Extending the ORM conceptual schema design procedure with the capturing of the domain ontology

Peter Bollen

Department of Organization and Strategy
Faculty of Economics and Business Administration
University of Maastricht

6200 MD Maastricht, the Netherlands
p.bollen@os.unimaas.nl

Abstract. In this paper we will extend the ORM conceptual modeling language with constructs for capturing the relevant parts of an application ontology in a list of concept definitions. In addition we give the adapted ORM meta model and an adaptation of the accompanying Conceptual Schema Design procedure (CSDP) to cater for the explicit modeling of the relevant parts of an application- or domain ontology in a list of concept definitions.

Keywords: Modeling languages, method engineering, meta-modeling

1 Introduction

A conceptual modeling methodology is a combination of a set of conceptual modeling constructs and an accompanying (set of) procedure(s) that 'prescribe' how these conceptual modeling constructs must be applied in practice. ORM is a conceptual modeling approach that models the world in terms of objects and roles that they play [1]. In the literature a number of definitions for Ontology can be found: "the definition of the basic terms and relations comprising the vocabulary of a topic area" [2], " an ontology is a description of the concepts and relationships for an agent or a community of agents." [3], " shared understanding of a domain that can be communicated between people and application systems." [4], "an ontology is a formal conceptualization of a real world, sharing a common understanding of this real world." [5, p.155]. Burton-Jones et al. [6] distinguish four types of material ontologies: *application*[1], *domain*[2], *generic*[3] and *representation*[4] ontologies.

In this paper we will extend the ORM conceptual modeling methodology and its *representation* ontology with additional modeling constructs to help us capture the relevant part of a *domain*- or *application ontology*.

[1] Application ontologies specify definitions needed for a particular application [6].
[2] Domain ontologies specify conceptualizations specific to a domain [6]
[3] Generic ontologies specify conceptualizations generic to several domains [6].
[4] Representation ontologies specify conceptualizations that underlie knowledge representation formalisms [6].

1.1 Related work

Weber and Zhang [7] analyze the extent in which a pre-decessor to ORM: NIAM complies to the Bunge-Wand-Weber (BWW) model for ontological expressiveness. Spyns, Meersman and Jarrar [8] introduce the DOGMA ontology engineering approach in which they apply ORM and where they separate an ontology base from the ontological commitment that contains the specific rules of the application domain. Dumas et al. [9] give an approach for the ontology markup in the context of the generation of web forms. Their approach, however, is limited to tasks that do not involve complex reasoning, and they claim that the rules that can be captured in a conceptual modeling language such as ORM or UML suffice for capturing the ontology. De Troyer et al. [10] state that semantic information is needed to solve possible conflicts between different ORM model chunks. They define 'reusable' concepts for designers, e.g. object concepts, relationships and tuples, the definition of the concepts, however is left to the designers: ' ..explaining in an informal way the meaning of the concept. Fliedel et al.[11] discuss a web-based glossary editor in which a description property is defined.

The rest of this article is organized as follows: In section 2 we will introduce the list of concept definitions and we will show how such a list of concept definitions can be used to augment a conceptual schema with an *application*- or *domain*- ontology [6]. In section 3 we will show how the modeling construct of the subtype can be partially contained in *the list of concept definitions* and how the list of concept definitions can be used to capture the precise semantics of naming conventions. In section 4 we will give a significant part of the augmented ORM meta model including the accompanying *representation* ontology. Finally, in section 5, conclusions will be given.

2 The list of concept definitions

Brasethvik and Gulla [12] give an approach for semantic document classification and retrieval. During the process of constructing domain models from document collections, terms are selected from the proposed list and defined in cooperation with domain users and other stakeholders from the domain [12, p. 50-51]. In the next section we will illustrate how such a list of terms or concept definitions can help us capture an *application*- or *domain* ontology in the conceptual modeling process and how it can be integrated with an existing ORM conceptual schema. We will use as running example the University Enrollment UoD (see section 2.1).

2.1 Example 1: University Enrollment part 1

The University Enrollment example will be used to illustrate the modeling concepts throughout this article. The University of Vandover offers a number of majors in education. Students can choose between majors in *Science, History* and *Economics* among others.

In figure 1 an example is given of a university enrollment document (example 1).

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 set of majors that are offered by the Vandover University.

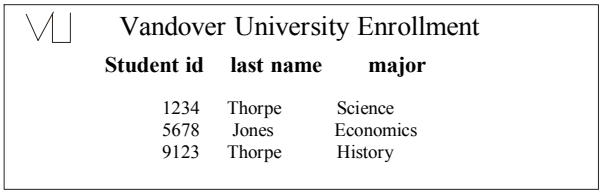


Fig. 1. Example Vandover University Enrollment (example 1).

In figure 2 we have given the standard ORM conceptual schema for the Vandover University enrollment UoD.

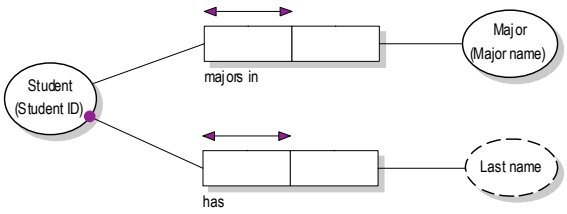


Fig. 2. Standard ORM conceptual schema for University Enrollment (example 1).

2.2 Adding the list of concept and definitions to the ORM conceptualh schema

If we examine the standard ORM conceptual schema for the University enrollment application subject area in figure 2 we notice that the definition of the vocabulary of the University Enrollment topic area is missing. To make the ontology of the application explicit, we need to incorporate a definition of the concepts in an ORM conceptual schema including the Entity Types. For example, the definition of the concepts *Student* and *Major.*

Student: *A student is a person that studies at Vandover University.*
Major: A major is a course program offered to students by Vandover University

The definition of the concept types in the list of concept definitions must specify how the knowledge forming the concept (*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 a different concept that must be previously defined in the list of concepts or it should be defined in a common business ontology. Brasethvik and Gulla use such a list of concept definitions in the context of a 'shared' or 'common information space' in which the semantics of information is locally constructed and '.. reflects the 'shared agreement' on the meaning of the information' [12, p.47]. To capture the relevant ontology of an application subject area in addition

to the conceptual schema we will document the relevant concepts and their definitions in a list (the list of concept definitions). In figure 3 we have given an example of such a list of concept definitions for the university enrollment UoD.

Concept	***Definition***
Student	*a person that studies at Vandover University*
Student ID	*a name class, instances of which can be used to identify a <Student> among the union of <students> that have ever been, are or will be enrolled at Vandover University*
Major	*a course program offered to <students> by Vandover University*
Major name	*a name class, instances of which can be used to identify a <major> among the union of <majors> offered by Vandover University*
Last name	*a name class*

Fig. 3. List of concept definitions for university enrollment application (example 1).

With respect to the fact type predicates in ORM it is recommended to add (parts of) them to the list of concept definitions in case the meaning differs from what is generally understood in business domains. For example, the predicate *..majors in* .. from the ORM conceptual schema that is given in figure 2 is considered to have clear semantics to all users in our example domain. On the other hand a predicate that might refer to a specific inter-university agreement in terms of master student's exchange between the Vandover University and for example the University of Uganda might read: *....participates in the Vandover-Uganda intercultural govenrnance program track.* In that case we can add (parts of) the predicate to the list of definitions. A second group of concepts that is candidate for incorporation into the list of concept definitions is formed by the ORM entity types (including subtypes and objectifications). We recommend to incorporate all of them into the list of concept definitions of the application UoD (see section 3). For *each* Entity type in the UoD, we must incorporate the name class(es) to be used as references for entity types (*Student ID*, *Major name*). The definition of a name class that can be used to identify an entity type must be given precisely, thereby explicitly focusing on the context in which the instances of a name class can be considered as identifiers for the defined entity type. For example the definition of the name class *student ID* in figure 3 contains a description of the context in which the Student ID can be used as an identifier for a student: only those students that have ever been, are or will be studying at Vandover University. Furthermore, we must incorporate name classes that are merely value types (for example *last name*).

In the Vandover University enrollment example the concept (in this example an entity type) *course program* is considered to be defined in a common business ontology for university education which implies that all 'agents' that are involved in this UoD have attached the same meaning to this concept. We remark that a list of concepts and definitions should be presented in order of comprehension [13, p.285]. This means that when reading the list from top to bottom, no forward 'jumps' need to be carried out to acquire the meaning of one or more *definiens* for a given *definiendum.* To facilitate the comprehension of the list of definitions we have put the

concepts in the definition that are definienda themselves between place-holder brackets (‘< >’).

3 Relationship between concept definition and subtyping in ORM

Entity types that are generally related to each other by means of super-sub type relationships in ORM can alternatively be captured in a list of definitions. In this section we will illustrate what this means for the existing modeling conventions in ORM regarding super/sub types or specialization and/or generalization. To illustrate these modeling issues we will extend our University enrollment example.

3.1 Example 2: University Enrollment part 2

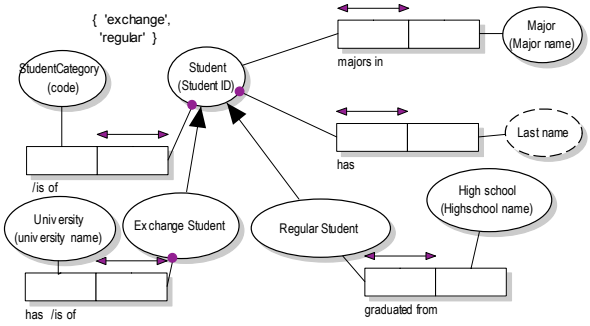
We will now extend our UoD with a second ‘real-life’ example of communication. This example contains additional background information on some students. The extended UoD is given in figure 4.

Additional student information

Student ID	University of origin	High school
1234	Harvard	--
5678	--	St. Paul new York

Fig. 4. Additional Example Vandover University Enrollment.

In the extended Universe of Discourse (examples 1 and 2 in combination) we see that for some students we will record only their University of origin. These students are generally known as exchange students. For some of the other students we will record the name of the high school from which they have graduated.

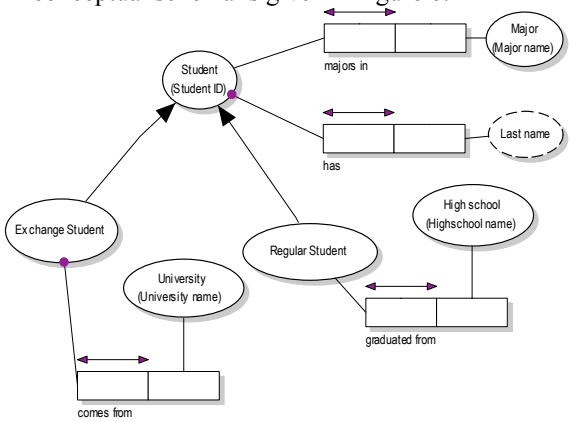


each Exchange Student **is a** student **who** has Student category ‘exchange’
each Regular Student **is a** student **who** has Student category ‘regular’

Fig. 5. ORM conceptual schema for extended University Enrollment examples .

For another group of *non-exchange* students no specific details will be recorded. The non-exchange students are are referred to as *regular* students. In figure 5 we have given the ORM conceptual schema for this UoD including the subtypes and a subtype defining fact type [1, p.253]. If we carefully inspect the ORM conceptual schema in figure 5 together with the user example from figure 4 we see that the examples on which this conceptual schema is based do *not* contain an explicit recording of the *Student Category*. It is during the CSDP step 1 that the verbalization of an example leads to natural language sentences that subsequently can be abstracted into (elementary) fact types. The addition of the *student category* fact type in the ORM conceptual schema in figure 5 is therefore not justified from the point of view of the application UoD because no explicit recording of the student type can be traced to the user examples.

We will now illustrate how the incorporation of a list of concept definitions will allow us to capture intentional subtype definitions in the list of definitions. This augmented ORM conceptual schema is given in figure 6.



Concept	***Definition***
Student	*a person that studies at Vandover University*
Student ID	*a name class, instances of which can be used to identify a <student> among the Union of <student>s that have ever been, are or will be enrolled at Vandover University*
Regular Student	*a <student> who has started his/her academic career on Vandover university*
Major	*a course program offered to students 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 name class*
Highschool	*a place where a form of secondary education is provided*
Highschool name	*a name class, instances of which can be used to identify a <highschool> among the union of <highschool>s in the world*
University	*a place where a form of tertiary education is provided.*
University name	*a name class, instances of which can be used to identify a <university> among the union of <universities> in the world*
Exchange student	*a <student> who has started his/her studies at another University than Vandover University but takes a <major> at Vandover University*
Graduated from	*the process of finishing a high school education*

Fig. 6. Augmented ORM conceptual schema for extended University Enrollment example

3.2 Definition of naming conventions in ORM

Another reason for incorporating a list of definitions into an ORM conceptual schema is the necessity of an explicit definition of the context in which a naming convention holds. In current ORM conceptual schemas it can only be recorded which name class can be used to identify an instance of an entity type in general. Even in the explicit recording of a ORM referencing scheme it can only be captured that an entity type has (exactly) one instance of a value type. What is not captured in such a mandatory 1:1 reference type in ORM (Halpin, 2001:186-189) is a description of the context in which a name class (e.g *Student ID, University name, High school name*) can be used to identify instances of a concept type (e.g. *Student, University, High school*). In figure 6 this is illustrated in the list of concept definitions for the name class *student ID* and the name class *major name*. If we study the definitions for the latter concepts we see that we have incorporated the explicit context under which the names from the name class can be used as identifiers for entities of the entity types *Student* and *Major*. A further reason to incorporate a precise definition of the semantics of a naming convention will be illustrated by an example in which one or more entity types need a compound reference type. We will illustrate the necessity of recording these explicit naming convention semantics by an extension of our running example.

3.3 Example 3: University Enrollment part 3

We 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 a student after the merger can no longer be identified by the existing student ID because a given *student ID* can refer to a student in the former Ohoa 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.* In figure 7 we have given the adapted ORM conceptual schema including a compound reference scheme [1, p.189] and an adapted list of definitions in which we have now incorporated the precise definition of the naming conventions and the conditions under which a name class is valid.

4 The meta model for the augmented ORM

In this section we will give a (segment of an) ORM meta model in which we have incorporated the list of concept definitions that allows us to capture the ORM representation ontology [6].

We will use one of the two the ORM meta models (model A) that were presented in [14]. One of the distinctive features of this meta-model is that it allows for inactive sub-types. Furthermore it enforces a subtype definition for each distinguished subtype. In our augmented meta-model we will relax the requirement in which it is stated that each subtype must have a subtype-definition in terms of a definition that is defined on a subtype defining fact type. We will furthermore remove inactive

subtypes from the conceptual schema but we will add their definitions in the list of concepts and definitions.

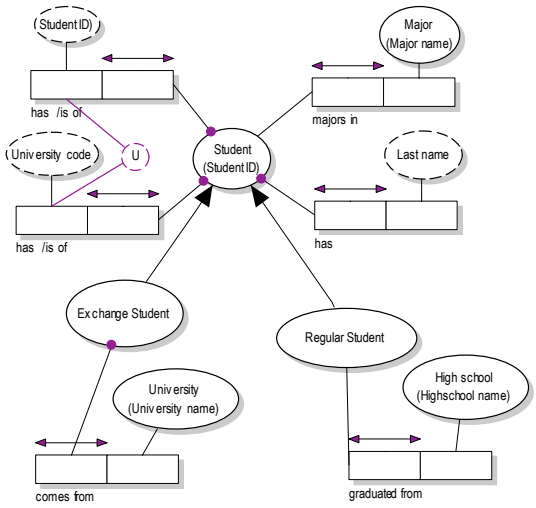
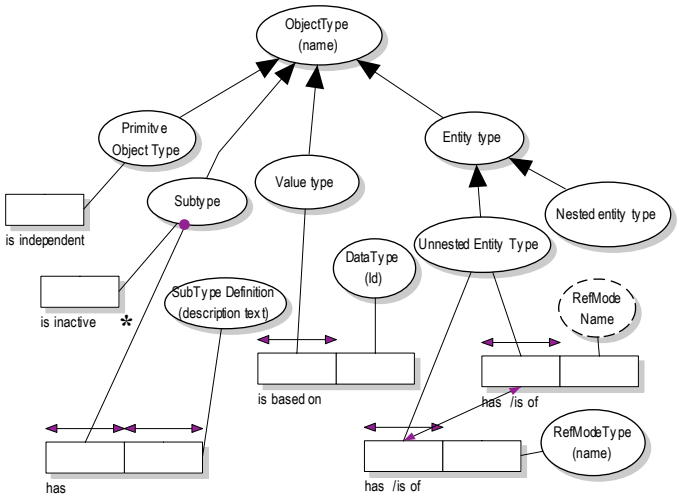


Fig. 7. ORM conceptual schema: fact diagram for integrated enrollment example



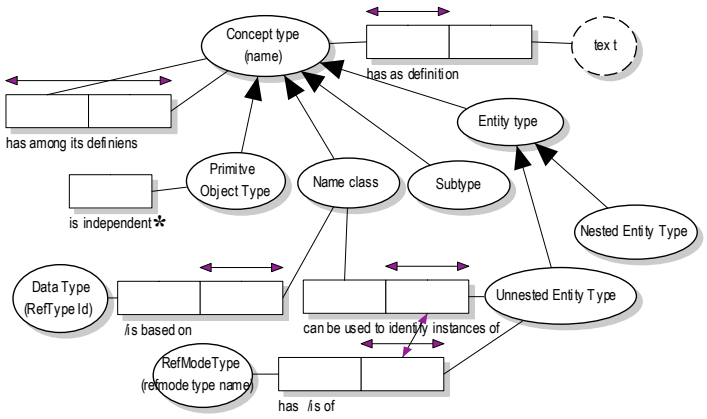
Derivation Rules:
Subtype is inactive **iff** Subtype **is an** ObjectType that plays **no** role

Subtype Definitions:
Each PrimitiveObjectType **is an** ObjectType that is a subtype of **no** ObjectType
Each Subtype is an ObjectType **that** is a subtype of **an** ObjectType

Fig. 8. Relevant segment of existing ORM conceptual meta schema

In figure 8 we have copied the relevant segment of the ORM meta model from [14, p.2]. We will illustrate how we can adapt this meta model for the list of definitions. We can derive the status of the concept types by inspecting the definition in the list of concepts. If the definition is a refinement of another concept type then such a concept type is a *subtype.* If the definition is the description of a condition under which instances of a concept can be used to identify entity types then such a concept type is a *name class.* A concept type can be involved in a definition fact type, but this is however not mandatory for every concept type. Furthermore, the subtype definition is no longer mandatory for a concept type that is a subtype but is contained in the general definition for a concept type.

Concept	***Definition***
Concept type	*a term that is used in the application UoD under consideration and that is not part of a common business ontology*
Concept type name	*a name class, instances of which can be used to identify a <concept type> among the union of concept types within the UoD*
Definien	*a <concept type> that is used in the definition of a different <concept type>*
Definition	*a specification how the knowledge forming a <concept type> is to be constructed from the <definiens>*
Name class	*a <concept type> whose <definition> starts with 'a name class'*
Primitive object typea	*<concept type> whose <definition> does not start with 'a name class' or 'a <concept type>'*
Subtype	*a <concept type> whose <definition> starts with 'a <concept type>'*
Entity type	*a <concept type>that defines an entity in an application UoD*
Nested Entity type	*a <entity type> that defines a relationship that we wish to think of as a <concept type>*
Unnested Entity type	*an <entity type>that defines an atomic entity in an application UoD*
RefModetype	*an instance of a classification for <name classes>*
Refmodetype name	*a name class, instances of which can be used used to identify a <refmodetype> among the union of <refmodetype>s within the UoD*
Data type	*an instance of a classification for denoting characterstrings*
Data type Id	*a name class, instances of which can be used used to identify a <datatype> among the union of <datatype>s*



Derivation Rules:
Prim. Object Type is independent **iff** Primitive Object Type is a concept type **that** plays no role

Fig. 9. Augmented (segment of) ORM conceptual meta schema

5 Conclusions

In this paper we have given modeling concepts together with increments for the CSDP. The new modeling constructs for the list of concept definitons that we have added to the ORM conceptual modeling allows us to capture the relevant part of a domain ontology of an application UoD. We have chosen our modeling constructs in such a way that the overall complexity of the ORM meta model has not increased, but is even reduced by using the list of concept definitions to capture subtype definitions. The augmentation of ORM with a list of concept definitions also preserves the methodological pureness of ORM in which every fact type in the ORM conceptual schema can be traced to (parts of) a user example. A further improvement to the current ORM standard of the augmentation in this article is the incorporation of the complete naming convention semantics in the list of concept definitions.

References

1. Halpin, T. *Information Modeling and Relational Databases; from conceptual analysis to logical design*. San Francisco, California: Morgan Kaufmann, 2001.
2. Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., and Swartout, W. Enabling technology for knowledge sharing. *AI magazine*, fall 1991 (1991), 36-56.
3. Gruber, T. A translation approach to portable ontologies. *Knowledge Acquisition*, 5, 2 (1993), 199- 220.
4. Fensel, D. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer Verlag, 2001.
5. Lammari, N., and Metais, E. Building and maintaining ontologies: a set of algorithms. *Data & Knowlege Engineering*, 48 (2004), 155- 176.
6. Burton-Jones, A., Storey, V., Sugumaran, V., and Ahluwalia, P. A semiotic metrics suite for assessing the quality of ontologies. *Data & Knowlege Engineering*, 55 (2005), 84-102.
7. Weber, R., and Zhang, Y. An analytical evaluation of NIAM's grammar for conceptual schema diagrams. *Information Systems Journal*, 6, 2 (1996), 147-170.
8. Spyns, P., Meersman, R., and Jarrar, M. Data modelling versus Ontology engineering. *SIGMOD record: special issue on semantic web and data management*, 31, 4 (2002), 12-17.
9. Dumas, M., Aldred, L., Heravizadeh, M., and Ter Hofstede, A. Ontology Markup for Web Forms Generation. *WWW '02 Workshop on Real-World applications of RDF and the semantic web*, 2002.
10. De Troyer, O., Plessers, P., and Casteleyn, S. Solving semantic conflicts in audience driven web-design. *WWW/Internet 2003 Conference (ICWI 2003)*, Portugal, 2003.
11. Fliedl, G., Kop, C., Mayer, H.-C., Salbrechter, A., Vohringer, J., Weber, G., and Winkler, C. Deriving static and dynamic concepts from software requirements using sophisticated tagging. *Data & Knowlege Engineering* (in press).
12. Brasethvik, T., and Gulla, J. Natural language analysis for semantic document modeling. *Data & Knowlege Engineering*, 38 (2001), 45-62.
13. Nijssen, G. *Kenniskunde 1A*. PNA, 2001.
14. Cuyler, D., and Halpin, T. Meta-models for Object-Role Modeling. *International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD '03)*, 2003.