The Great Work of Michael Senko as a Part of a Durable Application Model

Peter Bollen(✉)

Maastricht University, Maastricht, The Netherlands
p.bollen@maastrichtuniversity.nl

Abstract. This paper defines a language for expressing a durable semantic application model as major part of the specifications for business applications This language consists of the modeling constructs from Natural Language Modeling (NLM). It will be shown in this article how these modeling constructs constitute a hierarchy within any durable semantic application model.

Keywords: Semantic-conceptual modeling · Natural Language Modeling · Michael senko

1 Introduction

As the number of information services (e.g. web-services, application service providers and e-commerce applications) continues to grow, in a globalized economy and the need for interoperability between those services becomes apparent in the context of organizational resilience, the availability of a durable modeling language that enables analysts to capture a changing and evolving application business communication is necessary. This language therefore must be able to establish links with common business vocabularies and data in applications and should be easy to understand by the various groups of stakeholders including non-expert users. Such a language must be very close to the natural language that is used by people in their daily business communications.

Some people assert that an ontology can serve as a semantic-conceptual model. In the literature a number of definitions for ontology can be found " an ontology is a description of the concepts and relationships for an agent or a community of agents." [1] , " shared understanding of a domain that can be communicated between people and application systems." [2]. "an ontology is a formal conceptualization of a real world, sharing a common understanding of this real world." [3: p.155]. [4] distinguish four types of material ontologies: *application, domain*, *generic* and *representation* ontologies. We conclude that the unqualified term ontology is a first class homonym that we recommend to avoid.

In the context of the diffusion of IT for organizational resilience, we will furthermore, require that a language for a durable application model allows for fast adaptation to ever changing organizational environments, in terms of products and services delivered, changing market regulations and so forth.

DOI: 10.1007/978-3-319-26138-6_25

The application modeling constructs in NLM are based upon the axiom that all verbalizable information (computer screens, reports, note-books, traffic signs and so forth) can be translated into *declarative natural language sentences* [5]. It means that it is *neither* a real *nor* an abstract world that is the object described in the model, but the *communication about* such a real or abstract world. This will require the feasible modeling constructs to provide those constructs that enable analysts to model what in the seventies and eighties was called natural language sentences and today more often is referred to as facts. Since the early 1970's a number of conceptual semantic models approaches have emerged. A NLM based model was the fact based modeling approach. The most widely used, by far, is the entity-relationship (ER) approach. The first definition of the ER constructs can be found in Chen [6]. This approach has been developed and extended further in [7] and [8]. The first fact based modeling approach had the name ENALIM, Evolving Natural Language Information Model. As the name implies Natural Language Information modeling was at the very beginning in the seventies. Soon thereafter Control Data decided to call the then as a professional service offered modeling NIAM, later explained as Natural language Information Modeling. From this ORM emerged in the nineties, as well as FCO-IM and DOGMA.

The rest of this paper is structured as follows: section 2 gives an introduction of the NLM modeling constructs for expressing a durable application model; in section 3 the concepts for the durable application modeling are given; in section 4 conclusions regarding the construct hierarchy in durable semantic modeling will be drawn.

2 The Modeling Constructs in NLM for Capturing the Durable Application Model

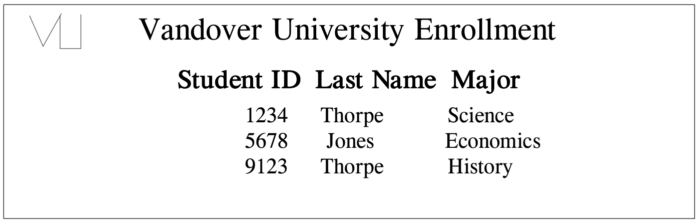
In this section, the modeling constructs for natual language modeling will be introduced. The first modeling construct is the *name.*

A name in human communication is used to refer to a concept or a thing in a real or abstract world [9]. A *name* is a sequence of words in a given language that is agreed upon to refer to *at least* one concept or thing in a real or abstract world, for example, *Jake Jones*, *567893AB*, *General Electric*. We will call the union of all names the *archetype.*

The choice of names used in communication is constrained by the reference requirement for effective communication. For example, the university registration office will use a *student ID* for referring to an individual *student.* The use of names from the name class *last name* in the university registration subject area for referring to individual students, however, will not lead to effective communication because in some cases two or more students may be referenced by *one* name instance from this name class. This is one of the reasons why not all names can be used for referencing entities, things or concepts in a specific part of a real or abstract world. On the other hand, it is evident that knowledge workers that are involved in activities in an application subject area have knowledge of the reference characteristic of the potential name classes for the different groups of "things" in a real or abstract world. This means that they should be able to tell an analyst whether a name from a specific name class can be used to identify a thing or concept among the union of similar things or similar concepts (in a specific part of a real or abstract world).

2.1 The Natural Language Axiom

In every (business) organization many examples of communication can be found. These examples can be represented on a computer screen, a worldwide web page, a computer report or even a formatted telephone conversation. Although the outward appearance of these examples might be of a different nature every time, their content can be expressed using natural language. We will refer to this class of examples of communication as *verbalizable* information.



Vandover University Enrollment

Student ID	Last Name	Major
1234	Thorpe	Science
5678	Jones	Economics
9123	Thorpe	History

Fig. 1. Example Vandover University Enrollment.

Now that we have defined the possible application areas for NLM we can start defining modeling constructs for durable application models that can take natural language sentences or facts as a starting point. In figure 1 an example of a *university enrollment document* is given. In this example the Vandover University wants to record information about the major for each of its students. It is assumed that the *student ID* can be used to identify a *specific* student among the *union* of students that are (and have been) enrolled in the Vandover University, and that a *major name* can be used as identifier for a *specific* major among the *union* of majors that are offered by the Vandover University. The application of the natural language axiom on the example of communication from figure 1 leads to facts 1.1 through 1.6.

student 1234 majors in Science………………………………………………………(fact 1.1)
student 5678 majors in Economics…………………………………………………(fact 1.2)
student 9123 majors in History……………………………………………………(fact 1.3)
student 1234 has last name Thorpe………………………………………………(fact 1.4)
student 5678 has last name Jones…………………………………………………(fact 1.5)
student 9123 has last name Thorpe………………………………………………(fact 1.6)

2.2 Variables and Roles: Both are Needed

If we analyze example facts 1.1 through 1.6 that have resulted from verbalizing the university enrollment example in figure 1, we can divide them into two groups according to the type of sentence predicate (..majors…, respectively ..has last name..). If we focus on the first group we can derive two fact group templates in which we have denoted the predicate as text, and the variable parts as text between brackets: *Student <enrolled student> majors in major <chosen major>* and *Student <enrolled student> has chosen the major <chosen major>*. We will refer to the variable parts as *roles, when we abstract from the naming convention*. However for modeling the communication we need to include the naming convention and then we speak of a variable. Figure 2 shows a graphical representation of the two fact groups in the University Enrollment example. Each role in the role representation is represented by a "box", e.g. *enrolled student*. Each abstract fact group is represented by a combination of role boxes. Fact group *Sg1* is represented by the combination of role boxes *enrolled student* and *chosen major*. Fact group *Sg2* is represented by the combination of "role" boxes *registered student* and *last name*. For each fact group one or more fact group templates are positioned underneath the combination of role boxes that belong to the fact group. In the diagram of figure 2, factgroup templates 1 and 2 belong to fact group *Sg1*. Fact group template 3 belongs to fact group *Sg2*.

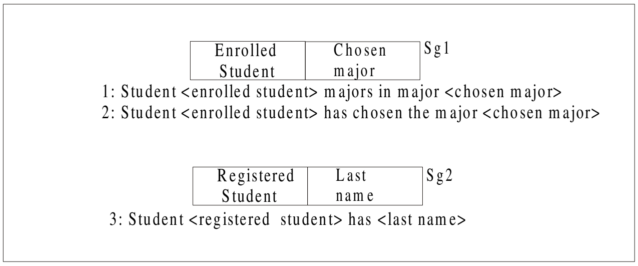

Enrolled Student
Chosen major
Sg1
1: Student <enrolled student> majors in major <chosen major>
2: Student <enrolled student> has chosen the major <chosen major>
Registered Student
Last name
Sg2
3: Student <registered student> has <last name>

Fig. 2. Roles, fact group and fact, group template(s) for university enrollment

If we examine figure 2 we will see that a fact group template can reveal additional information about the type of things that can be "inserted" into a *role*. For example, the word "student" specifies what type of thing (or concept) is allowed to play the role "enrolled student" but also what type of thing (or concept) is allowed to play the role "registered student". We will call the "student" part in the fact groups in figure 2 the *intension* of the roles "enrolled student" and "registered student". We will now illustrate how such a list of terms or concept definitions can help us capture a durable *application-* or *domain model*.

To make a durable application model explicit, we need to incorporate a definition of the concepts or pronouns in the fact group templates including the intensions. For example, the definition of the concept *Student*: *A student is a person that studies at a University*. The names of things or concept instances to which such a definition of intension applies within a *specific* application subject area at a *specific* point in time is called the *extension*. We can now give an example extension for the intension *Student*: *{1234, 5678, 9123}*. In the remainder of this paper we will use the term *intension* to denote the *type* of thing or concept to which a *specific* thing or concept belongs. For every application area we will document the relevant concepts and their definitions in a list. In the following illustration we have given an example of such a list for the university enrollment UoD (see table 1).

Table 1. List of definitions for university enrollment example

Concept	***Definition***
Student	*a person that studies at Vandover University*
Student ID	*an ID that identifies a specific student at Vandover University*
Major	*a course program offered to <student>s by Vandover University*
Major name	*a name class , instances of which can be used to identify a <major> among the union of <major>s offered by Vandover University*
Last name	*a characteristic of a student, namely his or her last name*

Such a list of concepts and their definitions should contain a definition for *each* intension in the UoD. The definition of an intension should specify how the knowledge forming the intension (*definiendum*) is to be constructed from the knowledge given in the definition itself and in the defining concepts (*definiens*). A defining concept should either be an intension or a different concept that must be previously defined in the list of concepts or it should be defined in a generic ontology. In the Vandover University student enrollment example the concept *course program* is considered to be defined in a durable application model for university education which implies that all 'agents' that are involved in this UoD have attached the same meaning to this concept.

The construct for modeling the meaning of a a surface structure sentence is a *fact instance*. A fact instance is expressed as a natural language sentence instance of a one of its corresponding *sentence group template(s)*. It is possible that the extensions of two different sentence group templates refer to the same *fact*. For example we can say that there exists a *fact* that a student is enrolled in a major (at the university of Vandover). Two sentence instances (from different sentence group templates) for communicating this *fact instance* can be:

Student 1234 has chosen the major Science
Student 1234 majors in major Science.

We need to make a distinction into the concept of *fact* (the deep structure sentence) and the concepts that we use to *represent a fact (surface structure sentence)*. A *specific* fact instance can be represented as one or more sentence instances from one or more sentence group templates. We can now conclude that an abstract fact type is a set of roles that can be represented by one or more sentence group template(s) in which these roles are contained. The roles that are contained in these sentence group templates are *identical* although they can have a different sequence.

2.3 Naming Convention Fact Types

In this section we will further formalize the outcome of the process of the selection of a name class for referring to things in a real or abstract world. The outcome of such a naming process will result in the utterance of sentences or facts for example facts 2.1, 2.2, 2.3 and 2.4.

1234 is a name from the student ID name class that can be used to identify a student within the union of students at Vandover University…………………………….(fact 2.1)

5678 is a name from the student ID name class that can be used to identify a student within the union of students at Vandover University……………………… …….(fact 2.2)
Science is a name from the major name name class that can be used to identify a major within the union of majors at Vandover University………………… …(fact 2.3)
Economics is a name from the major name name class that can be used to identify a major within the union of majors at Vandover University …………………….(fact 2.4)

We see that facts 2.1 through 2.4 express that a certain *name* belongs to a certain *name class* and that instances of the name class *student ID*, can be used to identify an instance of a *student,* and an instance of the name class *major name*, can be used to identify an instance of a *major* within the UoD of Vandover University. We can give, for example, the definition of the concept *Student ID*: *Student ID is a name class.* The 'intension' of the names in fact sentences 2.1 through 2.4 is a *name class* and NOT a type of *thing, entity* or *concept* in the real world. We will, therefore, refer to facts 2.1, 2.2, 2.3 and 2.4 as *naming convention facts*. The corresponding fact type will then be called a *naming convention fact type.*

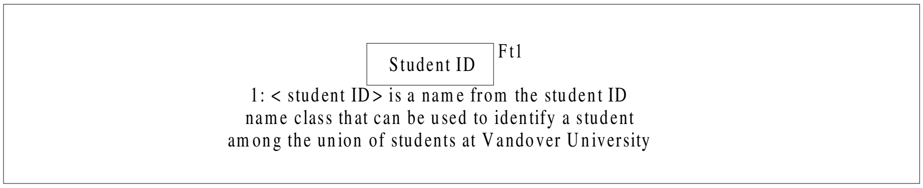

Student ID
Ft1
1: < student ID> is a name from the student ID name class that can be used to identify a student among the union of students at Vandover University

Fig. 3. Naming convention fact type for *Student*

A role representation is the preferred one if one is interested in formulating rules; a variable representation is the preferred one if one is interested in performing validation with realistic examples.

2.4 Compound Reference Schemes

In the Vandover University example the intension student has a "simple" reference scheme, namely: the single role "enrolled student" or "registered student". In many cases, however, a *simple* reference scheme will not be sufficient for referencing instances of a given intension as many people prefer to agree on names within a selected context, like city within a country, or street within a village, within a municipality within a province within a country. In those cases we will need *compound* reference schemes.

We can apply compound reference schemes in NLM in the same way as the simple reference schemes. To illustrate this we will first adapt our example UoD. We will assume that Vandover University has merged with Ohao University. In order to streamline the enrollment operations, it is decided to centralize them. This means that after the merger, a student can no longer be identified by the existing student ID because a given *student ID* can refer to a student in the (former) Ohao University, *and* to a different student in the (former) Vandover university. To capitalize on the existing naming conventions it is decided to add the qualification *O* (for Ohao) or *V* (for Vandover) to the existing student ID. This extension is the *university code.* The sentence group templates and the corresponding fact types in which such a compound reference scheme is implemented are given in figure 4.

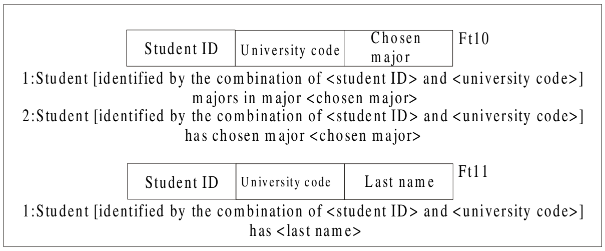

Student ID
University code
Chosen major
Ft10
1:Student [identified by the combination of <student ID> and <university code>] majors in major <chosen major>
2:Student [identified by the combination of <student ID> and <university code>] has chosen major <chosen major>
Student ID
University code
Last name
Ft11
1:Student [identified by the combination of <student ID> and <university code>] has <last name>

Fig. 4. Fact types and sentence group templates with compound reference scheme for student

We have introduced the [ ] ('brackets') symbol for capturing the definition of the compound reference scheme (see figure 4). For example, the reference scheme for student in fact type Ft10 consists of the roles *student ID* and *university code* and is defined as follows: *Student [identified by the combination of <student ID> and <university code>].*

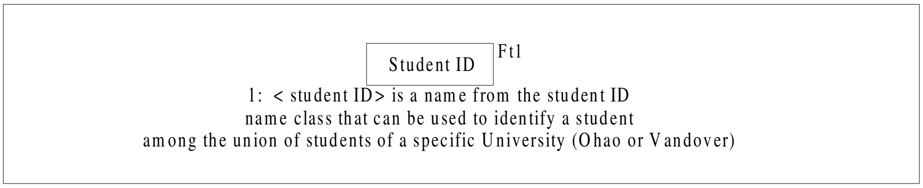

Student ID
Ft1
1: < student ID> is a name from the student ID name class that can be used to identify a student among the union of students of a specific University (Ohao or Vandover)

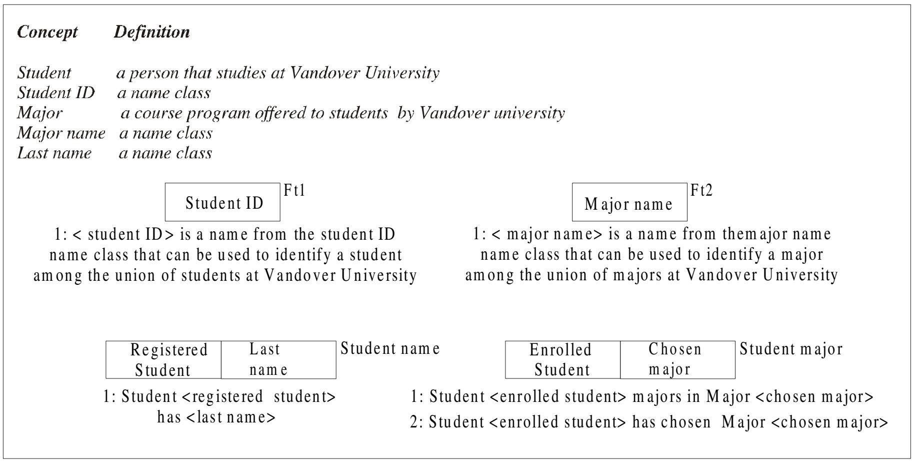
Fig. 5. Naming convention fact type for student in the integrated UoD

The case of a simple reference scheme is actually a special case of the compound reference scheme in which the brackets and description within (except for the role name used in the reference) are left out. In addition to this we need to adapt the naming convention fact types for the constituting intensions of the compound reference scheme. For example, the naming convention fact type for *student* should be adapted to reflect the application subject area in which it can be used to identify a specific student. In this case a student can be identified by his/her student ID within a *specific* University (Ohao or Vandover).

The unification of simple reference schemes and the different types of compound reference schemes into one uniform way of referencing, and the capability to capture the precise semantics of naming conventions are improvements in NLM to the predecessor methodologies.

2.5 The Basic Durable Application Model

A *basic durable application model* (DAM) for a Universe of Discourse *U* is defined by a list of concepts and their definitions applicable to that UoD, a set of roles, a set of variables, a set of fact types, and a set of sentence group templates for every fact type. The *extension* of a DAM his the union of the extensions of the fact types that are contained in that basic durable application model.



Concept	Definition
Student	a person that studies at Vandover University
Student ID	a name class
Major	a course program offered to students by Vandover university
Major name	a name class
Last name	a name class

Student ID
Ft1
1: < student ID> is a name from the student ID name class that can be used to identify a student among the union of students at Vandover University
Major name
Ft2
1: < major name> is a name from themajor name name class that can be used to identify a major among the union of majors at Vandover University
Registered Student
Last name
Student name
1: Student <registered student> has <last name>
Enrolled Student
Chosen major
Student major
1: Student <enrolled student> majors in Major <chosen major>
2: Student <enrolled student> has chosen Major <chosen major>

Fig. 6. Basic durable application model for University enrollment example

3 The NLM Representation Ontology

In this section we will narrow down the “real” or “abstract” world of interest (Universe of Discourse) to the UoD of NLM application ontology modeling or the NLM representation ontology [4]. First, the list of concepts for NLM representation ontology is provided (see table 2).

Table 2. List of concepts for NLM’s durable application modeling language

Concept	*Definition*
Sentence (instance)	*The result of the verbalization and qualification of a piece of verbalizable information by a domain user*
Sentence group	*The variable abstraction of a set of < sentence>s*
Sentence group template	*The ordering of fixed and variable parts of a group of < sentence>s that reflect domain semantics*
Role	*a variable part in one or more <sentence group template>s*
Role code	*a name class, instances of which can be used to identify a <role> among the union of <roles> in the application ontology*
Intension	*the meaning of a concept in a real or abstract world*
Intension name	*a name class, instances of which can be used to identify a <intension> among the union of <intensions> in the application ontology*
Extension(of intension)	*the set of names of things or concepts to which the definition of its < intension> applies*
Verb	*the parts of a < sentence group template> that are not variable*
Fact type	*a set of < role>s*
Fact type code	*a name class, instances of which can be used to identify a <fact type> among the union of <fact types> in the application ontology*
Basic information model	*a list of concepts and their definitions, a set of <role>s, a set of <fact type>s and a set of < sentence group template>s for each <fact type>*
Extension (of fact type)	*a set of < sentence instances> for that <fact type>*

4 Conclusion: A Hierarchy in the NLM Durable Semantic Modeling Constructs

As illustrated in figure 7, we can put the concepts of *archetype*, *name class, naming convention fact type* and *non-naming convention fact type* in a *hierarchical* perspective. In every application ontology (including NLM' durable semantic conceptual modelling language) we can distinguish three segments in an application ontology. The *top* segment consists of the world of names in which the *archetype,* (and, when applicable, the relevant *scale types)* and the *relevant name classes* for the application ontology are defined. In the *middle* segment we find the naming convention fact types for the intensions that are relevant for the application subject area. This hierarchy replaces the distinction between *concepts* and *names* by considering names as populations of fact types in the hierarchy. The verbs in the fact types then will explicitly show whether a fact type declares the existence of a concept, the existence of a name that can be used to identify a concept or a semantic relationship among concepts. The application of NLM for capturing an application's ontology will improve the organization's resilience, because changes, in products, services, product-life cycle, customers, regulations and other external conditions can be easily adapted in the application's ontology, by adding or redefining a concept and naming convention and/or adding or deleting a semantic relationship.

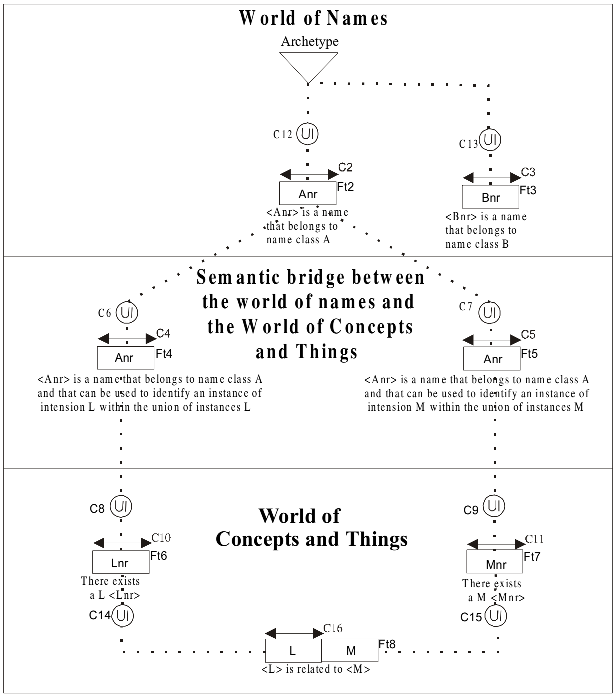

World of Names
Archetype
C12 UI
C13 UI
C2
Ft2
Anr
<Anr> is a name that belongs to name class A
C3
Ft3
Bnr
<Bnr> is a name that belongs to name class B
Semantic bridge between the world of names and the World of Concepts and Things
C6 UI
C4
Anr
Ft4
<Anr> is a name that belongs to name class A and that can be used to identify an instance of intension L within the union of instances L
C7 UI
C5
Anr
Ft5
<Anr> is a name that belongs to name class A and that can be used to identify an instance of intension M within the union of instances M
World of Concepts and Things
C8 UI
C10
Lnr
Ft6
There exists a L <Lnr>
C14 UI
C9 UI
C11
Mnr
Ft7
There exists a M <Mnr>
C15 UI
C16
L
M
Ft8
<L> is related to <M>

Fig. 7. Hierarchy in semantic conceptual modelling constructs.

References

1. Gruber, T.: A translation approach to portable ontologies. Knowledge Acquisition **5**(2), 199–220 (1993)
2. Fensel, D.: Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer verlag (2001)
3. Lammari, N., Metais, E.: Building and maintaining ontologies: a set of algorithms. Data & Knowledge Engineering **48**, 155–176 (2004)
4. Burton-Jones, A., Storey, V., Sugumaran, V., Ahluwalia, P.: A semiotic metrics suite for assessing the quality of ontologies. Data & Knowledge Engineering **55**(1), 84–102 (2005)
5. Nijssen, G.: A Framework for discussion in ISO/TC97/SC5/WG3 and comments on 78.04/01 and 78.05/03 (1978)
6. Chen, P.: The entity-relationship model: Towards a unified view of data. ACM Transactions on Database Systems **1**(1), 9–36 (1976)
7. Teory, T., Yang, D., Fry, J.: A logical design methodology for relational databases using the extended E-R model. *ACM Computing Surveys* **18**(2) (1986)
8. Markowitz, V., Shoshani, A.: Representing extended entity-relationship structures in relational databases: A modular approach. ACM Transactions on Database Systems **17**(3), 423–464 (1992)
9. Senko, M.: DIAM as a detailed example of the ANSI/SPARC architecture. In: Nijssen, G. (ed.), *Modeling in Database Management Systems*, North-Holland (1976)