An Architecture Ecosystem for the Whole Systems Perspective[1]

Including System Dynamics

Based on Logic & Set Theory

and

Controlled Natural Languages

Working paper for the OMG Architecture Ecosystem sig

Version 2011-02-28-1530

[a]Sjir Nijssen and [b]Serge Valera

[a]PNA Group, Geerstraat 105, 6411 NP Heerlen, The Netherlands
sjir.nijssen@pna-group.nl

[b]European Space Agency, Keplerlaan 1, 2200 AG Noordwijk, The Netherlands
serge.valera@esa.int

[1] **Acknowledgements**: This work has been partially supported by the European Commission under the ONTORULE (ICT 231875) project. (Disclaimer: the authors are solely responsible for information communicated and the European Commission is not responsible for any views or results expressed).

ABSTRACT

During the course of the last 50 years several modeling standards have been adopted in the IT and business world. Nearly all of these standards were developed in a silo environment, even within the same standards organization. To a large extent this is understandable as no standards organization had in its charter to provide integrated standards, or let us say federated. Functional specialization was one of the characteristics of the industrial economy; integration is an essential characteristic of the knowledge economy.

In most practical solutions or running systems the various standards provide part of the functionality that was required. The result is that systems and IT applications in many organizations make use of a substantial number of artifacts that are part of the overall solution. It is not an exception that a large system makes use of UML, BPMN, SoaML, SysML, XSD, RDF, RDFS, OWL, ER, SQL, BPEL, WSDL etc. The current situation is that information has to be exchanged between the various subsystems using different models. These information transformations are so far handcrafted. This costs lots of money, the largest part of which could be saved if there were a standard for the federation of models and systems.

By introducing a powerful Modeling and Exchange Language even more money could be saved as the famous n-square minus n problem could be reduced to 2(n+1). For the above listed 12 models this means an additional saving of 80 %.

Hence there is a great incentive to produce a standard that makes federation of business, information, process, events and semantics models, sub models and sub systems a well running federation in practice.

What is needed to have this standard soon available? An Architecture Ecosystem that makes it possible to gradually federate existing models and gradually replace procedural code with more productive declarative models. Such an ecosystem has to provide at the same time additional functionalities like version management, model integration and model merge, traceability, multi natural language and multi name support.

A major requirement for such an Architecture Ecosystem is that the Modeling and Exchange language is as stable as possible, and hence fully independent of (changing) technology. Hence satisfying the ISO TR9007 requirements of being an interpretation of logic, and conceptual. Also the ISO TR9007 requirement that the Conceptual Schema prescribes all the states and all the transitions of the information base should be a requirement for the Architecture Ecosystem. Another requirement is that the Modeling and Exchange Language provides an optimal transformation between its model and the models expressed in the many current silo languages. Another requirement of the Architecture Ecosystem is that it is understandable to a large group of people, having the capacity to follow logical reasoning expressed in natural language. This could be called Controlled Natural Languages. A Controlled Natural Language is "ordinary" natural language to the reader, i.e. it appears as if it is everyday language. However, the author of a Controlled Natural Language text needs professional expertise in applying logic due to the fact that a text in a Controlled Natural Language is nothing less than logic, only in a much more understandable format. The more widespread adoption of such Controlled Natural Languages is about to break through.

What is the Conceptual Architecture of an Architecture Ecosystem? Is it possible to derive such a Conceptual Architecture, starting from a few clearly defined and easy to accept axioms and from this basis only use logical reasoning to derive the entire architecture? One answer to this challenge is given in this working paper.

1 Contents

2 List of Figures

3 Introduction

Knowledge in the world of business and IT is increasing in volume as well as complexity, is the opinion of many professionals. The world is asking for new approaches, including standards, to be able to use the "whole systems perspective" (OMG 2010c). Therefore, there is a growing need to maximize the use of explicit and declarative knowledge over procedural knowledge (Sowa 2010). Moreover, in such an increasingly complex world it is recommended to develop and use *conceptual foundations* which are solidly based on *logical reasoning* (Sowa 1976, Sowa 2010), set theory and are *understandable* (Nijssen 1978) to a large number of people.

In the IT and business communities there is an increasing awareness of the need to have solid and enduring analysis and modeling processes, often called methodologies or protocols, including validation by the subject matter experts, in their language, to develop both the business domain-specific models as well as the generic or meta-models. The CogNIAM, ORM, NIAM and FCO-IM communities have been working, some since the early seventies, on the development of processes or protocols for developing conceptual models, including extensive validation procedures involving the subject matter experts with communication in their preferred language and continue to refine these as more and more industrial strength practices become available (Nijssen 1980).

We will apply the CogNIAM processes for developing a conceptual model for the Architecture Ecosystem. The CogNIAM processes for developing a conceptual model are the same for the domain-specific and generic conceptual model or conceptual schema; said otherwise the CogNIAM processes are the same for the domain-specific and generic level. In this paper we will start with an example that everybody can understand: facts about Nobel Prizes. We take a representative set of concrete situations (in OMG MOF terms M0) and apply hereupon the CogNIAM conceptual model specification processes. We then take the result of this process, the domain specific Conceptual Model (OMG MOF M1) or Schema and apply for the second time the CogNIAM conceptual model specification processes. The result of that process is the generic model (OMG MOF M2) or conceptual schema. We then take the result of that process, the generic conceptual model and apply for the third time the CogNIAM conceptual model specification processes and the result is …

There is a growing awareness that more integration is needed. It becomes clearer everyday that the advantages of the silo approach that have substantially contributed to the productivity increase in the industrial age are rapidly disappearing in the knowledge economy.

A fundamental aspect of business systems is the so-called system dynamics. The business systems have a knowledge base that is continuously changing and systems need to provide solid support for such change.

More and more it is becoming clear that communication between professionals as well as communication between professionals and computers will make use of controlled natural languages. Common Logic Controlled English is a controlled natural language based on the ISO Common Logic standard ISO-24707 and will be used in this paper – together with other CNLs, either textual or diagrammatic – to illustrate this paper.

4 What is the Value of an Architecture Ecosystem?

Business and government processes become more and more dependent on IT. It has been estimated (Morgan 2002, Sowa 2010) that hundreds of billions of dollars could be saved on a yearly basis, if business modeling would be performed much better than in the traditional way. At the yearly conference of the Dutch Business Architects, Business Analysts and IT Managers in November 2010, Sjir Nijssen held a survey, with the question: how much would the total project costs be if *you yourself* (as you know your work best, we ask you only to estimate your own work) would work according to the principles outlined in a long presentation just given before that can be summarized as follows: before for any part or increment of a new system or law funds are made available to start design a technical solution followed by programming, first 100% of the *what* need to be formally defined in a declarative model, that is validated by the modeler in close collaboration with the business person or in general subject matter expert, in the language of the business person, using a representative set of examples. The answer was 50%. If we extrapolate this figure we come to the same order of magnitude as given above by Morgan and Sowa.
There are a few major reasons why low productivity business modeling keeps being used by most IT organizations. One of them is that the most popular business modeling standards and architectures contain fundamental flaws; another one is that being paid by the hour does not invite higher productivity. Hence there is a need for innovation in the area of architectures and languages to express business models and the protocols or methodologies to develop a complete, understandable, validated and formal business model.

Languages for business models have so far been developed in the traditional stovepipe or silo approach. The result is that we now have a situation that can be characterized by Figure 4-1 (below). In the current business situation many different models are expressed in different languages, most often in isolation or silos. In this particular case models are developed in 10 different modeling languages. Such a silo approach costs too much and it has been conclusively shown that these costs can to a very large extent be avoided. Hence there is a need for innovation.

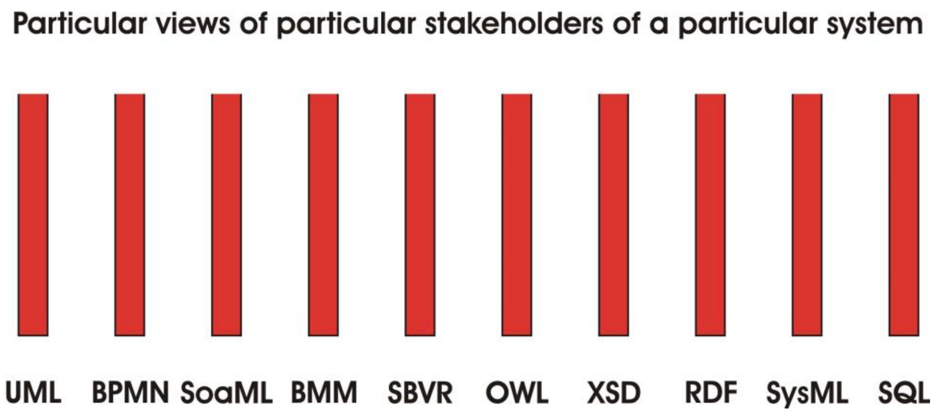


Figure 4-1 The traditional silo approach

Until now it was not possible to relate the various elements of various models expressed in various languages, although they are, from a business point of view, a part of the entire system. From a business point of view the goal is to have an integrating or federating modeling technology as can be represented in Figure 4-2; this is the basis for an Architecture Ecosystem Foundation. Here we see that there is a model base, a structural repository, in which elements of *all* models developed using such a traditional silo approach can be linked.

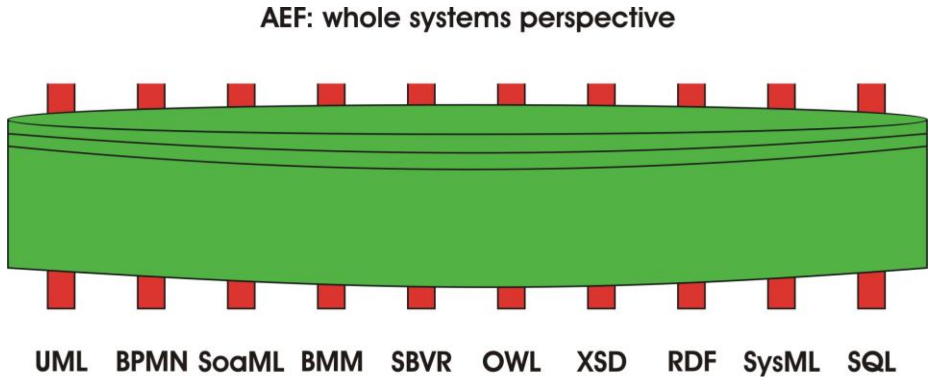


Figure 4-2 The whole systems' perspective: integrated models

Such an Architecture Ecosystem Foundation makes it possible to save yearly the enormous amount of money mentioned in the beginning of this chapter.

Integration of the silos enables much bigger targets: cross-validation of submodels, commonality in terminology between subcommunities, overlap detection, anchor points to link the enterprise model to external enterprise models, protect the investments in silo models. Furthermore:

1. Each methodology can benefit from the others, which means it becomes economically attractive to train more people better, especially across semantic/business communities.
2. The rigour of the underlying logical foundation allows for common tooling that can instantly verify a model for internal consistency.
3. The unification of silos leads to enhanced understandability which allows for easier validation by the actual knowledge owners, the business people, who in nearly all of the modeling languages shown above have little chance of understanding what they have to sign for.

The main architectural conceptual structure of such an Architecture Ecosystem Foundation will be *derived* in the subsequent chapters in an axiomatic and at the same time understandable way.

5 How to Develop an Architecture Ecosystem?

In this chapter we will describe that the selection of the axiom(s) to start with as well as of the procedure how to develop an Architecture Ecosystem from the selected axioms are crucial for the quality and durability of the Architecture Ecosystem and the systems that make use of it.
One of the most essential points in developing an architecture is the selection of the axiom or axioms to start from.

5.1 Axioms to start with

If one looks at the OMG MOF (OMG 2002) or ISO MOF (ISO 2005), or TOGAF (TOGAF 2010), or – although to a lesser extent – OWL (W3C 2009), one has difficulty detecting the axioms that have been used to start the development process of the associated architecture and language specification.

5.1.1 Controlled Natural Language

We have chosen an axiom that was the basis for a Controlled Natural Language business modeling approach that started in the seventies (Nijssen 1978). This axiom states that the communication between a human being and a computer can be considered to be an exchange of declarative facts or sentences. In 2011 language, one would say (declarative) ground facts, representing a proposition.
By way of example, we will look at facts associated with the Nobel Prize nomination and announcement process (cf. http://nobelprize.org/).
An example of such a set of ground facts, taken from the Nobel Prize website, is:

> The Nobel Peace Prize is awarded in Oslo.
> The Nobel Prize in Medicine is awarded in Stockholm.

> In Common Logic these two ground facts could be called atoms, in which each of the n terms of the predicate is a constant.
>
> The predicate in the first (and second) ground fact is:
> … is awarded in ….
> The two constants in the first ground fact are:
> “The Nobel Peace Prize” and “Oslo”.

In more traditional logic language (Halpin & Girle 1981) there are two propositions. Both share the same predicate. The first contains two individual constants: “The Nobel Peace Prize” and “Oslo”; the second contains the two individual constants “The Nobel Prize in Medicine” and “Stockholm”.

What is the advantage of selecting this axiom? Such examples of ground facts are abundantly available. People use them everyday in their business communication. The advantage of starting with the level of concrete facts as representing propositions (M0 in OMG MOF) is that one can start with a maximum of initial understanding between the parties involved in the communication.
Most often in the academic and standards world one starts at the other end of the knowledge spectrum, namely with a description at the generic (or meta-model) level (M2 in OMG MOF). The authors are convinced that starting from the solid rock bottom observable communication level consisting of everyday ground facts is a more productive way to derive an Architecture

Ecosystem. As a valuable additional feature, the derivation process from ground facts includes validation of the Architecture Ecosystem – it can be proven to serve a useful purpose for the knowledge owners (Architecture Ecosystem users).

5.1.2 Use logic and set theory as much as possible

Logic is fundamentally a declarative approach. This is in sharp contrast with a procedural approach. In a declarative approach one defines the goal; in a procedural approach one specifies meticulously every step in the procedure how to produce, starting from the input, via all intermediate stages, the goal. A declarative approach makes a number of procedural concrete approaches possible. A machine could select the most appropriate one, given the current situation. An everyday example of a declarative approach is a car navigation system, using the declarative knowledge of a map. A declarative approach is of course much more stable than a procedural approach.
To be of real use for modeling in practice logic needs to extended with set theory such as in Z or SQL.

5.2 Use a derivation protocol, including validation

An Architecture Ecosystem, a program and a model share at least one property: they all need to be validated or tested. For programs this has been widely accepted in business and is almost universally practiced. In IT considerable amounts of resources are dedicated to test a program. However, the authors have never seen any written evidence that e.g. the ISO and OMG MOF and TOGAF architectures have been tested. Such a test would involve selecting a representative number of actual architectures and validating whether they (a) do fit into the Architecture Ecosystem and (b) can be meaningfully mapped into each other.

6 The Solid Starting-Point

As the starting point in our quest to derive an Architecture Ecosystem, we will use the concrete level of the observable ground facts (M0). In this paper we will derive the necessary and sufficient number of levels of the Architecture Ecosystem and select an appropriate diagram to illustrate the resulting architecture.

We could look at an example of the Nobel Prize announcement, as represented in Figure 6-1.

The Nobel Prize in Physiology or Medicine 2002		
"for their discoveries concerning 'genetic regulation of organ development and programmed cell death'"		
Sydney Brenner	**H. Robert Horvitz**	**John E. Sulston**
1/3 of the prize	1/3 of the prize	1/3 of the prize

Figure 6-1 Level I (OMG M0, OWL A-Box, relational at data base level) example – Nobel Prize in Physiology or Medicine 2002

The business model for this kind of announcements would result 4 predicates in Interpreted Common Logic or several UML classes and associations depending on the scope to be selected like do we want to communicate some more facts about laureates such as their birth date, e.g. to be able to compute at which age the Nobel prize was awarded to the laureate. An abundance of such concrete examples of the Nobel Prize announcement process is readily available and it provides for repeatability of the procedure. We will use concrete examples to derive in the next chapters the conceptual model of the Nobel Prize to be used in the derivation and validation of the conceptual structure of the Architecture Ecosystem . However to derive the model for the main part of the Architecture Ecosystem we can use a simpler example of the Nobel Prize case as introduced in section 5.1.

7 From the World of Concrete Examples of Facts to the Domain-Specific Model or Conceptual Schema: Part I, Fact Types

The protocol to develop a complete conceptual model, sometimes called conceptual schema or a complete ontology, starts with the assignment of pieces of the use case text to a knowledge category of the Knowledge Triangle; the knowledge triangle will be discussed later in this paper. This assignment process ultimately results in a representative set of concrete permitted (to specify the scope and concept definitions) and not permitted examples (to specify the validation rules) at the level of the concrete communication with facts, in the OWL community called the A-box, in the OMG community called level M0 and in the relational community called the data base.

In the case of the Nobel Prize there is a representative set of *permitted* examples available on its website. Hence we can start with such a permitted example. This will lead to the *foundation* of the conceptual model, complete ontology or conceptual schema.
In another working paper we will illustrate how the CogNIAM protocol can be used to develop such concrete examples – both permitted and not permitted – from a general use case text, using a protocol.

We start with the example presented in a previous chapter:

The Nobel Laureates are announced at the beginning of October each year. A couple of months later, on 10 December, the anniversary of Alfred Nobel's death, they receive their prizes from the Swedish King – a Nobel diploma, a medal, and 10 million Swedish crowns per prize. All Nobel Prizes are awarded in Stockholm, Sweden, except for the Nobel Peace Prize, which is awarded in Oslo, Norway. (When Alfred Nobel was alive, Norway and Sweden were united under one monarch, until 1905 when Norway became an independent kingdom with its own king.)

"The Nobel Prizes - About". Nobelprize.org. 24 Feb 2011 http://nobelprize.org/educational/nobelprize_info/

Figure 7-1 Level I (OMG M0, OWL A-Box, relational at data base level) example – Nobel Prize Award Ceremonies

According to the CogNIAM conceptual analysis processes for developing a business model, an ontology (in the broadest sense), or a conceptual schema, available since a long time (Nijssen 1978, 1981) and more recently extensively described in (Halpin & Morgan 2008 and ONTORULE deliverable D1.3), the conceptual analyst asks a domain or subject matter expert to verbalize the relevant contents of the concrete example as if he were communicating to a colleague over the telephone. In this case we may get the following result:

The Nobel Peace Prize is awarded in Oslo.
The Nobel Prize in Medicine is awarded in Stockholm.

Please note that the second fact, about The Nobel Prize in Medicine, is actually derived from the statement "All Nobel Prizes are awarded in Stockholm, Sweden, except for the Nobel Peace Prize, which is awarded in Oslo, Norway".

In Common Logic these two are called atoms in which all variables are filled with constants. There is one predicate and two constants in each of these ground facts. Please note that these ground facts do not contain any elements that act as grammar for other sentences or facts.

In traditional logic, each of these two ground facts is called an atomic formula, in which each of the terms of the predicate is a (domain specific) individual constant. The predicate is: <…> is awarded in <…>; the two constants of the first atomic formula are: "The Nobel Peace Prize" and "Oslo".

The process of verbalization can be performed with the option to use an additional quality control procedure in which the corresponding parts in the original and in the verbalization are given the same color, with the aim to more easily detect missing parts (see Figure 7-2). The coloring shows that all parts of the representation of the first fact are included in the verbalization.

The Nobel Laureates are announced at the beginning of October each year. A couple of months later, on 10 December, the anniversary of Alfred Nobel's death, they receive their prizes from the Swedish King – a Nobel diploma, a medal, and 10 million Swedish crowns per prize. All Nobel Prizes are awarded in Stockholm, Sweden, except for the Nobel Peace Prize, which is awarded in Oslo, Norway. (When Alfred Nobel was alive, Norway and Sweden were united under one monarch, until 1905 when Norway became an independent kingdom with its own king.)

"The Nobel Prizes - About". Nobelprize.org. 24 Feb 2011 http://nobelprize.org/educational/nobelprize_info/

The Nobel Peace Prize is awarded in Oslo.

Figure 7-2 Correspondence between two representation forms

These two facts are part of the fact base in CogNIAM, part of the A-Box in OWL speak, are part of a database in relational terminology and are called ground facts in SBVR. In OMG MOF this is called M0.

The CogNIAM analysis protocol prescribes that the next knowledge elicitation sub process after verbalization is finding out which are the variable and non-variable parts in the verbalized sentences or facts. The business analyst makes a proposal to the business expert to check that he has understood this aspect. The proposal is given in Table 7.1.

Table 7.1 Result of identifying variable and non-variable parts at the level of ground facts

The Nobel Peace Prize	is awarded in		Oslo
The Nobel Prize in Medicine	”	”	” Stockholm

The business expert agrees that there are no other variable elements in the two given sentences and now the conclusion is that there are two placeholders or variables filled in the predicate with a constant.

The CogNIAM analysis protocol prescribes that the next knowledge elicitation sub process, after identifying the variable parts in the sentence (each of which is filled with a constant!) is to analyze whether each variable part needs some additional qualification or less qualification, in the specific speech community. It could be considered to add the qualification “the city of” in front of the variable describing the city (e.g. “The Nobel Peace Prize is awarded in the city of Oslo”). However, the business expert rejects that addition for his speech community.

According to the CogNIAM conceptual model specification processes, the predicate part is taken as the basis for the fact type form or reading (OMG 2008a) (also called sentential form in SBVR (OMG 2008a), or sentence pattern (Nijssen 2007)). This will be used at the *next level* conceptual model or schema, in this case the domain-specific model or domain specific conceptual schema or domain specific part of the OWL T-Box, and the variable parts in the ground facts are replaced by a variable.

Strong link between the level M0 and M1

Here we see a strong link between the ground fact level and the domain specific model or conceptual schema level, namely they *share* the predicate part of a fact (*instance*) and the corresponding fact *type*. This shared part between the 2 levels is extensively used in the CogNIAM validation procedure. It is this missing link between the M0 and M1 level that makes validation by some widely used standard graphical representation almost impossible. Hence for the many fact instances (at the ground fact level) sharing the same predicate part and sharing variable parts which perform the same function, we introduce a fact type form or sentence pattern at the domain specific conceptual model or schema level. If we apply this procedure in this concrete case we obtain:

1: <Category> is awarded in <City>.

> In logic, this could be called an atomic formula, in which each of the terms of the predicate is a (domain specific) variable. The predicate is: <…> is awarded in <…>; the two variables are: “Category” and “City”.

The reason to give the fact type form the identifier 1 is that in many business applications there are often thousands of these fact type forms and in such a situation a short ID is very convenient.

As a diagrammatic representation is in many situations used to increase the productivity of the communication, we will now use a diagram to represent the fact type and all its associated fact type forms as given below.

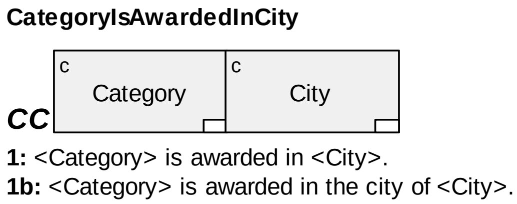


Figure 7-3 Intermediate fact type diagram

We may conclude that there is an easy to follow conceptual model specification process to arrive at the core of the domain-specific model or conceptual schema in which the subject matter expert and professional modeler work together according to a well-specified protocol. The core is taken to include primarily the data structure or in this case more appropriately called the communication structure. We take the CogNIAM knowledge triangle (see below in figure 7.4) to describe that we have traveled from the concrete example (in MOF called the information level or M0), identified in the knowledge triangle as Level I, to a part of the domain-specific model or conceptual schema (in MOF called the M1 model level). To be more precise to the fact type layer of level II.

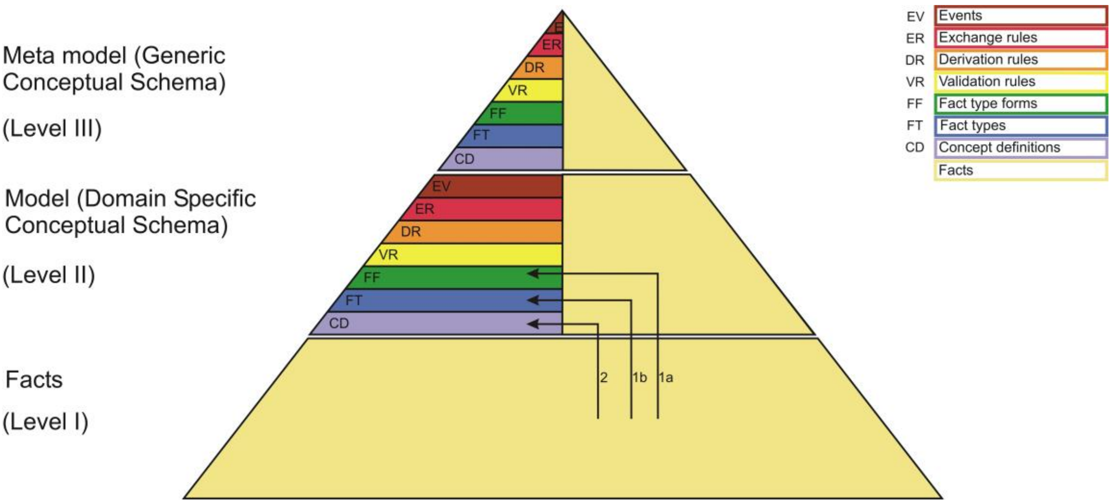


Figure 7-4 The process from the world of concrete examples to the domain-specific model

In general this part (level II in the knowledge triangle, MOF M1) also contains the concept descriptions for Category and City. In this case City is assumed to be known. The concept description for Category is given below:

> ***Category***
> The {Nobel Prize} is awarded each year in 6 [Categories]:
> Physiology or Medicine;
> Physics;
> Chemistry;
> Literature;
> Peace;
> Economic Sciences (Sveriges Riksbank Prize in Economic Sciences).

8 From the World of Concrete Examples to the Domain-Specific Model or Conceptual Schema: Part II, Integrity Rules

In the CogNIAM protocol it is strongly recommended to specify the primary uniqueness constraint for every fact type, even if there is hardly any time available to do so. What is the value of a function if it is not known which are the independent and which the dependent variables! And if there is a little more time just add the most relevant integrity rules right away. If we do this we get the contents of Figure 8-1 which will be discussed in the coming text.

The CogNIAM protocol provides a set of rule-specific protocols to derive the integrity rules. The most important integrity rule is the uniqueness constraint. It specifies that in every population of one or more variables of the fact type (hence at the ground facts level), no duplicates are allowed. The protocol to derive all the uniqueness constraints is to start with a reference fact. The diagrammatic representation of this crucial validation rule is first and for all intuitive as will be shown at the next page. To schedule all the questions that need to be asked by the business analyst to the subject matter expert or business expert, more facts are added below the reference fact, such that the value in the variable starting from the right is different with respect to the value of the variable in the reference fact.

It is recommended to make use of diagrammatic examples, whenever possible. The question to be asked for the combination of fact 1 and 2 of Table 8.3 is illustrated in Table 8.1**Fout! Verwijzingsbron niet gevonden.**.

Table 8.1 Illegal combination of statements concerning the Nobel Prize Award Ceremonies

The Nobel Prize in Medicine	is awarded in	Stockholm
The Nobel Prize in Medicine	is awarded in	Oslo

The subject matter expert recognizes very quickly that the content of Table 8.1 is not permitted.

The question to be asked for the combination of fact 1 and 3 of Table 8.3 is illustrated in Table 8.2.

Table 8.2 An allowed combination of statements concerning the Nobel Prize Award Ceremonies

The Nobel Prize in Medicine	is awarded in	Stockholm
The Nobel Prize in Physics	is awarded in	Stockholm

The answer by the subject matter expert is quickly a yes. This is the level of communication – the world of the concrete examples - that the subject matter expert uses almost all day.

Table 8.3 The register of answers given by the subject matter expert

Fact ID	Variable ‘Category	Variable ‘City	Subject Matter Expert’s answer regarding the simultaneous existence with fact 1	Conclusion by the Business Analyst
1 (reference fact)	The Nobel Prize in Medicine	Stockholm		
2	The Nobel Prize in Medicine	***Oslo***	No	Add a uniqueness constraint above the variable that has duplicates that are not permitted. Hence a uniqueness above the variable Category.
3	***The Nobel Prize in Physics***	Stockholm	Yes	No uniqueness constraint above the variable that has duplicates that are according to the subject matter expert, permitted. Hence no uniqueness above the variable City.

If we add these conclusions of the application of the uniqueness protocol to the fact type diagram of Figure 7-3 we obtain Figure 8-1.

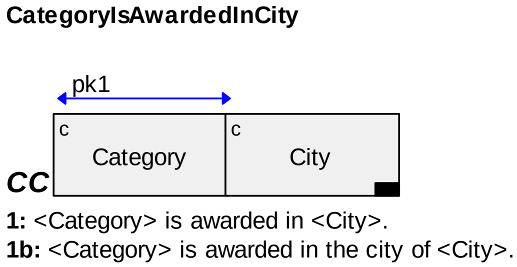


Figure 8-1 Complete fact type diagram at the domain-specific conceptual schema level

The arrow, labeled pk1, above the variable Category, specifies that in each population (= set of fact instances) of the facts (the level of the ground facts) conforming to this fact *type*, there may be no duplicates.

This is expressed in a Controlled Natural Language as:

r1: <Category> is awarded in **at most one** <City>.

Or in the context of the fact type CategoryIsAwardedInCity the following holds:

The instance of variable Category determines the instance of variable City.

Or

The variable City is a function of the variable Category.

Or

Category is the independent variable and City is the dependent variable.

The little black rectangle in the lower right corner of variable City indicates that for every fact in this fact population the city must be known. The value of the Category is already known for every fact as the primary uniqueness constraint implies the following two features: no duplicates beneath me and no empty cell in any fact beneath me.

If we would only use this diagrammatic convention for *elementary* fact types, then the little black rectangle is superfluous. However, in the CogNIAM graphical notation of predicates the same diagrammatic formalism is used for gracefully combined (n-ary) fact types.

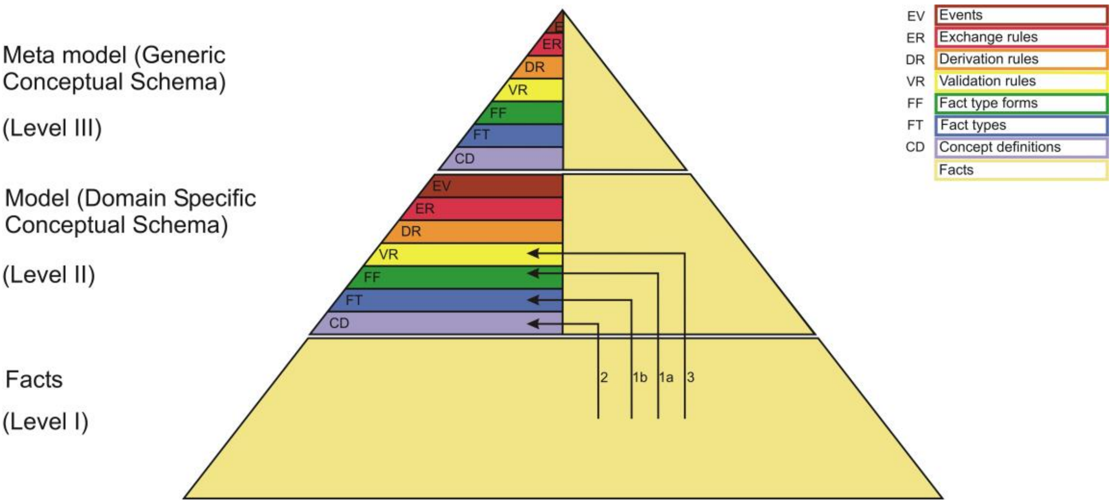


Figure 8-2 <TEXT>

We have now completed part 1 of our journey (from ground fact level (M0) to the level of the domain specific model (M1) or conceptual schema), and will start the second leg of this journey (from the domain specific conceptual model (M1) or schema level to the generic model (m2) or conceptual schema level).

9 From Domain-Specific Model to the Generic Model or Conceptual Schema: Part I, Fact Types

In the interpreted logic based CogNIAM approach an information bearing construct at any level (M0, M1, M2) is considered a fact (ISO 1978, Nijssen 2007). Hence we can take the result of the previous specification process (Figure 8-1) and take that as input for the next model specification process. If we do that we have the option to start with the verbalization of the fact types, variables, the primary uniqueness constraint, the mandatory constraints, or the fact type readings, all at the domain specific model level (OMG MOF M1).

This is probably the best time in this procedure to introduce the fact that there is room for two useful kinds of fact type forms. One fact type form has the aim to be used as basis for the specification of a rule. This is the fact type form that is primarily used in SBVR. For instance, the fact type form:

1: <Category> is awarded in <City>.

can be used to formulate the rule:

r1: <Category> is awarded in **at most one** <City>.

If we would have used the qualified fact type form:

1b: <Category> is awarded in the city of <City>.

and we would try to use this to formulate the rule, we would get the following:

r2: <Category> is awarded in the city of **at most one** <City>.

This is not what is considered a controlled, understandable natural language sentence. Hence, *unqualified* fact type forms are useful for the formulation of rules, *qualified* fact type forms are useful for enhancing the understandability of the facts that are used in the concrete examples. In the remainder of this chapter, the need for the qualified version of fact type forms will become clear at the meta-model level (M2).

Also note that one fact type can serve as the basis for any number of fact type readings or sentence patterns. This kind of diagram may contain furthermore a population as well as the verbalized facts using the fact type readings.

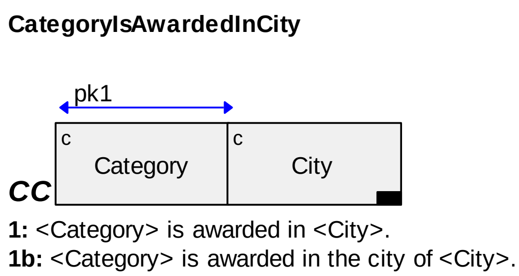


Figure 9-1 Domain specific conceptual schema

The result of the process verbalization, applied to the variables of the fact type in Figure 8-1 (for convenience here also presented in
Figure 9-1 above), is shown in Table 9.1.

Table 9.1 Result of verbalization determining of variable parts at the metadata (domain-specific conceptual schema) level

Variable	Category	of	fact	type	CategoryIsAwardedInCity	takes	position	1.
”	City	”	”	”	CategoryIsAwardedInCity	”	”	2.

In a Controlled Natural Language this can be expressed as:

> The first position of the fact type CategoryIsAwardedInCity is the variable Category.
> The second position of fact type CategoryIsAwardedInCity is the variable City.

One of the excellent quality control processes in the CogNIAM protocol – recommended for use in the more challenging modeling situations and hence also in meta modeling – is to apply a corresponding color to the part from the diagrammatic representation that is verbalized and the piece in the verbalization that represents that part. This procedure will help to spot parts that might have been overlooked. In Figure 9-2 is the color represented in the domain specific model and the verbalization.

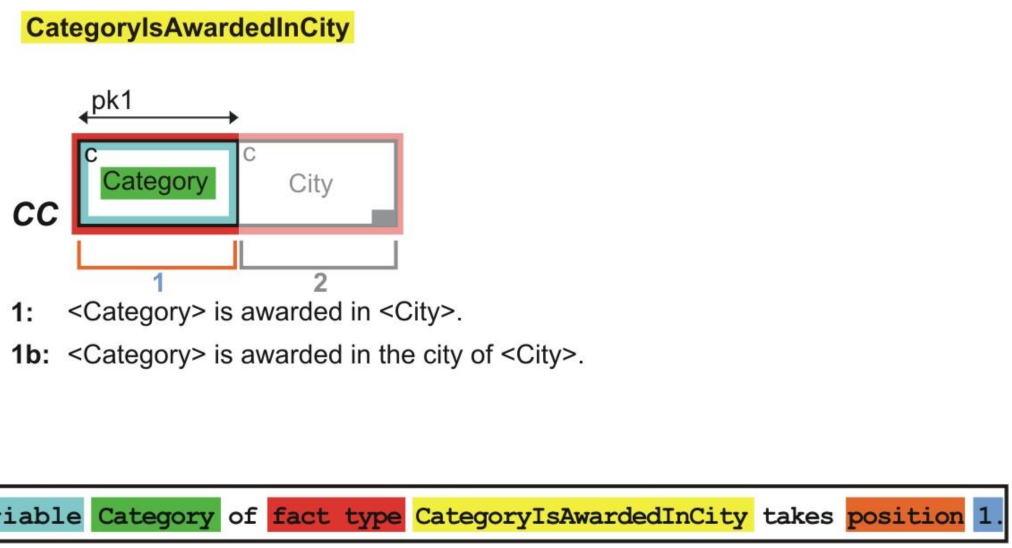


Figure 9-2 Correspondence between diagrammatic representation and verbalization

Verbalization including the corresponding color is presented in Figure 9-2.
The symbol for a variable is a rectangle (light blue), the name of that variable is in green, the symbol for a predicate or fact type is a rectangle (in red) containing all variables and the name of the predicate or fact type (CategoryIsAwardedInCity) is placed above the left end of the fact type symbol (in yellow).
A diagrammatic convention is often used to enhance the speed of communication.
Sometimes certain aspects have to be known. In this case the concept “position”. Every predicate or fact type has as many positions as it has variables. The concept of position makes it possible to guarantee that the same sequence is presented. The convention in the CogNIAM notation is that the first position is the left-most position.

The coloring procedure makes it easy to check that no part of the original is overlooked as the reader can easily verify.

Please note that at this level unqualified fact type forms are counterproductive for productive communication: the qualification now consists of meaningful words, not just recognizable proper nouns (names) or numbers. The unqualified form results in the following fact of which we assume that it is hardly understandable:

Category of CategoryIsAwardedInCity takes 1.

In fact, it will nearly always help to explicitly use qualification at the meta or M2 level (variable, fact type and position):

Variable “Category” of fact type “CategoryIsAwardedInCity” takes position “1”.

Hence at this level (the verbalization of MOF level M1) both fact type patterns, one to be used as a basis for rules and the other one to be used for producing sentences, are needed.

If we now apply the generalization process (as shown previously in section 7) we will produce a fact type reading, thereby arriving at the generic level, or the meta-model (M2) level in OMG MOF terminology:

1000: Variable <VariableName> of fact type <FactType> takes position <Position>.

The fact type diagram generated from the fact type reading is represented in Figure 9-3. This is level M2 in MOF.

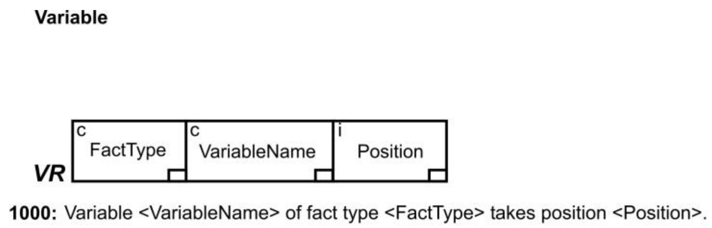


Figure 9-3 Intermediate fact type diagram

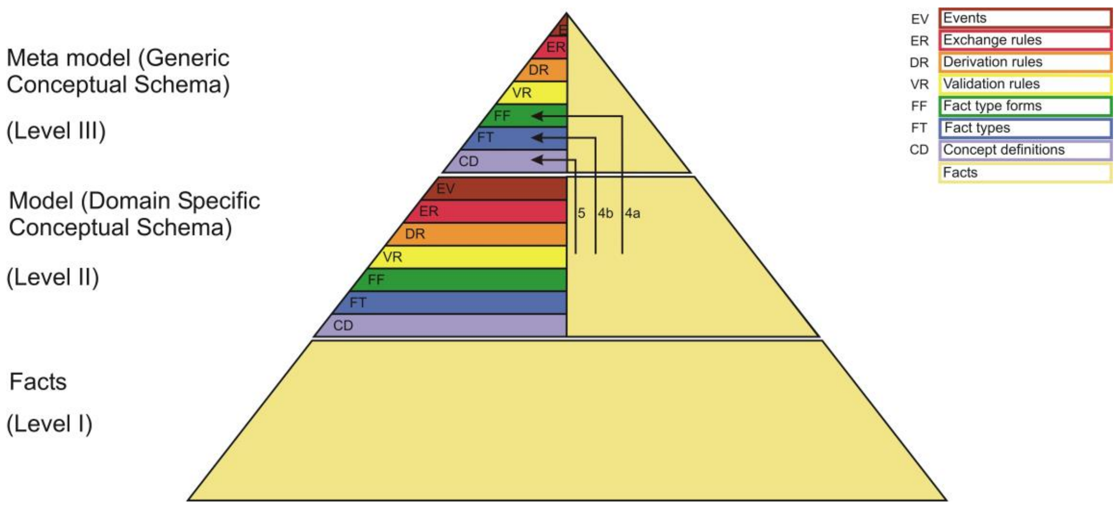


Figure 9-4 <TEXT>

10 From Domain-Specific Model to the Generic Model: Part II, Integrity Rules

We now utilize the same CogNIAM protocol to derive the uniqueness constraints, as demonstrated in section 8. In this case the questions are not posed to the subject matter expert but are answered by a business analyst (or knowledge engineer). Note that this business analyst now takes on the role of a subject matter expert in his own area of competence, so there is no fundamental difference in the methodology when applied to an input at the level of the ground facts or an input at the level of the domain specific conceptual schema. The intermediate steps and results of the uniqueness elicitation process are shown in Table 10.1.

It is recommended to make use of well defined diagrammatic examples. The question to be asked for the combination of fact 1 and 2 of Table 10.1 is illustrated in Figure 10-1 in a diagrammatic way.

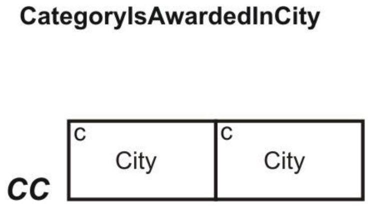


Figure 10-1 Diagrammatic example showing an illegal state of the model (M1)

The subject matter expert recognizes in a split-second that the content of Figure 10-1 is not permitted. Within a fact type each position has a unique name; if the name of the corresponding object type is used and it has a function more than once, then a prefix or postfix is added.
The question to be asked for the combination of fact 1 and 3 of Table 10.1 is illustrated in Figure 10-2.

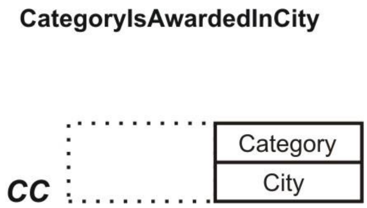


Figure 10-2 Diagrammatic example showing an illegal state of the model (M1)

Again, in a split-second the subject matter expert says “no”. In Table 10.1 this is all recorded in text. In each position in a predicate only one variable can function.

Table 10.1 The register of answers given by the business analyst (knowledge engineer)

Fact ID	Variable 'FactType'	Variable 'Variable-Name'	Variable 'Position'	Business Analyst's answer regarding the simultaneous existence with fact 1	Conclusion by the Business Analyst
1 (reference fact)	CategoryIsAwardedInCity	City	2		
2	CategoryIsAwardedInCity	City	***1***	No	Add a uniqueness constraint above the variables that have duplicates that are not permitted. Hence a uniqueness above the combination of the variables FactType and VariableName.
3	CategoryIsAwardedInCity	***Category***	2	No	Add a uniqueness constraint above the variables that have duplicates that are not permitted. Hence a uniqueness above the combination of the variables FactType and Position.
4	***XYZ***	City	2	Yes	No uniqueness constraint above the variable that has duplicates that are permitted according to the business analyst. Hence no uniqueness above the combination of the variables VariableName and Position.

If we add the conclusions of this uniqueness elicitation process to the fact type of Figure 9-3, we obtain the fact type plus uniqueness representation as shown in Figure 10-3. Primary uniqueness constraint pk1001 specifies that no duplicates are permitted in the fact population under the combination of the variables FactType and VariableName and alternate key ak1002 specifies that no duplicates are permitted under the combination of the variables FactType and Position. The black rectangle in the right lower part of the variable Position specifies that every fact in the fact population must have a value for the variable Position (note that the CogNIAM convention specifies that this symbol signifying that an element is mandatory is not necessary for every variable that is part of the primary uniqueness constraint, hence in this case for the variables "FactType" and "VariableName").

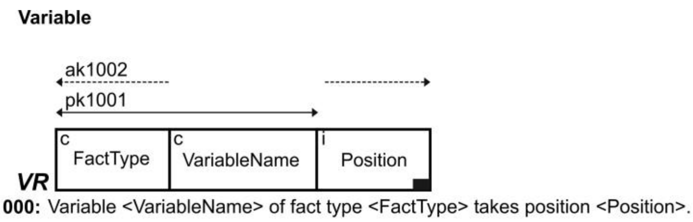


Figure 10-3 Fact type diagram at the generic conceptual schema level

In CLCE pk1001 is expressed as:

> The combination of the variable FactType and the variable VariableName determines the variable Position.

Or

> No duplicates are allowed for the combination of variable FactType and variable VariableName.

In CLCE ak1002 is expressed as:

> The combination of variable FactType and variable Position determines variable VariableName.

We have now finished two legs of our journey, starting at the solid level of ground facts, to methodically discover just by logical reasoning how many layers there should be in the conceptual structure of the Architecture Ecosystem (see Figure **10-4**).

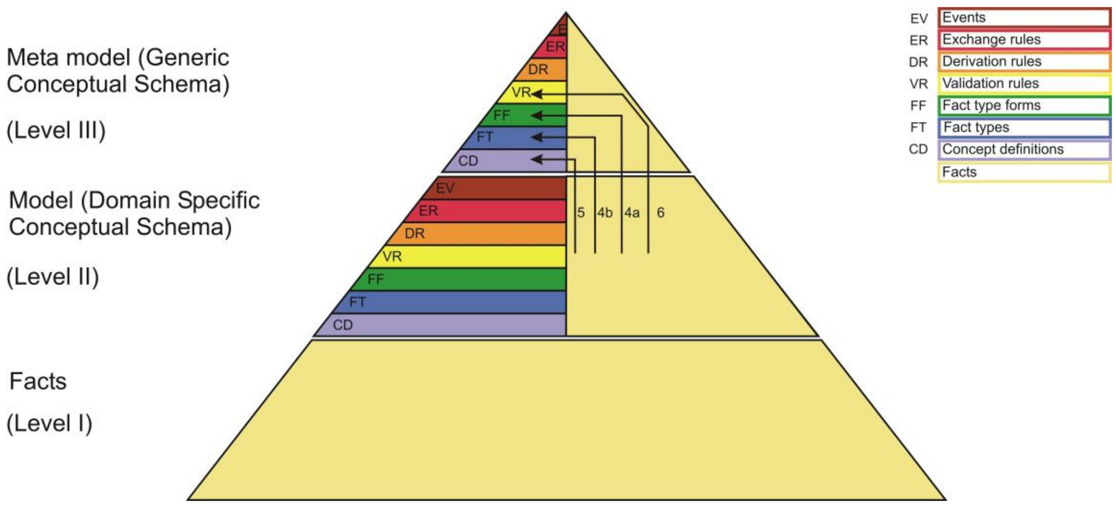


Figure 10-4 The elicitation process from the domain-specific model (Level II) to the generic model (Level III)

In the next section we will apply the same CogNIAM conceptual model specification processes to the fact type of Figure 10-3 to see where we end up, at the generic model level or …

11 From the Generic Model to its Model: Part I, Fact Types

We have applied the two CogNIAM processes verbalization and generalization into fact types and the derivation of the uniqueness validation rules, to the ground fact level and the domain-specific model level.

The question is: where will we end up if we apply *exactly the same* model specification processes to the meta model (M2)?

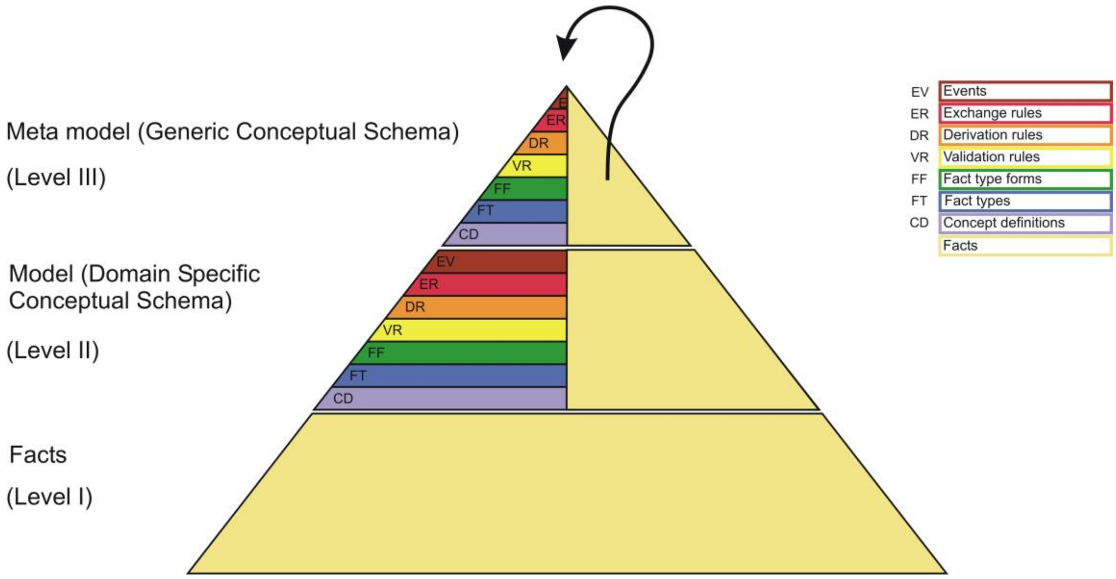


Figure 11-1 Where do we arrive if we start the conceptualization process at the generic model level (Level III)?

It is an assumption of the CogNIAM protocol that verbalization can be applied at any level of the knowledge triangle, hence also at the meta-model (M2) level. This was already accepted as a working axiom in 1980 (Nijssen 1980). It is a pleasure to read in a recent article: “One common characteristic of software analytics tools is that they view software as data.” (KDM Analytics 2010). Let us do this knowledge elicitation process for the facts about variables of Figure 10-3 (for the reader’s convenience represented in Figure 11-2. In Figure 11-2 the parts that are verbalized and their corresponding part of the verbalization are correspondingly colored.

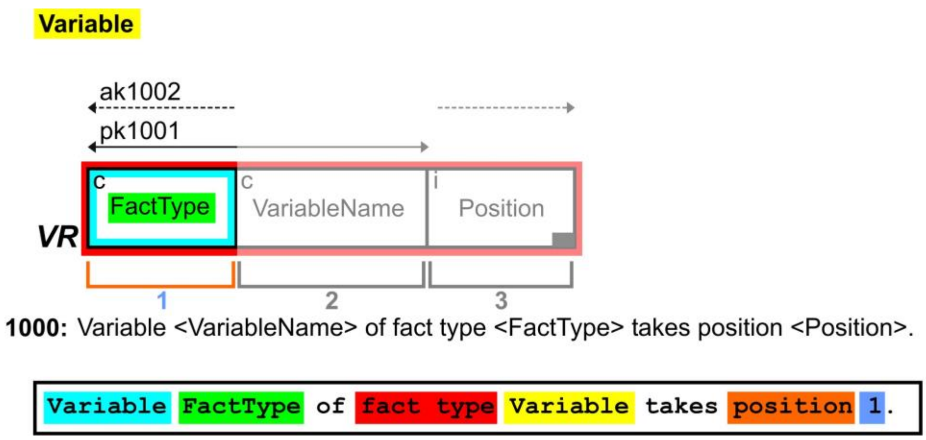


Figure 11-2 Correspondence between diagrammatic representation and verbalization

It is recommended to take a minute to check that the corresponding colors highlight the same information but represented in a different way. We have observed over the years that in the beginning, this kind of verbalization may be confusing for many. However, this is the exact verbalization. The (meta) fact type or predicate “Variable” has a variable with the name “FactType”. This verbalization is not using fact type form 1000! It is just consistent verbalization.

Table 11.1 Result of applying the processes of verbalization and determining of variable parts at the generic model (M2) level

Variable	FactType	of	fact	type	Variable	takes	position	1.
”	VariableName	”	”	”	Variable	”	”	2.
”	Position	”	”	”	Variable	”	”	3.

Be aware that at the meta-model level (MOF M2), terminology necessarily begins to loop: it is crucial to strictly follow the established protocol and observe carefully the produced patterns.

If we apply the generalization process to these three facts we obtain the associated fact type reading and we give this fact type reading a different identification from all existing identifications for fact type readings so far and therefore 1001.

1001: Variable <VariableName> of fact type <FactType> takes position <Position>.

If we apply the CogNIAM process of diagrammizing to this fact type reading, we obtain the fact type diagram as represented in Figure 11-3. This is also level M2 in MOF!

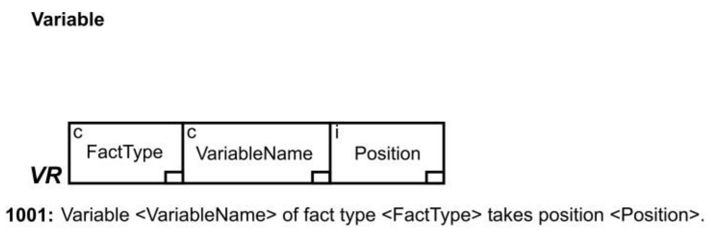


Figure 11-3 Intermediate fact type diagram

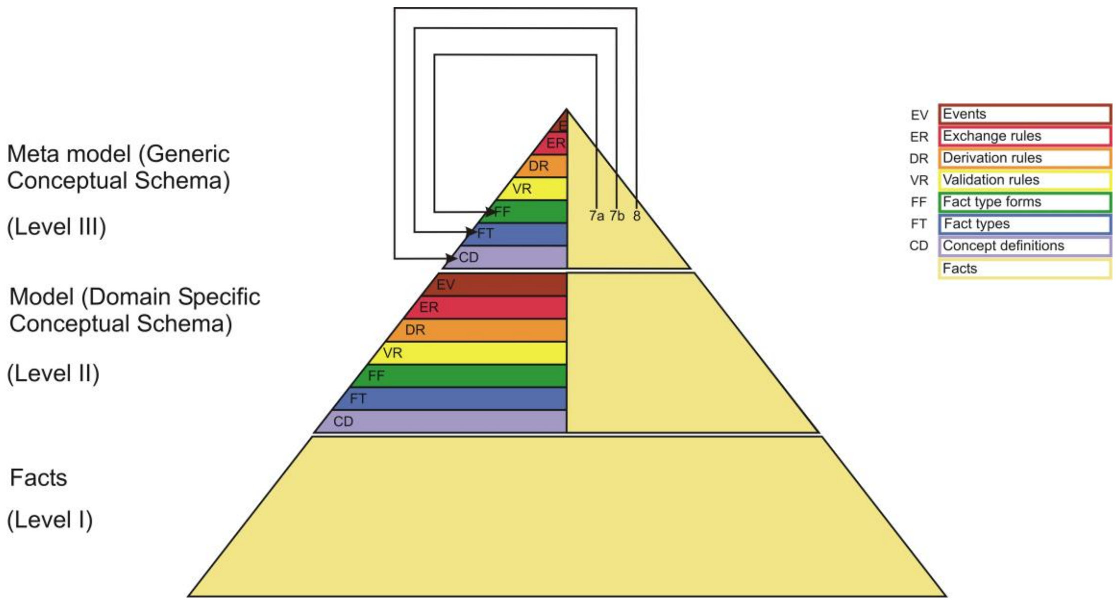


Figure 11-4 <TEXT>

12 From the Generic Model to its Model: Part II, Integrity Rules

If we apply the same uniqueness elicitation process as performed in sections 8 and 10, we obtain the fact type diagram including the two uniqueness constraints as represented in Figure 12-1.

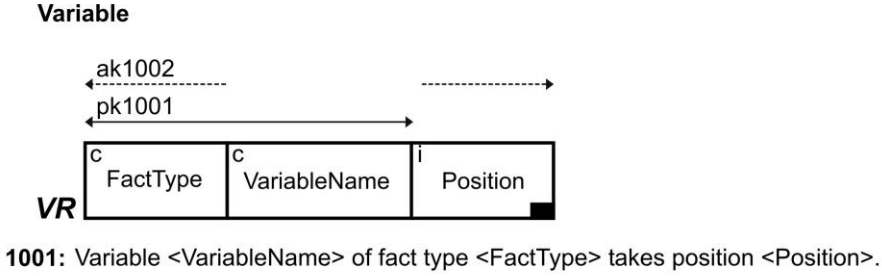


Figure 12-1 Fact type diagram at the generic conceptual schema level

Careful analysis of the fact type of Figure 10-3 and the fact type of Figure 12-1 results in the conclusion that these fact types are identical and that the fact type readings 1000 and 1001 are identical.
Of course we could apply the same CogNIAM model specification processes to arrive at the various other meta fact types like FactType, Constraint, PrimaryKey, AlternateKey, SubsetConstraint, EqualityConstraint, ExclusionConstraint, ValueRule, etc. The results lead to the same conclusion as has been drawn time and again.

Conclusion: if we apply in a consistent way the CogNIAM modeling processes of

1. Verbalization and
2. Generalization toward fact type reading,

the result is that the meta-meta-model of the MOF (M3) is exactly the same as the meta-model (M2).

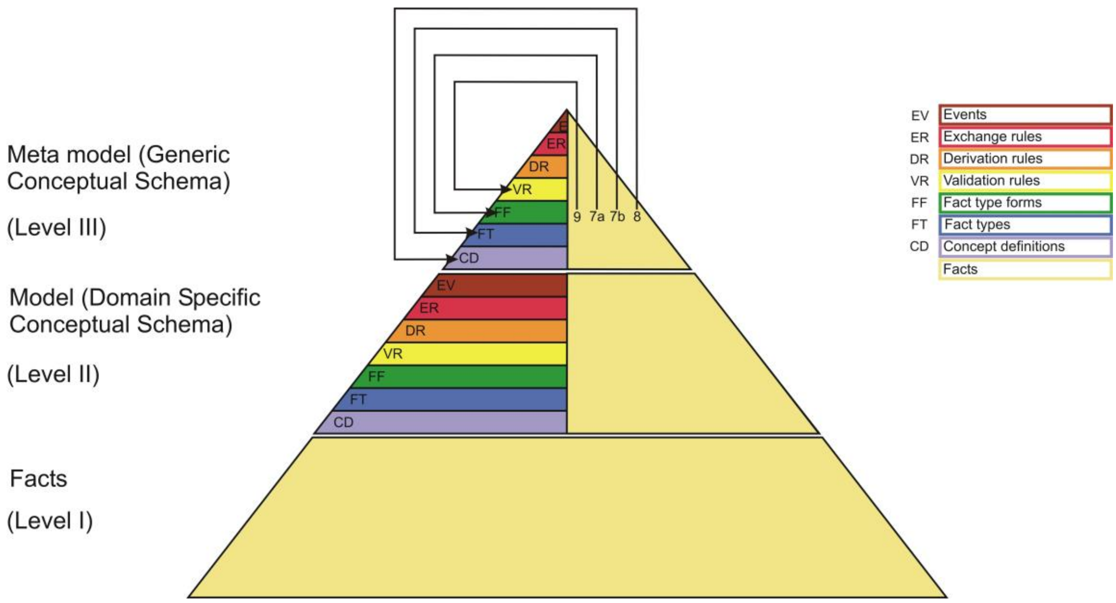


Figure 12-2 <TEXT>

13 The Knowledge Triangle as Knowledge Architecture

In another publication we have taken the same example as is used in the OMG Meta Object Facility (MOF) to analyze the OMG four-level MOF architecture. This example in itself was sufficiently representative to derive that an architecture of three levels, the ground level, the domain-specific model or domain-specific regulation level, and the generic model or generic regulation level, is sufficient. A major difference between this derived (CogNIAM) knowledge architecture and unvalidated OMG MOF architecture is the fact that the CogNIAM architecture is based on a clearly defined axiom and from there on logical reasoning.

We will now use another representative example to derive the *internal structure* of the knowledge triangle, consisting of three levels.

Many hours, perhaps millions could be saved in many organizations including top level standards experts groups, if in the communication consistent use would be made of a classification or stratification of knowledge elements into three categories or levels of knowledge:

I. Facts that have no grammatical function, called the ground level; e.g. The Nobel Peace Prize is awarded in Oslo, the Nobel Prize for Physics is awarded in Stockholm,
II. Domain specific rules, that specify which ground fact populations and transitions of ground fact populations are permitted and what each term means; this level is called the domain-specific regulations level; an example of a formulation at this level is: *The <NobelCategory> is awarded in <City>*, and
III. Generic rules that specify which domain specific rules and rule transitions are permitted and what each term means; this level is called the generic regulations level; an example at this level is: *<VariableName> of <FactType> takes <Position>*.

This is diagrammatically illustrated with the knowledge triangle represented before e.g. in figure 7.4.

Furthermore, we will classify the elements within the levels II and III into a small set of useful categories. This process in itself is an example of knowledge meta-modeling.

13.1 How to improve professional communication?

How is it possible to take advantage of these three levels of knowledge classification in business discussions (mostly at the OMG MOF level M0 and M1) and discussions among experts in modeling languages (mostly OMG MOF level M2)?
Many observations on the size and frequency of knowledge communication at each level in business, in the ontology community, in data bases and models for business systems, laws and regulations and standard specifications like BPMN, SBVR, OWL, Common Logic and UML bring us to the following conclusions:
For the volume of business and knowledge economy communication, the following holds:

I. More than 99.9% of all communication is at the level of the ground facts; this level is called M0 in OMG MOF, A-Box in OWL and database in relational technology;
II. Less than 0.1% is at the level of the domain specific regulation level; this level is called M1 in OMG MOF, the domain specific part of the T-Box in OWL and schema in relational technology and
III. Less than 0.001% is at the level of the generic regulation level; this level is called M2 in OMG MOF, the generic part of the T-Box in OWL and the meta-schema in relational technology.

Looking at these figures, it can be explained why discussions at level II (MOF M1) and III (MOF M2) are usually so ineffective.

To give an illustrating example: Level III (M2 at OMG MOF level) for relational is estimated at at most 1000 variables (10*3), domain specific relational schema descriptions are estimated to contain over 1,000,000,000 variables (10*9) and the databases associated with these relational schema descriptions are estimated to contain over 1,000,000,000,000,000,000 (10*18) facts. The M2 level of BPMN is around 150 variables and the M2 level of OWL2 is around 100 variables.
The same estimations apply to the knowledge available on the web.
Hence a minute volume of communication is at the OMG MOF M2 level. And as far as we know there is hardly any formal training at the M2 communication and modeling level.
However, standard modeling languages like BPMN, SBVR, OWL, Common Logic or UML are nearly exclusively defined at the generic regulation level (MOF M2). Hence one may expect that most business people do not understand communication at that level. Furthermore to perform a complete validation all three levels are needed.
And what to think of the lack of M1 level examples, considering validation of the M2 level specifications?

Our claim is that the productivity of professional communication can be significantly enhanced if all the relevant levels are systematically taken into account. This claim has been tested in business practice for the last 35 years and the observed results give ample support for the claim.

13.2 Assumption

It is our assumption that not recognizing and utilizing the possibilities offered by the knowledge triangle in professional daily work, both by the business community at the domain-specific level (MOF M1) and by the standards experts community at the generic level (MOF M2), is a major source of low productivity in the knowledge economy. For some this might be considered an inconvenient truth, for others it is a stimulating challenge on how to improve the situation. The very good news is: this can be improved and broadly implemented just as driving a car has become mainstream.

A similar reasoning applies to the M1 level. Effective validation of an M1-level domain-specific model, expressed in languages such as BPMN, OWL or SBVR, requires a representative set of M0-examples.

13.3 Requirements for a Representative Case or an Example to Be Used as M0 to Validate M1

It is the belief of the authors that a main reason for the problems clearly described by Cory Casanave on the OMG AE (Architecture Ecosystem) SIG Wiki is the fact the OMG and ISO, W3C, TOGAF and OASIS, in general and for MOF and for UML, BPMN, PRR, SBVR and the Business Architecture in particular, to mention a few silos, do not adhere to the following guidelines:

a. always start with objectively observable results or inputs of processes,
b. express as much as possible the results or inputs in the preferred natural language and notation of the business expert, knowledge owner or subject matter expert,
c. communicate as professional business analyst (BPMN) or knowledge engineer (SBVR, OWL, FOL, CL) as much as possible in natural language sentences, with the knowledge owners,
d. apply a procedure (process, methodology, protocol) how to go from the concrete communication (called ground facts in SBVR and assertions in OWL [or the contents of the A-Box) to the domain specific business model, and validate the model by having the user give his answer to a representative set of permitted and not permitted concrete examples developed by the professional business modeller in the preferred notation of the end user or subject matter expert,
e. make sure that all stakeholders receive the model description in their preferred language, usually a controlled natural language,

f. have a two-way trace from the first initial document, via all intermediate documents, to the elements in the final business model.

What are the requirements for a good use case for an Architecture Ecosystem (AE) or the development of a standard for Business Architecture? Please remember that a major goal of the AE and Business Architecture is to make *integration* or *federation* between models and, where possible, modelling languages a reality. Hence such an Architecture Ecosystem or Business Architecture needs to be build up from concrete examples as *validation* devices for the domain specific models (M1) and using a representative set of domain specific models to derive and validate the meta model (M2). The key word in the previous sentence was *derive* as opposed to *postulate*.

I. The ground facts (called contents of the A-Box in OWL, M0 in OMG MOF, database in relational) should be of interest to a large audience.

Why this requirement of ground facts (or parts of an A-Box)? A business model needs to be checked with the relevant stakeholders. A program or system is tested by having the program or system execute some input and analyze the output. Most stakeholders are very good at recognizing the good and bad pairs of input-output combinations, and even more if the input and output is represented in the preferred representation of the stakeholder.

II. The knowledge of the case should preferably be in the public domain.

Why the requirement of being in the public domain? A use case needs to be representative. Hence there is a need to discuss the complete system, or at least in that degree of completeness that business people accept it as a realistic example. A public domain problem makes this possible.

III. The knowledge should cover a long period of time, ideally starting before IT was introduced and cover the entire IT-period.

Why? A long period of time will in all probability introduce different domain-specific laws or rules at different periods in time. This is a good case for illustrating version control and operating with slowly changing models. Furthermore by having the period start before IT it is straightforward to show that the deep structure communication is the same before and during the IT period.

IV. There should be a number of business processes and actors (participants) that is representative for a substantial business.

Why? Some business people only look at the number of business processes and actors involved in a use case. If this number is below a certain threshold their interest is gone. It would be a pity to loose such attention by having a too small number of business processes and actors.

V. The knowledge should cover a representative set of non-trivial concepts.

Why? As the world moves more and more into the knowledge economy the number of abstract concepts increases. A representative use case should include all the difficulties of defining abstract concepts in such a way that the stakeholders can apply the definitions in their daily practice.

VI. The knowledge should cover a representative set of fact types.

Why? Most specifications and textbooks use tiny examples. Most business people are then not willing to read further. They are not in the toy business, is their opinion. Hence use an example that is regarded by the business people to be representative in term of fact types (information bearing constructs, populatable constructs, data structure in computer science terms) for an average business application.

VII. The knowledge should cover sufficient subtypes or a class hierarchy.

Why? In most business applications there are important examples of subtypes.

VIII. The knowledge should contain a representative set of validation rules.

Why? It is amazing to see that there are movements in the semantic web and the ontology community that grossly underestimate the value of validation rules. Why do people close their eyes for the meta derivation rule: garbage in, garbage out. Without the complete set of validation rules, there is no guarantee that the result of a derivation rule or inference engine has any value.

IX. The knowledge should cover a representative set of derivation rules.

Why? Most business people are most interested in the results of the derivation rules as this is what they consider of interest.

X. The knowledge should cover a representative set of events.

Why? Most business persons consider the event as the beginning of action by a business process the result of which is presented to the customer. Hence a use case should contain a representative set of events.

XI. There should be an AS-IS situation in practical use.

Why? There should be a process running in practice such that there is an observable AS-IS situation and the Architecture Ecosystem or Business Architecture can be used to develop the TO-BE situation

Unfortunately there are not many cases that satisfy all these requirements. The EU Rent case extensively used in OMG SBVR fails on more than one of the requirements listed above. Similar remarks can be made about the examples used in OMG UML, BPMN 1.0 and PRR, or TOGAF 9.

But luckily there is at least one case that satisfies all the eleven requirements. This is the case of the Nobel Prize:

1. Ground facts or assertions (in the ABox, or M0 level on OMG MOG) about the Nobel Prize are certainly of interest to a large audience. The website of the Nobel Prize Foundation gives many examples.
2. The knowledge about the Nobel Prize as well as the processes is in the public domain.
3. The knowledge and processes of the Nobel Prize cover the period since 1901, a long time before IT came into existence and it has been going since the beginning of IT and is still continuing.
4. There is a number of non-trivial business processes involved in the Nobel Prize business. A thorough analysis of the business processes associated with the Nobel Prize business reveals that the degree of difficulty is representative for a business.
5. The Nobel Prize knowledge contains a number of non-trivial concepts. If a discovery for which the Nobel Prize is awarded, is shared by three persons, do we speak of three Nobel Prizes, or three Nobel Laureates sharing the same price? How is the amount of money derived? Is it known that a Nobel Laureate can only receive $^{1}/_{1}$, ¾, ½, ⅓ or ¼ of the prize amount? On which rules is this based in the original statutes of Alfred Nobel?
6. The Nobel Prize use case contains more than 50 fact types, enough to make a representative case.
7. The Nobel Prize case contains several subtypes, some overlapping, some non-overlapping. There are many persons involved, nominators, nominees, laureates, organizations, and several groups of persons.
8. The Nobel Prize case contains quite a few validation rules, some of which cannot be expressed in OWL, RDF or UML. See e.g. the first three statements of Clause 4 of the Statutes of the Nobel Prize Foundation.

9. The Nobel Prize case contains some very interesting derivation rules. In all probability a few new derivation rules could be developed with modern software technology to help improve the quality of decision making.
10. The Nobel Prize contains some well described set of events.
11. The Nobel Prize process is running year after year in the current AS-IS situation. It would be very convincing if the application of the OMG Architecture Ecosystem or the Business Architecture could result in significant improvements to the Nobel Prize processes

13.4 Intuitive summary

An intuitive introduction is as follows:

In communication about business goals, business architecture, business processes, business models, business analysis, business communication, knowledge description languages (e.g. SBVR, RDF, RDFS, SKOS, or OWL), process description languages (e.g. BPMN), system engineering languages (SysML), meta models (KDM, IMM) or implementation data structure languages (e.g. UML), or in discussions about conceptual modelling, the architecture represented by the CogNIAM knowledge framework consisting of three layers has proven, over many years and in many practical situations, to be very productive. Sometimes a claim is made that it has the Russell's paradox problem but this is not true as the CogNIAM architecture is only intended to be used with finite sets, and hence the Russell's paradox problem can be avoided.

In this section we will provide another view on the architecture derived before.

13.5 Abstraction level

The solid basis of the knowledge triangle is the so-called level of the ground facts. Ground facts can logically be considered as a predicate with only constants. This level is very familiar to the vast majority of people. As said before more than 99.9 % of human communication is at that level.
For the other two levels this knowledge triangle or architecture abstracts from some aspects and focuses on:

a. which kinds of facts or assertions do we consider to be within scope and about which individuals do these facts give us information, for a selected (business) communication, law, system or subject,
b. concept definitions, for each term that may not be entirely clear,
c. fact type forms, to be used in speech community specific communication,
d. validation rules to restrict the many fact bases (A-Boxes) and their transitions, made possible by the business model, to the ones considered useful,
e. derivation rules to generate new facts from existing facts in the fact base,
f. assert facts from the environment into the system or withdraw facts from the system, called exchange rules and
g. event rules to specify when to start a derivation or exchange rule.

Hence in this model it is chosen to abstract from e.g. authorization rules and internal structure of processes like: which actor (participant) may assert which facts or which is the sequence of activities that may generate facts. Why is this abstraction level chosen? It has turned out to provide the necessary and sufficient functionality for modelling, including process modelling. To avoid any misunderstanding the aspects of authorization and sequence of activities (processes) can be adequately modelled with the logic based concepts of the Knowledge Triangle (see
Figure 13-1).

13.6 Level I: the Ground Facts or Assertions

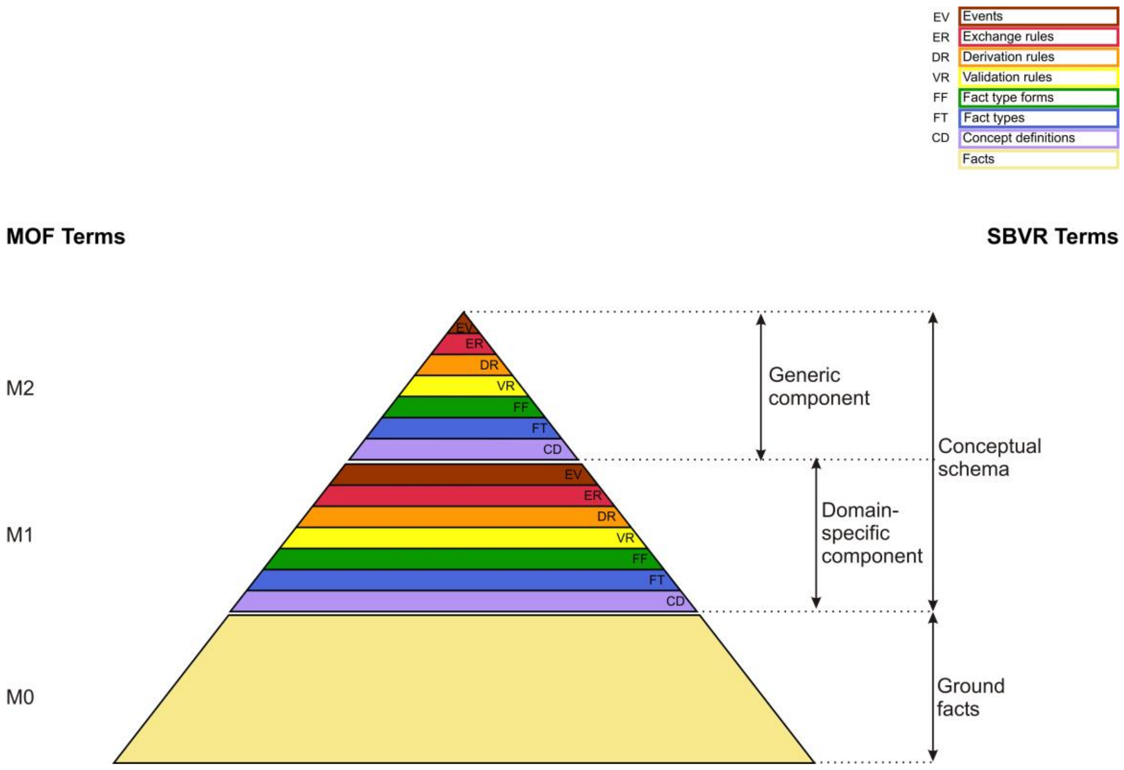


Figure 13-1 The Knowledge Triangle

The model consisting of all the different classes, their associations and the relevant integrity rules will be derived in this and later chapters. As a service to the various communities we provide in Table 13.1 the different terms for the same concepts.

Table 13.1 Translation of Knowledge Triangle terms into various standards

Level in CogNIAM	MOF	OWL	Relational	SBVR
III	M2 (meta-model)	Generic part of the T-Box	Meta-schema	Generic component of the conceptual schema
II	M1 (model)	Domain-specific part of the T-Box	Schema	Domain-specific component of the conceptual schema
I	M0 (information)	A-Box	Database	Ground facts

The assumption is that the most concrete level (level I) – the level at which the largest volume of inter-human, human-computer and inter-computer communication takes place – in any structured knowledge description consists of n-ary facts or n-ary assertions, with n = 1, 2, 3, etc. This level is called in SBVR the level of the ground facts, in relational technology based systems the data base, in OWL the level of assertions, restricted to unaries and binaries (the A-Box) and in UML the level of the objects and the links between pairs of objects (OMG MOF M0). For a summary see Table 13.2.

Table 13.2 Translation of Knowledge Triangle Level I terms into various standards

Level	MOF	OWL	Relational	SBVR
I	Objects and links	Assertions	Facts	Facts

We will now present a few examples at level I (MOF M0, OWL A-Box) which everybody can see at the Nobel Prize website (http://nobelprize.org/).
A few examples of this level in user notation are given below.

At the website of the Nobel Prize Foundation (http://nobelprize.org/nobel_prizes/chemistry/laureates/2007/) one can find the following:

The Nobel Prize in Chemistry 2007
"for his studies of chemical processes on solid surfaces"
Gerhard Ertl

Figure 13-2 Level I example – Nobel Prize in Chemistry 2007

A verbalization is: The Nobel Prize in Chemistry 2007 was awarded to Gerhard Ertl because of the work "for his studies of chemical processes on solid surfaces".

The example of

Figure 13-2 is representative for the situation where a Nobel Prize is awarded to a single laureate.

Let us bring in a few more examples from the Nobel Prize history such that the set of examples is representative for all the announcements made so far.

The Nobel Prize in Physiology or Medicine 2002		
"for their discoveries concerning 'genetic regulation of organ development and programmed cell death"		
Sydney Brenner	**H. Robert Horvitz**	**John E. Sulston**
1/3 of the prize	1/3 of the prize	1/3 of the prize

Figure 13-3 Level I example – Nobel Prize in Physiology or Medicine 2002

A verbalization is: The Nobel Prize in Physiology or Medicine 2002 was awarded to Sydney Brenner, H. Robert Horvitz and John E. Sulston, each for 1/3 of the prize because of the work "for their discoveries concerning 'genetic regulation of organ development and programmed cell death".

The example of

Figure 13-3 is representative for the situation where a Nobel Prize is shared by three laureates and each laureate receives 1/3 of the prize amount, and consequently only one work was awarded.

Another example of a Nobel Prize announcement is given below:

The Nobel Prize in Physics 2006	
"for their discoveries concerning 'genetic regulation of organ development and programmed cell death".	
John C. Mather	**George F. Smoot**
1/2 of the prize	1/2 of the prize

Figure 13-4 Level I example – Nobel Prize in Physics 2006

A verbalization is: The Nobel Prize in Physics 2006 was awarded to John C. Mather and George F. Smoot, each for 1/2 of the prize because of the work "for their discoveries concerning 'genetic regulation of organ development and programmed cell death".
This example is representative for the situation where two Nobel laureates receive each ½ of the prize, having contributed to one and the same work.

<table>
<tr><td colspan="2">The Nobel Prize in Chemistry 1998</td></tr>
<tr><td>"for his development of the density-functional theory"</td><td>"for his development of computational methods in quantum chemistry"</td></tr>
<tr><td>Walter Kohn</td><td>John A. Pople</td></tr>
<tr><td>1/2 of the prize</td><td>1/2 of the prize</td></tr>
</table>

Figure 13-5 Level I example – Nobel Prize in Chemistry 1998

A verbalization is: The Nobel Prize in Chemistry 1998 was awarded to Walter Kohn, for 1/2 of the prize because of the work “for his development of the density-functional theory” and to John A. Pope, for ½ of the prize because of the work “for his development of computational methods in quantum chemistry”.
This example is representative for the situation where two Nobel laureates receive each ½ of the prize, each having contributed to a different work.

<table>
<tr><td colspan="3">The Nobel Prize in Physics 2008</td></tr>
<tr><td>“for the discovery
of the mechanism
of spontaneous
broken symmetry
in subatomic physics”</td><td colspan="2">“for the discovery of the origin of the broken symmetry
which predicts the existence of at least three families
of quarks in nature”</td></tr>
<tr><td>Yoichiro Nambu</td><td>Makato Kobayashi</td><td>Toshihide Maskawa</td></tr>
<tr><td>1/2 of the prize</td><td>1/4 of the prize</td><td>1/4 of the prize</td></tr>
</table>

Figure 13-6 Level I example – Nobel Prize in Physics 2008

A verbalization is: The Nobel Prize in Physics 2008 was awarded to Yoichiro Nambu, for 1/2 of the prize because of the work “for the discovery of the mechanism of spontaneous broken symmetry in subatomic physics” and to Makato Kobayashi and Toshihide Maskawa, each for ¼ of the prize because of the work “for the discovery of the origin of the broken symmetry which predicts the existence of at least three families of quarks in nature”.

The example of

Figure 13-6 is representative for the situation where a Nobel Prize is shared by three laureates, two of whom receive ¼ of the prize because they contributed to the same awarded work and one receives ½ of the prize for another awarded work.

The previous 5 examples give a good overview of the patterns of announcements that are made on the Nobel Prize website.

A small excerpt from another list from the Nobel Prize Foundation website is given below:

2009, THE NOBEL PRIZE IN PHYSICS
Charles K. Kao, Willard S. Boyle, George E. Smith
2009, THE NOBEL PRIZE IN CHEMISTRY
Venkatraman Ramakrishnan, Thomas A. Steitz, Ada E. Yonath
2009, THE NOBEL PRIZE IN PHYSIOLOGY OR MEDICINE
Elizabeth H. Blackburn, Carol W. Greider, Jack W. Szostak
2009, THE NOBEL PRIZE IN LITERATURE
Herta Müller
2009, THE NOBEL PEACE PRIZE
Barack H. Obama
2009, THE SVERIGES RIKSBANK PRIZE IN ECONOMIC SCIENCES IN MEMORY OF ALFRED NOBEL
Elinor Ostrom, Oliver E. Williamson
2008, THE NOBEL PRIZE IN PHYSICS
Yoichiro Nambu, Makoto Kobayashi, Toshihide Maskawa
2008, THE NOBEL PRIZE IN CHEMISTRY
Osamu Shimomura, Martin Chalfie, Roger Y. Tsien
2008, THE NOBEL PRIZE IN PHYSIOLOGY OR MEDICINE
Harald zur Hausen, Françoise Barré-Sinoussi, Luc Montagnier
2008, THE NOBEL PRIZE IN LITERATURE
Jean-Marie Gustave Le Clézio
2008, THE NOBEL PEACE PRIZE
Martti Ahtisaari
2008, THE SVERIGES RIKSBANK PRIZE IN ECONOMIC SCIENCES IN MEMORY OF ALFRED NOBEL
Paul Krugman

Figure 13-7 Level I example – All Nobel Prizes in 2009 and 2008

The verbalization of the shaded lines in

Figure 13-7 is: The Nobel Prize in Physics 2008 was awarded to Yoichiro Nambu, to Makato Kobayashi and Toshihide Maskawa.
Please note that this derived compound fact has abstracted from the awarded work and part of the prize.
The previous 6 figures give a representative set of examples of core subset of the Nobel Prize Foundation.

Let us now go to the next level of the Knowledge Triangle, the level that is needed to fully understand the concrete communication or examples given above.

13.7 Level II: the Domain Specific Business Model

The next level of the knowledge triangle consists of seven knowledge categories, with the aim to provide knowledge to fully understand the facts at level I (MOF M0).

13.7.1 Concept Definitions

The first knowledge category of level II contains a list of concepts and their description for the terms occurring in facts at level I (M0), and the terms used at level II (M1). In SBVR this category is called concept definitions. In OWL, UML and relational they are absent. The function of this knowledge category is to describe every term or group of terms in the ground facts or assertions for which it is assumed that the meaning may not be fully clear to the intended audience or speech community as well as all the terms of M1.
In the case of the Nobel Prize examples given before, the following concept definitions may enhance the understanding of the ground facts.

Category
The {Nobel Prize} is awarded each year in 6 [Categories]:
Physiology or Medicine;
Physics;
Chemistry;
Literature;

Peace;
Economic Sciences (Sveriges Riksbank Prize in Economic Sciences).

Discovery
The {Discovery} is the achievement why a person is nominated and eventually may be awarded the [Nobel Prize].

Laureate
A {Laureate} is a [Nominee] that is a winner of a [Nobel Prize] in any [Year].

Nobel Prize
Every [Year] since 1901 the {Nobel Prize} has been awarded for achievements in physics, chemistry, physiology or medicine, literature and for peace. The {Nobel Prize} is an international award administered by the Nobel Foundation in Stockholm, Sweden. In 1968, Sveriges Riksbank established The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel, founder of the {Nobel Prize}. Each prize consists of a medal, personal diploma, and a cash award.

Nominee
A {Nominee} is a person that has participated in a [Discovery] that is nominated for a [Nobel Prize] by one or more nominators.

Work
A [Discovery] is called {Work} in the statute.

synonym:
Discovery

Year
Assumed to be known

A concept definition can furthermore be added for each object type, fact type, validation rule, derivation rule, exchange rule and event. It is recommended to add concept definitions for each term or verb of a fact type form. Also new concept definitions can be added to describe other concepts.

It is recommended in the CogNIAM protocol to avoid circular references between concept definitions. If it is likely to assume that a certain concept is clear to every reader, this concept can be indicated as “assumed to be known”. If a concept description is taken from an authoritative source, this source is also recorded.

13.7.2 Fact Types

For every system or subject there is a need to define which kinds of facts or assertions at level I (A-Box, database, M0 in MOF) are considered to belong to the system, subject or domain, and by definition all others are excluded from consideration, often also called “out of scope”. This is the populatable construct. This *type* construct at level II is the only one that has corresponding *instances* at level I.

This functionality is provided by
a. the fact types in SBVR,
b. the classes, and object and data properties in OWL,
c. the tables and columns in the relational model and
d. in UML by the classes, the properties and the associations and
e. in ER by the entity types, the attributes and the relationships.

The fact type is considered the only populatable construct in the conceptual approach of logic based Fact Based Modeling (NIAM, ORM2, CogNIAM, FCO-IM) and SBVR.
In IT language it is the only data structure at level II, the domain specific level.
Please note that in OWL, UML and ER there are three populatable constructs. If a dynamic change by the environment forces a migration of a fact type encoding from one type of construct to another, the inherent instability of models with more than one populatable construct becomes clear.

This construct has the name fact type in Fact Based Modeling and SBVR, class (unary fact type) or object property (binary fact type) or data property (binary fact type) in OWL, table and column in the relational model, and class (unary fact type or part of a n-ary fact type), property (binary fact type) or association (binary fact type) in UML.

Examples of (elementary and compound) fact types (in textual and diagrammatic format) are for the Nobel Prize case:

14: Within the collection of all laureates to whom Nobel prizes are awarded the laureate name <LaureateName> identifies a specific laureate.

7: Within the collection of all works for which Nobel prizes are awarded the work title <WorkTitle> identifies a specific work.

1: Within the collection of all categories for which Nobel prizes are awarded the category name <CategoryName> identifies a specific category.

3: Within the collection of all years which are relevant to the Nobel Prize Foundation the year number <YearNumber> identifies a specific year.

11: <Laureate> contributed to <Work>.

8: <Work> is awarded in <Category>.

9: <Work> is awarded in <Year>.

23: <Laureate> was awarded in <Category> in <Year>.

22: <Work> is awarded in <Category> in <Year>.

12: The prize in <Category> in <Year> was awarded to <Laureate> for the contribution to <Work> resulting in <PartOfPrize>.

In a diagrammatic format the fact types, the fact type forms, integrity rules, derivation rules and event rules are all represented in one (compound) diagram:

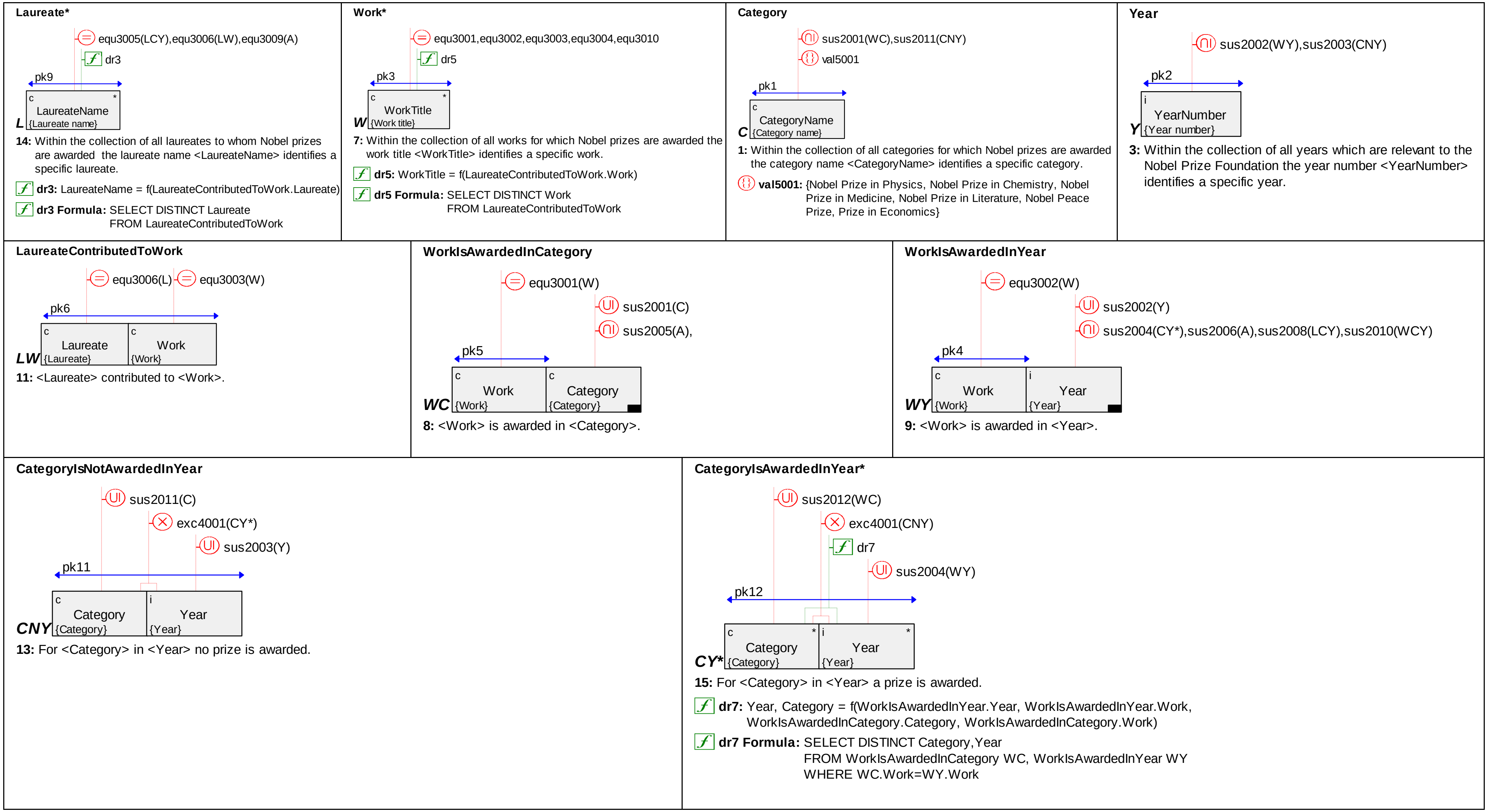
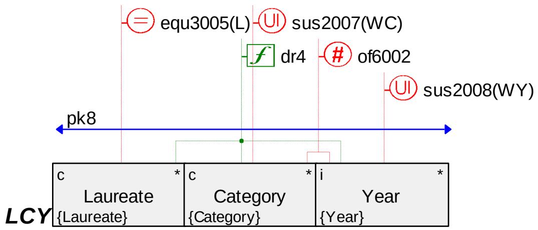


Figure 13-8 Fact type diagrams for Nobel Prize example (part 1)

LaureateWasAwardedInCategoryAndInyear *



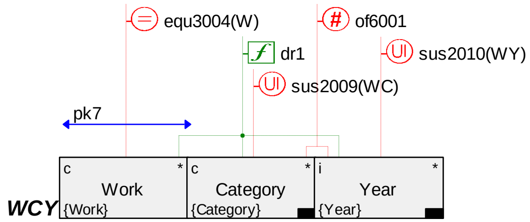
23: <Laureate> was awarded in <Category> in <Year>.

of6002: /..3/

dr4: Year, Category, Laureate = f(WorkIsAwardedInYear.Year, WorkIsAwardedInYear.Work, WorkIsAwardedInCategory.Category, WorkIsAwardedInCategory.Work, LaureateContributedToWork.Work, LaureateContributedToWork.Laureate)

dr4 Formula:
```
SELECT Laureate, Category, Year
FROM LaureateContributedToWork LW,
    WorkIsAwardedInCategory WC,
    WorkIsAwardedInYear WY
WHERE LW.Work = WC.Work
 AND WC.Work = WY.Work
```

WorkIsAwardedInCategoryAndInYear *



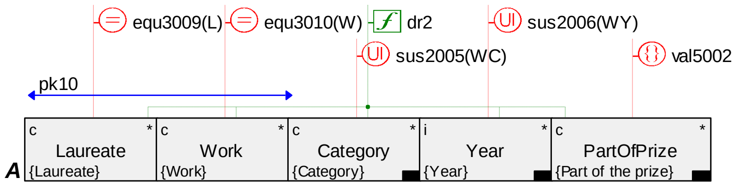
22: <Work> is awarded in <Category> in <Year>.

of6001: /..2/

dr1: Year, Category, Work = f(WorkIsAwardedInYear.Year, WorkIsAwardedInYear.Work, WorkIsAwardedInCategory.Category, WorkIsAwardedInCategory.Work)

dr1 Formula:
```
SELECT WC.Work, Category, Year
FROM WorkIsAwardedInCategory WC, WorkIsAwardedInYear WY
WHERE WC.Work = WY.Work
```

Announcement*



12: The prize in <Category> in <Year> was awarded to <Laureate> for the contribution to <Work> resulting in <PartOfPrize>.

val5002: {1, 1/2, 1/3, 1/4}

dr2: Year, PartOfPrize, Category, Laureate, Work = f(LaureateContributedToWork.Work, LaureateContributedToWork.Laureate, WorkIsAwardedInCategoryAndInYear *.Year, WorkIsAwardedInCategoryAndInYear *.Category, WorkIsAwardedInCategoryAndInYear *.Work)

dr2 Formula:
```
SELECT Laureate, LW1.Work, Category, Year, (
 SELECT 1/(
  SELECT COUNT(*)
  FROM WorkIsAwardedInCategoryAndInYear WCY2
  WHERE WCY2.Category=WCY1.Category
   AND WCY2.Year=WCY1.YEAR
 )
 /
 SELECT COUNT(*)
 FROM LaureateContributedToWork LW2
 WHERE LW2.Work=LW1.Work
) as PartOfPrize
FROM LaureateContributedToWork LW1,
    WorkIsAwardedInCategoryAndInYear WCY1
WHERE LW1.Work = WCY1.Work
```

Figure 13-9 Fact type diagrams for Nobel Prize example (part 2)

As a service to those readers who are not familiar with the CogNIAM conventions to express a knowledge grammar, we now provide a reading companion to the fact type diagrams shown in Figure 13-8 and Figure 13-9 above.

In the figures above there are a total of:

4 unary fact types (row 1 in the diagramseries),
5 binary fact types (rows 2 and 3 in the diagramseries),
2 ternary fact types (row 4 in the diagramseries) and
1 quinary fact type (row 5 in the diagramseries).

In OWL terminology one would say there are 4 classes and 5 object properties; the representation of the 2 ternaries and the 1 quinary is out of scope for this paper.

Some of the facts (instances) of these fact types are asserted, some are derived. A variable that is asserted is represented with a rectangle without an * in the upper right corner; a derived variable has an * in the upper right corner of its rectangle. When all the variables in a fact type are derived, the name of the fact type itself is followed by an asterisk.

Unary fact types L (Laureate) and W (Work) are derived from others, as their populations depend on decisions of the relevant Nobel Committees. Unary fact types C (Category) and Y (Year), on the other hand, can be populated ahead of time as the available Nobel Prize Categories and the available Years are known in advance.

Integrity rule (constraint) equ3020 (on fact type L and LW) specifies that the population of variable LaureateName of fact type L (Laureate) is at all times *equal* to the population of the variable Laureate in fact type LW (LaureateContributedToWork).

Integrity rule sus2007 (on fact type WC and C) specifies that the population of the variable Category of the fact type WC (WorkIsAwardedInCategory) is at all times a *subset* of the population of the variable CategoryName of fact type C (Category).

Value constraint val5001 (on fact type C) specifies that the variable CategoryName can only contain a value from the following list: {Nobel Prize in Physics, Nobel Prize in Chemistry, Nobel Prize in Medicine, Nobel Prize in Literature, Nobel Peace Prize, Prize in Economics}.

Occurrence frequency constraint of6002 on the combination of the variables Category and Year of derived fact type LCY (LaureateWasAwardedInCategoryAndInYear) specifies that at most 3 laureates can be rewarded in a category in a year. Source: Clause 4 of the Statutes of the Nobel Prize Foundation.

Occurrence frequency constraint of6001 on the combination of the variables Category and Year of derived fact type WCY (WorkIsAwardedInCategoryAndInYear) specifies that at most 2 works can be rewarded in a category in a year. Source: Clause 4 of the Statutes of the Nobel Prize Foundation.

Each of the various Nobel Committees makes an announcement for its category and a specific year. The real news (assertions) consists of the fact (instances) of the binary asserted fact type with name LaureateContributedToWork (and code LW), occupying the first position in row 2 of the combined diagram.
As soon as a new fact instance is inserted into the population of fact type LW, the following events are triggered: er3, er4 and er5.

Note that the SQL fragments are a convenient and widely understood notation to express queries and operations on sets of values in fact type populations. Any other declarative set-handling language will do.

er4. Event er4 (fact type LW) triggers derivation rule dr5, after (>>) every insert (I) in the variable Laureate of fact type LW.
The description of derivation rule dr5 is given under the fact type LCY (1st position in row 3) in SQL. Please note that the variable Laureate is preceded by an @; this means that the value of the variable Laureate is taken from the fact instance from where the event is triggered. Hence there is no reason to include the fact type LW in the SQL FROM clause. An INSERT is performed INTO the population of fact type LCY; all three variables of fact type LCY are derived. [In this case the UPDATE and DELETE of fact type LW is not taken into consideration.]

Derivation rule dr3 computes which part of the prize a certain Laureate receives for his contribution to a work.

13.7.3 Fact Type Forms

In order to provide the functionality that the fact instances or assertions can be communicated in a speech community specific jargon, the concept "fact type form" is used. This concept was introduced a long time ago in CogNIAM. In SBVR this concept has the name fact type form. A fact type form may be a sentence pattern (<President> married <Spouse>) or an object pattern (The marriage of <President> and <Spouse>). This communication concept is not available in OWL, RDF, RDFS, the relational model nor in UML.

13.7.4 Validation Rules (Integrity Rules, Constraints)

The set of facts at level I, and transitions between sets of facts, need to be restricted to those that are considered useful. It is the function of the validation rules to restrict the populations permitted by the fact types and their transitions to useful ones. In SBVR this knowledge category is a subset of the category rules. In OWL there are some validation rules but not enough to describe each validation rule requested to describe all kinds of practically relevant business models. OWL does not cover all the validation rules as are usually needed in a real life business model. In (Halpin 2008) there is some further description of what is available in OWL and how it relates to what is needed in a business model. In UML there is even less of this kind of functionality than in OWL. For example, the statutes of the Nobel Foundation, clause 4, first paragraph, states the following: "A prize amount may be equally divided between two works, each of which is considered to merit a prize. If a work that is being rewarded has been produced by two or three persons, the prize shall be awarded to them jointly. In no case may a prize amount be divided between more than three persons." To correctly model this constraint would call for an occurrence frequency constraint spanning multiple variables, something which is not possible in OWL (Halpin & Morgan 2008, section 16.6) or introduce an artificial OWL class resulting in constructions unfamiliar to the relevant stakeholders. This is a disadvantage that is grossly neglected by the OWL community.

13.7.5 Derivation Rules

Some facts are asserted by the outside world, often called the environment of the system, and are thereafter considered part of the set of the ground facts (SBVR) or A-Box (OWL) or database in relational database technology based systems. In the case of the Nobel Prize Website the Nobel Prize facts are asserted by the relevant institution in accordance with the Nobel Statutes and thereafter available on the website. It is also possible and very useful in practice to derive facts from other facts in the population, asserted or derived. This is done through the use of derivation rules. In SBVR there is the construct of definition that can be used as derivation rule. Derivation rules are a subset of rules in SBVR. However there is a need to extend SBVR with some functions, arithmetical and operational operators to make the derivation rules in SBVR practically useful. In OWL there are other ways of defining derived unary facts. UML has no formal provisions to derive facts.

13.7.6 Exchange Rules

Exchange rules describe the processes how facts from the environment will be accepted as asserted facts, or that the environment requests to remove certain asserted facts from the set of ground facts or assertion-box. The function of an exchange rule is to move sentences from the outside world into the system, or the layer of facts, or remove sentences from the layer of facts. This knowledge category gives an answer to the following question: How are the contents of the fact population inside the system altered, or kept in sync with the wishes of the environment?

13.7.7 Event Rules

To make a system useful in practice it is needed to have a functionality to define when to start a derivation rule or an exchange rule. There are no events in SBVR. However, in BPMN events are first class citizens. Hence it is recommended to extend a practically useful SBVR with the concept of event.

With these knowledge categories as described above it is possible to describe all the aspects of languages like OWL, RDF, RDFS, SKOS, SBVR, BPMN, UML, SysML, SoaML etc.

13.8 Level III: the Generic Model or Meta Model

The same knowledge categories as have been described for the domain specific schema (level II, OMG MOF M1) are part of the generic conceptual schema. Each element in the generic conceptual schema is independent of any specific domain.
Interestingly enough, the generic conceptual schema is a population of itself.

13.9 All Structured Knowledge is Declarative

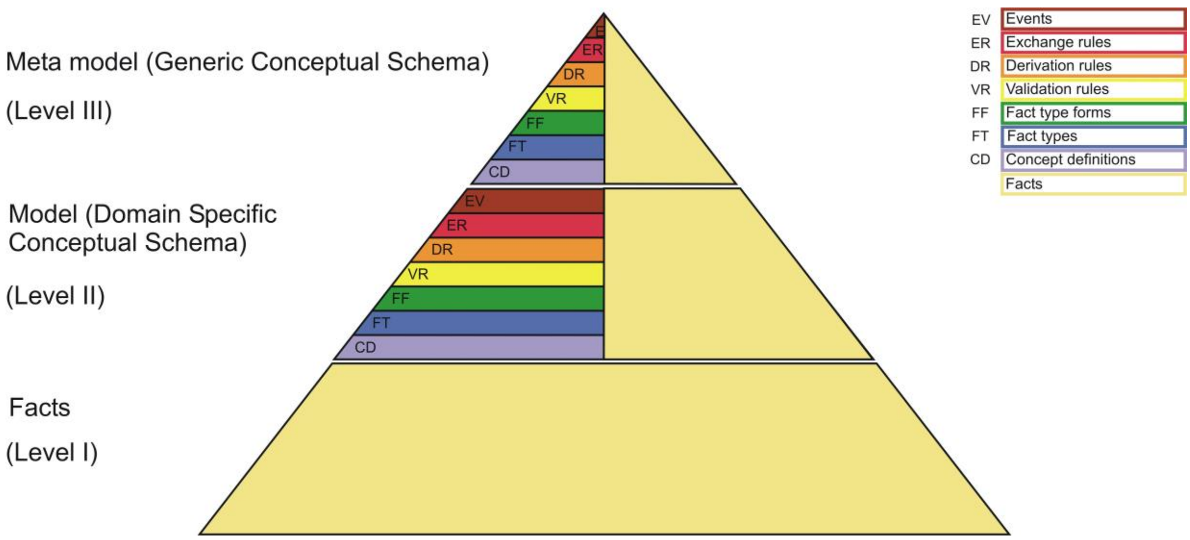


Figure 13-10

In
Figure 13-10 the knowledge framework is presented in a special way. It is assumed that all the knowledge at the domain specific model and generic model schema level is declarative. The consequence is that this knowledge can be verbalized and the result is a set of facts (Nijssen, 1980). Note that these facts are at level II and level III, hence these facts have a grammatical function. This means that four of the six sections of the knowledge triangle consist of facts (i.e. the rightmost parts of the generic and domain specific levels and both parts of the ground fact level).

13.9.1 Usefulness

In many practical modelling and rule application situations but also in discussions about business communications, new laws and regulations and modelling languages it is very helpful to have the Knowledge Architecture Framework as a tool to help in the communication or analysis. It helps in the communication between the analyst or ontologist and the business or subject matter expert. It gives a procedure to structure the communication as the expert in the Knowledge Triangle can easily switch between general rules and concrete illustrations of the rule and the other way around.

It also helps in the analysis of a new modelling language. Suppose one wants to study SKOS or UML. With the help of the Knowledge Triangle each element of the language to be studied can be mapped to the elements of the Knowledge Triangle. This gives very quickly a good insight what the modelling power of a new language is. The Knowledge Triangle has proven its usefulness in particular in difficult meta model discussions and the validation of meta models.

13.9.2 Summary and review

The Knowledge Triangle has been developed in the professional modelling practice as a tool for the modelling engineer, be it at the domain specific or generic level.

It is based on the assumption that there is a level of knowledge that consists of facts that have no grammatical function. This assumption is the same for OWL with the proviso that the OWL community has selected to restrict the OWL facts or assertions to unaries and binaries.

The next level is the level of the domain specific conceptual schema. The definition is as follows: a domain specific conceptual schema defines all the states and all the transitions of the fact bases at level 1 as well as all terms that are deemed to need a description. This level consists of a 7 knowledge classes,

1. the fact types,
2. the concept definitions of terms,
3. the fact type forms,
4. the validation rules,
5. the derivation rules,
6. the exchange rules and
7. the events.

With the Knowledge Triangle the same knowledge categories are available at the generic level.

A useful practical application of the Knowledge Triangle is to analyze a new modelling language and relate every new element to the existing structure of the Knowledge Triangle.

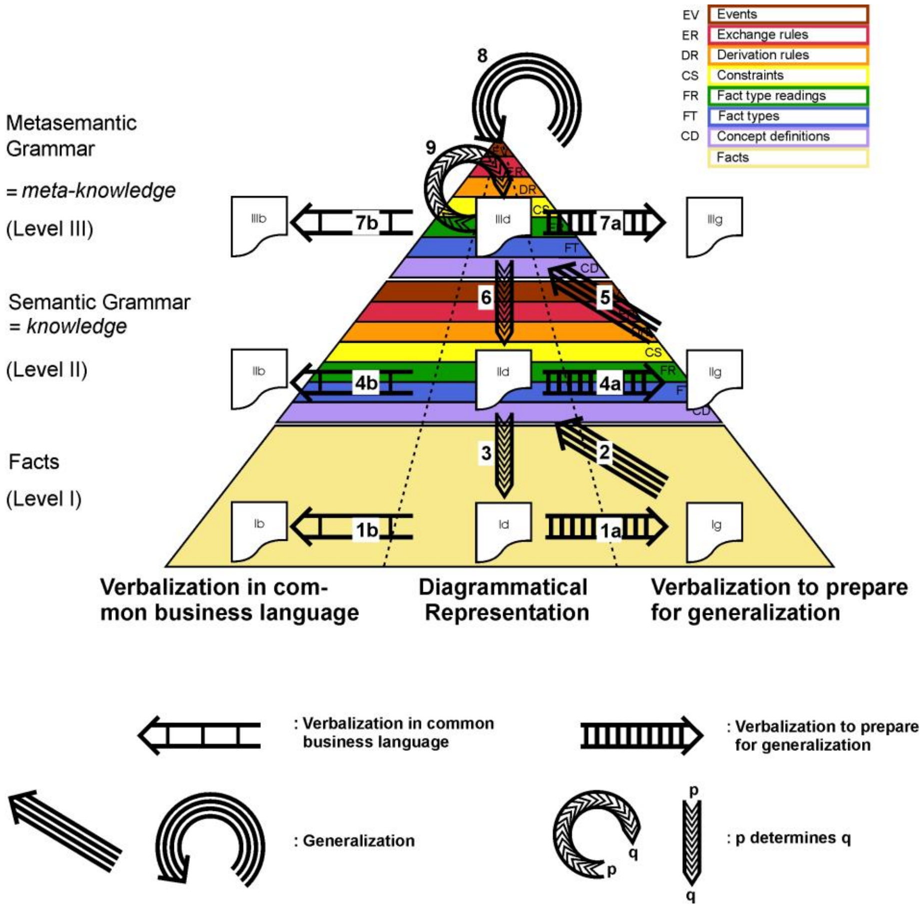


Figure 13-11 Knowledge triangle with three forms of representation and two kinds of verbalization processes

14 Using the Knowledge Triangle to Derive Clarity

In this chapter we will briefly discuss the concepts model, ontology, system, view, viewpoint, stakeholder and concern as we now have the equipment to define these concepts and describe how they relate to the knowledge architecture derived before. Most of the concepts are described in terms of the IEEE 1471 (ANSI/IEEE 2000).

14.1 What is a model?

The term model is quite differently used in logic, business, the ontology community and the data base world.
In logic model means the OMG MOF M0 level, the database level in relational technology, the A-Box in OWL and the set of ground facts in SBVR.
Model as used in many business discussions *either* refers to the database level (OMG MOF M0 level), e.g. an organization chart *or* the domain specific schema level (OMG MOF M1 level). It is a pity to read that the term model is used as a homonym on the OMG website. Hence in business discussions the term model often results in misunderstanding.
Model as used in OMG MOF is the domain specific conceptual schema level (OMG MOF M1).

14.2 What is an ontology?

The term ontology is one of the most misunderstood terms in the ontology community.
For some an ontology is the set of object types, the subtypes and the fact types.
For others, such as an OWL ontology, it means in addition a very restricted set of validation rules and a more extensive set of derivation rules. For some an ontology is a synonym for Conceptual Schema satisfying the 100% Principle and Conceptualization Principle of ISO TR9007 (ISO 1987).
It is recommended in productive discussions to ask what is meant by ontology. As long as there is so much confusion we recommend to use a prefix resulting in OWL-Ontology, SBVR-Ontology, ISO-TR9007-Ontology or CogNIAM Ontology.

14.3 What is a system?

A system is a collection of components organized to accomplish a specific function or set of functions. The term *system* encompasses individual applications, systems in the traditional sense, subsystems, systems of systems, product lines, product families, whole enterprises, and other aggregations of interest

14.4 What is a view?

A view is a representation of a whole system from the perspective of a related set of concerns.

14.5 What is a viewpoint?

A viewpoint is a specification of the conventions for constructing and using a view. 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.

14.6 What is a system stakeholder?

A system stakeholder is an individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system.

14.7 What is a concern?

Concerns are those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders. Concerns include system considerations such as performance, reliability, security, distribution, and evolvability.

14.8 What is a meta model?

If one takes the time to find out the semantics of the term meta model used at the OMG website one unfortunately comes to the conclusion that certain groups use meta model in the sense of M1 (e.g. the Business Architecture) while others use meta model in the sense of M2. It is recommended to OMG to stop this use of homonym as it results in needless frustration and loss of productivity. If M1 is called a model, then M2 is the meta model.

15 Comparing the Knowledge Triangle Architecture with existing architectures

In this chapter we will briefly compare the Architecture derived in this paper, the Knowledge Triangle with various architectures that play an important role in practice.

15.1 OMG MOF

So far we have not been able to find any record that contains evidence that the MOF architecture has been validated. We believe that no such validation has been applied to the MOF architecture. The reason for this is that it can be shown in a fairly straightforward way that the third and fourth level of MOF are the same. Furthermore the diagram used in MOF (see Figure 15-1) suggests that there are even more models. The knowledge triangle derived in this paper from a single axiom is clear: there are three levels.

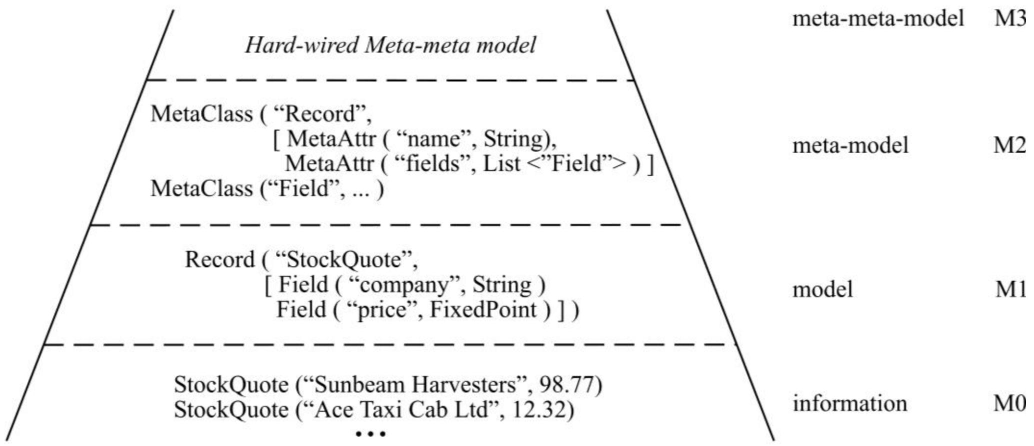


Figure 15-1 Four Layer Metadata Architecture

15.2 ISO MOF

ISO MOF is a copy of the OMG MOF. Hence everything presented in 15.1 applies.

15.3 TOGAF

The TOGAF architecture is not adequately described. So far there is no description of that architecture available that can be used in a validation procedure.

15.4 OWL

OWL assumes that the contents of the A-Box consists of unary facts (instances of OWLclasses) and binary facts (instances of OWL object properties or data properties). However nowhere in OWL is there a derived architecture with a specific number of levels and within a level a clear classification of the elements. Furthermore the meta model of OWL is incomplete in the specifications.
We would welcome the explicit use of the knowledge stratification in OWL specifications as described in this paper.

15.5 BPMN

For BPMN there is no architecture described in the specification. The vagueness of the BPM approach is used throughout in BPMN. Nevertheless, BPMN is a smashing success from the numbers of persons using it.

15.6 UML

For UML the same can be said as in 15.1.

16 System dynamics

In this section it will be argued that we need in general the capability to know for every fact when it starts to be valid and if known, when it has ended to be valid. As we have demonstrated that all knowledge at the three levels of the Knowledge Triangle consists of facts, dynamics applies at all three levels of the knowledge triangle.

16.1 At the ground fact level (OMG M0; OWL A-Box)

It is amazing that there is virtually no mention in the professional ontology literature of the fact that OWL assumes that all facts are constant at the A-Box level. This is the most rigid on the spectrum. OWL is therefore not useful in the vast majority of real life applications. In the relational database world the model permits full dynamics at the M0 level.

16.2 At the domain specific level (OMG M1; OWL domain specific part of the T-Box)

In most practical applications that range over a reasonable period, the domain specific conceptual schema needs an update every now and then. The Nobel prize case illustrates this very clearly (e.g. the addition of the Sveriges Riksbank Prize in Economic Sciences to the set of available Nobel Prize categories in 1968). The original relational model did not cover this. However a CogNIAM dynamics discipline make it possible for many kinds of changes to use relational technology in a way that the system can cope with changes in the domain specific conceptual schema.

16.3 At the generic level (OMG M2; OWL generic part of the T-Box)

In the well known relational technology implementation the meta model or schema can not be updated. This is a vestige from the early eighties and could be relaxed today. After all, the knowledge at the generic level also can be described as a finite set of facts and then the engineering solution mentioned under 16.2 is applicable.

17 Summary, Conclusions and Recommendation

In this paper we have consistently applied the CogNIAM model specification processes to the information level (OMG MOF M1), the model level (M1) and the meta-model level (M2) of the knowledge architecture. In doing so it became obvious that the meta-meta-model of MOF (M3) is the same as the meta-model of MOF (M2). Why then introduce a fourth ghost level in such an important specification? Our recommendation is to remove this inconsistency, as soon as practicable, from the MOF and ISO specifications in order to obtain a more solid understanding of the advantages of the MOF approach in the world at large. "The central theme of the MOF approach to metadata management is extensibility." (OMG 2006) This can much easier be obtained if the foundations are solid and do not lead to contradiction.

We have introduced in this working paper a conceptual architecture that we offer to the OMG Architecture Ecosystem SIG as a working paper. The modeling language itself is based on logic and set theory. It provides all the functionality to derive the federation and modeling functionality listed in the AE SIG so far. Of course it includes the mapping to and from UML and OWL and many others such that the investments in current models can be optimized. As such an Architecture Ecosystem should be made widely understood we recommend to describe it in e.g. an set theory version of CLIF but also in a set theory enhanced version of CLCE.

We welcome any and all feedback on this working paper. Our aim is not at all to present this working paper as a religious dogma but as a working paper to be used as a starting point for a productive and respectful exchange of ideas how to arrive at an Architecture Ecosystem.

We hope the decision makers at OMG understand that the world is looking for leadership in federated and integrated business modeling based on an enduring architecture that is technology independent.

18 Acknowledgements

A paper one sixth the size of this paper was presented at the ORM2007 conference. A special word of thanks for the excellent reviews to that version is presented here.
One review states: “I like the fearless challenge to the omni-accepted 4-level architecture; and for me, at least, this paper has put in grave doubt the legitimacy of that architecture. (I am embarrassed to say that I've never seriously questioned it until now.) Although someone could perhaps come up with a coherent, intelligible distinction between level 3 and level 4, I think this paper makes it very doubtful that that distinction could be relevant enough to legitimize a division into these two levels.” Another review says: “The paper proves well that CogNIAM can be used as a meta-meta model to define itself as a meta model, but it stops short of proving that it is capable of defining all meta models (such as: SysML, CWM, SBVR, BPMN, UML, ER) - which is the goal of the meta-meta model in the MOF 4-layer view.” In (Nijssen *1980*) one can read: “All grammars of the Coexistence information systems framework can be considered as information bases.” In current day language this means that the internal schema, be it UML or ER, can be and has been expressed by the authors in facts. BPMN and SBVR have also been expressed in facts by the authors. Space does not permit us to deliver these “derivation” proofs here.

19 Bibliography and References

ANSI/IEEE 2000, ANSI/IEEE Std 1471-2000, *Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE, NY, ISBN 0-7381-2518-0.

Abrial, Jean-Raymond, 1974, *Data Semantics*. IFIP Working Conference Data Base Management 1974: 1-60

Bakema, G., Zwart, J. & van der Lek, H. 2000, *Fully Communication Oriented Information Modelling*, Ten Hagen Stam, The Netherlands.

Boehm, B. 1981, *Software Engineering Economics*, Englewood Cliffs, NJ, Prentice-Hall, ISBN 0-13-822122-7.

Chen, P. 1976, 'The Entity-Relationship Model: Toward a Unified View of Data', *ACM Transactions on Database Systems*, March 1976, pp. 9-36.

CMMI Product Team 2006, 'CMMI for Development, Version 1.2', CMMI-DEV v1.2, CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute, August 2006, URL: http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf.

Codd, E. 1970, 'A Relational Model of Data for Large Shared Data Banks', *Communications of the ACM*, Vol. 13, No. 6, pp. 377-387.

Dearden, J. 1966, *Myth of Real-time Management information*, Harvard Business Review, May-June 1966.

Dietz, J. 2006, *Enterprise Ontology: Theory and Methodology*, Springer, Berlin, ISBN 978-3-540-29169-5.

Falkenberg, E. 1975, Strukturierung und Darstellung von Information an der Schnittstelle zwischen Datenbankbenutzer und Datenbank-Management-System.

Fillmore, C. 1988, 'The Case for Case', in *Universals in Linguistics*, Bach, E. and Harms, E.T. (eds.), pp. 1-88.

Frege, G. 1879, Begriffsschrift, eine der arithmetischen nachgebildete Formalsprache des reinen Denkens, Halle a. S.

Griethuysen, J. and Jardine, D. 1989, *Information Modelling with INFOMOD, Volume I – Concepts and Information Structure*, Philips, Eindhoven.

Habrias, H. 1988, *Le modèle relationnel binaire: Méthode I.A. (NIAM)*, Eyrolles, Paris, ISBN 978-2212081695.

Halpin, T. and Girle, R. 1981, *Deductive Logic*, 2nd edn, Logiqpress, Brisbane.

Halpin, T. and Morgan, T. 2008, *Information Modeling and Relational Databases*, Morgan Kaufmann, San Francisco, ISBN 978-0-12-373568-3.

ISO 1987, ISO/TC97/SC5/WG3 (J.J. van Griethuysen, ed.), *Concepts and Terminology for the Conceptual Schema and the Information Base*, Report No. ISO/TC97/TR9007, International Standards Organization, Central Secretariat, Geneva, Switzerland, 1987 (originally published in 1982).

ISO/IEC 13568:2002, *Z Formal Specification Notation*, ISO/IEC JTC 1, Information technology, Sub-committee SC 22, Programming languages, their environments and system software interfaces.

ISO/IEC 19502:2005, *Meta Object Facility*, JTC 1/SC 32, International Organization of Standards, 2005.

ISO/IEC 24707:2007, Common Logic, *Information technology - Common Logic (CL): a framework for a family of logic-based languages*, First Edition, Geneva, Switzerland: International Organization of Standards, October 1, 2007.

KDM Analytics 2010, 'Why Knowledge Discovery Metamodel?', URL: http://www.kdmanalytics.com/kdm/why_kdm.php, accessed on June 16, 2010.

Kendall, E. 2007, 'The Ontology Definition Metamodel: Motivation & Brief Introduction', URL: http://ontolog.cim3.net/file/work/DatabaseAndOntology/2007-01-04_ElisaKendall/ODM--ElisaKendall_20070104.pdf, accessed on June 16, 2010.

Kent, W. 1978, Data and Reality, basic assumptions in data processing reconsidered, North-Holland, ISBN 0-444-85187-9.

Lemmens, I., Nijssen, M. and Nijssen, G. 2007, 'A NIAM2007 Conceptual Analysis of the ISO and OMG MOF Four Layer Metadata Architectures', *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, Meersman, R., Tari Z., Herrero, P. (eds..), Springer-Verlag, ISBN 978-3-540-76887-6, pp. 613-623.

Lemmens, I., Sgaramella, F. and Valera, S. 2009, 'Development of Tooling to Support Fact-Oriented Modeling at ESA', *On the Move to Meaningful Internet Systems 2009: OTM 2009 Workshops*, Meersman, R., Tari Z., Herrero, P. (eds.), Springer-Verlag (to appear).

Meersman, R. 1982, *The RIDL Conceptual Language*, Research Report, Int. Centre for Information Analysis Services, Control Data, Brussels, Belgium.

Meersman, R. and Steel, T. (eds.) 1986, *Data semantics (DS-1) : proceedings of the IFIP WG 2.6 Working Conference on Data Semantics (DS-1), Hasselt, Belgium, 7-11 January, 1985*, North-Holland, Amsterdam, ISBN 0-444-70004-8.

Montague Grammar, Wikipedia entry, URL: http://en.wikipedia.org/wiki/Montague_grammar.

Morgan, T. 2002, *Business Rules and Information Systems: Aligning IT with Business Goals*, Addison-Wesley, Boston, p. 5.

Nijssen, G. 1978, A Framework for Discussion in ISO/TC97/SC5/WG3, ISO/TC97/SC5/WG3 78.09/01.

Nijssen, G. 1980, *A Framework for Advanced Mass Storage Applications*, page 1-21, Conference Medinfo, Tokyo.

Nijssen, G. 2007, 'SBVR: Semantics for Business', *Business Rules Journal*, Vol. 8, no. 10. URL: http://www.BRCommunity.com/a2007/b367.html.

OMG 2001, *Object Management Group*, *Model Driven Architecture* (MDA), ormsc/2001-07-01.

OMG 2002, *Object Management Group*, *Meta Object Facility (MOF) Specification, version 1.4*, Available as document formal/02-04-03 at http://www.omg.org/spec/MOF/1.4/.

OMG 2006, *Object Management Group*, *Meta Object Facility (MOF) Core Specification, version 2.0*, Available as document formal/06-01-01 at http://www.omg.org/spec/MOF/2.0/.

OMG, 2007, *Object Management Group*, *Architecture Driven Modernization (ADM): Knowledge Discovery Meta-Model* (KDM), KDM 1.0.3, KDM FTF ptc/2007-03-15.

OMG 2008a, *Object Management Group, Semantics of Business Vocabulary and Business Rules (*SBVR*)*, Version 1.0, Jan. 2008. Available as document formal/08-01-02 at http://doc.omg.org/formal/08-01-02.

OMG 2008b, *Object Management Group, Business Process Modeling Notation (*BPMN*) Specification*, v1.1. Available as document formal/2008-01-17 at http://www.omg.org/spec/BPMN/1.1/PDF.

OMG 2010a, *Object Management Group, Business Process Model and Notation (*BPMN*) FTF Beta for Version 2.0*. Available as document dtc/2010-05-03 at http://www.omg.org/spec/BPMN/2.0.

OMG 2010b, *Object Management Group*, 'Model Driven Architecture (MDA), The Architecture of Choice for a Changing World', URL: http://www.omg.org/mda/executive_overview.htm, updated on March 24, 2010.

OMG 2010c, *Object Management Group*, 'Architecture Ecosystem SIG Wiki', URL: http://www.omgwiki.org/architecture-ecosystem.

ONTORULE 2009, *ONTOlogies meet business RULEs*, 'Annex I: Description of Work', Grant Agreement for large-scale integrating project no. FP7 231875, funded by the EC under ICT call 3 FP7-ICT-2008-3.

Reusch, P. and Wintraecken, J. 1990, *Systemanalyse und Systemspezifikation*, BI-Wiss.-Verl., Mannheim, ISBN 3-411-14141-7.

Reusch, P. and Wolf, A. 1994, *Natürlich-sprachliche Informations-Analyse-Methode: NIAM in Theorie und Praxis*, IBIES, Dortmund.

Ross, R. 2009, *Business Rule Concepts: Getting to the Point of Knowledge, Third Edition*, Business Rule Solutions, ISBN 0-941049-07-8.

Senko, M. 1975, Information Systems: Records, Relations, Sets, Entities and Things, Information Systems 1, pp.1-13.

Senko, M. 1976, *DIAM as a Detailed Example of the ANSI SPARC Architecture*, in: Proceedings of the IFIP Conference Modelling in Data Base Management Systems, Nijssen (eds.), North-Holland.

Sowa, J. 1976, *Conceptual Graphs for a Data Base Interface*, IBM Journal of Research and Development, July 1976.

Sowa, J. 2009, *Controlled English*, URL: http://www.jfsowa.com/logic/ace.htm, accessed on 31/08/2009.

Sowa, J. 2010, 'Integrating Semantic Systems', tutorial held at 2010 Semantic Technology Conference, San Francisco.

Suppes, P. 1957, *Introduction to Logic*, Van Nostrand Reinhold Company, New York .

TOGAF, 2010, The Open Group, TOGAF Version 9, ISBN 978 90 8753 230 7

Valera, S., et al., ECSS 2009, *European Cooperation for Space Standardization*, ECSS-E-TM-10-23A, 'Engineering Database Version 2.0'.

Vanthienen, J. 2008, 'SBVR: The ABCs of **A**ccurate **B**usiness **C**ommunication', *Business Rules Journal*, Vol. 9, No. 3, URL: http://www.BRCommunity.com/a2008/b403.html.

Verheijen, G. and Bekkum, J. van 1982, *NIAM: An Information Analysis Method*, Proceedings of the IFIP WG 8.1 Conference on Comparative Review of Information System Design Methodologies, North-Holland, pp. 537-590.

W3C 2009, *World Wide Web Consortium*, OWL 2 Web Ontology Language: Overview, W3C Recommendation, URL: http://www.w3.org/TR/owl2-overview/.

Watson, A. 2008, 'Visual Modelling: past, present and future', OMG Whitepaper, URL: http://www.uml.org/Visual_Modeling.pdf.

Watson, P. 2006, *Ideas: A History of Thought and Invention, from Fire to Freud*, Harper Perennial, New York, ISBN 978-0-060935641.

Wittgenstein, L. 1921, *Logisch-Philosophische Abhandlung*, Annalen der Naturphilosophie, 14, translated as *Tractatus Logico-Philosophicus*, Ogden, C. 1922.

Zur Muehlen, M. and Recker, J. 2008, 'How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation', In: *Proceedings 20th International Conference on Advanced Information Systems Engineering*, CAiSE'08, Montpellier, France, Springer, ISBN 978-3-540-69533-2.