István Szakadát, Miklós Szőts, György Gyepesi

MEO Ontology Infrastructure

Abstract

MEO project (Hungarian Unified Ontology) started in 2005. The most important goals of the project were 1) building an upper level ontology, 2) developing a special application supported by ontological knowledge, and 3) planning, establishing and distributing an ontology infrastructure. All achievements of the project were intended to be freely accessible and usable according to the Creative Commons philosophy. In the first part of the paper we outline the ontology building process, briefly describe the most important ontological concepts, present and interpret our most important methodological decisions and our ontological commitments, and show the applied layered solution of the project: how the conceptual and the language layers can be separated and integrated. While the conceptual layer is always language independent, the language layer can contain any number of individual languages without restriction. In the second part of the paper we discuss the structure and the logic of MEO ontology, and present some interesting problems we had to solve in our project.

Introduction

The MEO project (Magyar Egységes Ontológia/Hungarian Unified Ontology) started in 2005. Based on a wide academic and industrial partnership with seven consortium members the project was sponsored by NKFP (a Hungarian governmental R&D Program). The most important goals of the project were:

1) building an upper level ontology,
2) developing a special application supported by domain level ontological knowledge in the field of the telecommunication call center activities,
3) planning, establishing and distributing an ontology infrastructure, and
4) forming a framework for cooperation, consensus management during ontology building processes.

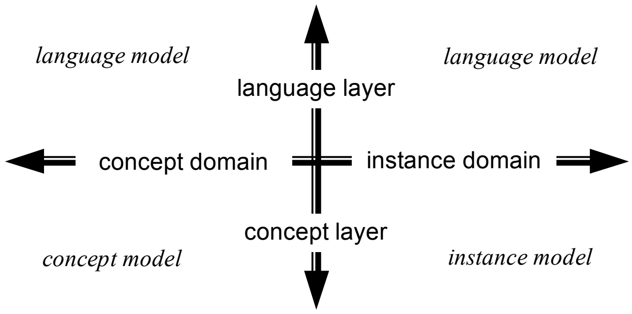
Although in recent years the category of ontology has functioned as a relatively new buzzword, we put the emphasis on our third goal. In spite of the continuously increasing popularity of ontologies, it seemed to be evident to us,

that our most important, current task is to learn how we can handle our ontologies, rather than build ontologies that we do not know how and for what purpose we can use. Of course, by the end of the project we shall build a top level and a domain ontology, but it will be more important to have a tool set (a special integrated, consistent ontology infrastructure component set), with the help of which we or anybody else can start a new ontology building process.
While the knowledge we build into our ontologies consists of words, expressions, concepts which are the results of constant human cooperation over the history of mankind, we decided at the beginning of the project that all achievements of the project would be freely accessible and usable by the public (in line with the Creative Commons philosophy we retain only the Attribution licence to all important achievements of the project). The project communication was fitted to our Creative Commons based commitment, and we launched a MEO session on the Hungarian ontology portal (ontologia.hu/meo), where we made all our project achievements (official project reports, working papers, models, manuals, ontology components etc.) freely accessible.
Of course we did not want to reinvent the wheel. From the beginning of the project it was obvious for us, that we would have to try to reuse all freely available resources. After composing the first version of our upper ontology, we compared our top level concepts to elements of possibly reusable upper level ontologies (like SUMO, WordNet, Dolce). For example we mapped SUMO‘s time and location concepts to the ones in MEO. Sometimes we adopted good solutions, useful concepts from SUMO and Dolce. But it was the area of ontology building methodology, which had really great influence on our project. We learnt a lot especially from the OntoClean methodology. Although we started to develop our own lexicon and ontology editor (MEOditor), a very popular, widely used ontology editor (Protégé) was also included into our suggested ontology infrastructure tool set. Our editor, of course, supports the two most important ontology related formal languages (OWL and DL), and we provided export/import utilities between MEOditor and Protégé as well.

Layered approach

In the MEO project we adopted a layered solution in order to ensure the language neutrality of the ontology. The task is obvious. An ontology – per definitionem – is language independent, but if we (human contributors) would like to use ontological knowledge, we necessarily need a language that is bound to it (a natural or an artificial language). In our MEO model we distinguished a language and a concept layer. We can build our ontologies within the concept layer – totally independently of any language, and in the language layer we can connect as many language dependent words, expressions to our concepts, as we wish. Due to this separation we can build our ontologies in a truly language independent way, and we do not have any language limits. An ontology can be built in any number of languages, and it can not present a problem if a certain language lacks a linguistic construction for a particular concept.

In the MEO model not only the language and the conceptual layers were separated, but we made another distinction as well. In the first phase of the project we could deal with our concepts only on a general level which meant that we divided our concept layer into two parts: a generic level and an instance level. The generic part, the so-called concept domain contains concepts (not instances), and in the other part of the model, what we called instance domain, we can build – in the future – our instance level knowledge base.
Based on the two dimensions briefly introduced in the previous paragraphs we show in the following figure how the different layers of the MEO model can be separated and integrated.

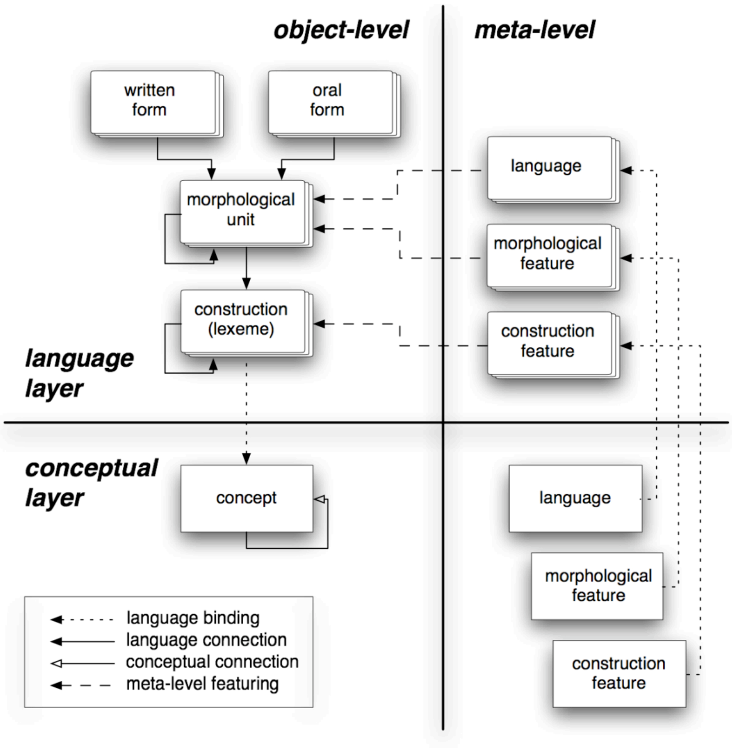


The language layer has its own model (a linguistic ontology). The elements of the language model provide a meta-ontology for our object level ontology. Although our model is valid at the levels of both spoken and written language, in our project we restricted our work to the written language domain.
The most important entities of the MEO linguistic ontology are:

- wordform
- morphological unit
- construction (similar to lexical unit)

Wordform is a simple string without any language binding. This is an entity where we need not differentiate the English word ‘nap‘ (as an activity) from the ambiguous Hungarian word ‘nap‘ (as a period (day) or a star (sun)). This level is needed because it is only here that we can assign word frequencies to the simple strings independently of the languages the wordforms may be found in. If we bind a language and a set of morphological features to a selected wordform, we can talk about a *morphological unit*. Based on this type of information we can predict the behavior of the wordform in sentences. But this grammatical information is not enough if we wish to know the sense of the selected wordform, so we need – on a third level – the entity of the *construction*. With a construction we can bind a concept (and its sense as well) and a wordform or morphological unit (a language specific utterance). Construction is very similar to the category of *lexical unit*, but the two notions are not identical. In our adopted linguistic theory lexical units are *construction constants*, but we need *construction functions* as well if we want to grasp all other types of linguistic phenomena (for example if we want to describe the productive derivation of words).

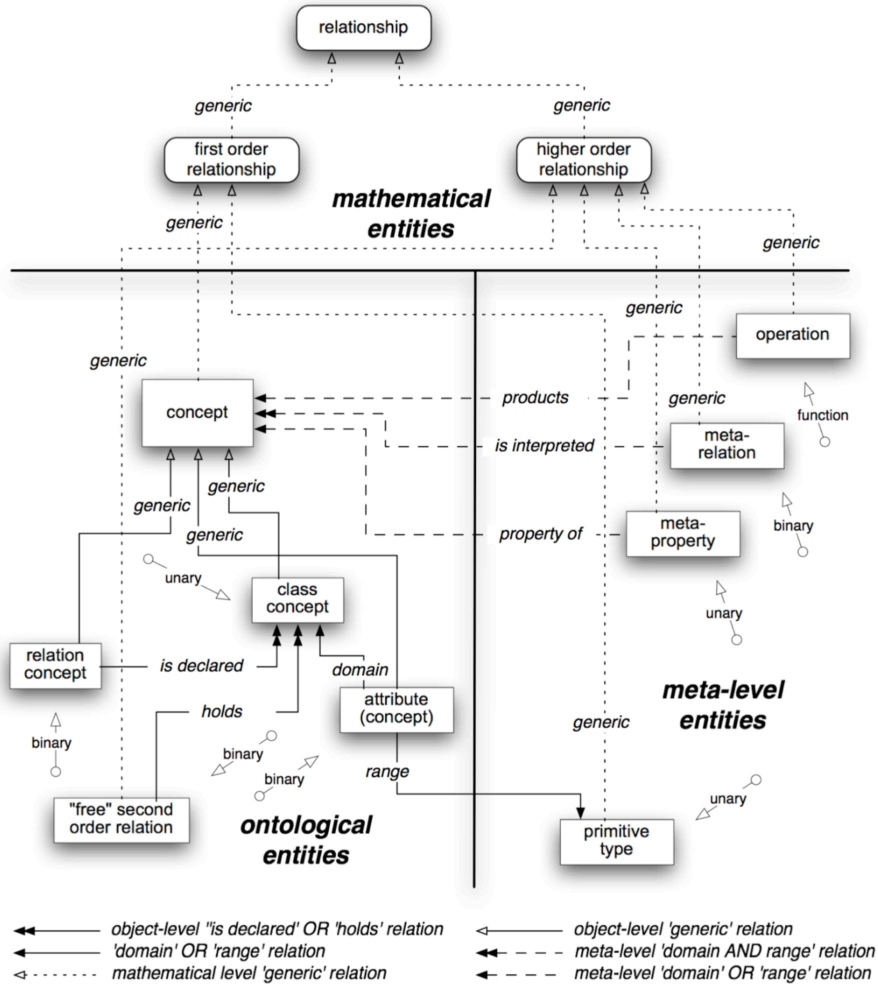
MEO has a strong commitment to construction grammar, which is not particularly widely used either in the ontology building community or in the field of linguistics. This is a unique feature of our project.
The next figure shows the most important linguistic relationships:



Concept model

The concept model is the most important part of the entire MEO model in the intersection of the concept layer and the concept domain. Within our model we needed to separate precisely the object level and the different meta-level concepts and concept areas. 1) At the most abstract level we have some mathematical concepts, of course, 2) we have an object level concept domain where we can build our ontological concepts, and 3) we have another meta-level area, where we separate and handle those meta-concepts that we need to describe our ontological concepts.
The function of the mathematical level having only three main categories is only to transform the concepts of ontology into the language of mathematical logic. We use the notion of *relationship* as is usual in mathematics, and we have two other subordinate categories, the notions of *first order relationship* and *higher order relationship* (with the usual interpretations).
Our basic meta-level entities are *meta-properties*, *meta-relations*, *operations* and *primitive types*. We declared and used the same primitive types that the

programming world uses – these are roughly same the as the data types used in OWL. If it is necessary or simply useful, some meta-properties can be introduced into this meta-level of our model, such as rigid, semi-rigid or dependent in the OntoClean methodologies, but, of course, we can define any other types of meta-property. On this level we defined the most important relation properties (symmetric, asymmetric, transitive, intransitive etc.), and based on these properties and some consequences of them we can build and use a special relation property checking mechanism without using any complex inference system. On this meta-level of the MEO system meta-relations can be declared and used as well, such as the *generic* or *disjunction* relations (this term came from thesauri, and instead of 'generic relation' we could use a lot of other alternative terms like subclass, is-a-kind-of, AKO, subsumption etc.). The generic relation has a special and very important role in our model, we use it on various levels with partly different interpretations (we connect different level entities with the generic relation, for example it is used between object-level concepts or between object-level concepts and mathematical entities). Finally, operations can be declared and implemented on this meta-level in order to create concepts from concepts with the help of operations on the object level. We introduced a lot of OWL and DL operators into the MEO model, such as union, intersection, complement, minimum or maximum cardinality, inverse, composition etc.
The main entity of the object level is the *concept*. The most important goal of the ontology building activity is creating new and new concepts, characterizing them with the help of different features, and establishing relations among the concepts. We differentiated three subtypes of concept, and declared *class concepts*, *relation concepts*, and *attributes*. The difference between class and relation concepts is obvious, we interpreted these categories as it is common in the database and programming world. Attribute is a relation-like concept, but attribute and relation differ from each other in their ranges. While the domain and the range of the relation concept can come from class concepts, in the case of attribute the range can come from primitive types. In this respect attribute overlaps the object level and meta-level area of our model. In our approach the three main subconcepts have totally equal status which is not so common in the ontology building community. Most ontologies concentrate on class concepts. Our ontology contains not only the concept of 'father' (which is a class concept), but the 'is father of' relation is considered as concept. In our universe there can be another relation type, a *„free" second order relation*. 'Free' means it can be freely declared and defined by the editors of the special ontology. A possible example for it can be the so-called evolutional development relation, with which we can describe the relationship between Horse and Przewalski's Horse in the taxonomy of animals. This relation is not generic (between the instances of the two concepts there are no real connection), but because it can be interpreted between two concepts, we have to handle it as a second order relation.
The different parts and the most important entities of the concept model are the following:

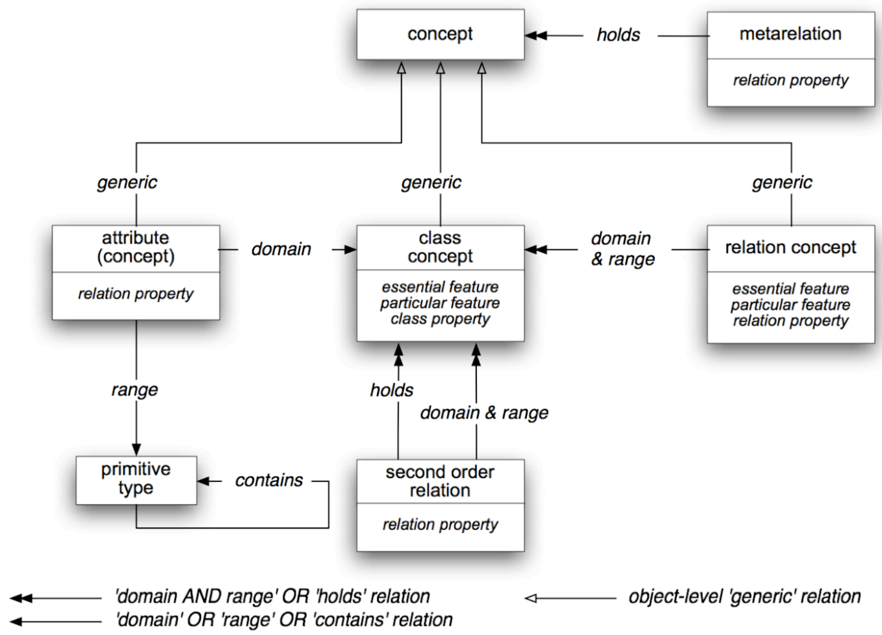


Under the highest level, most abstract conceptual entities, the model has some other concepts which can be used in all types of ontology. Very closely connected to the generic relation, we defined and applied the notions of *partition* and *taxonomy*. We declared a very important ontology building rule: in the object level area of the model, during the ontology building process we have to subordinate our new concepts to the already existing concepts in a given ontology. As we have three concepts subordinate to our main category (concept), all new concepts have to be subordinated to these (class, relation and attribute) concepts, and therefore potentially every ontology can have three „independent" taxonomies.

For each taxonomy we introduced another general concept which we called *category*. Category (in our model) is a concept immediately subordinate to the root concept of the possible taxonomies. This notion (or more exactly this interpretation) came from the field of the classification systems and thesauri. In general 'category' is a widespread and overused term, but in our interpretation it has a definite, unambiguous meaning, and we can say, that categories are the first level concepts in the concept hierarchies.

Concepts and features

In the MEO project we tried to reuse a very old philosophical category first applied by Aristotle. When Aristotle described how we could create new concepts, how we could build taxonomies, he used the notion of *differentia specifica*. When we create a new concept with the help of the generic relationship we have to provide a feature which unambiguously characterizes the new concept. This idea is very simple and seems to be very promising, although the well-known ontologies (or thesauri) did not adopt it. The probable explanation is again very simple. Providing new features every time we create new concepts would require much more effort. Based on this principle we should build a dual taxonomy, because we have to provide the same number of new features (differentia specifica) as the number of concepts we have. It is not an easy task.
In our project we did not require differentia specifica to be assigned to every new concept, but we incorporated this „featuring" possibility into our model. We can characterize our concepts in different ways. The attributes and the relation concepts can be described with the help of the well-known relation algebraic properties, such as symmetric, irreflexive, transitive etc. In the case of class concepts we can use meta-properties (the kind, for example, that OntoClean methodology introduced) in order to characterize our concepts. We can bind two other types of feature to class concepts and relation concepts. Essential features are those which are usually inherited via generic relation (although we have to provide exception handling capabilities as well, because inheritance is not a necessary requirement), while particular features are not inheritable. The following figure shows the most important connections between concepts and features:



MEO and OWL

Comparing the MEO concepts to the OWL concepts we can map the most important categories to each other. For example the MEO primitive type has the same meaning and role as the OWL Datatype, and we can MEO attribute to OWL DatatypeProperty, MEO relation concept to OWL ObjectProperty. MEO contains all DL constructors and relation properties which exist in OWL (except those concepts that have individual connections, for example OWL hasValue, because in the first phase we did not implement instance level system components). In MEO we can differentiate ObjectProperty (relation concept) and DatatypeProperty (attribute). Finally axioms are not allowed in OWL which can handle only comments (both exist in MEO), and the concept of feature is completely missing from OWL while it has a very important role in the MEO model.

Conclusions

In the MEO project we are building a top level and a domain ontology, but our main focus is on planning, building, using and distributing a robust, consistent ontology infrastructure. In order to ensure the real language dependency of our ontologies, we clearly separated the language and the concept levels of our model, and we designed and implemented an integrated and institutionalized set of language management capabilities with which we can easily bind as many languages to our concepts as we wish. This objective and our implementation of it is, in our opinion, unique, we are not aware of any similar projects. While we have built our inner linguistic ontology, we based our work on construction grammar, which is, again, a less frequently used theoretical approach among the ontology building community. Of course, it is usual (maybe necessary), to have the entity of the concept stand in the centre of the model, but the fact, that we differentiated three subtypes of our main entity (concept) and evaluated these as three subordinated concepts with totally equal status is unusual and can be regarded as a novelty. And finally in contrast with the widely applied approach we attributed an important role to relation concepts, and we tried to provide an institutionalized role for features.

References

[Corcho et.al 2003]
Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez: Methodologies, tools and languages for building ontologies. Where is their meeting point?, in: *Data & Knowledge Engineering*, Vol. 46, 2003, pp.41-64.

[Green et al. 2001]
Green, Rebecca, Bean, C.A., Myaeng, S. Hyon, *The Semantics of Relationships: An Interdisciplinary Perspective*, Dordrecht: Kluwer, 2001.

[Guarino 2000]
Guarino, N., Welty, C., A Formal Ontology of Properties. *Proceedings of 12th Int. Conf. on Knowledge Engineering and Knowledge Management* Lecture Notes of Computer Science, Springer Verlag

[Guarino 2001]
Guarino, N., Welty, C., Supporting ontological analysis of taxonomic relationships. *Data & Knowledge Engineering* 39 pp 51-74.

[Staab 2004]
Handbook of Ontologies eds. S. Staab, R. Studer, Springer Verlag