

2.3.5 Concept Management

The next step would be to think about 'managing concepts', for example to improve the concepts that are part of information systems. Logically, when you want to create better concepts, you need to get your conceptualization processes right. And this is where management is needed. So if you manage your conceptualization processes, you are performing concept management. This is also where the problem comes in. Because organizations often do not realize that what they are doing is conceptualization. Concept management cannot be found on the agendas of organizations. The architectural viewpoint for conceptualization that will be presented in chapter 5 should be a good approach to start thinking about conceptualization. Additionally, based on how the viewpoint is filled in, concept management strategies can be created.

3 Architecture

As chapter one already explained, one of the goals of architecture is reducing complexity. Therefore architecture is very useful in creating a descriptive framework for conceptualization. This chapter will take a better look at architecture. First a couple of ways of looking at architecture are discussed. Next the goals of architecture the context of architecture and the relation to business – IT alignment will be discussed. Later architecture is defined and a method for architectural description is given.

3.1 Interpretations of Architecture

Because there is no agreement on what architecture is, the concept of architecture is open to many different interpretations. This section discusses a couple of those interpretations.

Architecture as vision

Architectures may express a certain vision, as the definition of de Bruijn (2000) suggests: *"Architectures describe parts of complex things and their relations and at the same time express a vision"*

Architectuur as a collection of guidelines

Other definitions like the definition by Dietz (1999) view architectures as a set of guidelines for a development process: "G*iving context and focus to the development process*".

Architectuur as description

Many definitions, of which the definition by IEEE (2000) is the most important, view architecture as the description of a system: "*The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.*"

3.1.1 Goals of Architecture

Van der Poel (2000) gives a couple of reactions to definitions of architecture. From this reactions the following two goals for architecture can be derived:

- Offer additional value to the system to be designed
- Architecture can serve as means of communication

A third goal of architecture is to reduce complexity (van Vliet, 2000).

Additional value

Examples of additional value architecture can offer are the reduction of development costs and increased usability of the system.

Means of communication

Architecture can be used as a means of communication. Architecture offers a common language, and can serve as a framework to refer to for different groups. (Kruit, 2000)

Complexity

One of the oldest uses of architecture is reducing complexity. Architectures describe parts of complex systems and their relations. Every part by itself is less complex than the system as a whole.

3.2 Context of Architecture

Many of the reactions in the article by van der Poel (2000) refer to the context of architecture. The system that is described by an architecture is an open system, and therefore this system has relations to its environment. This is why architecture cannot be viewed apart from its environment.

3.3 Architecture and Alignment

Alignment is a hot item in business and computer science nowadays. Because ICT more than ever fulfills the role of enabling technology, a good fit between the organization and the ICT it uses is essential. This concept of fit is known as alignment, or business – IT alignment and is shown in figure 3.3.1.

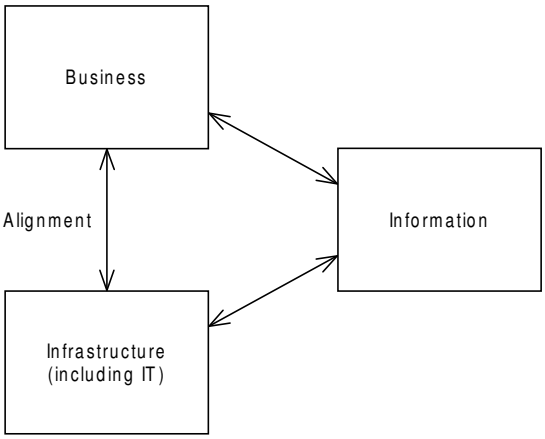


Figure 3.3.1. Alignment

But how do architectures fit in this picture? To every area in the model belongs an architecture. This is shown in figure 3.3.2.

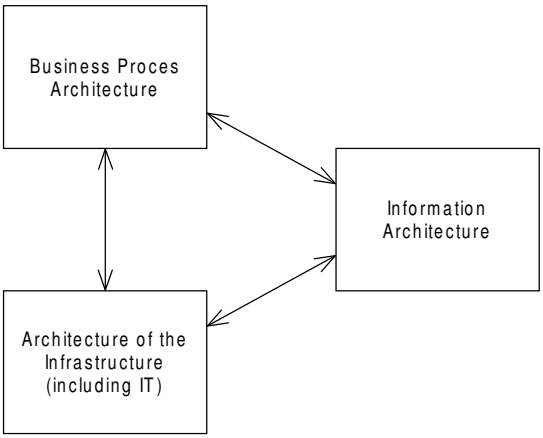


Figure 3.3.2. Architecture

Of these three architectures, the information architecture is the most important. This architecture relates the business process architecture with the infrastructure (it translates the goals of the organization into a concrete (IT) infrastructure) by models and guidelines.

3.4 What is Architecture?

Now we know how you can see an architecture, the goals of architecture, and how architecture stands in relation to business and ICT, it is time to give a definition for architecture. In the last couple of years many definitions of architecture have been given (Vermeulen, 2000). Many of them were very vague. But most of the definitions agreed on a few important points:

- An architecture describes a system
- An architecture describes all aspects of a system (holistic)
- An architecture connects the system with its environment (context)
- An architecture shows the relations inside a system (structure)

Because the IEEE (2000) definition of architecture contains all of these elements, we will use this definition as a definition for architecture in this thesis:

"The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution."

To this definition we will add the note that the IEEE definition is very software-oriented. In this thesis the system in question can vary from a whole organization to a software system. Therefore a system should not solely be seen as a software system that can be captured in models only.

3.5 Kinds of Architecture

As figure 3.3.2 showed, there are various kinds of architecture, which describe all subsystems of the all-encompassing architecture, the enterprise architecture (a way of describing an organization, also known as city planning):

- The *business process* architecture describes the primary business process of an organization. These business processes are organized in one or more supply chains. The environment of the organization is seen as a network, connecting the organization with customers, suppliers and other parties.
- An *information-architecture* is a coherent vision of an organization[1] on their existing and future information household. An information architecture is formed by a collective process of negotiation between people involved.
  In an information architecture elements of the information household in their context are expressed, together with the way they correspond to the enterprise architecture[2] and the IT-architecture, as well as to the why.
  Inherent to an information-architecture are choices in the area of information- functionality and -structure. These choices are recorded in the form of principles, standards and models.
  With that an information architecture is the development plan for the renewal of the information household of an organization. (GIA, 1998)
- The *infrastructure architecture* is the architecture of software, data storage systems, the hardware etc.

Next to these main types of architecture, there are some special viewpoints that should be regarded from a holistic perspective (Goedvolk, 1999). A good example of such a viewpoint is security. Security is important the entire enterprise architecture, so security should be taken into account in every architecture. Another special area where business and ICT meet is conceptualization. Therefore conceptualization should be seen from a holistic perspective too. Chapter five will create such a viewpoint for conceptualization.

The main goal of this viewpoint is to provide a descriptive framework for conceptualization. Other architectural approaches to the topic could have involved providing methods (i.e. protocols) to describe conceptualization (though some parts of this thesis do serve this purpose up to a point) approaches to the evaluation and managing of conceptualization. These two last activities are very useful but could not be covered in this thesis; we restrict our focus to providing a descriptive conceptual framework and some additional descriptive instruments.

3.6 Architectural Methods

Now we know what an architecture is, we would like to know how to describe one. For software architecture, quite a lot of methods have been created. Examples are the "4+1" View Model of Software Architecture (Kruchten, 1995) that was created by Rational and the Open Group Architectural Framework (TOGAF) (The Open Group, 2001). A recent and widely accepted method for describing architectures is IEEE standard 1471: *Recommended Practice for Architectural Description of Software Intensive Systems.*

[1] *In this context an "organisation" is a enterprise / institution / agency, or part of such, or a collaboration between more enterprises / institutions / agencies. The formal term for this notion is Universe of Discourse (UoD).*

[2] *One may consider talking about "enterprise architectures" (plural). Many aspects may be modelled in or of an organisation. All of them may be called architecture.*

3.6.1 IEEE 1471

IEEE standard 1471 offers a conceptual framework for architecture. Maier (2001) describes IEEE 1471's conceptual framework for architecture as follows:

- A system has 1 architecture
- An architecture is described by 1 or more architecture descriptions
- An architecture description is composed of 1 or more of stakeholders, concerns, viewpoints, views, and models
- A stakeholder has 1 or more concerns
- A concern has 1 or more stakeholders
- A viewpoint covers 1 or more concerns and stakeholders
- A view conforms to 1 viewpoint
- A viewpoint defines the method of a model
- A view has 1 or more models and a model is part of 1 or more views
- A viewpoint library is composed of viewpoints

This framework is also displayed in figure 3.6.1.1.

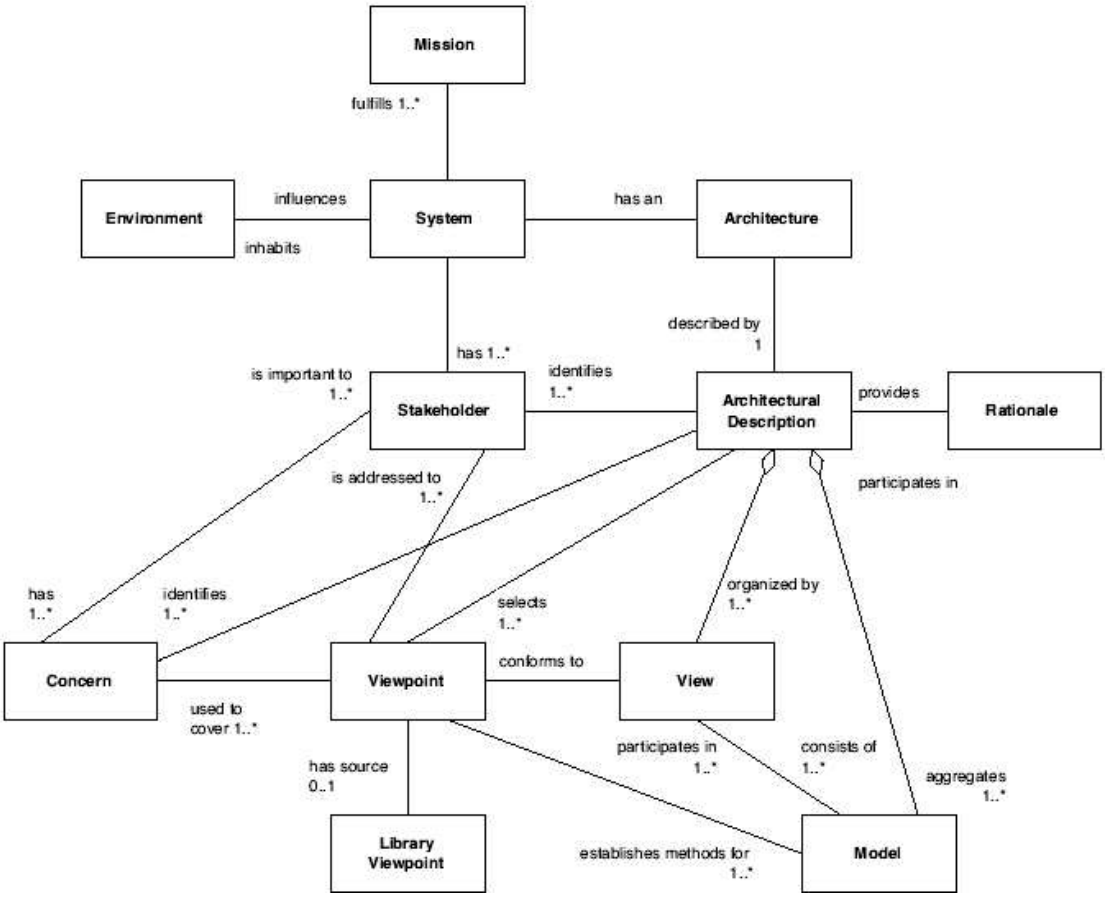


Figure 3.6.1.1. IEEE 1471's conceptual framework for architectural description (IEEE, 2000)

3.6.1.1 Elements of the Framework

This paragraph gives a short introduction to IEEE 1471's conceptual framework for architectural description (IEEE, 2000). It explains the most important terms regarding the content and use of architectural descriptions.

System
A system is a collection of components organized to accomplish a specific function or set of functions. (IEEE, 1990) A system does not have to be a computer system, but can vary from an individual application, to it is subsystems or an entire organization.

Mission
A system fulfils one or more missions. A mission is a general expression of the overall purpose of a system. It is the use or operation for which a system is intended to meet some objectives.

Environment
A system has an environment, or context, in which the system is developed, operated and used. Architecture should be sensitive to that environment and to the influences of that environment upon the system. The environment determines the scope of a system and how the system interacts with other systems.

Stakeholder
A system has one or more stakeholders. A stakeholder is a person, group, or organization that has interests in or concerns relative to a system. Stakeholders should include users of the system, acquirers of the system, developers of the system and maintainers of the system (IEEE, 2000).

Concerns
A stakeholder has one or more concerns which are important to that stakeholder. One or more stakeholders can share a concern. Concerns are those interests that refer to the system's development, operation or any other aspects that are critical or otherwise important to one or more stakeholders. Concerns can refer to performance, reliability, security, etc.

Concerns should include the following (IEEE, 2000):
- The purpose or missions of the system
- The appropriateness of the system for use in fulfilling its missions
- The feasibility of constructing the system
- The risks of system development and operation to users, acquirers, and developers of the system
- Maintainability, deployability, and evolvability of the system

Architectural Description
An architecture is described by an architectural description consists of a collection of products to document an architecture.

Rationale
An architectural description should provide rationale. With rationale is meant that an architectural description should explain the choices that were made in the architectural description and should include evidence of the consideration of alternative architectural concepts.

Viewpoint
A view conforms to a viewpoint. A viewpoint is a specification of the conventions for constructing and using a view. It is a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.

According to (IEEE, 2000) each viewpoint shall be specified by:
1. A viewpoint name,
2. The stakeholders to be addressed by the viewpoint,
3. The concerns to be addressed by the viewpoint,
4. The language, modeling techniques, or analytical methods to be used in constructing a view based upon the viewpoint,
5. The source, for a library viewpoint (the source could include author, date, or reference to other documents, as determined by the using organization).

A viewpoint specification may include additional information on architectural practices associated with using the viewpoint, as follows:

- Formal or informal consistency and completeness tests to be applied to the models making up an associated view
- Evaluation or analysis techniques to be applied to the models
- Heuristics, patterns, or other guidelines to assist in synthesis of an associated view

View
An architectural description is organized in one or more views. A view is a description of the entire system from the perspective of a set of related concerns. A view is composed of one or more models.

Viewpoint vs. View
The viewpoint is where you look from, the view is what you see. For example: A chair and a table have different front views, but the concept of a front view is general. (Maier, 2001))

Model
A view may consist of one or more architectural models. Each such architectural model is developed using the methods established by its associated architectural viewpoint. An architectural model may participate in more than one view.

Library Viewpoint
A viewpoint may have a library viewpoint as its source. A library viewpoint is a previously defined viewpoint.

According to Maier (2001) an IEEE 1471 compliant architectural description must:
- Define the stakeholders for the system (whose architecture is being described)
  - A minimal set should be considered
- Define those stakeholders concerns
- Select a set of viewpoints for the description
- Describe the system in a series of models
  - The models are organized into views
  - Each view much be conformant to exactly one selected viewpoint

As stated above, an architectural description contains many viewpoints of a system, for example security, maintainability, costs, but also conceptualization. Therefore a conceptualization viewpoint is only a part of the architecture of a system (in this case an organization). But an architectural viewpoint should follow the ‘rules’ of the architectural description it is part of, therefore the above points will also be taken into account when creating an architectural viewpoint for conceptualization.

4 Methodology and Approach

This chapter describes the methodology and approach that was used in this research. First it is explained why we have decided to create an architectural viewpoint for conceptualization and the meta-concepts that should be included in that viewpoint are discussed. The next paragraph describes the questions that need to be answered to create an architectural viewpoint. Paragraph 4.3 describes the models the viewpoint should include and the relations between those models. It also explains why certain models were not created. Paragraph 4.4 gives a short introduction to the modeling methods and languages that were used to create the viewpoint that is described in chapter five. In paragraph 4.5 the approach of the assignment is described and finally the last paragraph explains why a case was used and how this case was selected.

4.1 An Architectural Viewpoint for Conceptualization

After giving a short introduction to architecture and its uses, chapter three described the IEEE Recommended Practice for Architectural Description of Software Intensive Systems. This section will examine whether IEEE 1471 can be used in this research as a guidline to help chart conceptualization processes and their environment.

When we take a closer look at the title of IEEE 1471, we notice some important things about the method. First IEEE 1471 is a *recommended practice* and not a standard. This means IEEE 1471 is to serve as a guideline in creating an architectural description and should not be adhered to the letter. The second thing we notice the goal of IEEE is to create an *architectural description* of a *software intensive system*. A *software intensive system* is a system in which software plays an important part. This is also the case in our ICT supported communities. Therefore we can conclude IEEE 1471 should be useful as a guide for this research. As we learned in paragraph 3.6.1.1, an *architectural description* is used to describe an architecture and is composed of one or more of stakeholders, concerns, viewpoints, views, and models. As paragraph 3.5 explained, viewpoints are (special) areas of a system that should be regarded from a holistic perspective. It was already made clear that conceptualization is such a viewpoint.

Since a *complete* architectural description would include multiple viewpoints for different aspects of the system, it is decided here to create a *specific* architectural viewpoint for conceptualization. Because IEEE 1471 is a recommended practice, we can use this method to create an IEEE 1471 compliant architectural viewpoint, as longs as we clearly state which parts of IEEE 1471 we are going to use and why we are using them.

System

An architectural description describes the architecture of a system. Therefore the viewpoints that are part of the architectural description should also be based on that system. That is why the system should be described. Because the viewpoint we are going to create is not supposed to be used to create a view for one special system, this already hints the viewpoint we are going to create is a library viewpoint.

Mission

A system fulfils one or more missions. Therefore the mission(s) of a system should be described as well.

Environment

Every system has an environment. Therefore the environment of the system too should be described. This environment should not be confused with the Discourse Environment.

Concerns

A stakeholder has one or more concerns which are important to that stakeholder. The viewpoint we are going to create covers one or more of those concerns regarding conceptualization. Therefore only those concerns should be taken into account.

Stakeholders

A system has one or more stakeholders. The stakeholders who share the concerns regarding conceptualization should be described.

Architectural Description
A viewpoint is part of an architectural description. Because conceptualization is just one viewpoint, the entire architectural description will not be given.

Rationale
An architectural description should provide rationale. The same goes for an architectural viewpoint.

Viewpoint
A view conforms to a viewpoint. A viewpoint is a specification of the conventions for constructing and using a view. It is a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis. This is exactly what we are going to create for conceptualization.

According to (IEEE, 2000) each viewpoint shall be specified by:
1. A viewpoint name,
2. The stakeholders to be addressed by the viewpoint,
3. The concerns to be addressed by the viewpoint,
4. The language, modeling techniques, or analytical methods to be used in constructing a view based upon the viewpoint,
5. The source, for a library viewpoint (the source could include author, date, or reference to other documents, as determined by the using organization).

These things will also be taken into account when creating a viewpoint for conceptualization.

View
A complete view will not be created in this research. Chapter five will contain some examples that are part of a view, to illustrate the working of the viewpoint.

Model
The viewpoint mainly consists of some 'template-models', which can be filled in to be part of an architectural view.

Library Viewpoint
The viewpoint we are going to create is not supposed to serve in creating a view for just one system or situation. Therefore the viewpoint that will be created in chapter five will in principle serve as a library viewpoint.

4.2 Methodological Questions
To create an architectural viewpoint for conceptualization, a lot of questions need to be answered. The main questions are:

1. What can be concluded about the elements of an architectural viewpoint, based upon theory about conceptualization and architecture?
2. How can these findings be translated into an architectural viewpoint for conceptualization?
3. What are the models of which this architectural viewpoint consists?

To answer these questions, the following supporting questions should be asked:

1.1.	What is the environment?
1.2.	Who are the stakeholders?
1.3.	What are stakeholders' concerns?

2.1.	How do stakeholders, concerns and the environment fit together?

3.1.	How is the Discourse Environment modeled? - How are the most important elements and relations of a Discourse Environment modeled? - How can agent functions be described? - How can different language use be categorized and described? - How can artifacts and documentation be classified and described? - How are conceptualization processes modeled? - How can the input and output for conceptualization processes be modeled? - How can the conceptualization processes themselves be modeled?
3.3.	Which modeling language(s) should be used?

4.3 Models and Descriptions

The questions 3.1 and 3.2 in paragraph 4.2 need some explanation. First should be explained why exactly the items mentioned in those questions should be modeled / described. Next should be explained how those models are related. Figure 4.3.1 shows the relationships between the models.

Conceptual Model

The conceptual model serves as a conceptual framework in which the central elements and relations that have to do with conceptualization processes and their environment are modeled. This is the most important model in the viewpoint, because it contains the basic elements that are used in the other models.

Functions / Stakeholders

Discourse Environments contain agents who perform certain functions. It is important to know who fulfills certain conceptualization functions, because agents with certain functions take part in conceptualization processes. Also agents with certain functions have the power to authorize other functions and the use of artifacts and even may be responsible for rejecting the process of creating an architectural view.

Language Use Model

It is important to know what types of language are used in a Discourse Environment, because the type of language use influences the type of conceptualization required in meta-communication. For example it is likely that when the language use is very official, the concepts will be made explicit, will be widely distributed and documented, and some authority is closely involved.

Artifacts

Artifacts play a vital role in the conceptual model as it is described in chapter five (meta-communicational documentation, general documentation, medium system specification, etc.). Therefore it should be known what kind of artifacts exist. Artifacts are also related to the language use model. When we look at the example of official language use again, the artifacts will also be more official and will not be handwritten notes.

Input / Output Model

Artifacts serve as the input and output of an explicit conceptualization process. It is possible that that certain aspects of artifacts (like the officialness) change during a conceptualization process.

Conceptualization Processes

Of course conceptualization processes themselves should be modeled too. Most elements that are defined in the main model play a role in conceptualization processes.

Models that were not created

Other models that could have been created were models for the capacity and performance of, and resources used in conceptualization. This model was not created because the time planned for the assignment did not allow for it. A model for methods, tools and techniques used in conceptualization was not included because the larger research effort that served as input for this research had not yet advanced to this stage. The same goes for heuristics for concept management.

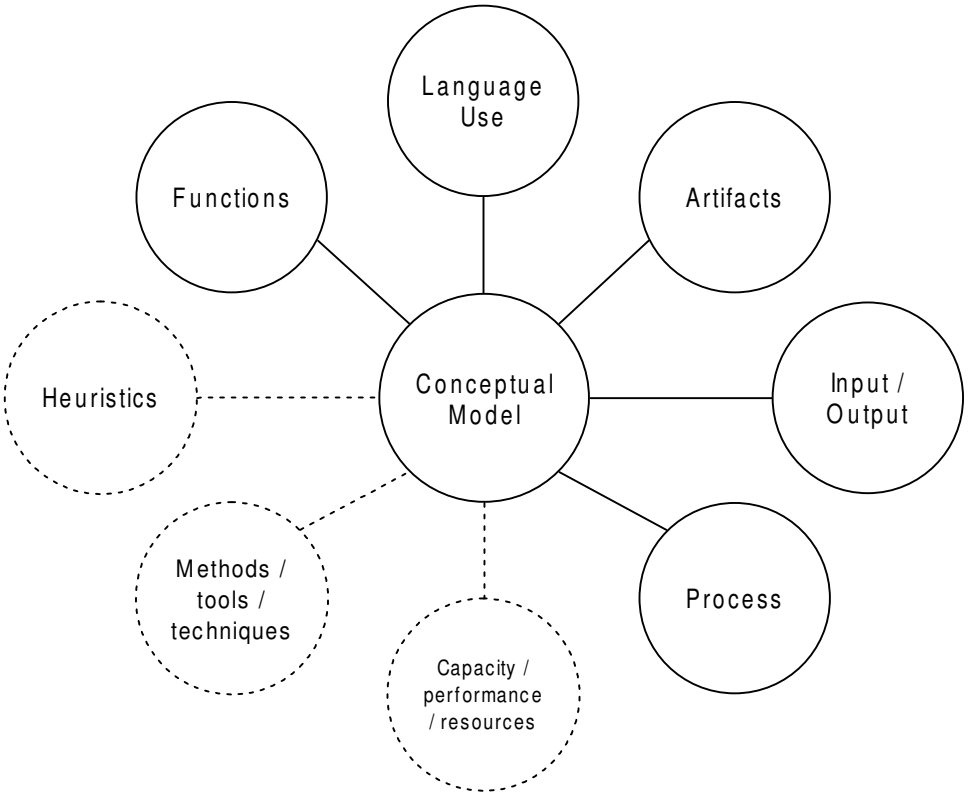


Figure 4.3.1. Relations between the models

4.4 Modeling Methods

This section will give a short overview of a couple of modeling methods and languages that were used in this research. Chapter five will explain why a certain language or method is chosen for each model. No exhaustive study of modeling languages has been done, because of the relatively short time span in which the research took place.

4.4.1 NIAM

NIAM (also known as Object Role Modeling or ORM) stands for Nijssen' s Information Analysis Method and was originally developed by G.M. Nijssen in 1974 (Nijssen, 1993). NIAM is a method that is very useful for conceptual data modeling. NIAM offers a structured method to create information systems through a conceptual structure abstracted from the real-world system. Appendix A contains a description of NIAM. Paragraph 5.3 will explain why NIAM was used to create the conceptual model.

4.4.2 Flowcharts

A flowchart is a graphic illustration of the steps involved in a process. A flowchart consists of a set of various boxes of a standard shape that are interconnected by lines. These lines have arrows to indicate the direction of the flow between the boxes. The description of the activity within the boxes is written in plain text. Paragraph 5.8 will explain why flowcharts were used to create the conceptualization process model.

4.4.3 Plain Text

The field of computer science has the habit to try to capture every aspect of a system in models. This will in many cases lead to problems, because there are certain aspects of systems (especially those aspects of systems that exist in the area where business and ICT meet) that just cannot be captured in models. Therefore in chapter five, structured plain text (for example, template forms) will be used to model those parts of the system that cannot be usefully modeled otherwise given the scope of this thesis. Sometimes plain text will be illustrated by images.

4.5 Approach: Modeling Exercise

There are many ways by which the assignment of creating an architectural viewpoint for conceptualization could take shape. For this thesis the approach of a modeling assignment was chosen. In that approach, Stijn Hoppenbrouwers played the role of the problem owner and domain expert, Erik Proper the role of advisor and Angenita Heijmans played the role of performer of the modeling assignment.

4.6 Case Study

After it was decided a case would be used, the first question that came to mind was whether to use a real case (for example the GAK example) or a fictional one. The GAK example that was used in the larger research effort (Hoppenbrouwers, forthcoming), on which the conceptual framework in paragraph 5.3 is based, is very complex. Also the GAK does not acknowledge conceptualization as a processes, which also makes the GAK case less suitable. The latter problem is the most important problem with nearly every existing case. That is why in some cases, a fictional case is used instead of the GAK case. The fictional case features an insurance company (Eridani Insurance) that is about to start developing a new system for processing declarations made by clients, and starts a conceptualization process about the meaning of errors in the system. This case is described in appendix B.

5 Architectural Viewpoint

This chapter describes the result of the research for this thesis: an architectural viewpoint for conceptualization.

In chapter four the decision was made to describe the viewpoint according to (IEEE, 2000) by:

1. A viewpoint name,
2. The stakeholders to be addressed by the viewpoint,
3. The concerns to be addressed by the viewpoint,
4. The language, modeling techniques, or analytical methods to be used in constructing a view based upon the viewpoint,
5. The source, for a library viewpoint (the source could include author, date, or reference to other documents, as determined by the using organization).

Chapter four also suggested that the system, its mission and the environment were important, because a viewpoint is part of an architectural description. Therefore these should be taken into account too. We will discuss each of these elements shortly.

Viewpoint name

The viewpoint will simply be named 'Viewpoint for Conceptualization'. Since the viewpoint may function as library viewpoint in the future, the viewpoint name may be changed depending on the system the viewpoint is used for.

Source

Our viewpoint is not derived from a library viewpoint, therefore no source has to be given.

Concerns

The viewpoint for conceptualization addresses all concerns of stakeholders regarding conceptualization. The exact concerns may be different in different organizations are dependent of the organization's view on conceptualization. Therefore no exact method to identify concerns is given.

Stakeholders

In this viewpoints all stakeholders of the system that have concerns regarding conceptualization are described. These stakeholders can differ from organization to organization. Therefore paragraph 5.1 also gives a method of identifying stakeholders.

System, mission and environment

The system in question may be any company for which conceptualization processes are researched. The mission of the system is the mission of the company. The environment of the system consists of the environment of the organization and – at a lower level – the Discourse Environments in which the conceptualization processes take place.

Language, modeling techniques and methods

Paragraphs 5.3 through 5.8 describe methods and techniques that can be used when creating a view. Here two of the uses of architecture that were described in paragraph 3.5 can be recognized. The architectural viewpoint both serves as a descriptive model for conceptualization and offers methods by which conceptualization can be described (in a view).

5.1 Stakeholders

Stakeholders are persons, groups, or organizations that have interests in or concerns relative to a system. In this thesis the system is the conceptualization process. In the next paragraphs some examples of conceptualization functions will be given and two methods to determine the most important stakeholders will be explained. These methods can be used when creating a conceptualization viewpoint based on this conceptualization viewpoint as a library viewpoint. This section will be concluded with an example stakeholder analysis for Eridani Insurance, which uses both steps.

5.1.1 Conceptualization Functions

This paragraph offers some questions that help discover conceptualization functions and thus identify possible stakeholders. Of course there can be more conceptualization functions than can be found using these questions and in some situations not every question will result in finding conceptualization functions.

Some possible conceptualization functions and the questions to identify them are:

- Language informants: Who take part in discussions about language/terminology?
- Language authors: Who are authors of certain language specifying texts?
- Language authority: Who decide upon "official" terminology?
- Concept manager: Who are responsible for conceptualization management?
- Conceptualization facilitator: Who is a facilitator for the conceptualization process?

5.1.2 Determining the Most Important Stakeholders

Stakeholders and their concerns are a key element of an Architectural Description that is compliant with IEEE 1471. Since the Architectural Viewpoint in question is conceptualization, all agents with concerns relating to conceptualization are stakeholders in this viewpoint. IEEE 1471 does not provide an exact method for stakeholder analysis, but such a method would be very useful.

A method for stakeholder analysis can be useful for various purposes. Many different stakeholders can be identified. In this case stakeholder analysis could be used to identify the most important stakeholders. Additionally stakeholder analysis can be used to contribute to better conceptualization, by comparing the current and desired (future) situations. For example in the existing situation problems with concepts are caused by the lack of a facilitator for conceptualization processes. A desired future situation could then be designed in which this function does exist.

5.1.2.1 Internal Versus External Stakeholders

Internal stakeholders are stakeholders that are located inside the organization, external stakeholders are located outside the organization. In a stakeholder analysis, power of a stakeholder is defined as the extent to which they are able to persuade, induce, or coerce others into following certain courses of action. The power of stakeholders can base on various sources (Recklies, 2001):

Internal stakeholders	External stakeholders
• Hierarchy (formal power) *e.g. authority, senior position* • Influence (informal power) *e.g. leadership style* • Control of strategic resources *e.g. responsibility for strategic products* • Possession of knowledge and skills *e.g. expert knowledge that forms the organizations core competence* • Control of the environment *e.g. network of relationships to external stakeholders* • Involvement in strategy implementation *e.g. as a change agent or responsibility for strategic projects*	• Control of strategic resources *e.g. monopolistic supplier* • Involvement in strategy implementation *e.g. strategic partners in distribution channels* • Possession of knowledge and skills *e.g. cooperation partners, subcontractors* • Through internal links *e.g. networking*

5.1.2.2 Power / Interest Matrix

A way of stakeholder mapping is the power / interest matrix. The power / interest matrix in this paper was adapted by Johnson and Scholes (1999) from an idea by Mendelow (1981). This method identifies stakeholder expectations and power and helps establishing priorities.

It consists of two dimensions:

1. How interested is each stakeholder (group) to impress its expectations on the organization's choice of strategies.
2. Whether they have the means to do so. This is concerned with the power of stakeholder groups.

These two dimensions are combined to form a matrix:

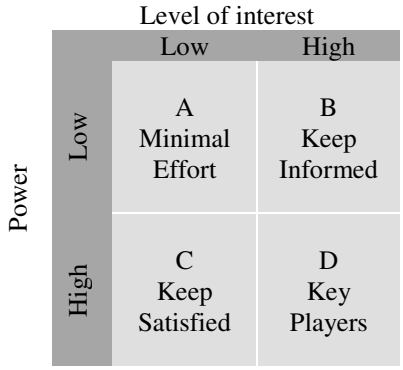


Figure 5.1.2.2.1. Power / interest matrix

Stakeholders in **A** neither have a high interest in the organization's strategy nor do they have the power to have much impact on the strategy. Organizations should not invest too much effort into them.

Stakeholders in **B** have a high interest in the organization's strategy, but they have limited means to influence things. This type of stakeholders could be a valuable ally in important decisions. Therefore, they should be kept informed about the objects they are concerned about.

Examples of stakeholders of type **C** are investors and legislators. Most of the time they behave passively and do not show a lot of interest in the organization's strategy. But they can exert a lot of power, for example when it comes to investments. Therefore reactions of this group should be taken into account at all times and they should be kept satisfied.

But the most important type of stakeholders can be found in sector **D**. They have both high interest and high power. That is why they have to be involved in all relevant developments.

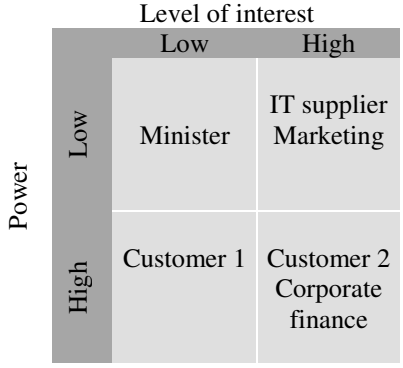


Figure 5.1.2.2.2. Example power / interest matrix

This form of stakeholder mapping can easily be adapted by changing 'the organizations strategy" in 'the architectural description". That way the dimensions are:

1. How interested is each stakeholder (group) to impress its expectations on the architectural description.
2. Whether they have the means to do so. This is concerned with the power of stakeholder groups.

The power / interest matrix is very useful when an outside party tries to identify the most important stakeholders in an organization. It's mainly a descriptive way of stakeholder analysis, from which an conceptualization architect can derive his own way of stakeholder mapping.

5.1.2.3 Project Risk Score

Sachs (2001) recognized the problem that IEEE 1471 does not contain a method for stakeholder analysis and proposed an extension of the IEEE 1471 model to include a project risk score. A project risk score consists of a stakeholder's influence and their willingness to accept project results. Figure 5.1.2.3.1 shows the main ingredients of a project risk score and figure 5.1.2.3.2 gives an example of how a project risk score can be calculated by multiplying willingness and influence.

Influence	**Willingness**
0. No influence 1. Some influence, need to listen to 2. Some influence, need to listen and act 3. In collusion with others, can cause the Architectural Description to be rejected 4. Can single-handedly cause the Architectural Description to be rejected	1. Sees need for an enterprise Architectural Description and inclined to accept results 2. Neutral 3. Does not see a need for an Architectural Description and would likely reject the results)

Figure 5.1.2.3.1. Project risk score ingredients

Stakeholder title	Name	Willingness	Influence	Risk score
Vice president	Jones	3	3	9
Board Chairman	Smith	2	4	8
Financier	Lawson	1	2	2

Figure 5.1.2.3.2. Example of project risk score calculation

Sachs' method of determining the most important stakeholders for an Architectural Description is also useful in developing an architectural viewpoint. This will be shown by an example in paragraph 5.1.3.

This method is a prescriptive method, that is particularly useful when a large number of stakeholders is identified, so a precise hierarchy of the most important stakeholders can be created.

5.1.3 Example Stakeholder Analysis

This section features an example stakeholder analysis for our case. First the conceptualization functions are identified, then both ways of stakeholder analysis are used to determine the most important stakeholders.

5.1.3.1 Conceptualization Functions

In this section the conceptualization functions in our case are identified, with the help of the questions proposed in paragraph 5.1.1.

Who take part in discussions about language/terminology?
In the Eridani example, these are all agents in the Health Care Administration department and the systems design department.

Who are authors of certain language specifying texts?
These are the authors of the guidebook "declaration processing": the process design department.

Who decide upon "official" terminology?
This is mainly the process design department, but the department of health also creates a lot of terms.

Who are responsible for conceptualization management?
The operational responsibility lies with the head of the healthcare administration and the final responsibility at the board of directors.

Who is a facilitator for the conceptualization process?
The head of the health care administration is the facilitator for the conceptualization process in the example.

5.1.3.2 Determining the Most Important Stakeholders
In this example the stakeholders that were found in the previous paragraph are analyzed by the two methods of stakeholder mapping.

Power / interest matrix
In this power / interest matrix all agents from our example are placed:

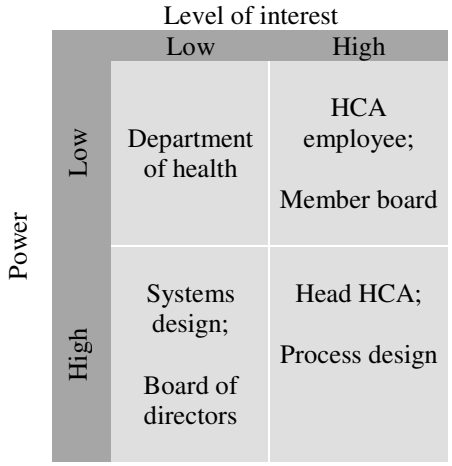


Figure 5.1.3.2.1. Power / interest matrix for Eridani Insurance

Project risk score
In this project risk score matrix in figure 5.1.3.2.2 all agents from our example are placed.

Stakeholder title	Willingness	Influence	Risk score
Government: Department of health	2	0	0
Member board	2	0	0
Board of directors	2	4	8
Head health care administration	1	3	3
Process design	2	2	4
Systems design	2	2	4
Health care administration employee	1	1	1

Figure 5.1.3.2.2. Project risk score for Eridani Insurance

5.2 Models and Descriptions
This paragraph describes the models and descriptions of which the architectural viewpoint consists. The models are not models of actual conceptualization situations (views) but models by which views can be created. Paragraphs 5.3. through 5.6 describe the Discourse Environment and its elements. Paragraphs 5.7 and 5.8 describe conceptualization processes.

5.3 Conceptual Framework

The conceptual framework (or conceptual model) is the most important model of the Architectural Viewpoint, because it contains the central, essential elements and relations that have to do with conceptualization processes and their environment. It provides the elements and relations that the models that are described in the next paragraphs also use. Without the conceptual framework it would be very hard to create those models because the elements they use would not be placed in their context.

The conceptual framework is clearly a descriptive model. It does not provide an exact method for filling in the concept when creating a view, but this is where the other models come in. The conceptual model that is described in this paragraph could in principle be transformed into a database of 'conceptualization relations' that could be populated using the other models and description templates.

The conceptual framework was created using NIAM. Not only is NIAM very useful because it uses natural language in modeling, the most important reason for using NIAM to create the conceptual framework was that NIAM creates a relational model, and examining the elements and relation in a conceptualization environment was exactly the goal of the conceptual framework. The next paragraphs explain how main model was created by modeling example clauses that were provided by the domain expert. The most important constraints were added to the model by asking questions to the domain expert.

The domain expert based the example clauses that were modeled on the GAK case that was used in the larger research effort. Here the complexity of the GAK case was not a problem, because the final model abstracts from the case, and the domain expert was capable of understanding it.

5.3.1 Main Model

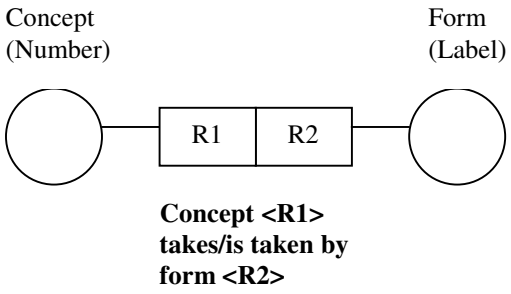
The main model was created by modeling example clauses. Since this part of the paper takes the form of a modeling assignment, the domain expert created the example clauses. Because the domain expert had some experience with modeling languages the step from deriving the example clauses from a domain description was done by him. The example clauses are shown in Appendix C. An example of how example clauses were modeled is shown in figure 5.3.1.1.

The domain expert created the following two clauses:

concept (number) "C2" takes/is taken by form (label) "cancellation code 2"

agent (name) "CA1" can write/is writable form (label) "cancellation code 2" in artefact (name) "RESA/FASA 1201"

The first clause is modeled as follows:



The second clause is modeled in a ternary fact type:

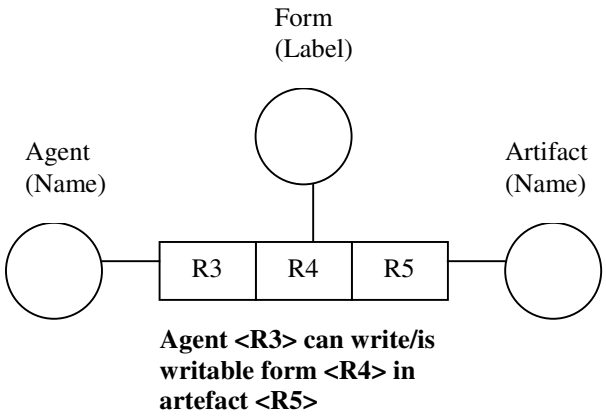


Figure 5.3.1.1. Example clause modeling

The entire main model is shown in figure 5.3.1.2.

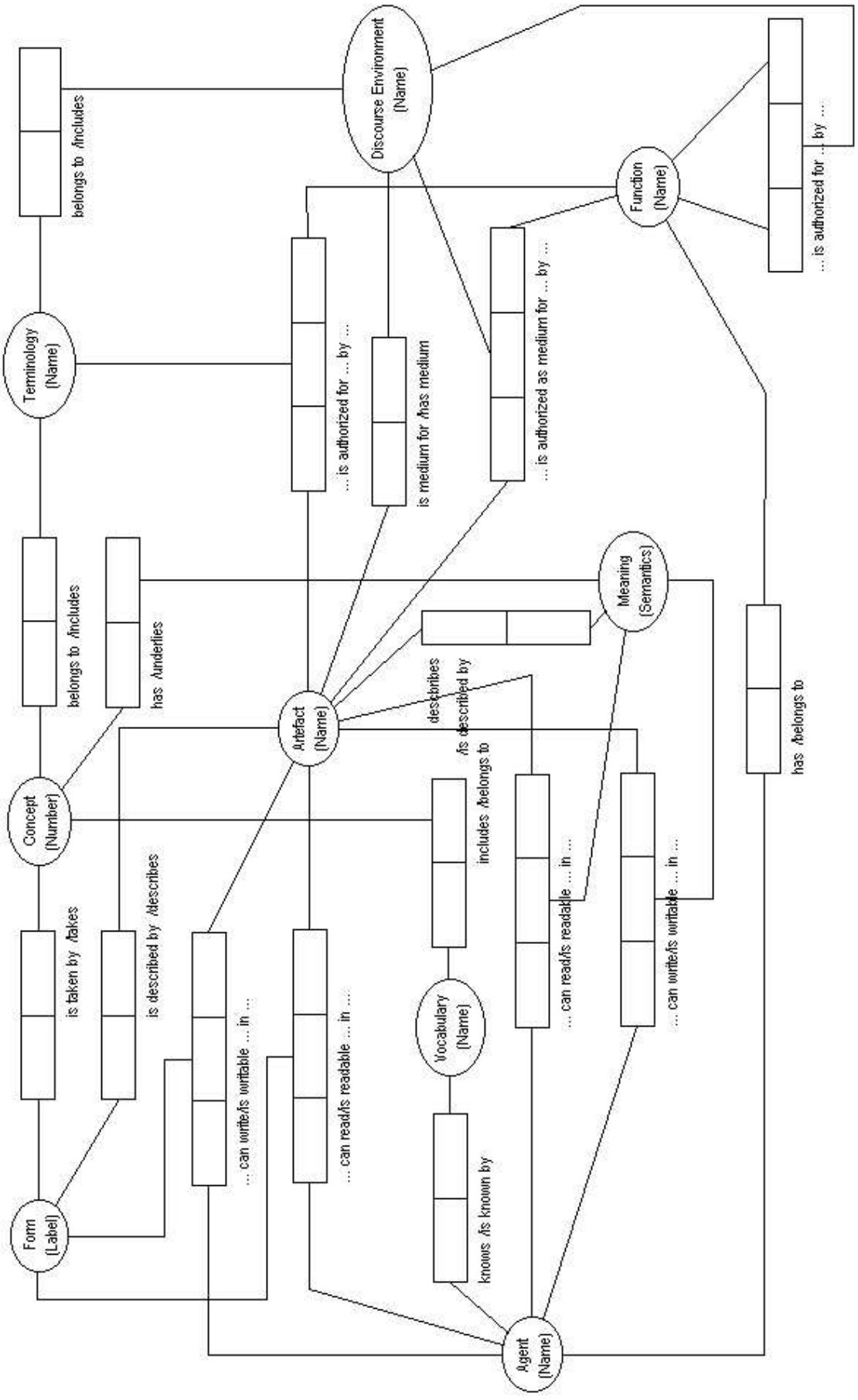


Figure 5.3.1.2. The main model

5.3.2 Constraints

This section describes the modeling of following types of constraints: uniqueness, mandatory role, equality, exclusion, and subset constraints.

Uniqueness constraints

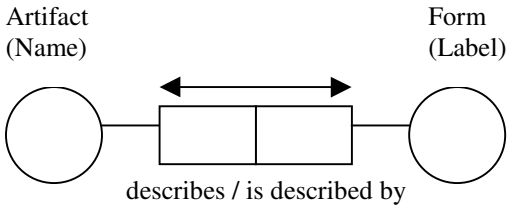
Uniqueness constraints are identified by asking the domain expert questions. These questions and their answers are shown in appendix D. Figure 5.3.2.1. illustrates how questions are used to find uniqueness constraints.

These are two of the questions that were asked to the domain expert:

1a. Can artefact a describe form f and g? – yes
1b. Can artefacts a and b describe form f? – yes

2a. Can concept c belong to terminologies t and u? – no
2b. Can concepts c and d belong to terminology t? – yes

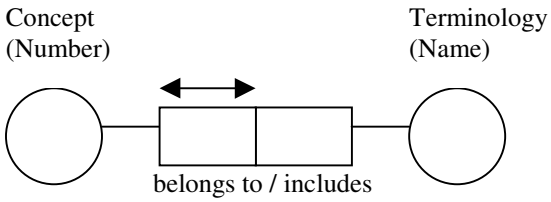
From question one can be concluded that one artifact can describe two different forms, and that one form can be described in two different artifacts. Therefore there is no single-role uniqueness constraint and the rule that orders every fact type to have at least one uniqueness constraint applies and the following uniqueness constraint is placed:



a	f
a	g
b	f

All of the examples below the fact type are valid.

The second question tells us that a terminology may include two different concepts, but that each concept may only belong to one terminology. This implies a uniqueness constraint on the concept role:



c	t	Not allowed
c	u	
d	t	

In short this uniqueness constraint forbids double concepts below the constraint.

There are also situations in which there is a one on one relation between different roles. A good example is the fact that vocabularies are ‘personal’ to agents. An agent knows only one vocabulary and a vocabulary is known by a single agent.

Figure 5.3.2.1. Example uniqueness constraints

All uniqueness constraints that were identified using questions and directly from the examples are displayed in figure 5.3.2.2.

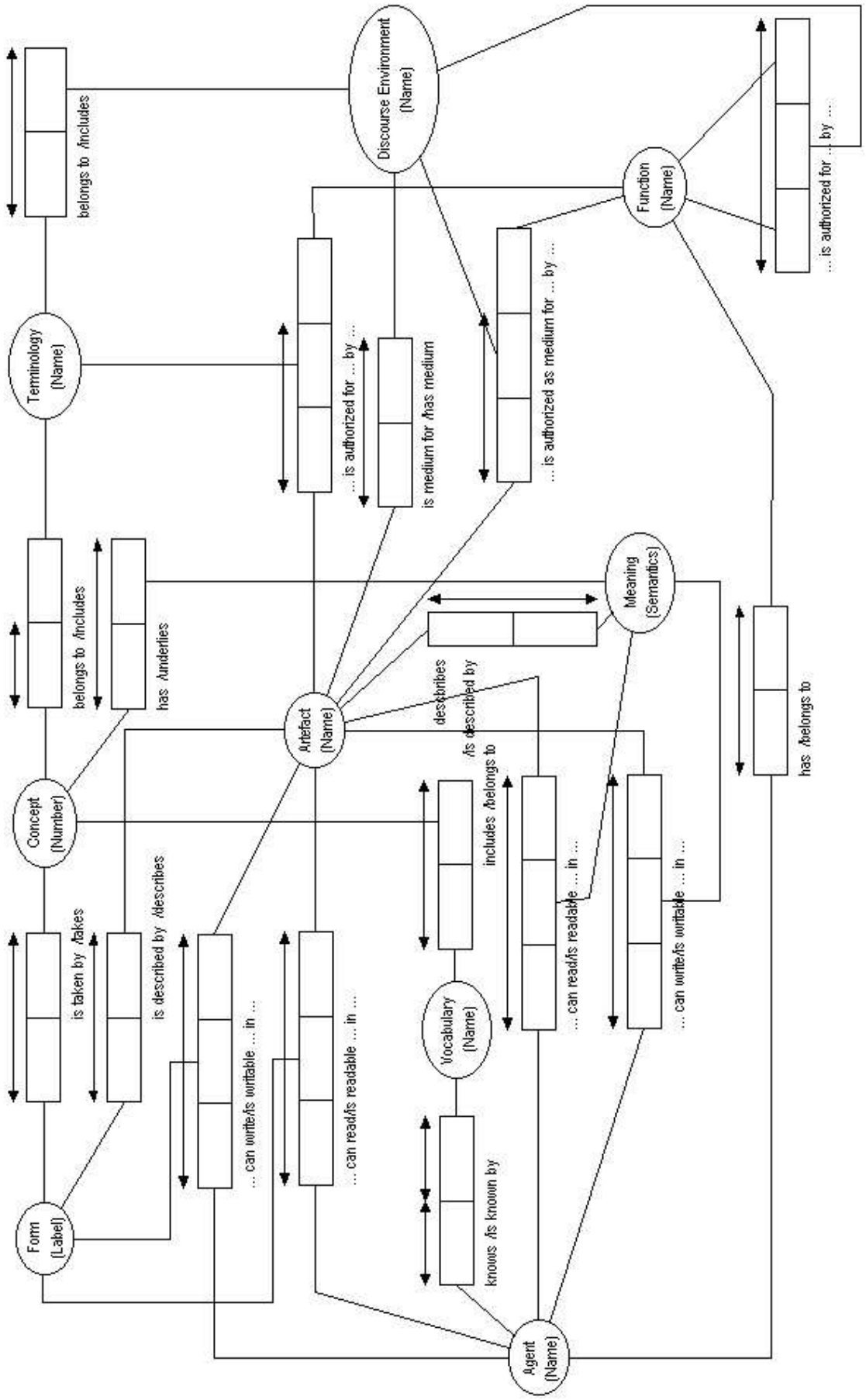


Figure 5.3.2.2. Uniqueness constraints

Mandatory role constraints

Mandatory role constraints too were identified by asking questions to the domain expert. The questions are displayed in appendix E. Figure 5.3.2.3. shows how mandatory role constraints can be derived from those questions.

Some of the questions that were asked to the domain expert were:

Concept

Are there concepts that don' t have form? – NO
Are there concepts that aren' t part of any terminology? – YES
Are there concepts that don' t have meaning? – NO
Are there concepts that aren' t part of any vocabulary? – NO

These answers suggest that: each concept has form, each concept has meaning, and each concept is part of a vocabulary, but not every concept is part of a terminology. Therefore mandatory role constraints are placed on every role except the role "belongs to/includes":

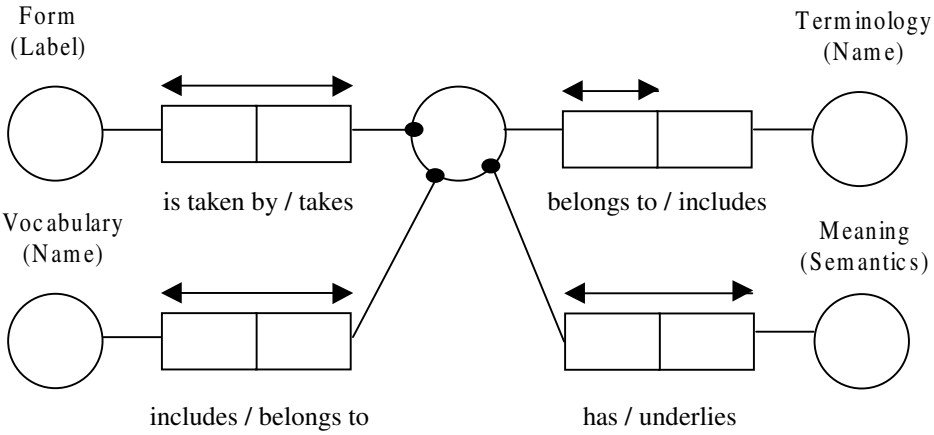


Figure 5.3.2.3. Example mandatory role constraints

All mandatory role constraints that were identified are displayed in figure 5.3.2.4.

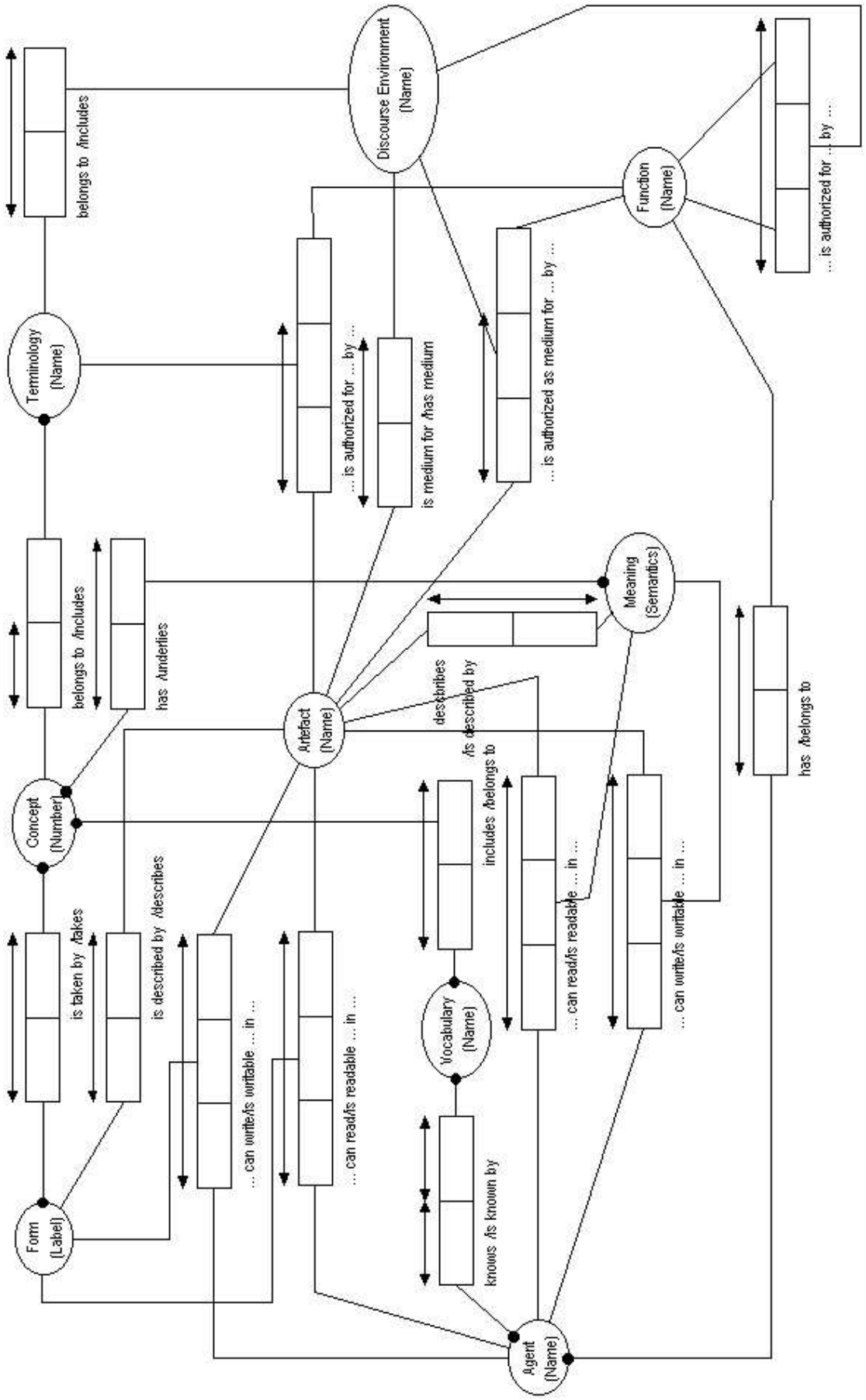


Figure 5.3.2.4. Mandatory role constraints

Equality, exclusion, and subset constraints

Mandatory role constraints were the hardest to discover. They too were identified by asking questions to the domain expert. The questions are displayed in appendix F. Figure 5.3.2.5. shows how mandatory role constraints can be derived from those questions.

Some of the questions that were asked to the domain expert were:

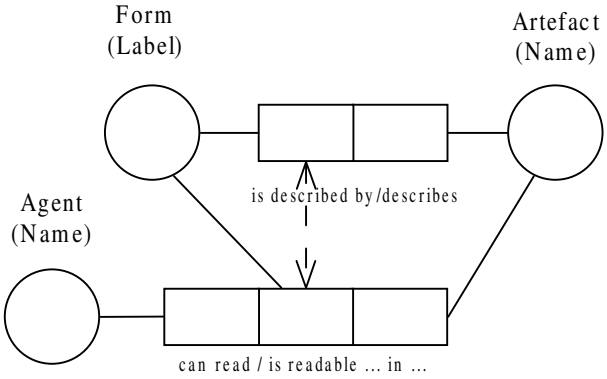
Form

[sub] Is every form f that can be read by an agent in an artifact also described by an artifact? - YES
[sub] Can every form f that is described by an artifact also be read by an agent in an artifact? - YES
[x] Is NO form f that can be read by an agent in an artifact also described by an artifact? - NO

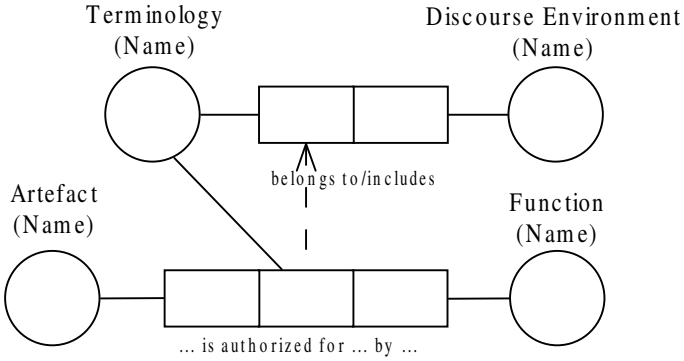
Terminology

[sub] Does every terminology t that belongs to a Discourse Environment also have an artifact authorized for it by a function? - NO
[sub] Does every terminology t that has an artifact authorized for it by a function also belong to a Discourse Environment? - YES
[x] Does NO terminology t that belongs to a Discourse Environment also have an artifact authorized for it by a function? - NO

The answers to the three questions for form imply the following equality constraint, because of double inclusion:



The answers to the questions for Terminology imply the following subset constraint:



No exclusion constraints were identified.

Figure 5.3.2.5. Example equality and subset constraints

All equality and subset constraints that were identified are displayed in figure 5.3.2.6.

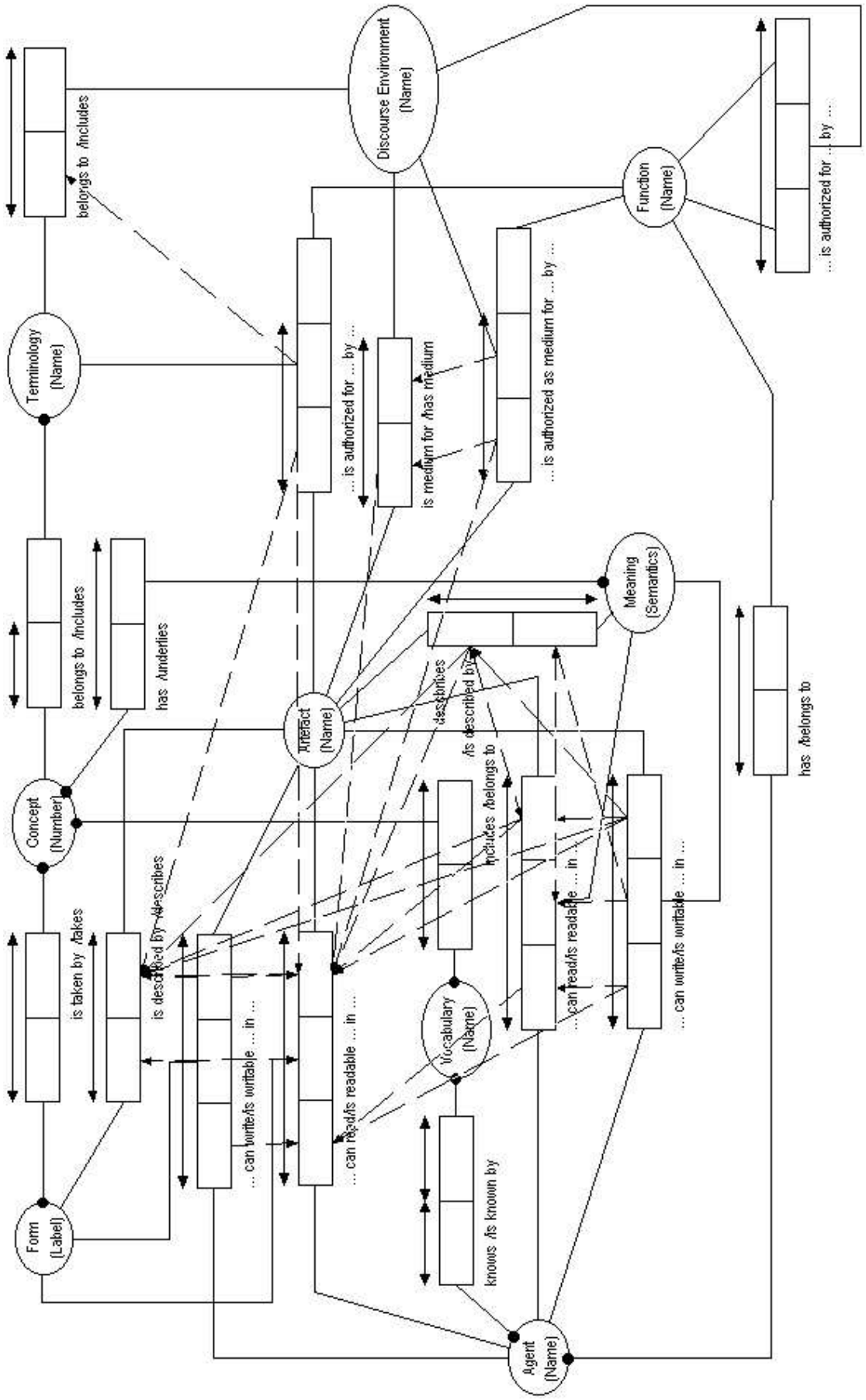


Figure 5.3.2.6. Equality and subset constraints

5.4 Functions

As the conceptual framework already showed, Discourse Environments contain agents who perform certain functions. It is important to know who fulfills certain conceptualization functions, because agents with different functions perform different roles in a conceptualization process. You may have noticed an overlap between functions and stakeholders. This is not a coincidence. We already explained that the stakeholders that are addressed by the viewpoint for conceptualization have concerns regarding conceptualization. It is clear that agents who perform certain conceptualization functions also have concerns regarding conceptualization. Therefore they can also be considered stakeholders. Because of the overlap, identifying functions is the logical step to be performed before identifying stakeholders. That is why paragraph 5.1.1 already provided a method for identifying conceptualization functions.

Paragraph 5.1.1 identified the following conceptualization functions:

- Language informants
- Language authors
- Language authority
- Concept manager
- Conceptualization facilitator

Functions can be described by the following aspects:

- Degree of expertise in DE
  - How much knowledge does a person with the function have about the DE?
- Degree of expertise in conceptualization
  - How much expertise with explicit conceptualization processes, methods and techniques does a person with the function have?
- Degree of expertise in concept management
  - How much expertise with concept management processes, methods and techniques does a person with the function have?
- Authority in operational matters
  - Strategic, operational, no authority, …
- Authority in conceptual matters
  - How much power does the person with the function have to allocate resources to conceptualization?
- Amount of resources allocated for conceptualization
  - How much time is available for conceptualization?

Like stakeholders, functions depend on the organization for which a conceptualization viewpoint is created. Therefore it is not possible to create a comprehensive descriptive model of functions. However a method for identifying functions is provided by this checklist:

<table>
<tr><td colspan="3">1. Degree of expertise in DE</td></tr>
<tr><td colspan="3">How much knowledge does a person with the function have about the DE?</td></tr>
<tr><td colspan="2">❑ None
❑ Little
❑ Normal</td><td>❑ High
❑ Expert</td></tr>
<tr><td colspan="3">2. Degree of expertise in conceptualization</td></tr>
<tr><td colspan="3">Expertise with explicit conceptualization processes, methods and techniques?</td></tr>
<tr><td><u>Processes</u>:
❑ None
❑ Little
❑ Normal
❑ High
❑ Expert</td><td><u>Methods</u>:
❑ None
❑ Little
❑ Normal
❑ High
❑ Expert</td><td><u>Techniques</u>:
❑ None
❑ Little
❑ Normal
❑ High
❑ Expert</td></tr>
</table>

3. Degree of expertise in concept management		
Expertise with concept management processes, methods and techniques?		
Processes:	Methods:	Techniques:
❑ None	❑ None	❑ None
❑ Little	❑ Little	❑ Little
❑ Normal	❑ Normal	❑ Normal
❑ High	❑ High	❑ High
❑ Expert	❑ Expert	❑ Expert
4. Authority in operational matters		
How much authority does a person with the function have in operational matters?		
❑ None	❑ Other:	
❑ Operational	.....	
❑ Strategic		
5. Authority in conceptual matters		
How much power does the person with the function have to allocate resources to conceptualization?		
❑ None	❑ Normal	
❑ Little	❑ High	
6. Amount of resources allocated for conceptualization		
How much time is available for conceptualization?		
❑ 0 – 2 hours a day	❑ 6 – 8 hours a day	
❑ 2 – 4 hours a day	❑ > 8 hours a day	
❑ 4 – 6 hours a day		

Figure 5.4.1. Functions checklist

Example: Eridani insurance

To illustrate the use of the checklist, the checklist is filled out for the head of the health care administration:

1. Degree of expertise in DE		
How much knowledge does a person with the function have about the DE?		
❑ None	☑ High	
❑ Little	❑ Expert	
❑ Normal		
2. Degree of expertise in conceptualization		
Expertise with explicit conceptualization processes, methods and techniques?		
Processes:	Methods:	Techniques:
❑ None	❑ None	❑ None
☑ Little	☑ Little	☑ Little
❑ Normal	❑ Normal	❑ Normal
❑ High	❑ High	❑ High
❑ Expert	❑ Expert	❑ Expert
3. Degree of expertise in concept management		
Expertise with concept management processes, methods and techniques?		
Processes:	Methods:	Techniques:
☑ None	☑ None	☑ None
❑ Little	❑ Little	❑ Little
❑ Normal	❑ Normal	❑ Normal
❑ High	❑ High	❑ High
❑ Expert	❑ Expert	❑ Expert
4. Authority in operational matters		
How much authority does a person with the function have in operational matters?		
❑ None	❑ Other:	
☑ Operational	.....	
❑ Strategic		
5. Authority in conceptual matters		
How much power does the person with the function have to allocate resources to conceptualization?		
❑ None	☑ Normal	
❑ Little	❑ High	

6. Amount of resources allocated for conceptualization	
How much time is available for conceptualization?	
☑ 0 – 2 hours a day ❑ 2 – 4 hours a day ❑ 4 – 6 hours a day	❑ 6 – 8 hours a day ❑ > 8 hours a day

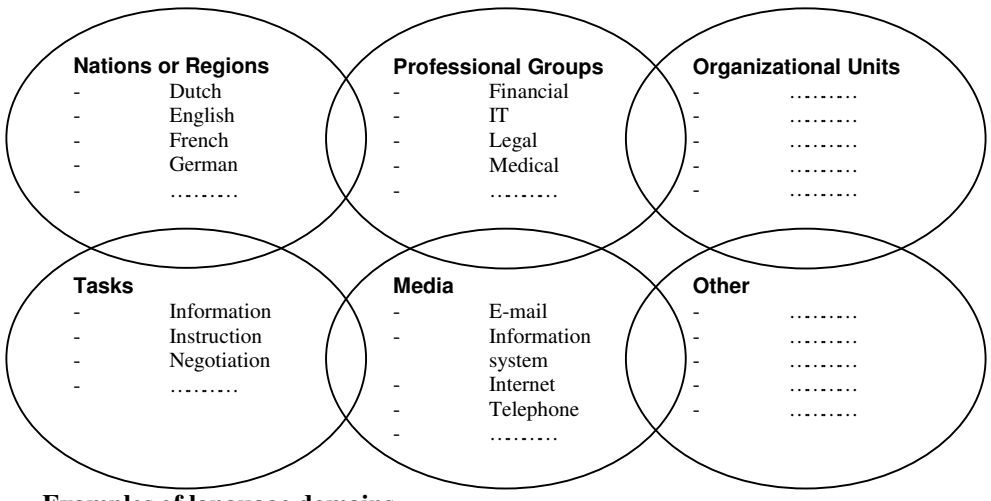
Figure 5.4.2. Functions checklist for head health care administration

5.5 Language Use Typology

The language use typology is used to determine what different types of language use are present in the Discourse Environment. It is important to know what types of language are used, because the type of language use influences the type of conceptualization in meta-communication. For example it is likely that when the language use is very official, the concepts will be made explicit, will be widely distributed and documented, and some authority is closely involved. More examples of language use will be given in paragraphs 5.5.1 through 5.5.10. Together they form a descriptive model or typology of language use. Paragraph 5.5.11 provides a method that can be used when creating a conceptualization view.

5.5.1 Language Domains

This paragraph gives a number of important distinctions by which language domains can categorized. Of course there are a lot of other origins of language domain categorization, as long as people actively and successfully invent labels for language domains, and use these to single out some subset of terms they consider relevant, for whatever reason. There can also be an overlap between two or more domains. Language domains can also be found in the main model. The vocabularies of (groups of) agents can be based on a certain language domain and artifacts can serve as a medium. Some language domains are displayed in figure 5.5.1.1.



Examples of language domains

- Language domains may overlap

Figure 5.5.1.1. Language domains

Nations or regions

Often language domains are categorized by the nation or region where the language originated. For example "English" or "Dutch".

Professional groups

Another distinction can be made in terms of the group of people using them. Take "legal language" used by people in legal professions and "medical language" used in medical professions.

Organizational units

Yet another way of categorizing language domains is in terms of organizational units. (IBM, South-East Branch, Projects)

Tasks

Language domains can also be categorized by the tasks that have to be performed, for example negotiation or instruction.

Media used

And finally by the medium system or the communication environment that has to be used, for example the Internet.

5.5.2 Communicationally Used Language versus Computationally Used Language

This is the distinction between *the construction of symbolic representations as part of a computational machine* and *the use of symbols for human-to-human communication*. It is important to note that these two types of communication refer to the primary use of the language. For example when you look at a programming language, a higher level programming language contains a lot of symbols that are borrowed from natural language that are intended to make the programming language easier to understand. When using a programming language that resembles natural language, the programmer can also communicate with other programmers (human-to-human communication). However the main purpose of the programming language is a computational one. The distinction between communicationally used language and computationally used language is relevant when regarding conceptualization processes, because a computationally used language has greater need for making concepts explicit. The distinction between communicationally used language and computationally used language is displayed in figure 5.5.2.1.

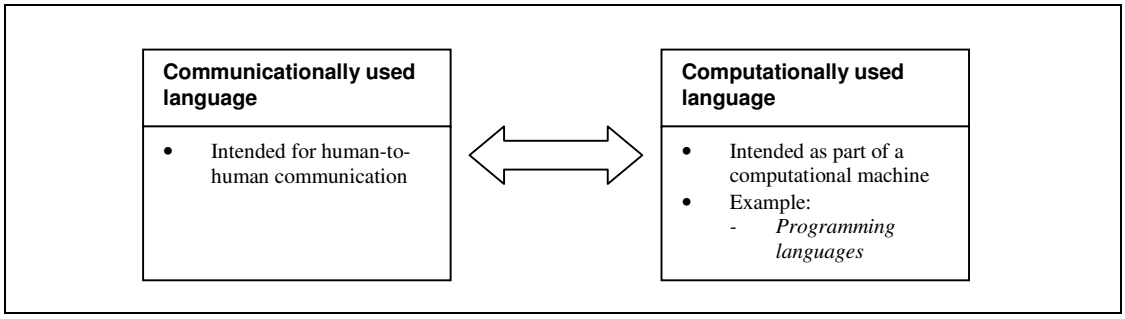


Figure 5.5.2.1. Communicationally used language vs. computationally used language

5.5.3 Operational Language Use versus Auxiliary Language Use

Operational language use concerns the use of language for the execution of the task at hand. Auxiliary language use is meant to support operational language use. Examples of such supporting languages are meta communication, technical instruction language, management language and language for socialization. While it can easily be stated the focus should be on operational language use, the importance of auxiliary language use should not be underestimated. Without auxiliary language, conceptualization processes would not be possible. The distinction between operational versus auxiliary language use is displayed in figure 5.5.3.1.

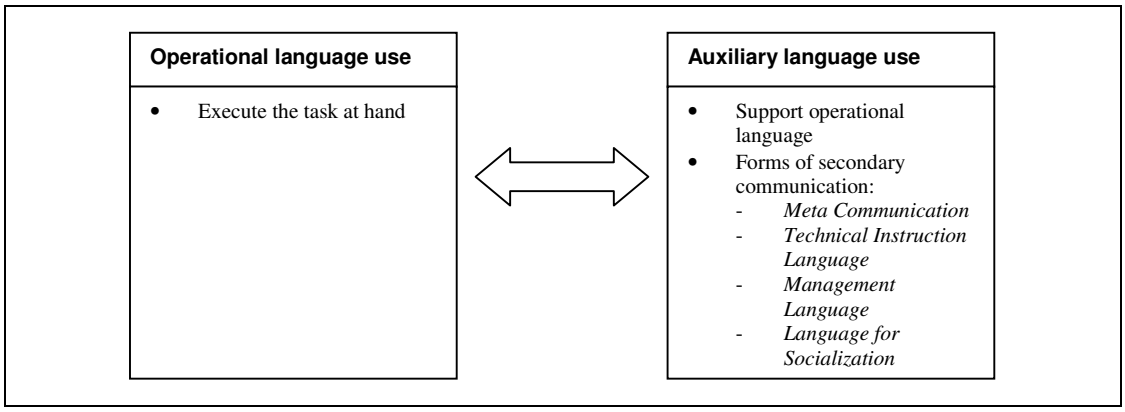


Figure 5.5.3.1. Operational language use vs. auxiliary language use

5.5.4 Local versus Global Language Use

Language may be intended by a few (or even one) agents (local) or for a more widespread use (global). Globality is usually related to geographical space, but can also refer to cyberspace. A group of agents in different countries can be separated in geographical space, but close in cyberspace, if connected by a medium system.

When physical distance is no longer an issue it's better to measure distance using the notion of 'cultural space'. Using this notion globality is more linked to the number of different agents than to the physical distance between agents.

The distinction between local language use and global language use is important. It is easy to imaging that when language is used globally problems with concepts can appear when a lot of different nationalities meet. This distinction is displayed in figure 5.5.4.1.

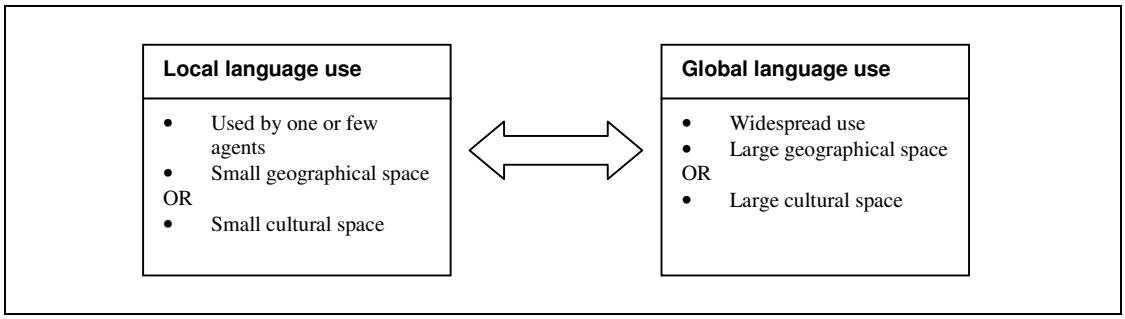


Figure 5.5.4.1. Local language use vs. global language use

5.5.5 General versus Specialized Language Use

This distinction is based on the degree of expertise of agents. General language asks little specific expertise of agents, while specialized language assumes considerable expertise.

The notion of general vs. specialized use should not be confused with global vs. local use or technical (computational) vs. communicational use. Language can be specialized and still not technical (for example specialized legal language) and spoken by a small group, but still general (or vise versa). It can be imagined that specialized language requires that concepts are made explicit. The distinction between general and specialized language use is displayed in figure 5.5.5.1.

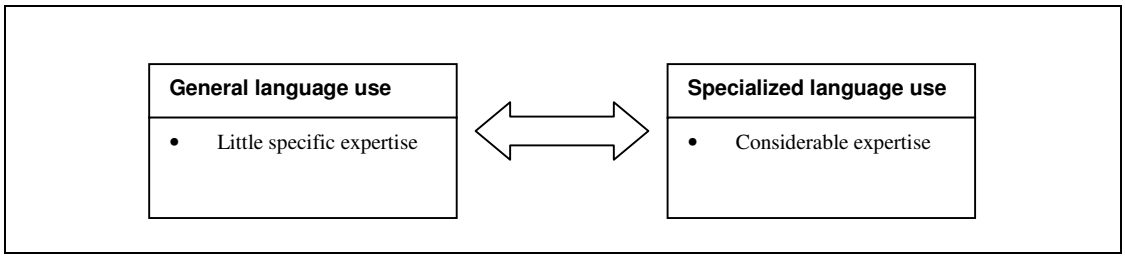


Figure 5.5.5.1. General language use vs. specialized language use

5.5.6 Degree of Persistency of Representations Generated

It is important whether language used applies to a single interpretation (single speech act) or to persistent representation (text). When used for a single interpretation it can be better estimated what the audience will be. A text may be around much longer and distributed more widely, therefore it's harder to predict the audience and also flexibility will become an issue. This is why it's important to know how long the product of some language will last. The degree of persistency is displayed in figure 5.5.6.1.

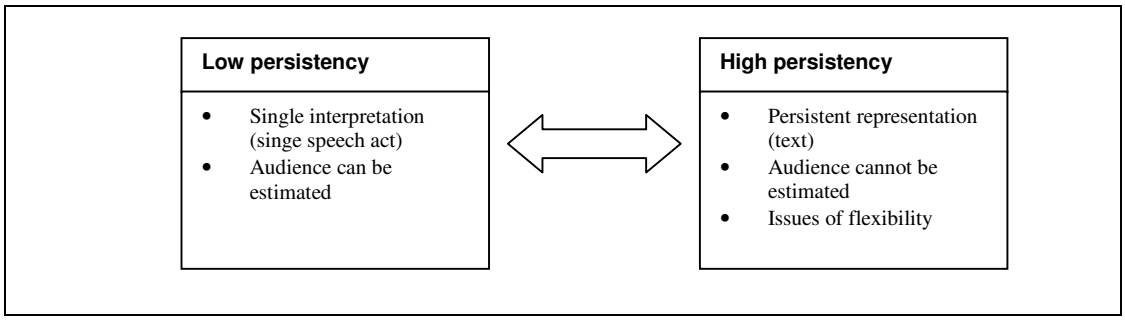


Figure 5.5.6.1. Degree of persistency of representations generated

5.5.7 Productive Language Use versus Receptive Language Use

This is not a very important distinction, it's sole purpose is to determine whether a particular vocabulary to be used productively (in speech or writing) or receptively (listening or reading). When language is used productively, the demands on concepts are higher. The distinction between productive and receptive language use is displayed in figure 5.5.7.1.

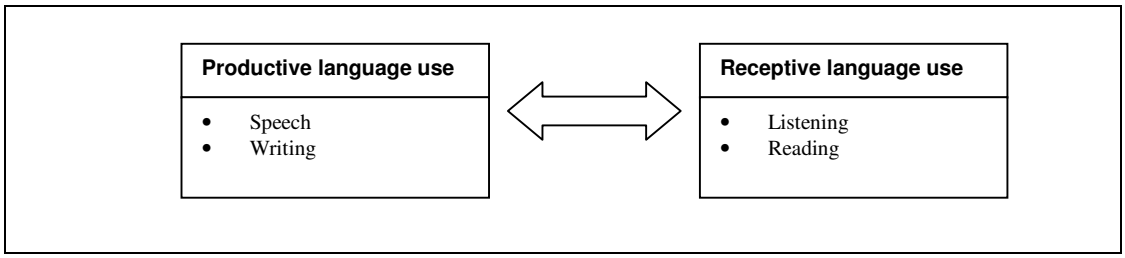


Figure 5.5.7.1. Productive language use vs. receptive language use

5.5.8 Standardization versus Flexibility

Standards are not only used often, but also very important to language use. Standards are often used to restrict communication, but may also facilitate communication. When a Discourse Environment has a lot of agent with different vocabularies, or when the language in a DE is rapidly changing, standards may be difficult to keep. Circumstances like these ask flexible languages. The distinction between standardization and flexibility is displayed in figure 5.5.8.1.

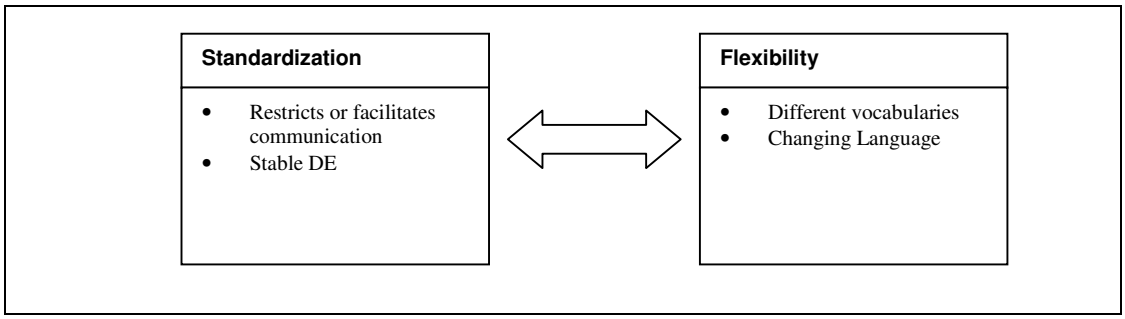


Figure 5.5.8.1. Standardization vs. flexibility

5.5.9 Official Language Use

Official language is defined as the use of the language that is usually preferred by authority. Often this means a terminology is used.

5.5.10 Open Language Use versus Closed Language Use

This is a very important distinction. It is about whether or not the language use is actively restricted by the medium system. When language use is closed a medium system is involved that restricts language. It is not enough for explicit agreement on language to exist. The distinction between open and closed language use is displayed in figure 5.5.10.1.

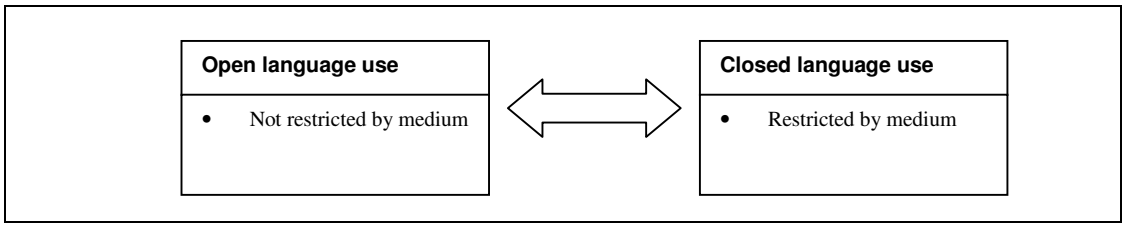


Figure 5.5.10.1. Open language use vs. closed language use

5.5.11 Language Use Checklist

The language use checklist can be filled out when creating a conceptualization view to do an inventory of the types of language used in an organization. The language use checklist can be used when population the conceptual model from paragraph 5.3 and is also useful when describing artifacts.

1. Domains	
What are the language domains that are part of the DE in question?	
1.1. Nations or regions	
Which of the following languages is spoken in the DE?	
❑ Dutch ❑ English ❑ French	❑ German ❑ Other: ……
1.2. Professional groups	
Which professional groups are represented in the DE?	
❑ Financial ❑ Information Technology ❑ Legal	❑ Medical ❑ Other: ……
1.3. Organizational units	
What are the most important organizational units in the DE?	
❑ IT ❑ Sales ❑ Marketing	❑ Finance ❑ Other: ……
1.4. Tasks	
What are the most important tasks that have to be performed in the DE?	
a. General tasks:	
❑ Performative action (making new 'facts') ❑ Informative action (communicating 'facts')	❑ Precondition creating action (e.g. conventions) ❑ Socializing
b. Domain-specific tasks:	
❑ …… ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
1.5. Media	
What types of media are used in the DE?	
❑ E-mail ❑ Information system ❑ Internet	❑ Radio ❑ Telephone ❑ ……
1.6. Other	
What other language domains can be found in the DE?	
❑ …… ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
2. Communicationally used language vs. computationally used language	
For each vocabulary in the DE is its primary use communicational or computational?	
Communicational	Computational
❑ …… ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
3. Operational language use vs. auxiliary language use	
Which forms of secondary language use can be found in the DE?	
❑ Meta Communication ❑ Technical Instruction Language ❑ Management Language	❑ Language for Socialization ❑ Other: ……

4. Global language use vs. local language use (cultural space)	
What is the number of different agents in the DE?	
❑ 0-10 ❑ 10-100 ❑ 100-500	❑ 500-1000 ❑ > 1000
In what space do the agents in the DE communicate?	
❑ Task ❑ Department ❑ Company	❑ National ❑ International (one or more countries) ❑ Global (worldwide)
5. General language use vs. specialized language use	
What is the degree of expertise in the DE?	
❑ Low ❑ Medium ❑ High	
6. Degree of persistency of representations generated	
What are the most important ways of communication in the DE?	
❑ Speech (recorded) …% ❑ Speech (not recorded) …% ❑ Texts (digital) …%	❑ Texts (paper) …% ❑ Other: …… …%
For each document in the DE: How long does the document last?	
❑ …… ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
7. Productive language use vs. receptive language use	
For each vocabulary in the DE, is it meant for productive use (writing, speech) or receptive use (listening, reading)?	
Productive	Receptive
❑ …… ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
8. Standardization vs. flexibility	
Does the DE require standardized or flexible languages?	
❑ Standardized	❑ Flexible
What standards are used in the DE?	
❑ …… ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
9. Official language use	
Is there official language use in the DE?	
❑ Yes	❑ No
If yes: What level imposes the official language use?	
❑ Governmental (e.g. national or local legislation) ❑ Strategical (e.g. the board of directors)	❑ Operational (e.g. department head) ❑ Other: ……
10. Open language use vs. closed language use	
For every medium system in the DE define whether it is open or closed (restricts possible language use or not).	
Open	Closed
❑ …… ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……

Figure 5.5.11.1. Language use checklist

Example: Eridani insurance

To do an inventory of the types of languages used in Eridani Insurance, the checklist is filled in for the Discourse Environment 'declaration processing':

1. Domains	
What are the language domains that are part of the DE in question?	
1.1. Nations or regions	
Which of the following languages is spoken in the DE?	
❑ Dutch ☑ English ❑ French	❑ German ❑ Other: ……
1.2. Professional groups	
Which professional groups are represented in the DE?	
☑ Financial ☑ Information Technology ☑ Legal	☑ Medical ❑ Other: ……
1.3. Organizational units	
What are the most important organizational units in the DE?	
☑ IT (Systems design department) ❑ Sales ❑ Marketing	❑ Finance ☑ Other: Process design department
1.4. Tasks	
What are the most important tasks that have to be performed in the DE?	
a. General tasks:	
❑ Performative action (making new 'facts') ❑ Informative action (communicating 'facts')	☑ Precondition creating action (e.g. conventions) ❑ Socializing
1.5. Media	
What types of media are used in the DE?	
❑ E-mail ☑ Information system ❑ Internet	❑ Radio ❑ Telephone ❑ ……
2. Communicationally used language vs. computationally used language	
For each vocabulary in the DE is its primary use communicational or computational?	
Communicational	Computational
☑ Declaration processing ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
3. Operational language use vs. auxiliary language use	
Which forms of secondary language use can be found in the DE?	
☑ Meta Communication ❑ Technical Instruction Language ☑ Management Language	❑ Language for Socialization ❑ Other: ……
4. Global language use vs. local language use (cultural space)	
What is the number of different agents in the DE?	
❑ 0-10 ☑ 10-100 ❑ 100-500	❑ 500-1000 ❑ > 1000
In what space do the agents in the DE communicate?	
❑ Task ☑ Department ❑ Company	❑ National ❑ International (one or more countries) ❑ Global (worldwide)
5. General language use vs. specialized language use	
What is the degree of expertise in the DE?	
❑ Low ❑ Medium ☑ High	

6. Degree of persistency of representations generated	
What are the most important ways of communication in the DE?	
❑ Speech (recorded) …% ☑ Speech (not recorded) 60 % ❑ Texts (digital) …%	☑ Texts (paper) 40 % ❑ Other: …… …%
For each document in the DE: How long does the document last?	
☑ Guidebook Declaration Processing: 6 months or until a change in the products of Eridani Insurance.	❑ …… ❑ …… ❑ ……
7. Productive language use vs. receptive language use	
For each vocabulary in the DE, is it meant for productive use (writing, speech) or receptive use (listening, reading)?	
Productive	Receptive
☑ Transaction processing ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
8. Standardization vs. flexibility	
Does the DE require standardized or flexible languages?	
☑ Standardized	☑ Flexible
What standards are used in the DE?	
☑ Governmental guidelines for health care insurance ❑ …… ❑ ……	❑ …… ❑ …… ❑ ……
9. Official language use	
Is there official language use in the DE?	
☑ Yes	❑ No
If yes: What level imposes the official language use?	
☑ Governmental (e.g. national or local legislation) ❑ Strategical (e.g. the board of directors)	☑ Operational (e.g. department head) ❑ Other: ……
10. Open language use vs. closed language use	
For every medium system in the DE define whether it is open or closed (restricts possible language use or not).	
Open	Closed
❑ …… ❑ …… ❑ ……	☑ Declarations 96 ❑ …… ❑ ……

Figure 5.5.11.2. Language use checklist

5.6 Artifacts

Artifacts play an important role in the conceptual framework. Commonly used artifacts in Discourse Environments are dictionary-like texts, conceptual specifications and auxiliary documents. Examples of dictionary like texts are, pocket dictionaries, translation dictionaries, historical dictionaries, glossaries, thesauri, legal definitions and data dictionaries. Examples of conceptual specifications are data structures. Examples of auxiliary documents are manuals, procedural descriptions etc. Artifacts are also related to the language use model. When we look at the example of official language use again, the artifacts will also be more authoritative and will not be handwritten notes. To describe an artifact the following aspects can be used:

- *Medium*: The artifact is based on what medium? (e.g. paper, web page, etc.)
- *Relation to medium system*: Is the artifact the documentation of a system, part of a specification of a system (a data structure), a process description, a data dictionary etc?
- *Source*:
  - Author: Who created the artifact?
  - Organization: What organization is responsible for the artifact?
- *Authoritativeness*: How authoritative is the artifact? (Is it a guideline, or a 'law'?)

- *Distribution / availability*: How widespread is the artifact in the organization? How many agents use the artifact?
- *Version*: How frequently is the artifact renewed? How can a new version of the artifact be obtained?

These aspects are captured in the following checklist, which can be filled out when creating a conceptualization view:

1. Medium	
The artifact is based on what medium?	
❑ Paper (unstructured text) ❑ Online document ❑ CD-ROM	❑ Book (structured text) ❑ Other: ……
2. Relation to medium system	
How is the artifact related to a medium system?	
❑ Documentation ❑ Specification of a system ❑ Process description	❑ Data dictionary ❑ Other: ……
3. Source	
Who is/are the author(s) of the artifact?	
❑ …… ❑ ……	❑ …… ❑ ……
What organization(s) is/are responsible for the artifact?	
❑ …… ❑ ……	❑ …… ❑ ……
4. Authoritativeness	
How authoritative is the artifact?	
❑ Guideline ❑ Recommended ❑ Obligatory	❑ By law ❑ Other: ……
5. Distribution/availability	
What percentage of the organization uses the artifact?	
❑ 0 – 20% ❑ 20 – 40% ❑ 40 – 60%	❑ 60 – 80% ❑ 80 – 100%
6. Version	
How frequently is the artifact renewed?	
❑ Never ❑ Every ……hours / days / months / years	❑ In the following situation: ……
How can a new version of the artifact be obtained?	
❑ Online ❑ Automatically ❑ Artifact is delivered	❑ Look for updates manually ❑ Other: ……

Figure 5.6.1. Artifact checklist

This list can be used when creating a viewpoint and therefore is (part of) a method to describe conceptualization.

Example: Eridani insurance

Eridani Insurance does not use a lot of artifacts. The healthcare administration employees use dictionaries, but these are very common and are not maintained by the organization. The main artifact in the Eridani case is the guidebook "Declaration Processing". For this document the following checklist is created:

1. Medium	
The artifact is based on what medium?	
❑ Paper (unstructured text) ❑ Online document ❑ CD-ROM	☑ Book (structured text) ❑ Other: ……
2. Relation to medium system	
How is the artifact related to a medium system?	
❑ Documentation ❑ Specification of a system ☑ Process description	❑ Data dictionary ❑ Other: ……
3. Source	
Who is/are the author(s) of the artifact?	
☑ Process design department ❑ ……	❑ …… ❑ ……
What organization(s) is/are responsible for the artifact?	
☑ Eridani Insurance ❑ ……	❑ …… ❑ ……
4. Authoritativeness	
How authoritative is the artifact?	
☑ Guideline ❑ Recommended ❑ Obligatory	❑ By law ❑ Other: ……
5. Distribution/availability	
What percentage of the organization uses the artifact?	
☑ 0 – 20% ❑ 20 – 40% ❑ 40 – 60%	❑ 60 – 80% ❑ 80 – 100%
6. Version	
How frequently is the artifact renewed?	
❑ Never ☑ Every 6 ~~hours~~ / ~~days~~ / ~~weeks~~ / months / ~~years~~	☑ In the following situation: In case of a change in the products Eridani Insurance offers.
How can a new version of the artifact be obtained?	
❑ Online ❑ Automatically ☑ Artifact is delivered	❑ Look for updates manually ❑ Other: ……

Figure 5.6.2. Artifact checklist for the guidebook Declaration Processing

5.7 I/O Model

This section will describe an input/output (or I/O model) for conceptualization processes. An I/O model is a 'black box model" of a system. In a black box model only the in- and output of a system are known, the system boundaries and the relations between the components of the system are not. All that is known about the system, is that it contains a transformation process that transforms certain input into certain output.

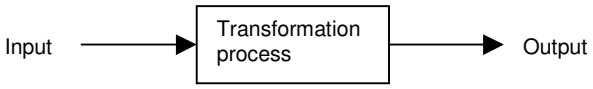


Figure 5.7.1. A black box model

In this case the transformational process is the conceptualization process. The input of the conceptualization process are the artifacts that contain the concepts the organization wants to look at in the conceptualization process. The output of the conceptualization process are the same artifacts, but

possibly containing new concepts. The I/O model is descriptive rather than prescriptive, because it does not provide methods by which the output can be compared with the input. To compare the output of conceptualization with the input, concepts should be compared on a textual level. Because this thesis does not look at the description of specific concepts, no methods will be given to compare concepts, but it is possible to compare artifacts at a more general level, by comparing the descriptions of artifacts.

Example: Eridani insurance

In case of Eridani insurance, the input and output for the conceptualization process is the guidebook "Declaration processing". Figure 5.7.2 shows how this artifact has been changed after the conceptualization process. Changes are marked in grey.

Input	Output
Medium	**Medium**
The artifact is based on what medium?	*The artifact is based on what medium?*
☑ Book (structured text)	☑ Book (structured text)
Relation to medium system	**Relation to medium system**
How is the artifact related to a medium system?	*How is the artifact related to a medium system?*
☑ Process description	☑ Process description
Source	**Source**
Who is/are the author(s) of the artifact?	*Who is/are the author(s) of the artifact?*
☑ Process design department	☑ Process design department
What organization(s) is/are responsible for the artifact?	*What organization(s) is/are responsible for the artifact?*
☑ Eridani Insurance	☑ Eridani Insurance
Authoritativeness	**Authoritativeness**
How authoritative is the artifact?	*How authoritative is the artifact?*
☑ Guideline	Recommened
Distribution/availability	**Distribution/availability**
What percentage of the organization uses the artifact?	*What percentage of the organization uses the artifact?*
☑ 0 – 20%	☑ 0 – 20%
Version	**Version**
How frequently is the artifact renewed?	*How frequently is the artifact renewed?*
☑ Every 6 ~~hours~~ / ~~days~~ / ~~weeks~~ / months / ~~years~~ ☑ In the following situation: In case of a change in the products Eridani Insurance offers.	☑ Every 6 ~~hours~~ / ~~days~~ / ~~weeks~~ / months / ~~years~~ ☑ In the following situation: In case of a change in the products Eridani Insurance offers.
How can a new version of the artifact be obtained?	*How can a new version of the artifact be obtained?*
☑ Artifact is delivered	☑ Artifact is delivered
Concept descriptions before conceptualization process	**Concept descriptions after conceptualization process**
☑ Concept A ☑ …… ☑ ……	☑ Concept A' ☑ …… ☑ ……

Figure 5.7.2 Adaptation of an artifact

5.8 Conceptualization Process Model

As the I/O model is a black box model, the conceptualization process model could be considered a white box model. In a white box model the boundaries of a system and the system components are known.

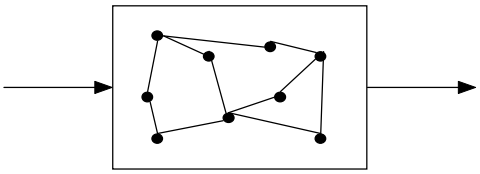


Figure 5.8.1. A white box model

The elements of conceptualization processes and their relations are known now. This paragraph describes how these elements interact to form a conceptualization process. To describe a conceptualization process, this section describes four steps that are involved in explicit conceptualization (Hoppenbrouwers, forthcoming). Some of these steps may be implicit or even optional. They aren't meant to show how a conceptualization process should go, but rather how it could go. Therefore the conceptualization process model is a descriptive model. The conceptualization process model also serves as a framework for reference, based on which conceptualization processes can be described in a view based on this viewpoint. That is why the conceptualization process model also serves as a method for the description of conceptualization.

Though some of the steps in a conceptualization process are optional, the process itself could largely be considered sequential. That is why flowcharts are very useful to create this model. Each step is described in a box connected by arrows to show the sequence. For each step the information will be added whether the step is *optional* (the conceptualization architect may determine whether it should be performed), *compulsory (implicit)* (the step must be performed, but this may happen intuitively), or *compulsory (explicit)* (the step must be performed and this should happen explicitly). The four steps are shown in figure 5.8.2.

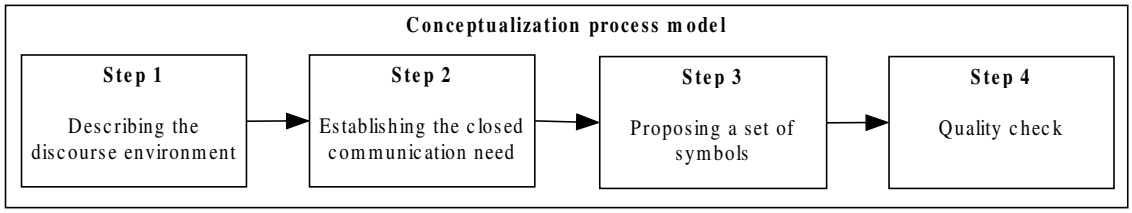


Figure 5.8.2. Conceptualization process model

5.8.1 Step 1: Describing the Discourse Environment

This step is compulsory (implicit).

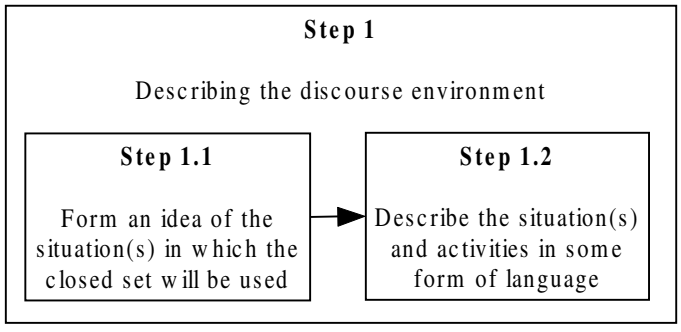


Figure 5.8.1.1. Step 1. Describing the Discourse Environment

Step 1.1: Form an idea of the situation(s) in which the closed set will be used
The goal of this step is to make sure the DE and its boundaries become clear and that the situational factors and insights that apply to the DE as a whole are noted and charted. In this step it may be required to distinguish sub-DEs.

Step 1.2: Describe the situation(s) and activities in some form of language
In this step a description of the situation and the relevant activities that take place in it is produced. This may be done using natural language or diagrams. In this step the first representations of individual concepts take shape.

5.8.2 Step 2: Establishing the Closed Communication Need

In this step, a decision is made about which concepts to freeze. It is also possible that the decision is made that no closed communication is required. This step is compulsory (implicit).

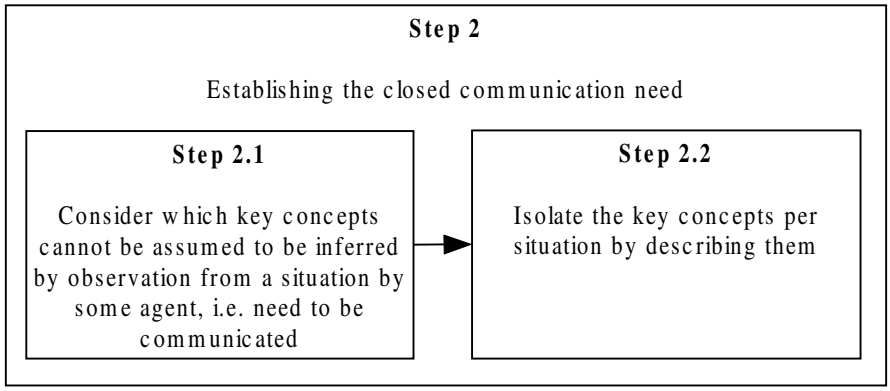


Figure 5.8.2.1. Step 2. Establishing the closed communication need

Step 2.1: Consider which key concepts cannot be assumed to be inferred by observation from a situation by some agent, i.e. need to be communicated
In step 2.1. the descriptions of the DE situations produced in step 1 are scanned for relevant concepts. In an ideal situation each word in the description should be evaluated as to its communicational value, and whether it can be used to convey information that is expected not the be part of common knowledge. When this criterion isn't adhered to strictly, there's a certain risk that some concept cannot be expressed through the medium. Therefore, how step 2.1. is performed, should depend on the communicative situation as analyzed in step 1.1.

Step 2.2: Isolate the key concepts per situation by describing them
This step is very essential. Someone has to suggest which concepts will constitute the closed set. Candidate symbols may be suggested, but definitive symbols should not yet be proposed.

5.8.3 Step 3: Proposing a Set of Symbols

The concepts selected in step 2 may be represented in a number of ways by 'definitive symbols'. Choosing definitive symbols involves different considerations than isolating key concepts. This is why step 2 and step 3 should be different steps.

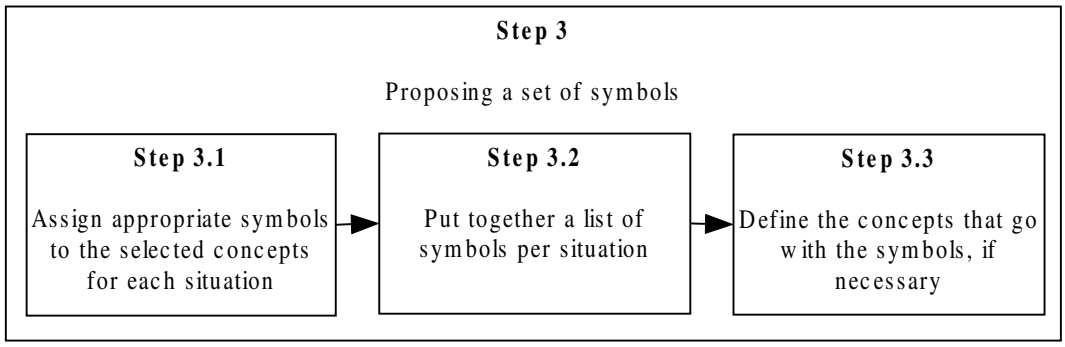


Figure 5.8.3.1. Step 3. Proposing a set of symbols

Step 3.1: Assign appropriate symbols to the selected concepts for each situation

This step is compulsory (implicit).

Step 3.2: Put together a list of symbols per situation

This step is compulsory (explicit).

Step 3.3: Define the concepts that go with the symbols, if necessary

This step is optional.

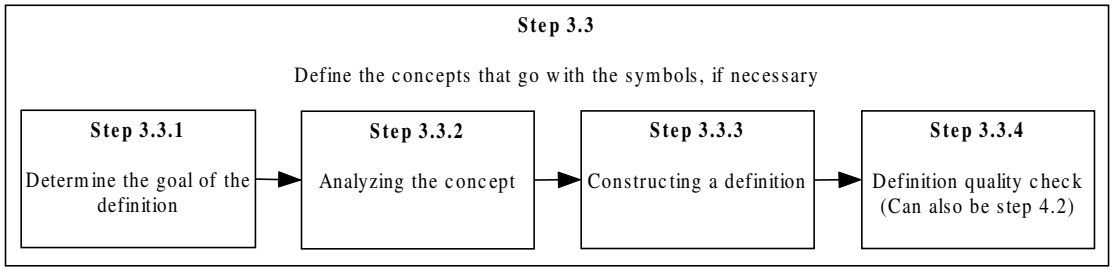


Figure 5.8.3.2. Step 3.3. Define the concepts that go with the symbols, if necessary

Step 3.3.1: Determining the goal of the definition

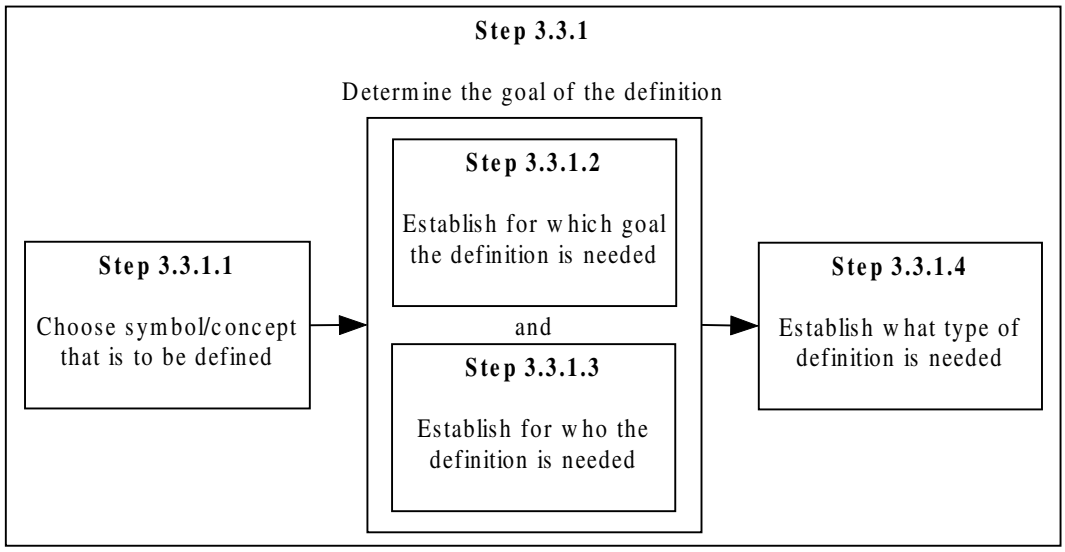


Figure 5.8.3.3. Step 3.3.1. Determine the goal of the definition

Step 3.3.1.1: Choose symbol/concept that is to be defined

Part of this step is to determine whether or not to engage in definition. This is an important choice, because once a definition exists it becomes a term. A term carries some authority (i.e. it can be used to cut short discussions about meaning). A definition may become a burden: it may awkwardly or needlessly fix interpretation of a word that would otherwise be treated more 'liberally', or the definition may overrule other useful meanings (homonyms) of the word. This is why in some cases it may be better not to engage in definition, or abandon a definition.

Step 3.3.1.2: Establish for which goal the definition is needed

This step and step 3.3.1.3. are best performed in parallel, because they are closely related. In these two steps some of the questions that should be answered are:

- For what purpose will the definition be used?
- What authority will the definition bear?
- What is the risk if the definition is insufficiently precise?
- What is the risk if the definition is too precise?
- Who will use the definition, will all agents be able to understand it, and will they benefit from it or be burdened by it in some respect?

Step 3.3.1.3: Establish for who the definition is written

Step 3.3.1.4: Establish what type of definition is needed

In this step a choice is made between stipulative definition (describes what a person or group 'chooses to mean' with a word. Asserts a meaning rather than observing it.), lexical definition (describes what a group of people can be observed to mean with a word) or stipulative-lexical definition (Asserts a meaning, but uses an observated (lexical) meaning as object of stipulation). Also some other important choices are made. In these choices the following questions are important:

- Does the definition aim to describe an existing convention shared by many?
- Or does the definition aim to make clear a particular meaning to be used in a particular DE or involving some particular task?
- Will the lexical knowledge of agents confronted with the definition be changed substantially by it, or will the definition mostly confirm what the agents already 'knew' implicitly?

Step 3.3.2: Analyzing the concept

This step has been divided into two options. Option a. is very useful if it is not clear what the meaning of a word is and an explorative definition process is needed. But in general option b. will be used.

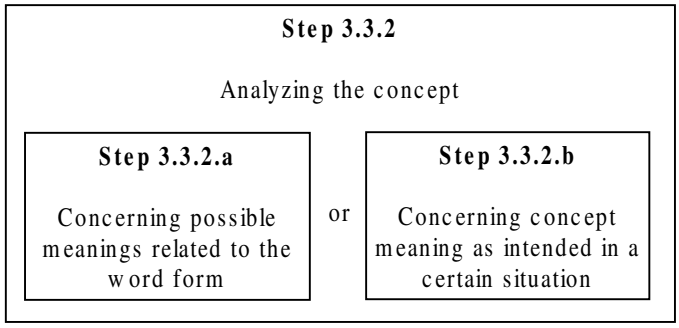


Figure 5.8.3.4. Step 3.3.2. Analyzing the concept

Step 3.3.2.a: Concerning possible meanings related to the word form

1. Check out existing meanings of the word
2. Check out existing derivations of the word, or the stem form of the word; also consider compound forms including the word
3. Check for synonyms of the word in other languages or domains
4. Find out in which combinations the word regularly occurs
5. Find out what are the 'relata' of the word (when the word is 'see', the relata are x sees y, hence the word 'see' has two relata)
6. Check for existing antonyms, hypernyms, and hyponyms of the word
7. Find out the etymology of the word

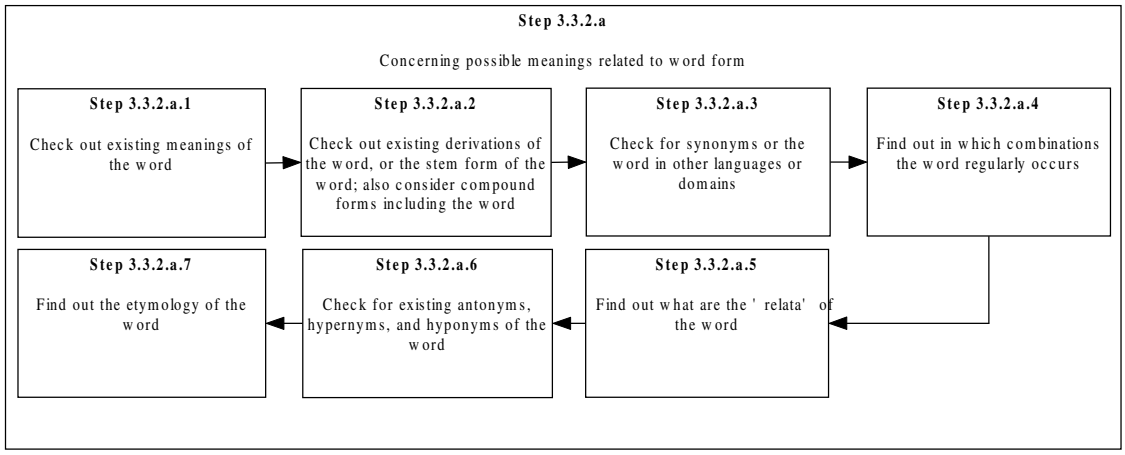


Figure 5.8.3.5. Step 3.3.2.a Concerning possible meanings related to the word form

Step 3.3.2.b: Concerning concept meaning as intended in a certain situation

1. Describe possible meanings of the word in the intended situation, by the defining party or by others (stipulative or lexically), as relevant
2. Find which concepts are related to the intended concept in the intended situation
3. Determine to which semantic categories the intended concept may belong in the intended situation

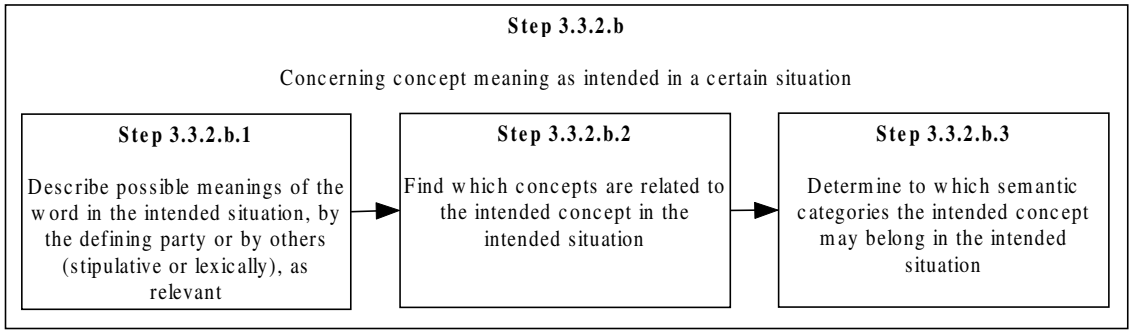


Figure 5.8.3.6. Step 3.3.2.b. Concerning concept meaning as intended in a certain situation

Step 3.3.3: Constructing a definition

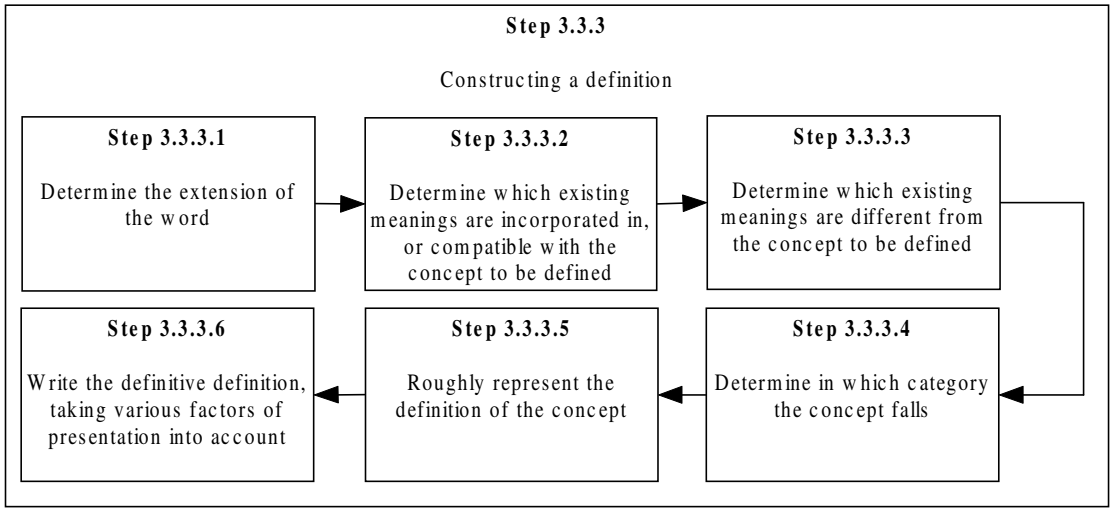


Figure 5.8.3.7. Step 3.3.3. Constructing a definition

Step 3.3.3.1: Determine the extension of the word

By focusing on the extension of the word, it is emphasized that we should not blindly follow some intentional description. Instead we should consider what actual occurrences, references and situations the definition is to cover.

Step 3.3.3.2: Determine which existing meanings are incorporated in, or compatible with the concept to be defined

Step 3.3.3.3: Determine which existing meanings are different from the concept to be defined

Step 3.3.3.4: Determine in which category the concept falls
Categories can cover anything from domains of use in all their variation, but also semantic categories, and often syntactic categories (i.e.: noun, verb, adjective).

Step 3.3.3.5: Roughly represent the definition of the concept

Step 3.3.3.6: Write the definitive definition, taking various factors of presentation into account

Step 3.3.4: Definition quality check
In this step, factors determining the quality of the definition proposed are systematically checked. This could be also done as step 4.2. This step is optional.

5.8.4 Step 4: Quality Check

After the first 3 steps we have a clear set of symbols (possibly including various sub-sets of symbols) representing carefully selected concepts. If it is possible and needed this set should be checked. This step is optional.

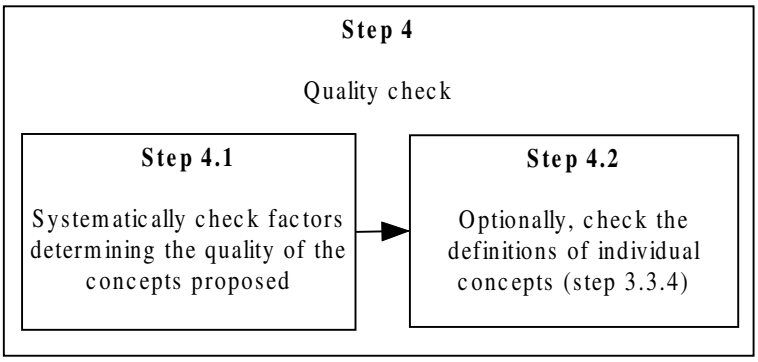


Figure 5.8.4.1. Step 4. Quality check

Step 4.1: Systematically check factors determining the quality of the concepts proposed

Step 4.2: Optionally, check the definitions of individual concepts

5.8.5 Example Conceptualization Process: Eridani Insurance

This paragraph describes an example of how the conceptualization process could have gone at Eridani Insurances. The conceptualization process is completely fictional, because it is hard to find a real conceptualization process, because they are often performed implicitly, if performed at all.

Step 1: Describing the Discourse Environment

Step 1.1: Form an idea of the situation(s) in which the closed set will be used
Eridani insurance wants to use the closed set of concepts in the new declaration processing system. In the part of Eridani Insurance we are studying, two DE's can be identified: declaration processing for the normal clients and for the health care workers.

Step 1.2: Describe the situation(s) and activities in some form of language
The systems design department shows a couple of flow-charts in which the processes in the new declaration processing system are captured. Some health care administration employees comment on the 'error handling' section and the terms that are used there.

Step 2: Establishing the Closed Communication Need

Everyone in the meeting agrees upon the fact that all errors should be redefined. These are the concepts that should be made explicit.

Step 2.1: Consider which key concepts cannot be assumed to be inferred by observation from a situation by some agent, i.e. need to be communicated

The systems department does not want to reconsider all concepts in the DE, because they do not get a lot of complaints about other concepts than the error handling concepts.

Step 2.2: Isolate the key concepts per situation by describing them

A list is made of all error handling concepts, numbered 1 through 50.

Step 3: Proposing a set of symbols

The concepts selected in step 2 may be represented in a number of ways by 'definitive symbols'. Choosing definitive symbols involves different considerations than isolating key concepts. This is why step 2 and step 3 should be different steps.

Step 3.1: Assign appropriate symbols to the selected concepts for each situation

Systems design explains that numbered errors are essential for easy use of the program. The health care administration employees don't object to numbered errors, as long as they can easily look up the meaning of the errors. It is agreed that step 3.3. should be performed.

Step 3.2: Put together a list of symbols per situation

The list of 'error 1' through 'error 51' becomes definitive.

Step 3.3: Define the concepts that go with the symbols, if necessary

Step 3.3.1: Determining the goal of the definition

One of the demands of the Health Care Administration employees was that the errors used in the system should be well-defined.

Step 3.3.1.2: Establish for which goal the definition is needed

It is decided that:

- The error handling concepts will be defined for easy use by the Health Care Administration employees
- The error handling concepts will act as no more than a reference guide for the Health Care Administration employees

This step is performed implicitly.

Step 3.3.1.3: Establish for who the definition is written

The definition is written for the Health Care Administration employees.
This step is performed implicitly.

Step 3.3.1.4: Establish what type of definition is needed

This step is performed implicitly, because everyone intuitively agrees on the type of definition: stipulative definition (describes what a person or group 'chooses to mean' with a word.)

Step 3.3.2: Analyzing the concept

In general option b is used, and Eridani insurance is no exception.

Step 3.3.2.b: Concerning concept meaning as intended in a certain situation

The health care administration employees all have their own opinion about the meaning of the error handling codes. Everyone can give his opinion and if necessary a higher person in the organization ("who knows things like that") is asked about his opinion.

Step 3.3.3: Constructing a definition

Step 3.3.3.1: Determine the extension of the word

The step is not performed.

Step 3.3.3.2: Determine which existing meanings are incorporated in, or compatible with the concept to be defined
The step is not performed explicitly.

Step 3.3.3.3: Determine which existing meanings are different from the concept to be defined
The step is not performed explicitly.

Step 3.3.3.4: Determine in which category the concept falls
The step is not performed.

Step 3.3.3.5: Roughly represent the definition of the concept
During the conversation one of the health care administration employees takes notes about the considered definitions and the final definitions.

Step 3.3.3.6: Write the definitive definition, taking various factors of presentation into account
The health care administration employee who has taken notes creates a list of the definitive concepts and sends it to the systems department to be used in the new system.

Step 3.3.4: Definition quality check
A quality check will not be performed right now, because the system will be tested thoroughly when it is finished.

Step 4: Quality check

A quality check is not performed.

Figure 5.8.5.1 contains an image of the conceptualization process at Eridani Insurance.

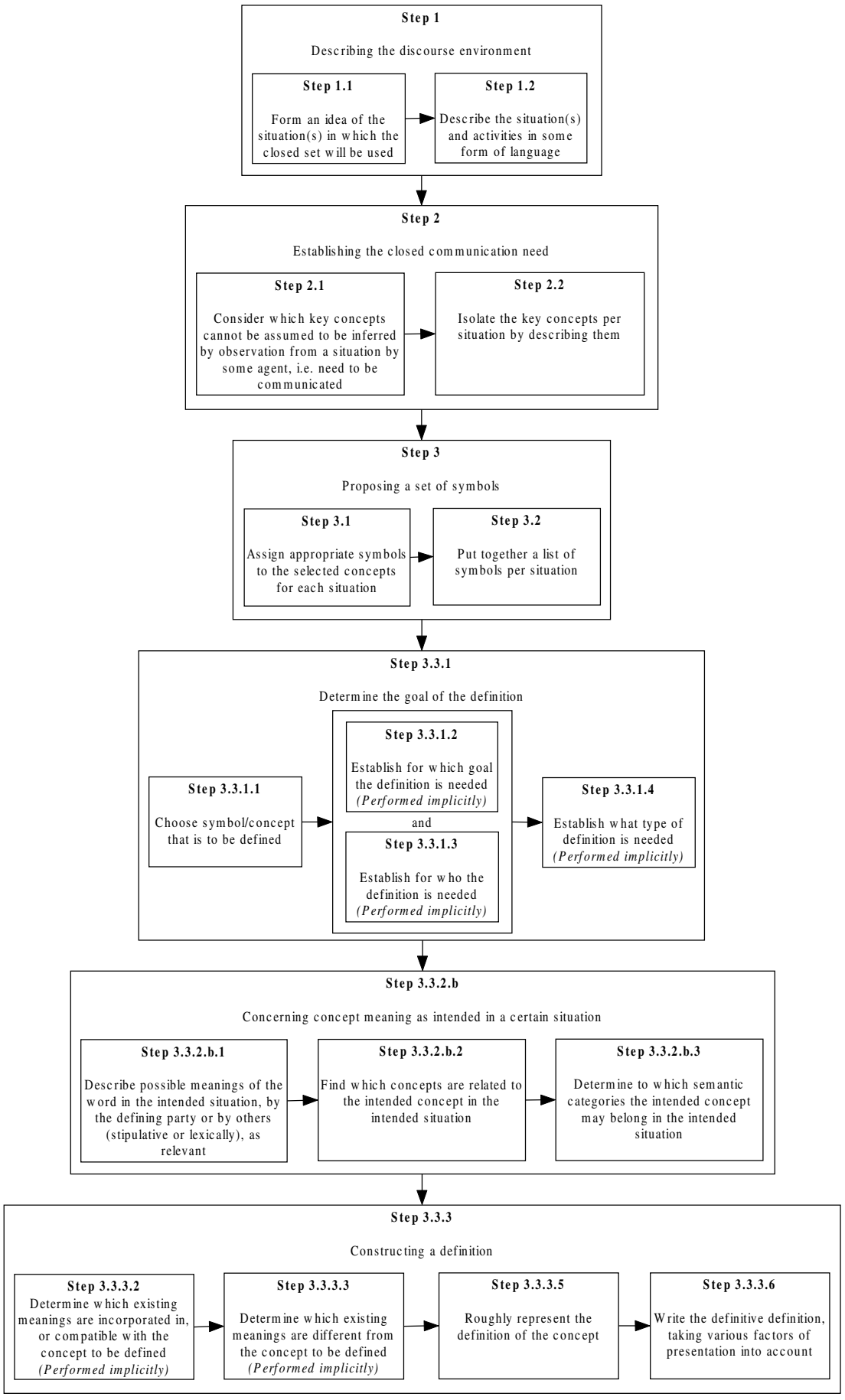


Figure 5.8.5.1. Conceptualization process at Eridani Insurance

6 Conclusion

The research described in this thesis aimed at finding out how architecture could be used to chart conceptualization processes and the complex environment they take place in. Chapter five describes an architectural viewpoint that contains two types of descriptive instruments. The first type models elements of conceptualization and the environment of conceptualization (Main Model). The second type can be used when creating a conceptualization view, gathering information that can be boiled down to the essential concepts as described in the Main Model. It constitutes instruments for charting conceptualization processes in organizations. Though the interest for conceptualization in organizations is not very big yet, it is recommendable to apply the viewpoint in modeling a real situation. Not only will the applying of the viewpoint create more awareness of conceptualization, but it will also help to improve the viewpoint itself.

In the conceptualization viewpoint no real attention has been paid to concept management. Concept management was not omitted by accident. Not a lot of theory about concept management is known yet, and therefore it is hard to set up concept management theories. Again this is something where the conceptualization viewpoint may come in handy. One of the demands of being able to manage a process is knowing in general what the process is about. And this is exactly what the conceptualization viewpoint does: give more insight in conceptualization processes. Based on the conceptualization view(point) heuristics can be created on how to deal with conceptual problems and demands can be made regarding things like time and costs. This is where the conceptualization viewpoint contributes to the bigger research effort.

It is also interesting to note that although IEEE standard 1471 is very software-oriented, it can still be used as a method for creating a viewpoint like one for conceptualization. However this will require the use of some business modeling methods (for example stakeholder analysis) and plain text for what cannot be modeled (for example the artifacts).

In short it can be said that though there is a lot of research still to do, the conceptualization viewpoint gives more insight in conceptualization processes and brings managing of those processes a step closer.

Appendixes

A NIAM

NIAM (also known as Object Role Modeling or ORM) stands for Nijssen' s Information Analysis Method and was originally developed by G.M. Nijssen in 1974 (Nijssen, 1993). NIAM acknowledges the importance of communication for actual systems development. In paragraph 5.3 NIAM was be used to create the conceptual framework.

A.1 Goal of NIAM

Creating an information system (computerized or otherwise) is a complicated process. Therefore if one wants to be successful in creating an information system, the developing process of that information system should be very structured. NIAM is one of the methods that were developed to assist in this process. As its name already suggests, NIAM is a method for information analysis. Information analysis contains all activities that lead to complete and formal specifications of the requirements of an information system.

The main purpose of NIAM is to create a usable information system through a conceptual structure abstracted from the real-world system. This is done in a number of steps creating an information structure diagram.

A.2 Practice of NIAM

This section describes the steps that are used in this thesis to create an information structure diagram. It uses an example to illustrate the process.

A.2.1 Modeling sentences

The first step in creating an information structure diagram was modeling sentences. These sentences are a lot like natural language. Some examples of sentences are:

“The person with name Adams has student number (decimal number) 1”
“The person with name Williams has student number (decimal number) 2”

“The person with name Adams has email address (string) d.adams@student.kun.nl”
“The person with name Williams has email address (string) r.williams@student.kun.nl”

These sentences are classified, using the following terms:
LOT: Lexical object type
NOLOT: Non-lexical object type
IN: Individual name
VP: Verb part

This is done here:

| NOLOT | LOT | IN |VP | NOLOT | LOT |IN|
“The person with name Adams has student number (decimal number) 1”

| NOLOT | LOT | IN |VP | NOLOT | LOT | IN |
“The person with name Adams has email address (string) d.adams@student.kun.nl”

To transform the qualified sentence into a diagram, a set of symbols is used. This set is displayed in figure A.2.1.1.

Symbol	Meaning
R1	Unary facttype
R1 R2	Binary facttype
R1 R2 R3	Ternary facttype
R1 … Rn	n-ary facttype
A (B)	Non-lexical objecttype A with lexical objecttype B (1 on 1)
C	Lexical objecttype C

Figure A.2.1.1. symbols used in information structure diagrams

Using these symbols, a simple information structure diagram can be created from the sentences. This diagram is displayed in figure A.2.1.2.

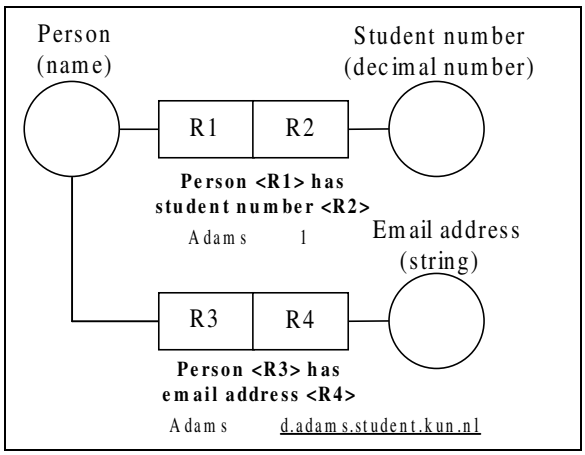


Figure A.2.1.2. Information structure diagram

Figure A.2.1.2 shows that:
1. The qualification NOLOT corresponds with a circle.
2. The name of NOLOT has been written above the circle.
3. The qualification LOT corresponds with '(' and ')'.
4. The name of Lot has been written between '(' and ')'.
5. There are as many rectangles as there is IN in the modeled sentence.
6. Individual names are written below the text in bold.
7. VP's are written in bold under the rectangles.
8. When you replace the parts between '<' and '>' (for example R1 or R2), or 'roles', with the individual names, the original sentences are returned. (For example: 'Person Adams has email address d.adams@student.kun.nl)

A.2.2 Constraints

Now the main model has been created, we need to know whether we can fill in just everything as individual name. Often this is not the case, because certain combinations just aren't allowed. This is where constraints come in. Five types of constraints were modeled in paragraph 5.3. That is why they are discussed here shortly.

Uniqueness constraints

The internal uniqueness (denoted by an arrow above a role) means that an individual name can only appear once in that role. More practically: the internal uniqueness constraint does not accept duplicates 'below the arrow'. There are four types of internal uniqueness constraints. They are displayed in figure A.2.2.1. through A.2.2.4.

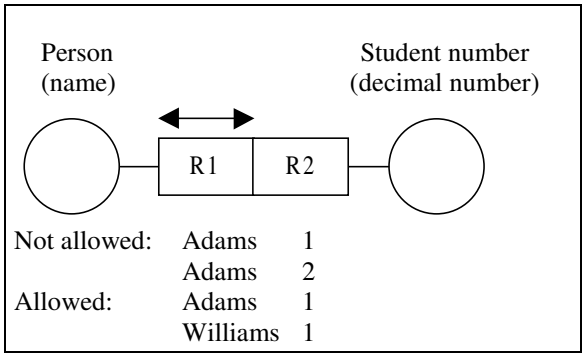


Figure A.2.2.1. Uniqueness constraint type 1

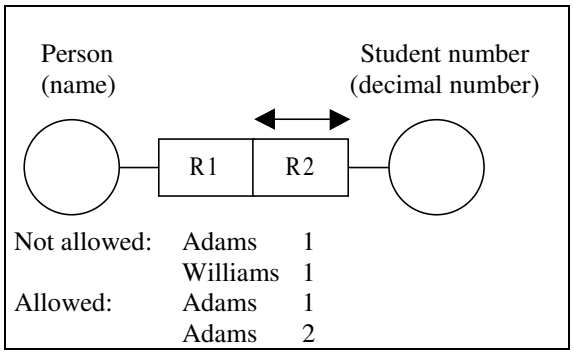


Figure A.2.2.2. Uniqueness constraint type 2

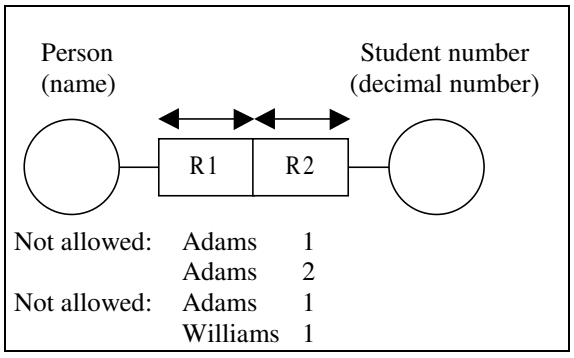


Figure A.2.2.3. Uniqueness constraint type 3

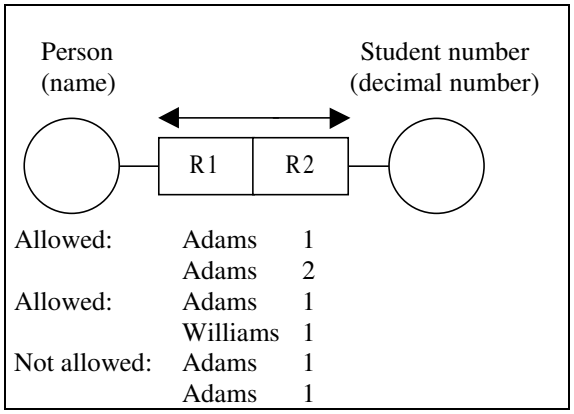


Figure A.2.2.4. Uniqueness constraint type 4

To identify uniqueness constraints in our example, we would ask a domain expert the following two questions:
1. Can a person have two different student numbers?
2. Can two persons share the same student number?

The domain expert would answer both questions with 'no' and we would place the third type of constraint.

Mandatory role constraints

The internal mandatory role constraint is used to denote that for a certain role all objects in the population of the object type have to occur at least once. This is shown in a diagram by a black dot on the line connecting object type and role. In our example we ask the domain expert the following two questions:
1. Are there persons that do not have a student number?
2. Are there student numbers that do not belong to a person?

The domain expert's answer to the first question is yes. Teachers are persons, but don't have student numbers. The second question is answered with no, no numbers are 'skipped'. From this answers we identify the mandatory role constraint in figure A.2.2.5.

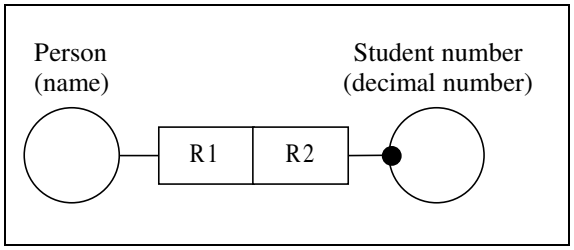


Figure A.2.2.5. mandatory role constraint

Subset constraints

The subset constraint indicates that the population of a role is a subset of the population of another role. This is denoted by an arrow with a dotted line pointing from the subset to the set. Of course both populations must be of the same type. In figure A.2.2.6. all persons with student numbers also have an email addres. Had it been the case that all persons who had email addresses also had student numbers, the arrow had been drawn in the opposite direction.

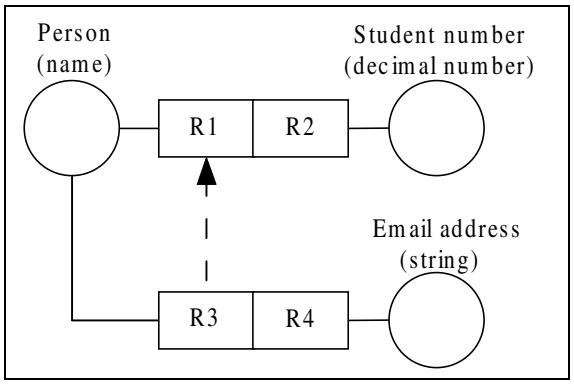


Figure A.2.2.6. subset constraint

Equality constraints

The equality constraint shows that the populations of two roles are equal. Again, the populations must be of the same type. In figure A.2.2.7. every person that has a student number, also has an email address, and the other way around.

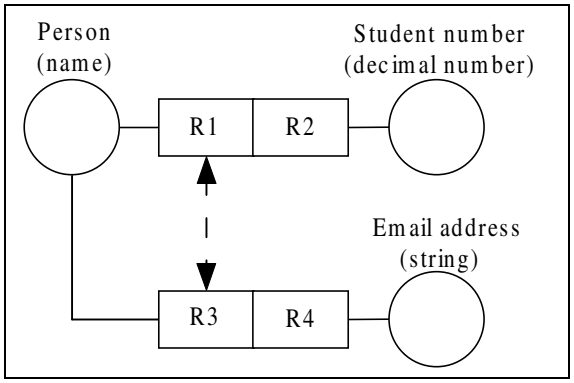


Figure A.2.2.7. equality constraint

Exclusion constraints

The exclusion constraint can be used when an instance of a certain type can only be part of one of the two roles. The rule is shown as an encircled ' x' . In figure A.2.2.8. no person that has a student number also has an email address (and no person that has an email address also has a student number).

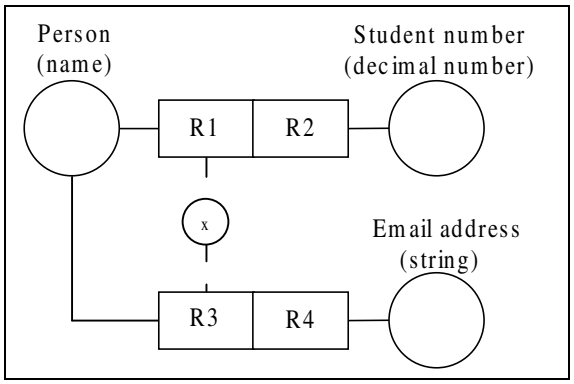


Figure A.2.2.8. exclusion constraint

When asked, the domain expert informed us the subset constraint in figure A.2.2.6. is correct, because all students (persons with student numbers) get an email address at the university.

There are some other constraints, but since the case modeled in paragraph 5.3. is very complex, only the types of constraints discussed in paragraph A.2.2. shall be modeled.

B Case: Eridani Insurance

Eridani Insurances was created in 1989 after a merger of 4 local Health Insurance companies. With about 1000 employees and about 1 million insurance takers, Eridani Insurance is a large organization.

B.1 Organizational structure

Eridani Insurance consists of a board of directors, some departments supporting the entire organization, and some functional departments. This situation is displayed in figure B.1.1.

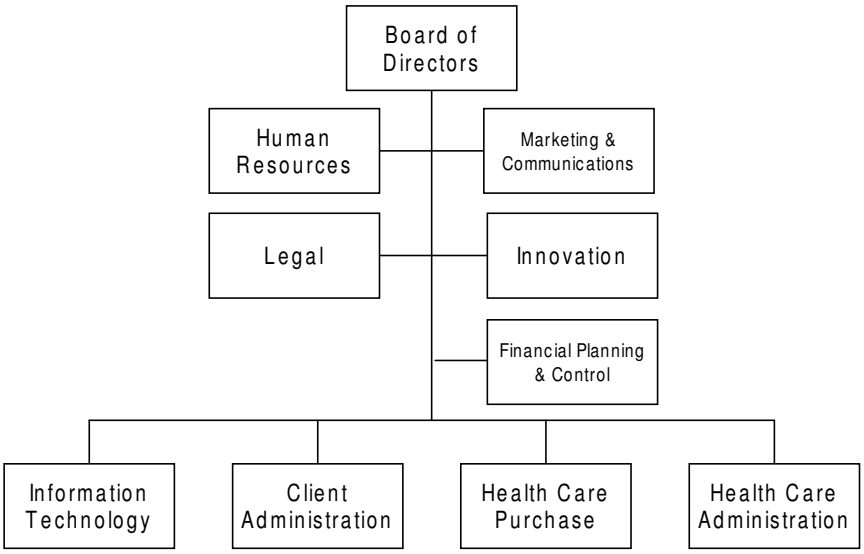


Figure B.1.1. Organogram Eridani Insurances

B.2 Health Care Administration

The Health Care Administrations (HCA) processes declarations of Health Care institutions (hospitals, etc.) and of individual insurance takers. (After a visit to the dentist, the insurance taker has to pay himself, and sends the bill to Eridani Insurance to receive (some part of) the bill back.) The Health Care Administration also checks whether the insurance company has to pay (part of) the bill, usually depending on the type of insurance.

B.3 Important agents

This paragraph describes some important agents and departments that are part of or important to the Health Care Administration.

Government: Department of Health

The department of Health gives out legislation by which every Health Insurance organization is bound and performs periodical checks of Health Insurance organizations.

Member board

In the member board, all clients of Eridani Insurance are represented. This way, the clients have some influence in the decisions of the organization.

Board of directors

The board of directors is responsible for the entire organization.

Head of the Health Care Administration (Head HCA)

The department head has the final responsibility for the activities of the Health Care Administration. He reports directly to the board of directors.

Proces design department
The proces design department creates the procedures according to which the declarations are processed. Based on new legislation and new products Eridani Insurance offers this department keeps the guidebook "Declaration processing" up to date.

System design department
The system design department works closely with the process design department. Every time there is an important change in the processing of declarations, they make sure the system that is used to process the declarations is up to date. The System design department is a special group inside Eridani's ICT department.

Health Care Administration Employee (HCA employee)
These are the people that actually work with the system. They process declarations and answer calls of insurance takers. The HCA employees report to the Head HCA.

B.4 The Information System
The Health Care Administration works with a system called 'Declarations 96' to process declarations sent in by health care institutions and individuals. Declarations 96 was created and is maintained by the system design department.

Declaration 96 operates as following:

First the HCA employee has to fill out the field 'client number'. Numbers for health care institutions start with a 0.

Then the name, address data and insurance data of the client or the institution are retrieved from the database. The employee checks whether this data is correct. If this is not the case, the employee has the option to change the client number.
If the number is correct, the employee is taken to a screen where he can fill out the following data for each bill that was declared on the billing form:

Bill information
- Bill filed by
- Bill number
- Bill date

Information about the person treated
- Initials
- Date of birth

Amount of money
- Currency

Caused by accident
- Yes or no

After the Employee has entered all declarations into the system, he presses ok and the system starts checking which of the bills should be paid for, and which part of each bill.

Usually there are no problems and the system fills out a letter template, which is sent to the insurance taker or the institution, to inform them of the decisions regarding the declaration. The financial department does a final check and the bill is paid for.

But sometimes, there is a small error somewhere in the declaration, and the system gives back an error, with the option to change or add data.

B.5 The situation
The innovations department of Eridani Insurance has come up with a new product: health insurance especially for people who work in the health care sector. Therefore the process design department has developed some new declaration processing procedures. This requires some large adaptations to the Declarations 96 system. The system development department wants to use this chance to create a new version of the Declarations program: Declarations 2002.

B.6 The problem

The system design department often gets complaints of the HCA employees about the errors the system produces when there' s a small error in one of the declarations. Often they do not understand the error, and cannot find any useful information in the help file. It sometimes takes up to 15 minutes to find out what caused the error. They inform the head HCA of this problem.

B.7 The solution

The head HCA discusses the problem with the process design department, who initially created the text for the errors in the system. Together they decide it would be a good idea to have a meeting with the entire HCA department and the system design department to discuss the meaning of the errors.

C Example Clauses

Example clauses for NIAM modelling of Conceptualization Viewpoint, inspired on GAK case (Hoppenbrouwers, forthcoming, chapter 6).

Main inspiration should be: a concept management information system.

The six "domains" within the model:

- 3 levels of communication
  - Primary communication
  - Utterance-level meta communication
  - Language-level meta communication
- 2 levels of organization
  - Operational level (individuals, concepts, use)
  - Organizational level (concrete processes, Discourse Environments, functions, terminologies)

We'll present the sentences in the six domains mentioned above.

1a: Primary communication, operational level

DE = "Claim Assessment"

concept (number) "C2" belongs to/includes terminology (name) "Claim Assessment" Discourse Environment (name) "Claim Assessment"

concept (number) "C2" takes/is taken by form (label) "cancellation code 2"

concept (number) "C2" has/underlies meaning (semantics) ["at entry the CA involved asserts that a certain benefit review is cancelled because it was requested illegally"]

artefact (name) "Claim assessment process description" describes form (label) "cancelleationcode 2" with meaning "(semantics) ["at entry the CA involved asserts that a certain benefit review is cancelled because it was requested illegally"]
Split into:
artefact (name) "Claim assessment process description" describes form (label) "cancelleationcode 2"
And:
artefact (name) "Claim assessment process description" describes meaning "(semantics) ["at entry the CA involved asserts that a certain benefit review is cancelled because it was requested illegally"]

agent (name) "CA1" can write/is writable form (label) "cancellation code 2" in artefact (name) "RESA/FASA 1201"

agent (name) "CA1" can reads/is readable by form (label) "cancellation code 2" in artefact (name) "RESA/FASA 1201"

agent (name) "CA1" has/belongs to function (name) "Claim Assessor"

agent (name) "CA1" knows/is known vocabulary (name) "CA1"

concept (number) "C2" belongs to/includes vocabulary (name) "CA1"

agent (name) "SPD1" knows/is known vocabulary (name) "SPD1"

concept (number) "C2" belongs to/includes vocabulary (name) "SPD1"

1b: Primary communication, organizational level

DE = ''Claim Assessment''

artefact (name) "RESA/FASA 1201" is medium for Discourse Environment (name) "Claim Assessment"

artefact (name) "RESA/FASA 1201" is authorized as medium for Discourse Environment "Claim Assessment" by function (name) "Directeur DZ"

terminology (name) "claim assessment" belongs to Discourse Environment (name) "Claim Assessment"

"Process description claim assessment" is authorized for terminology (name) "Claim Assessment" by function (name) "Directeur DZ"

function (name) "Claim Assessor" is athorized for Discourse Environment "Claim Assessment" by function (name) "Directeur DZ"

2a: Utterance-level MC, operational level

2b: Utterance-level MC, organizational level

3a: Language-level MC, operational level

DE = ''Claim Assessment Process Design''

concept (number) "C20" takes/is taken by form (label) "cancellation code 2"

concept (number) "C20" has/underlies meaning (semantics) ["at entry the CA involved asserts that a certain benefit review is cancelled because it was requested illegally"]

agent (name) "SPD1" can write/is writable form (label) "cancellation code 2" in artefact (name) "Claim Assessment Process Design"

agent (name) "SPD1" can write/is writable meaning (semantics) ["at entry the CA involved asserts that a certain benefit review is cancelled because it was requested illegally"]

agent (name) "SPD1" has/belongs to function (name) "System and Process Designer"

concept (number) "C20" belongs to/includes terminology (name) "Claim Assessment Process Design"

agent (name) "SPD1" knows/is known vocabulary (name) "SPD1"

concept (number) "C20" belongs to/includes vocabulary (name) "SPD1"

artefact (name) "Draft Process Description Claim Assessment" is medium for Discourse Environment (name) "Claim Assessment Process Design")

DE = Dissemination LISV Guidlines

terminology (name) "LISV Guidelines" belongs to/includes Discourse Environment (name) "Dissemination LISV Guidlines"

concept (number) "L25" belongs to/includes terminology (name) "LISV Guidelines"

agent (name) "SPD1" knows/is known vocabulary (name) "SPD1"

concept (number) "L25" belongs to/includes vocabulary (name) "SPD1"

DE = ''RESA/FASA system implementation''

concept (number) "C200" takes/is taken by form (label) "cancellation code 2"

concept (number) "C200" has/underlies meaning (semantics) ["Code that is (to be) part of RESA/FASA screen 1201 table such-and-such"]

concept (number) "C200" belongs to/includes vocabulary (name) "SPD1"

agent (name) "SPD1" can write/is writable form (label) "cancellation code 2" in artefact (name) "RESA/FASA Functional Description"

agent (name) "SPD1" can write/is writable meaning (semantics) ["Code that is (to be) part of RESA/FASA screen 1201 table such-and-such"] in artefact (name) "RESA/FASA Functional Description"

artefact (name) "RESA/FASA Functional Description" describes concept (number) "C200"

agent (name) "PROG1" can read/is readable form (label) "cancellation code 2" in artefact (name) "RESA/FASA Functional Description"

agent (name) "PROG1" can read/is readable meaning (semantics) ["Code that is (to be) part of RESA/FASA screen 1201 table such-and-such"] in artefact (name) "RESA/FASA Functional Description"

agent (name) "PROG1" has/belongs to function (name) "Programmer"

agent (name) "PROG1" knows/is known by vocabulary name "PROG1"

concept (number) "C200" belongs to/includes vocabulary (name) "PROG1"

agent (name) "PROG1" can write/is writable form (label) "cancellation code 2" in artefact (name) "RESA/FASA System Code"

3b: Language-level MC, organizational level

DE = ''Claim Assessment Process Design''

function (name) "Directeur DZ" authorizes artefact (name) "LISV Guidelines" for terminology (name) "Claim Assessment Process Design"

terminology (name) "Claim Assessment Process Design" belongs to/includes Discourse Environment (name) "Claim Assessment Process Design"

function (name) "Directeur DZ" authorizes function (name) "Process and System Designer" for Discourse Environment "Claim Assessment Process Design"

artefact (name) "Draft Process Description Claim Assessment" is medium for Discourse Environment (name) "Claim Assessment Process Design")

DE = RESA/FASA System Implementation

terminology (name) "RESA/FASA System Implementation" belongs to/includes Discourse Environment (name) "RESA/FASA System Implementation"

function (name) "Directeur DZ" authorizes function (name) "Programmer" for Discourse Environment "RESA/FASA System Implementation"

function (name) "Directeur DZ" authorizes function (name) "System and Process Designer" for Discourse Environment "RESA/FASA System Implementation"

artefact (name) "Draft RESA/FASA Functional Description" is medium for Discourse Environment (name) "Claim Assessment Process Design"

D Questions for uniqueness constraints

This appendix shows the questions that were used to identify the uniqueness constraints, and their answers by the domain expert. Some of the uniqueness constraints were already identified by the examples, so no questions are asked for those contraints.

1a. Can artefact a describe form f and g? - YES
1b. Can artefacts a and b describe form f? - YES

2a. Can concept c belong to terminologies t and u? - NO
2b. Can concepts c and d belong to terminology t? - YES

3a. Can terminology t belong to Discourse Environments d and e? - YES
3b. Can terminologies t and u belong to Discourse Environment d? - YES

4a. Can agent x write meaning m in artefacts a and b? - YES
4b. Can agent x write meanings m and n in artefact a? - YES
4c. Can agents x and y write meaning m in artefact a? - YES

5a. Can agent x read meaning m in artefacts a and b? - YES
5b. Can agent x read meanings m and n in artefact a? - YES
5c. Can agents x and y read meaning m in artefact a? - YES

6a. Can artefact a be authorized for terminology t by function f and g? - NO
6b. Can artefact a be authorized for terminologies t and u by function f? - YES
6c. Can artefacts a and b be authorized for terminology t by function f? - YES

7. Can artefacts a and b be a medium for Discourse Environment d? - YES

8a. Can artefact a be authorized as medium for Discourse Environment d by function f and g? - NO
8b. Can artefact a be authorized as medium for Discourse Environments d and e by function f? - YES
8c. Can artefacts a and b be authorized as medium for Discourse Environment d by function f? - YES

9a. Can function f authorize function f for DE d? - NO
9b. Can function f authorize function g for DE d and function g authorize function f for DE e? - NO
9c. If function f authorizes function g for DE d and function g authorizes function h for DE e, can function h authorize function f for DE z? - YES
9d. If function f authorizes function g for DE d and function g authorizes function h for DE e, can function f authorize function h for DE z? - YES

E Questions for mandatory role constraints

This appendix shows the questions that were used to identify the mandatory role constraints, and their answers by the domain expert.

Concept

Are there concepts that don' t have form? - NO
Are there concepts that aren' t part of any terminology? - YES
Are there concepts that don' t have meaning? - NO
Are there concepts that aren' t part of any vocabulary? - NO

Form

Are there forms that aren' t part of any concept? - NO
Are there forms aren' t described by any artefact? - YES
Are there forms that cannot be written in any artefact? (by an agent) - YES
Are there forms that cannot be read in any artefact? (by an agent) - YES

Agent

Are there agents who cannot write meaning in any artefact? - YES
Are there agents who cannot read meaning in any artefact? - YES
Are there agents who cannot write form in any artefact? - YES
Are there agents who cannot read form in any artefact? - YES
Are there agents who don' t know any vocabulary? - NO
Are there agents who don' t have a function? - NO

Vocabulary

Are there vocabularies that aren' t known by any agent? - NO
Are there vocabularies that don' t include concepts? - NO

Terminology

Are there terminologies that don' t include concepts? - NO
Are there terminologies that aren' t part of any Discourse Environment? - YES
Are there terminologies that don' t have any artifact authorized for? (by a function) - YES

Discourse Environment

Are there Discourse Environments that don' t have any terminology? - YES
Are there Discourse Environments that don' t have any artifacts as medium? - YES
Are there Discourse Environments that don' t have any artifacts authorized as medium? (by a function) - YES
Are there Discourse Environments that don' t have any functions authorized? (by a function) - YES

Function

Are there functions that don' t authorize any other function for a Discourse Environment? - YES
Are there functions that aren' t authorized for a Discourse Environment by any other function? - YES
Are there functions that don' t authorize any artefact for a terminology? - YES
Are there functions that don' t authorize any artefact as medium for a Discourse Environment? - YES

Meaning

Are there meanings that aren' t described in any artefact? - YES
Are there meanings that cannot be written (by an agent) in any artefact? - YES
Are there meanings that cannot be read (by an agent) in any artefact? - YES
Are there meanings that don' t underly a concept? - NO

Artefact

Are there artefacts where no agent can write any form? - YES
Are there artefacts where no agent can read any form? - NO
Are there artefacts where no agent can write any meaning? - YES
Are there artefacts where no agent can read any meaning? - YES
Are there artefacts that don' t describe any form? - NO
Are there artefacts that don' t describe any meaning? - YES
Are there artefacts that haven' t been authorized for a terminology by any function? - YES
Are there artefacts that haven' t been authorized as medium for a Discourse Environment by any function? - YES
Are there artefacts that aren' t a medium for any Discourse Environment? – YES

F Questions for equality, exclusion, and subset constraints

This appendix shows the questions that were used to identify the equality, exclusion, and subset constraints, and their answers by the domain expert. Questions with [sub] are meant to identify subset constraints and questions with [x] identify exclusion constraints. Equality constraints follow logical from mutual inclusion.

Form

[sub] Is every form f that can be read by an agent in an artifact also described by an artifact? - YES
[sub] Can every form f that is described by an artifact also be read by an agent in an artifact? - YES
[x] Is NO form f that can be read by an agent in an artifact also described by an artifact? - NO

[sub] Is every form f that can be written by an agent in an artifact also described by an artifact? - NO
[sub] Can every form f that is described by an artifact also be written by an agent in an artifact? - NO
[x] Is NO form f that can be written by an agent in an artifact also described by an artifact? - NO

[sub] Can every form f that can be read by an agent in an artifact also be written by an agent in an artifact? - NO
[sub] Can every form f that can be written by an agent in an artifact also be read by an agent in an artifact? - NO
[x] Is NO form f that can be read by an agent in an artifact also written by an agent in an artifact? - NO

Terminology

[sub] Does every terminology t that belongs to a Discourse Environment also have an artifact authorized for it by a function? - NO
[sub] Does every terminology t that has an artifact authorized for it by a function also belong to a Discourse Environment? - YES
[x] Does NO terminology t that belongs to a Discourse Environment also have an artifact authorized for it by a function? - NO

Discource Environment

[sub] Does every Discourse Environment d that includes a terminology also have an artifact that is medium for it? - NO
[sub] Does every Discourse Environment d that has an artifact as medium for it also include a terminology? - NO
[x] Does NO Discourse Environment d that includes a terminology also have an artifact that is medium for it? - NO

[sub] Does every Discourse Environment d that includes a terminology also have an artifact that is authorized as medium for it by a function? - NO
[sub] Does every Discourse Environment d that has an artifact authorized as medium for it by a function also include a terminology? - NO
[x] Does NO Discourse Environment d that includes a terminology also have an artifact that is authorized as medium for it by a function? – NO

[sub] Does every Discourse Environment d that includes a terminology also have a function authorized for it by a function? - NO
[sub] Does every Discourse Environment d that has a function authorized for it by a function also include a terminology? - NO
[x] Does NO Discourse Environment d that includes a terminology also have a function authorized for it by a function? - NO

[sub] Does every Discourse Environment d that has an artifact as medium also have an artifact authorized as medium for it by a function? - NO
[sub] Does every Discourse Environment d that has an artifact authorized as medium for it also have an artifact as medium? - YES
[x] Does NO Discourse Environment d that has an artifact as medium also have an artifact authorized as medium for it by a function? - NO

[sub] Does every Discourse Environment d that has an artifact as medium also have a function authorized for it by a function? - NO
[sub] Does every Discourse Environment d that has a function authorized for it by a function also have an artifact as medium? - NO
[x] Does NO Discourse Environment d that has an artifact as medium also have a function authorized for it by a function? - NO

[sub] Does every Discourse Environment d that has an artifact authorized as medium for it also have a function authorized for it by a function? - NO (such an artefact may be authorized by a function, but this function does not need to be auhorized by a function)
[sub] Does every Discourse Environment d that has a function authorized for it by a function also have an artifact authorized as medium for it? - NO
[x] Does NO Discourse Environment d that has an artifact authorized as medium for it also have a function authorized for it by a function? - NO

Function

[sub] Does every function f that authorizes an artifact for a terminology also authorize an artifact as medium for a DE? - NO
[sub] Does every function f that authorizes an artifact as a medium for a DE also authorize an artifact for a terminology? - NO
[x] Does NO function f that authorizes an artifact for a terminology also authorize an artifact as medium for a DE? - NO

[sub] Does every function f that authorizes an artifact for a terminology also belong to an agent? - NO (this is an interesting one!)
[sub] Does every function f that belongs to an agent also authorize an artifact for a terminology? - NO
[x] Does NO function f that authorizes an artifact for a terminology also belong to an agent? - NO

[sub] Is every function f that authorizes an artifact for a terminology also authorized for a DE by a function? - NO
[sub] Does every function f that is authorized for a DE by a function also authorize an artfact for a terminology? - NO
[x] Is NO function f that authorizes an artifact for a terminology also authorized for a DE by a function? - NO

[sub] Does every function f that authorizes an artifact for a terminology also authorize a function for a DE? - NO
[sub] Does every function f that authorizes a function for a DE also authorize an artifact for a terminology? - NO
[x] Does NO function f that authorizes an artifact for a terminology also authorize a function for a DE? - NO

[sub] Does every function f that authorizes an artifact as a medium for a DE also belong to an agent? - NO (again, interesting; of course it *should* belong to an agent, but it is possible it does not, which describes a *problem*.)
[sub] Does every function f that belongs to an agent also authorize an artifact as a medium for a DE? - NO
[x] Does NO function f that authorizes an artifact as a medium for a DE also belong to an agent? - NO

[sub] Is every function f that authorizes an artifact as a medium for a DE also authorized for a DE by a function? - NO
[sub] Does every function f that is authorized for a DE by a function also authorize an artifact as a medium for a DE? - NO
[x] Is NO function f that authorizes an artifact as a medium for a DE also authorized for a DE by a function? - NO

[sub] Does every function f that authorizes an artifact as a medium for a DE also authorize a function for a DE? - NO
[sub] Does every function f that authorizes a function for a DE also authorize an artifact as a medium for a DE? - NO
[x] Does NO function f that authorizes an artifact as a medium for a DE also authorize a function for a DE? - NO

[sub] Is every function f that belongs to an agent also authorized for a DE by a function? - NO
[sub] Does every function f that is authorized for a DE by a function also belong to an agent? - NO
[x] Is NO function f that belongs to an agent also authorized for a DE by a function? - NO

[sub] Does every function f that belongs to an agent also authorize a function for a DE? - NO
[sub] Does every function f that authorizes a function for a DE also belong to an agent? - NO
[x] Does NO function f that belongs to an agent also authorize a function for a DE? - NO

[sub] Does every function f that is authorized for a DE by a function also authorize a function for a DE? - NO
[sub] Is every function f that authorizes a function for a DE also authorized for a DE by a function? - NO
[x] Does NO function f that is authorized for a DE by a function also authorize a function for a DE? - NO (though this is a very extreme case...)

Meaning

[sub] Can every meaning m that is described by an artifact also be read in an artifact? - YES
[sub] Is every meaning m that can be read in an artifact also described by an artifact? - YES
[x] Can NO meaning m that is described by an artifact also be read in an artifact? - NO

[sub] Can every meaning m that is described by an artifact also be written in an artifact? - NO
[sub] Is every meaning m that can be written in an artifact also described by an artifact? - YES
[x] Can NO meaning m that is described by an artifact also be written in an artifact? - NO

[sub] Can every meaning m that can be read in an artifact also be written in an artifact? - NO
[sub] Can every meaning m that can be written in an artifact also be read in an artifact? - YES
[x] Can NO meaning m that can be read in an artifact also be written in an artifact? - NO

Agent

[sub] Can every agent a that can write form in an artifact also read form in an artifact? - YES
[sub] Can every agent a that can read form in an artifact also write form in an artifact? - NO
[x] Can NO agent a that can write form in an artifact also read form in an artifact? - NO

[sub] Can every agent a that can write form in an artifact also read meaning in an artifact? - NO
[sub] Can every agent a that can read meaning in an artifact also write form in an artifact? - NO
[x] Can NO agent a that can write form in an artifact also read meaning in an artifact? - NO

[sub] Can every agent a that can write form in an artifact also write meaning in an artifact? - NO
[sub] Can every agent a that can write meaning in an artifact also write form in an artifact? - NO
[x] Can NO agent a that can write form in an artifact also write meaning in an artifact? - NO

[sub] Can every agent a that can read form in an artifact also read meaning in an artifact? - NO
[sub] Can every agent a that can read meaning in an artifact also read form in an artifact? - YES
[x] Can NO agent a that can read form in an artifact also read meaning in an artifact? - NO

[sub] Can every agent a that can read form in an artifact also write meaning in an artifact? - NO
[sub] Can every agent a that can write meaning in an artifact also read form in an artifact? - YES
[x] Can NO agent a that can read form in an artifact also write meaning in an artifact? - NO

[sub] Can every agent a that can read meaning in an artifact also write meaning in an artifact? - NO
[sub] Can every agent a that can write meaning in an artifact also read meaning in an artifact? - YES
[x] Can NO agent a that can read meaning in an artifact also write meaning in an artifact? - NO

Artfifact

[sub] Is every artifact that describes form also authorized for a terminology by a function? - NO
[sub] Does every artifact that is authorized for a terminology by a function also describe form? - YES
[x] Is NO artifact that describes form also authorized for a terminology by a function? - NO

[sub] Is every artifact that describes form also medium for a Discourse Environment? - NO
[sub] Does every artifact that is a medium for a Discourse Environment also describe form? - NO
[x] Is NO artifact that describes form also medium for a Discourse Environment? - NO

[sub] Is every artifact that describes form also authorized as medium for a Discourse Environment? - NO
[sub] Does every artifact that is authorized as medium for a Discourse Environment also describe form? - NO
[x] Is NO artifact that describes form also authorized as medium for a Discourse Environment? - NO

[sub] Does every artifact that describes form also describe meaning? - NO
[sub] Does every artifact that describes meaning also describe form? - YES
[x] Does NO artifact that describes form also describe meaning? - NO

[sub] Can an agent read meaning in every artifact that also describes form? - NO
[sub] Does every artifact in which an agent can read meaning also describe form? - YES
[x] Can NO agent read meaning in every artifact that also describes form? - NO

[sub] Can an agent write meaning in every artifact that also describes form? - NO
[sub] Does every artifact in which an agent can write meaning also describe form? - YES
[x] Can NO agent write meaning in every artifact that also describes form? - NO

[sub] Can an agent read form in every artifact that also describes form? - YES
[sub] Does every artifact in which an agent can read form also describe form? - YES
[x] Can NO agent read form in every artifact that also describes form? - NO

[sub] Can an agent write form in every artifact that also describes form? - NO
[sub] Does every artifact in which an agent can write form also describe form? - YES
[x] Can NO agent write form in every artifact that also describes form? - NO

[sub] Is every artifact that is authorized for a terminology by a function also a medium for a Discourse Environment? - NO
[sub] Is every artifact that is a medium for a Discourse Environment also authorized for a terminology by a function? - NO
[x] Is NO artifact that is authorized for a terminology by a function also a medium for a Discourse Environment? - NO

[sub] Is every artifact that is authorized for a terminology by a function also authorized as medium for a Discourse Environment? - NO
[sub] Does every artifact that is authorized as medium for a Discourse Environment also authorized for a terminology by a function? - NO
[x] Is NO artifact that is authorized for a terminology by a function also authorized as medium for a Discourse Environment? - NO

[sub] Does every artifact that is authorized for a terminology by a function also describe meaning? - NO
[sub] Is every artifact that describes meaning also authorized for a terminology by a function? - NO
[x] Does NO artifact that is authorized for a terminology by a function also describe meaning? - NO

[sub] Can an agent read meaning in every artifact that is authorized for a terminology by a function? - NO
[sub] Is every artifact in which an agent can read meaning also authorized for a terminology by a function? - NO
[x] Can NO agent read meaning in every artifact that is authorized for a terminology by a function? - NO

[sub] Can an agent write meaning in every artifact that is authorized for a terminology by a function? - NO
[sub] Is every artifact in which an agent can write meaning also authorized for a terminology by a function? - NO
[x] Can NO agent write meaning in every artifact that is authorized for a terminology by a function? - NO

[sub] Can an agent read form in every artifact that is authorized for a terminology by a function? - YES
[sub] Is every artifact in which an agent can read form also authorized for a terminology by a function? - NO
[x] Can NO agent read form in every artifact that is authorized for a terminology by a function? - NO

[sub] Can an agent write form in every artifact that is authorized for a terminology by a function? - NO
[sub] Is every artifact in which an agent can write form also authorized for a terminology by a function? - NO
[x] Can NO agent write form in every artifact that is authorized for a terminology by a function? - NO

[sub] Does every artifact that is a medium for a Discourse Environment also authorized as medium for a Discourse Environment? - NO
[sub] Is every artifact that is authorized as medium for a Discourse Environment also a medium for a Discourse Environment? - YES
[x] Does NO artifact that is a medium for a Discourse Environment also authorized as medium for a Discourse Environment? - NO

[sub] Does every artifact that is a medium for a Discourse Environment also describe meaning? - NO
[sub] Is every artifact that describes meaning also a medium for a Discourse Environment? - NO
[x] Does NO artifact that is a medium for a Discourse Environment also describe meaning? - NO

[sub] Can an agent read meaning in every artifact that is a medium for a Discourse Environment? - NO
[sub] Is every artifact in which an agent can read meaning also a medium for a Discourse Environment? - NO
[x] Can NO agent read meaning in every artifact that is a medium for a Discourse Environment? - NO

[sub] Can an agent write meaning in every artifact that is a medium for a Discourse Environment? - NO
[sub] Is every artifact in which an agent can write meaning also a medium for a Discourse Environment? - NO
[x] Can NO agent write meaning in every artifact that is a medium for a Discourse Environment? - NO

[sub] Can an agent read form in every artifact that is a medium for a Discourse Environment? - YES
[sub] Is every artifact in which an agent can read form also a medium for a Discourse Environment? - NO
[x] Can NO agent read form in every artifact that is a medium for a Discourse Environment? - NO

[sub] Can an agent write form in every artifact that is a medium for a Discourse Environment? - NO
[sub] Is every artifact in which an agent can write form also a medium for a Discourse Environment? - NO
[x] Can NO agent write form in every artifact that is a medium for a Discourse Environment? - NO

[sub] Does every artifact that is authorized as a medium for a Discourse Environment also describe meaning? - NO
[sub] Is every artifact that describes meaning also authorized as a medium for a Discourse Environment? - NO
[x] Does NO artifact that is authorized as a medium for a Discourse Environment also describe meaning? - NO

[sub] Can an agent read meaning in every artifact that is authorized as a medium for a Discourse Environment? - NO
[sub] Is every artifact in which an agent can read meaning also authorized as a medium for a Discourse Environment? - NO
[x] Can NO agent read meaning in every artifact that is authorized as a medium for a Discourse Environment? - NO

[sub] Can an agent write meaning in every artifact that is authorized as a medium for a Discourse Environment? - NO
[sub] Is every artifact in which an agent can write meaning also authorized as a medium for a Discourse Environment? - NO
[x] Can NO agent write meaning in every artifact that is authorized as a medium for a Discourse Environment? - NO

[sub] Can an agent read form in every artifact that is authorized as a medium for a Discourse Environment? - YES
[sub] Is every artifact in which an agent can read form also authorized as a medium for a Discourse Environment? - NO
[x] Can NO agent read form in every artifact that is authorized as a medium for a Discourse Environment? - NO

[sub] Can an agent write form in every artifact that is authorized as a medium for a Discourse Environment? - NO
[sub] Is every artifact in which an agent can write form also authorized as a medium for a Discourse Environment? - NO
[x] Can NO agent write form in every artifact that is authorized as a medium for a Discourse Environment? - NO

[sub] Can an agent read meaning in every artefact that describes meaning? - YES
[sub] Does every artifact in which an agent can read meaning describe meaning? - YES
[x] Can NO agent read meaning in every artefact that describes meaning? - NO

[sub] Can an agent write meaning in every artefact that describes meaning? - NO
[sub] Does every artifact in which an agent can write meaning describe meaning? - YES
[x] Can NO agent write meaning in every artefact that describes meaning? - NO

[sub] Can an agent read form in every artefact that describes meaning? - YES
[sub] Does every artifact in which an agent can read form describe meaning? - NO
[x] Can NO agent read form in every artefact that describes meaning? - NO

[sub] Can an agent write form in every artefact that describes meaning? - NO
[sub] Does every artifact in which an agent can write form describe meaning? - NO
[x] Can NO agent write form in every artefact that describes meaning? - NO

[sub] Can an agent write meaning in every artefact in which an agent can read meaning? - NO
[sub] Can an agent read meaning in every artefact in which an agent can write meaning? - YES
[x] Can NO agent write meaning in every artefact in which an agent can read meaning? - NO

[sub] Can an agent read form in every artefact in which an agent can read meaning? - YES
[sub] Can an agent read meaning in every artefact in which an agent can read form? - NO
[x] Can NO agent read form in every artefact in which an agent can read meaning? - NO

[sub] Can an agent write form in every artefact in which an agent can read meaning? - NO
[sub] Can an agent read meaning in every artefact in which an agent can write form? - NO
[x] Can NO agent write form in every artefact in which an agent can read meaning? - NO

[sub] Can an agent read form in every artefact in which an agent can write meaning? - YES
[sub] Can an agent write meaning in every artefact in which an agent can read form? - NO
[x] Can NO agent read form in every artefact in which an agent can write meaning? - NO

[sub] Can an agent write form in every artefact in which an agent can write meaning? - NO
[sub] Can an agent write meaning in every artefact in which an agent can write form? - NO
[x] Can NO agent write form in every artefact in which an agent can write meaning? - NO

[sub] Can an agent write form in every artefact in which an agent can read form? - NO
[sub] Can an agent read form in every artefact in which an agent can write form? - YES
[x] Can NO agent write form in every artefact in which an agent can read form? - NO

Bibliography

- Berlo, D. (1960), *The Process of Communication*, Holt, Rinehart and Winston Inc., New York.
- Bruijn, L. de, Karnebeek, S. and Krabben, A. van der (2000), 'Sneller innoveren met bedrijfsarchitecturen", *White Paper, Landelijk Architectuur Congres 2000*.
- Dietz, J., Mallens, P., Goedvolk, H, Rijsenbrij, D. (1999), "A Conceptual Framework for the Continuous Alignment of Business and ICT", *Whitepaper Technical University Delft & Cap Gemini*.
- Goedvolk, J.G. (1999), "White Paper Integrated Architecture Framework", *Cap Gemini*, Presentation with lecture notes.
- Hoppenbrouwers, S. (2001), 'Communicative Aspects of Information Systems", *Syllabus for course DI05 of the 2001/2002 Informatiekunde Curriculum*, Nijmegen University.
- Hoppenbrouwers, S. (forthcoming), 'Freezing Language: Conceptualization Processes across ICT-supported Organizations", *PhD thesis*, Tilburg University.
- IEEE (1990), 'IEEE Standard Glossary of Software Engineering Terminology. (IEEE-Std-610-12-1990)", *IEEE Standards Collection, Software Engineering, IEEE*, New York.
- IEEE (2000), 'IEEE Recommended Practice for Architectural Description of Software Intensive Systems. (IEEE-Std-471-2000)", *IEEE Standards Collection, Software Engineering, IEEE*, New York.
- Johnson, G. and Scholes, K. (1999), *Exploring Corporate Strategy*, Fifth Edition, Prentice Hall.
- Kingston, M (1992, April 13), in *The Independent*, cited by:
- Czerniawska F. (1997), *Corporate speak. The use of language in business*, MacMillan Business, Houndmills.
- Kruchten, Philippe (November 1995), "Architectural Blueprints – the "4 + 1" View Model of Software Architecture", *IEEE Software*, vol. 12.
- Kruit, D. et. al. (2000), "Een stap verder in zorgautomatisering: architectuur", *White Paper, Landelijk Architectuur Congres 2000*.
- Maier, M., Emery, D.F. and Hilliard R.F. (July 23, 2001), 'The IEEE 1471-2000 Standard – Architecture Views and Viewpoints", *INCOSE 2001 Tutorial*.
- Mendelow, A. (1981), 'Environmental scanning - The impact of the stakeholder concept", *Proceedings of the Second International Conference on Information Systems*, Boston.
- Nijssen, G.M. (1993), *Universele Informatiekunde*, PNA Publishing, Beutenaken.
- Open Group, the (2001). 'The Open Group Architectural Framework (TOGAF) version 7". http://www.opengroup.org/togaf/
- Poel, K. van der (2000), 'Stap voor stap naar een definitie van Informatie Architectuur", *White Paper, Landelijk Architectuur Congres 2000*.
- Plato, (380 BC), *Meno*, translated by Benjamin Jowett. http://eserver.org/philosophy/plato/meno.txt
- Recklies, D. (2001), *Stakeholder Management*. http://www.themanager.org/Resources/Stakeholder%20Management.htm
- Redeker, G. (June 22, 1999), 'Communicatie in institutionele contexten", *Inaugurele rede uitgesproken bij de aanvaarding van het ambt van hoogleraar in de Communicatiekunde aan de Rijksuniversiteit Groningen*. http://odur.let.rug.nl/~redeker/oratie.htm
- Sachs, I. (2001), 'Extended Stakeholder Analysis Based on IEEE Standard 1471-2000", *ICSE 2001 Workshop Architecture & UML*.
- Underwood M. (2001), *Introductory models & basic concepts: semiotics.* http://www.cultsock.ndirect.co.uk/MUHome/cshtml/semiomean/semio1.html
- Vermeulen, E. (2000), 'Definitie", *Presentation, Landelijk Architectuurcongres 2000*.
- Vliet, R.I.M. van and Portier, R.J. (2000, June), 'Technische architectuur: verfijning of uitbreiding?", *Informatie*.
- Webster (2002), *Webster online dictionary.* http://www.webster.com