

# Towards a Hierarchy in Domain Ontologies

Peter Bollen

Department of Organization and Strategy  
School of Business and Economics  
Maastricht University  
P.O. Box 616  
6200 MD Maastricht, The Netherlands  
p.bollen@maastrichtuniversity.nl

**Abstract.** This paper defines a language for modeling ontologies in business applications and application services. This language consists of the modeling constructs from Natural Language Modeling (NLM). It will be shown in this article how the application of this modeling language will enable us to model a hierarchy for a domain ontology.

## 1 Introduction

As the number of information services (e.g. web-services, application service providers and e-commerce applications) continues to grow and the need for interoperability between those services becomes apparent (i.e. the semantic web), the availability of a modeling language that enables analysts to capture the precise semantics of such an application information service will become a necessity. Such a language should not only be able to capture the application semantics for the user-group that is the provider of the information service, but it should also contain modeling provisions that enable comprehension by the potential client groups of such an information service (e.g. autonomous agents). The language; therefore; must be able to establish links with common business ontologies and should be easy to understand by non-expert users. Such a language must be very close to the natural language that is used by people in their daily communications.

We will define ontology in this paper as “the definition of the basic terms and relations comprising the vocabulary of a topic area”[1], or according to Gruber [2]: “an anthology is a description of the concepts and relationships for an agent or a community of agents.” We will, therefore, provide modeling constructs in this paper that can be used within the context of a specific universe of discourse in which in principle different users are involved. The information 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* [3]. It means that it is *neither* a real *nor* an abstract world that is subject to modeling, but the *communication about* such a real or abstract world. This will constrain the feasible modeling constructs to those constructs that enable analysts to model natural language sentences.

## 2 The Modeling Constructs in NLM for the Information Structure

A name in human communication is used to refer to a concept or a thing in a real or abstract world [4]. 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.

### 2.1 The Natural Language Axiom

In every (business) organization examples of communication can be found. These examples can be materialized as 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 [5].

VU Vandover University Enrollment		
Student ID	Last Name	Major
1234	Thorpe	Science
5678	Jones	Economics
9123	Thorpe	History

**Fig. 1.** Example Vandover University Enrollment

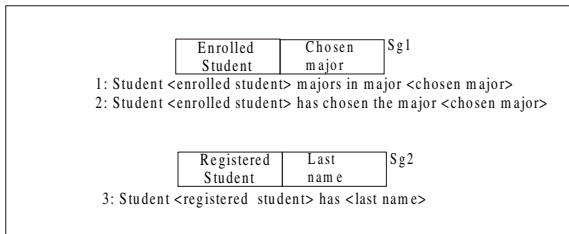
Now that we have defined the possible application areas for NLM we can start defining modeling constructs that can take natural language sentences as a starting point. In figure 1 an example of a *university enrollment document* is given. In this example the Vandover University wants to record information 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 can lead to for example following sentence instances.

*The student 1234 majors in Science.....(sentence 1.1)*

*The student 5678 majors in Economics.....(sentence 1.2)*

## 2.2 Roles

If we analyze example sentences 1.1 and 1.2 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 sentence 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*. Figure 2 shows a graphical representation of the two sentence groups in the University Enrollment example. Each role is represented by a “box”, e.g. *enrolled student*. Each sentence group is represented by a combination of role boxes. Sentence group *Sg1* is represented by the combination of role boxes *enrolled student* and *chosen major*. Sentence group *Sg2* is represented by the combination of “role” boxes *registered student* and *last name*. For each sentence group one or more sentence group templates are positioned underneath the combination of role boxes that belong to the sentence group. In the diagram of figure 2, sentence group templates 1 and 2 belong to sentence group *Sg1*. Sentence group template 3 belongs to sentence group *Sg2*.



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

If we inspect figure 2 we will see that a sentence group template can reveal additional information about the type of things that can be “inserted” into a *role* variable. 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 sentence groups in figure 2 the *intension* of the roles “enrolled student” and “registered student”.

To make an application’s ontology explicit, we need to incorporate a definition of the concepts in the sentence 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, 912). 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. In the following illustration we have given an example of such a list for the university enrollment UoD.

<b>Concept</b>	<b>Definition</b>
<i>Student</i>	<i>a person that studies at Vandover University</i>
<i>Student ID</i>	<i>a name class</i>
<i>Major</i>	<i>a course program offered to [student]s by Vandover University</i>
<i>Major name</i>	<i>a name class</i>
<i>Last name</i>	<i>a name class</i>

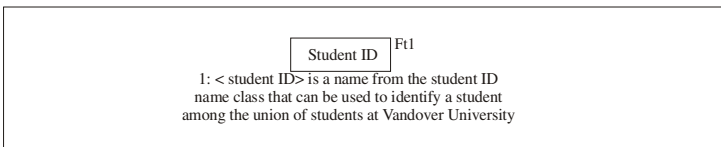
Such a list of concepts and their definitions should contain a definition for *each* intension in the UoD. The definition of an intension should specify how the knowledge forming the intension (*definiendum*) is to be constructed from the knowledge given in the definition itself and in the defining concepts (*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 common business ontology.

### 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, for example sentence 2.1

*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.....(sentence 2.1)*

We see that sentences 2.1 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 sentence 2.1 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 as a *naming convention facts*. The corresponding fact type will then be called a *naming convention fact type*.



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

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.

### 2.4 The Basic Information Model

A *basic information model* (BIM) 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 fact types, and a set of sentence group templates for every fact type. The *extension* of a BIM is the union of the extensions of the fact types that are contained in that basic information model (see figure 4).

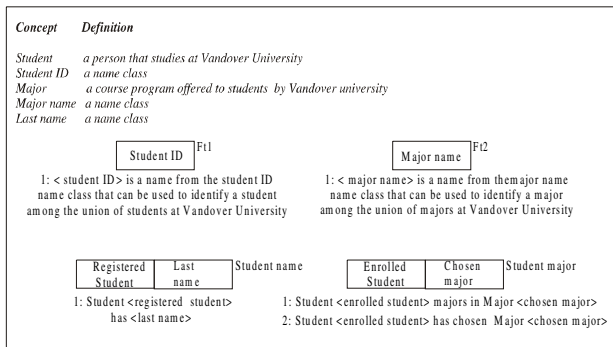


Fig. 4. Basic information model University enrollment

## 3 Population Constraints

In this section we will introduce the modeling constructs that will make it possible to explicitly show that some extensions of a *basic information model* are **not** allowed to exist. In order to make a distinction between an extension of a basic information model (regardless of the fact whether it is allowed to exist), and an extension of a basic information model that is allowed to exist, we will introduce the concept of *population state*. A *population state* is an extension of a basic information model that is allowed to exist.

We can consider a *population state* as a further reduction of the extensions in the set of *possible* extensions of a basic information model. After the restriction of the names to the name classes that can be used to identify a specific *thing*, *entity* or *concept* in the application UoD, we will further restrict the extensions that are allowed to exist by incorporating specific *domain knowledge* or those *domain rules* (sometimes called *business rules*) that can be expressed as propositions on the basic information model that must be true for *every* population state. We will call such a proposition a *population state constraint*. A *population state constraint*  $p$  in a basic information model  $BIM$  limits the allowed extensions of the basic information model  $BIM$  to those extensions that comply to the proposition specified in the population state constraint  $p$ .

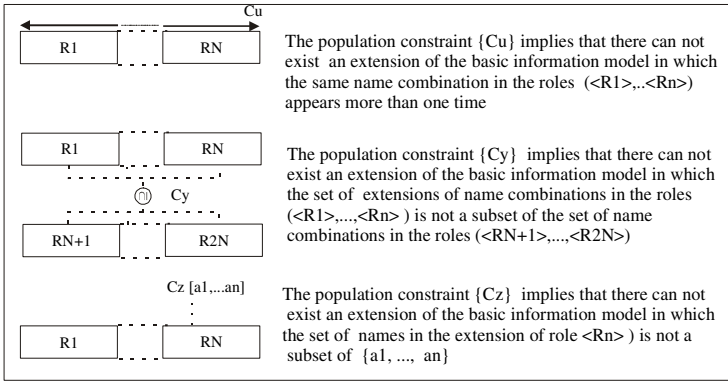


Fig. 5. Legend for uniqueness, subset and value population state constraints

The business rule: a student can be enrolled in at most one major, can be expressed as the following constraint instance from the constraint legend in figure 5: The population constraint c7 implies that there can not exist an extension of the basic information model in which the same name in the role **enrolled student** appears more than one time. If we check this example thoroughly we can conclude that the addition of a population state constraint onto a (basic) information model actually eliminates those extensions from the set of extensions that do **not** comply to the proposition. In this example we can see that the example extension : {student 1234 majors in major science, student 1234 majors in major economics} does **not** comply to the proposition of population state constraint c7.

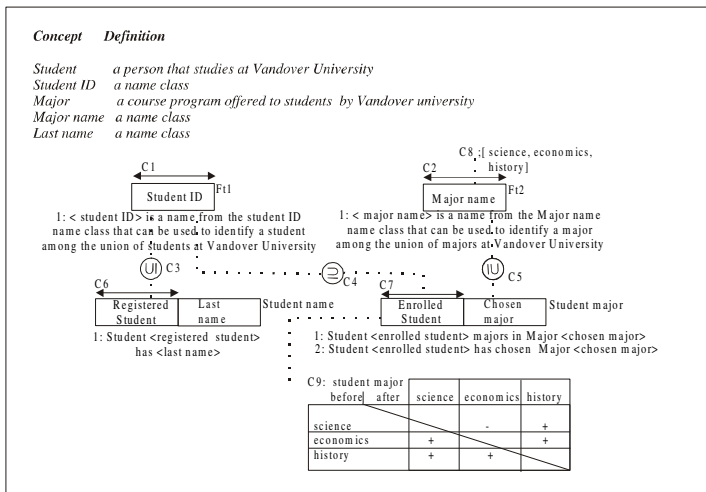


Fig. 6. Information model for university enrollment example

We can now define an information model. An information model referring to a universe of discourse is a *basic information model* for that UoD together with all

*population constraints* that reflect the business rules in that UoD and that can be defined on the roles of the basic information model for that UoD. In figure 6 the resulting information model for the student enrollment UoD application area is shown. The legend for the interpretation of the population state constraint symbols that are used in figure 6 is given in figure 5.

#### 4 Integration of the NLM Information Model and the Application Ontology

We can put the concepts of *archetype*, *name class*, *naming convention fact type* and *non-naming convention fact type* in a *hierarchy*. In every application subject area (including the analyst's subject area) we can distinguish three segments in an information model.

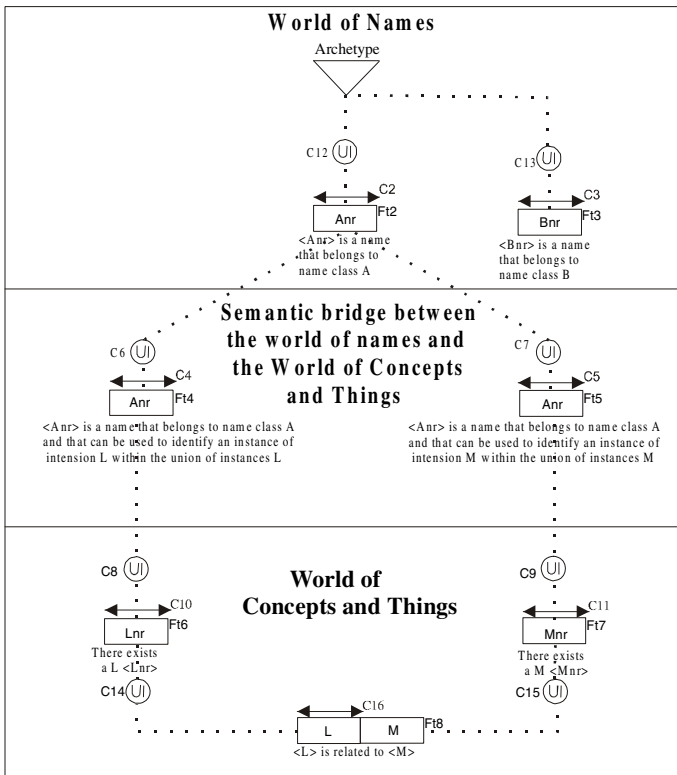


Fig. 7. Hierarchy in information models

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 subject area are defined. In the *middle* segment we find the naming

convention fact types for the intensions that are relevant for the application subject area. How this ‘semantic bridge’ is created should be laid down precisely in an information modeling procedure. The *lowest* segment in the application information grammar consists of the application fact types that can be derived whenever the verbalization axiom is applied to an application subject area.

## 5 Conclusion

In this paper we have defined the modeling constructs in *natural language modeling*. The defined constructs will enable us to define a hierarchy in every information model. 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. In case the Universe of Discourse refers to an application service environment in which different (partly unknown) user groups are involved, the list of concept definitions serves as a semantic connection between the provider of the application service and the different user- “agents” that do not necessarily share the same ontological background. NLM enables an application analyst to create multiple lists of concept definitions, if necessary, to share the application’s semantics with users having a different ontological background.

## References

1. Neches, R., et al.: Enabling technology for knowledge sharing. *AI Magazine*, 36–56 (Fall 1991)
2. Gruber, T.: A translation approach to portable ontologies. *Knowledge Acquisition* 5(2), 199–220 (1993)
3. Nijssen, G.: An axiom and architecture for information systems. In: *Information systems concepts: an in-depth analysis* (1989)
4. Senko, M.: DIAM as a detailed example of the ANSI/SPARC architecture. In: Nijssen, G. (ed.) *Modelling in Database Management Systems*. North-Holland (1976)
5. Nijssen, G.: On experience with Large-Scale Teaching and Use of fact-based Conceptual Schema’s in Industry and University. In: Meersman, R., Steel, J. (eds.) *IFIP Conference on Data Semantics (DS-1)*, pp. 189–204. North-Holland (1986)