Methodological Aspects of a Data Reference Model for Campus Management Systems

Marco Carolla, Thorsten Spitta
Bielefeld University, Faculty for Business Administration and Economics, D-33501 Bielefeld,
{marco.carolla | thSpitta}@wiwi.uni-bielefeld.de

Abstract: It can be doubted, whether the student is a university's customer, but if he is regarded as such, a campus management system is a very important customer-induced service system. Our paper proposes an incremental ontology-based method for reference models, applied in the domain of German-speaking higher education institutions. The method is applied to gain a reference data model for campus management system systems being deployed since about three years as competing solutions for large universities. The model is developed validating it empirically using publicly available institutions' regulations. An important goal of our paper is, to give the readers of our community basic insights into the construction of an ontology, which is an important foundation of semantic modelling.

1 Introduction

A fundamental structural change in German-speaking universities (Austria, Germany, Switzerland) caused an urgent demand for software, which could handle the new processes, mainly in teaching and assessing: The so-called *Bologna Process*. This “process”, consisting of many sub-processes, can no longer be handled by office components, it needs professionally developed and maintained standard software. Except the very old and monopolisticly grown German set of packages HIS (*Hochschul-Informationssysteme*) (see [32; 34]) we only find pilot systems implemented in few large universities (e.g. Hamburg, TU Munich, RWTH Aachen), which have the potential to *become* a standard, but in our opinion earliest after the fifth successful implementation. Bode & Borgeest [2], Bick et al. [1] and Schillbach et al. [28] give overviews about the software 'landscape' in this field. Meanwhile a new 'name' for such systems was created, *Campus Management Systems* (CaMS), used by the pioneers of new software with arbitrary semantics. The multitude of aspects in this field can not even be mentioned here, but it seems fundamental in the situation of the ongoing Bologna Process, that reference models are needed to facilitate the communication of *all* stakeholders, the developing enterprises and the large number of users, from the teaching to the learning and the administrative members of universities. Meanwhile it seems as a common scientific rule [7, p. 333]; [36, p. 2f], that standard systems should be developed by means of *reference models*.
However, as in the case of SAP's R/2 in the 1980s [25], the evolution of the above mentioned new systems does *not* follow a well communicated semantic foundation. We try to fill this gap with a reference model for campus-management system's databases. The paper presents the methodological foundation of this project, demonstrated by examples of our final results. We use an ontology to design our universe of discourse (UoD), apply a language-based method to construct normalized statements about the UoD and validate the findings against a sample of published regulations of representative universities. Figure 1 shows our main modelling steps.

We hope our paper can help to promote a discussion in the university's own business, the modelling of their internal affairs.

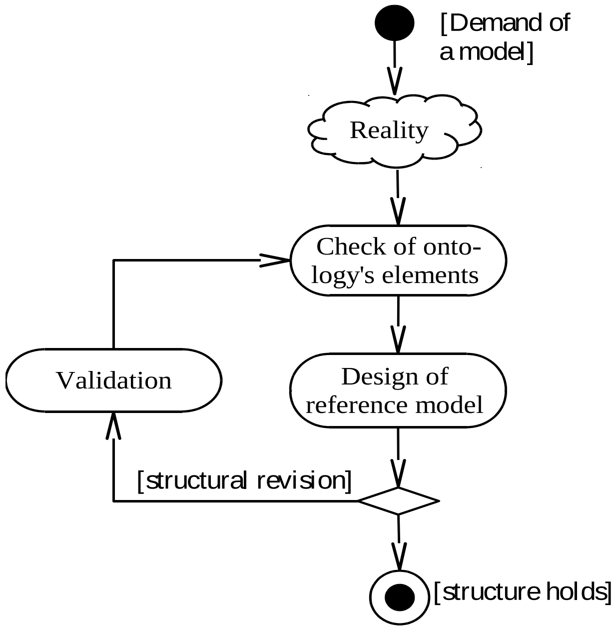

Figure 1: Development of the reference model

2 Terms and Research of Semantic Modelling

A *model* is a set of statements about a universe of discourse [29, p. 27] replacing something that has to be understood, created, undertaken or run [15, p. 377]. A *conceptual model* is a description of some aspects from the physical and social world to understand and communicate its concepts [21, p. 51]. Modelling terms have to underlie agreed semantics to avoid misunderstandings in communication about a UoD. The term *semantic modelling* is common as well [15, p. 384f]. Because the database of an information system is a core element, a *semantic data model* is a relevant concept of the UoD. It is represented as object-types, their attributes and integrity constraints. We claim four components for a semantic model: a *language*, based on a *grammar* defining rules how to use language and grammar constructs, a *modelling method* offering a methodical framework, a *script*, as the result of the modelling process, and a *modelling context* describing its circumstances (e.g., see [23]; [40, p. 364]).

Language and Grammar have been treated by Wand & Weber, who state that the language should be based on an ontology defining an agreed fundamental view of the world [39, p. 220]. This is a first step to a common understanding of the UoD. In its traditional philosophical meaning, ontology is a discipline dealing with the structure and order of the real and the cognitive world and its fundamental components. Ontology (as a discipline) creates an ontology (as an artefact): a theory about the nature of being and the existence of things and their relationships [10, p. 1]. Such ontologies are called *Top-Level-Ontologies* [9, p. 9]. Examples are the *Bunge-Wand-Weber-Ontology* [38;39], *Unified Foundational Ontology* (UFO) [11;12] or the *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) [20]. We chose UFO for our purposes, described in the next section, because its scope covers social systems as well as material ones and concentrates on human perception and representation adequacy for means of communication rather than machine computability [11, p. 90ff].

As mentioned above, language is the second foundation of our method. Once discovered, we have to find adequate and well defined names for our modelling constructs to avoid misunderstandings. Ortner [22]describes a language-based method dealing with this problem, based on the ideas of the “Erlangener Schule” [19]. We use this method to define modellings terms being used to name real-world concepts that are categorised by means of a Top-Level-Ontology.

In our first modeling steps, we use UFO and Ortner's method to create a conceptual model that can be called a domain-ontology (see [9]). It focusses on real-world structures. In the sequel, this model will be transformed into a relational data model representing a concrete data structure. This structure is finally used to validate our findings by means of emprical data. The idea of a “pre-conceptual model” [18] can be found in other publications as well (e.g.[8;14;24]).

3 The UFO Ontology

To give the reader of our community an impression about the detailed subjects of a Top-Level-Ontology, we describe the UFO ontology in a rather detailed way, following the main source, developed by Guizzardi, 2005 [11].

Particulars and universals. *Universals* are property patterns that are realised in *particulars* [p. 94].

Endurants and perdurants. *Endurants* are particulars being wholly observable whenever they are present [p. 210]. *Perdurants* are particulars, composed of temporal parts, e.g. a business process. When a perdurant is present, it is never the case that all its temporal parts are present. As a structural view, our approach concerns endurants.

Substantial individuals and moments. Endurants are split into substantials and moments. A *substantial* S has direct spatial-temporal qualities and is existentially independent of other particulars that have no common parts with S [p. 215]. Substantials are further specialised in *objects* and *amounts of matter*. *Moments* are objectified properties, such as a colour or an electric charge [p. 212]. They depend existentially on other individuals, their *bearers* [p. 213f]. *Intrinsic moments* have exactly one bearer (e.g. the age of a person), *relational moments* depend on more than one individual. Substantials are instances of *substantial universals*; moments are instances of *moment universals*.

Sortal universals and mixin universals. Substantial universals are specialised in *sortal universals* and *mixin universals*. Both carry a *principle of application* that assigns a particular to a specific universal [p. 98]. In addition, sortal universals carry a *principle of identity* for their instances that allows to judge whether two particulars are identical.

Rigidity, non-rigidity, and anti-rigidity. Sortal universals can be rigid, non-rigid, or anti-rigid [p. 101f]: We consider a particular x that is instance of a universal U. If x can not cease to be an instance of U without ceasing to exist, and if x is also an instance of U in every other conceivable world w', then U is called a *rigid* universal. U is called *non-rigid* if it does not apply necessarily to *at least one* of its instances in every conceivable world w'. U is called an *anti-rigid* universal if it does not necessarily apply to *all* of its instances in every conceivable world w'. Anti-rigid sortals applying possibly to an individual only during a certain phase of its existence are called *phase sortals* and are specialised in *phases* and *roles* [p. 102f]. An individual can play several roles at the same time or pass through several phases, hence there exists a substantial universal S for every phase sortal PS, which is specialised by PS and on which the identity of the individual is based.

Parts and Wholes. UFO differentiates three types of part-whole relationships: *Mass/Quantity* [p. 174ff], *Member/Collection* [p. 180ff] and *Component/Functional Complex* [p. 183]. We neglect the first case in this paper, as we could not observe a mapping example in the UoD up to this point. As (integral) wholes, collections and functional complexes are held together by a *unifying condition* [p. 156] (e.g. a social one, such as "enrolled in the same university"). The difference between them is constituted by their structure [p. 183]: The parts of a collection all play the same role type (all members of a football team can be considered as football team members). On the other hand, the parts of a complex can play a variety of roles, each role contributing to the whole's functionality (we can speak of a football team as a complex, when we consider the different roles the members play, e.g. goal keeper or defender). We gain three types of part-whole relations related to collections and complexes: *M-parthood* [p. 185f] relating a singular entity and a collection; *C-parthood* [p. 186f], relating subcollections and collections, and the *relation between a component and its functional complex* [p. 187f]. Parthood relations have secondary properties, such as shareable parts or essential and mandatory parts (see [11, chap. 5.4]).

Formal and material relationships. Formal relationships hold directly between two or more individuals, e.g. comparative relations, such as "taller than" [p. 236]. Material relationships have an own material structure; examples are an employment or an enrolment. They are *mediated* by a third individual, called *relator* [p. 236]: an employment connects a person and his employer, an enrolment connects a student and a university.

Social concepts. For our purposes we focus on the extensions of UFO with social concepts concerning endurants ([12, p. 132ff]): Objects are specialised in agents and non-agentive substantial particulars, hereinafter called *agent* respectively *object*. Agents can be *physical* (a person) or *social* (a faculty as an organisational unit). This also applies to objects, e.g. a book or a chair as physical objects, and money, a language or a normative description as social objects. Normative descriptions, e.g. examination and study regulations in our UoD, define rules and norms that are noticed and accepted by at least one social agent. They create exactly what Searle calls "institutional reality" [26, p. 19]. They form new "social entities" [27, p. 277], such as a 10 € bill as an instrument of payment, a social role like a student or a normative description.

Used constructs. A *kind* [11, p. 108] represents a rigid sortal and supplies a principle of identity to its instances. Its parts contribute to the functionality of the whole. A *subkind* specialises a kind. Subkinds are rigid sortals; their instances are self-contained and have an own identity. A *collective* [p. 181] is a rigid sortal with its own principle of identity. Its parts have a homogeneous structure and are held together by a unifying relation. A *phase* [p. 103f] is an anti-rigid sortal without an own principle of identity. It represents phases in the history of an individual and is based on an intrinsic specialisation condition. An individual can never be in two distinct phases at the same time. A *role* [p. 103f] is an anti-rigid sortal without an own principle of identity. An individual plays a role in a specific context, which is delimited by a relationship to another entity. A role results from a specialisation that is based on an external (relational) specialisation condition. A *relator* [p. 240] is a moment universal, i.e. it represents properties. Relators mediate the relata of a material relationship, bearing all properties that only exist in this context. Their instances have an own identity, but they are existentially depending on the connected entities. *Categories* [p. 112] are rigid mixin universals without own principle of identity. Categories are abstract, never having direct instances. They sum up common properties of kinds being realised in their instances. A c*omponentOf-relationship* [p. 350] is a parthood relationship that holds between a kind, as a functional complex and its components. A *memberOf-relationship* [p. 352] holds between a collective and its members. A *material-relationship* is a material relationship as explained above. Such relationships always derive from a relator. A *Mediation-relationship* [p. 241] is a formal relationship that relates such a relator and one of the entities it relates. A *Derivation-relationship* [p. 241] is a formal relationship that holds between a material relationship and the relator this material relationship is derived from.

OntoUML (e.g. see [11; 13]) is a modelling language whose constructs and grammar are based on UFO, hence it implies the concepts explained above. To describe reality structures by means of UFO, we have to translate natural language (NL) constructs to modelling language (ML) constructs. The already provided mapping of UFO-constructs to OntoUML-constructs (see [11, chap. 8]) leads directly to an ontology based structural model. We describe this translation process in the next section.

4 Reality Construction

Language carries semantics which a model of an UoD has to capture. Descriptions of an UoD are verbalised in NL. Hence these descriptions are mostly vague, incomplete and contradictory – misunderstandings occur e.g. due to homonyms and synonyms [23, p. 35]. [3] deals with this issue by breaking complex sentences of NL descriptions into *kernel sentences* (see [4]) of the form *subject* + *verb* + *object*. Those sentences constitute the semantics of a text [3, p. 489]. Their components can be directly categorised by means of *semantic types* in accordance to [5, p. 7], who states that all words in the *open word classes* (nouns, adjectives, verbs and adverbs) can be grouped into types sharing semantics and some properties. The idea in the approach of [3] consists of identifying conceptually significant NL signs (nouns, verbs, adjectives) in a normalised list of statements about the UoD, linking them to semantics types, and mapping the semantic types to UFO-/ML-constructs. Now, the structural model can be created following the grammatical rules of the ML.

We adopt this idea, but instead of semantic types we use the older approach of Wedekind, Ortner et al. [37; 22; 23], well known in the German-speaking literature, dealing with the issue of unclear NL descriptions and normalising it by a (re-)construction process clarifying the semantics extensionally and 'intensionally'. The German pair of philosophical terms *Extension – Intension* does not exist in English. The translation of Intension – semantics – is too weak. We use in this paper 'intension' to express the semantics, like intended by Ortner. This (re-)construction process results in a normalised terminology that limits the common speech in the UoD and unifies the communication about it. We map the results of four construction acts [22, p. 23ff] to UFO-constructs. In our opinion, this is more feasible and reproducible than choosing from a large number of mutually not excluding semantic types, of which Dixon uses more than fifty (see [5, part B]). Following [23, p. 34] and [3, p. 489f] we split our approach into seven steps.

4.1 Collect relevant statements about the UoD

There are several resources for statements about the UoD, e.g. software- and database documentations of productive systems in the UoD, legal texts, interviews with domain experts or data models (see e.g. [30] for CaMS). The result of the first step is a collection of NL statements about the UoD.

4.2 Clear and reconstruct relevant modelling terms

From the NL statements we gain reconstructed and clear *terms*. [23, p. 35f] describe several defects that have to be corrected before a new term can be introduced: Synonyms have to be pointed out, homonyms have to be terminated, equipollencies have to be illustrated, vague terms have to be cleared and properly defined, wrong or unclear identifiers suggesting another meaning have to be replaced by an adequate name. For the introduction of a new term, we use one of the four construction acts listed by [22, p. 23ff]: *Predication, inclusion, connexion* or *aggregation*. Every construction act results in a new term with particular (meta-)properties forming its type (see section 4.4). As a result, we get a list of (re-)constructed terms, which either stand for classes of information objects (objects types) or properties of object types (attributes).

4.3 Find relationships between terms

After the reconstruction of modelling terms we have to point out their relationships. We do this by means of kernel sentences (see first paragraph of section 4), derived from our UoD statements. We get a list of kernel sentences, where subject and object are replaced by our reconstructed terms and the *verb* denotes the relationship between them (see example in table 1 on page - 22 -).

4.4 Map resulting terms and relationships to ML constructs

We map terms resulting from step 4.2 to UFO constructs by comparing their meta-properties to the definition of UFO constructs: The ***predication act*** [22, p. 23] introduces a new term, hence its result has to be cleared in the specific context. The ***inclusion act*** [ibid.] either specialises or generalises a term into a new term. The resulting term of a specialising inclusion carries new properties in its 'intension' – the specialisation condition. If the specialisation condition is extrinsic, e.g. a relationship with another term, we map the resulting term to an *UFO-role*, else we choose an *UFO-phase* to represent it. The resulting term of a generalizing inclusion sums up common properties of the generalised terms. At the end of an inheritance chain it is mapped to an *UFO-kind* or an *UFO-category,* else it can be an *UFO-category*, *UFO-subkind*, *UFO-role* or *UFO-phas*e. The ***connexion act*** [ibid.] connects partial terms to a complex whole, whose parts contribute to the functionality of the whole. This corresponds to the notion of an *UFO-kind*. The ***aggregation act*** [22, p. 24] sums up terms to form a relatively homogeneous whole. This corresponds to the notion of an *UFO-collective* with homogeneous parts that are held together by a unifying relation. Next, we map relationships found in step 4.3 to UFO-relationships: Parthood relationships result from or are indicators for the application of connexion- or aggregation acts. They are mapped to c*omponentOf*- resp. *memberOf*-relationships. Verbs denoting comparative relations are mapped to *formal-relationships*. Verbs indicating an external relationship with an own material structure (e.g. “is enrolled in” and the resulting enrolment) are mapped to *material-relationships*. In table 2 on page - 22 - we show the terms of our example, mapping UFO-constructs to the steps of Ortner.

4.5 Create a structural view (Entity type view)

After mapping the terms to UFO-constructs we can create an OntoUML diagram showing the structure found. In figure 2 on page - 23 - we show an example, relevant for our UoD *campus management*.

4.6 Show the 'intension' of the terms in an object view (Attribute type view)

For pragmatic reasons (good readability and first consistency checks) we use a relational representation and not the common UML class diagrams for the object view of the entity's attributes. It is *not* intended to drift into a physical database design. Especially the large amount of enumeration types in information system's databases (see case study from a large industrial development in [31]) simplifies the structural view of a data model significantly. The compact presentation of relations and related attributes clarifies the 'intension' of a term and allows first statements about the meaning of an object type. Section 5.6 will show a core part of the reference model's object view.

As the step from the OntoUML-diagram to the relational model is the first step to a logical model as well, new object types can be found being not considered in the previous structural view. We found some rules that can be applied within this transformation process: Relators, mediating material relationships, are modelled as discrete relations in the object view. *Roles* are not modelled as relations in the object view, their names are used in the resulting relator. Such a relator results always from a role, since a role's specialisation condition is an external one. *Mereological structures* (whole-part-relationships) with wholes consisting of multiple parts of the same type are modelled as a new relation in the object view (suffix "_Struct").

4.7 Empirical Validation

We validate the relational model that results in the object view by filling it with real data gained from examination and study regulations published by universities. We proceed stepwise until we have a sample large enough or the latest n steps showing a convergent structural view. We can speak of a *reference model*, if the number of mapping test cases fulfils our sample size (see [17, ch. 5], *cluster sample*]). Due to space restrictions we can not show our sampling strategy and results in depth. We selected randomly 10 of the 60 large German speaking universities [16], within which we analysed 3 study courses for each, selected randomly as well. As 'large' we define > 15.000 students [6].

5 Some Sample Results

We show for one case an excerpt of our UoD's core, which is the regulation modelling.

5.1 Statements

We start with a set of statements about the UoD: **Regulations**, e.g. examination and study regulations, regulate contents, progress and examinations of a course of study. Regulations can concern e.g. only one or a group of degrees and all related courses of study, or only particular courses of study of a university. **Subjects** have several variants. **Variants** embody "courses of study" in a sequel. They are defined as closed study programs leading to different **degrees.** A variant usually consists of different **phases** describing a set of cohesive modules. Profiling phases, for instance, consist of modules that teach knowledge of a special area of expertise within a subject. Phases can be divided into sub-phases which can be divided into sub-phases as well. A **module** is a learning unit that relates several (module-)elements. **Elements** are the smallest learning units; types of elements are courses, seminars or examinations.

5.2 Modelling Terms

We gain the terms *regulation*, *subject*, *variant*, *degree*, *phase*, *module* and *element*. shown in table 2 with the terms' semantic definitions.

5.3 Relationships

In our set of statements we find the following relationships between our modelling terms: Regulations *regulate* degrees; regulations *regulate* variants; variants *have* a subject; variants *have* a degree; variants *consist of* phases; phases *divide* phases; modules are *attached to* phases and elements are *attached to* modules.

Relationship	***Semantics***	***UFO Relation***	***Resulting Terms***
regulation, degree	regulation regulates degree	material	Degree regulation, regulation for degree
regulation, variant	regulation regulates variant	material	Variant regulation, regulation for variant
variant, subject	variant has subject	component of	-
variant, degree	variant has degree	component of	-
variant, phase	variant consists of phases	component of	-
phase, phase	phase divides phase	material	Sub_Phase, upper phase, sub-phase,
module, phase	module attached to phase	material	Module_Phase, phase with module
element, module	element attached to module	material	Element_Module

Table 1: Mapping relationships to UFO-constructs

5.4 Mapping terms and relationships to modelling language

Table 2 shows the (re-)constructed terms in the first column and their mapping to UFO-constructs. Table 1 shows detected relationships, corresponding UFO-constructs and resulting terms. Resulting new terms were added to table 2.

Term	***Semantics***	***UFO-construct***	***TermCal***
Regulation	A regulation, e.g. a study regulation.	Complex whole with an own identity, hence *kind*.	Predication act
Degree	A concrete degree related to a regulation. Part of a variant that leads to this degree.	*Kind* (see regulation).	Predication act
Subject	A subject, e.g. information systems.	*Kind* (see regulation).	Predication act
Phase	A learning unit, as a part of variants. Phases can be divided into other phases. Modules are attached to phases.	*Kind* (see regulation).	Predication act
Variant	Variant of a subject. Parts of a variant are a subject, a degree and two or more phases.	Complex whole with own identity composed of kinds as functional parts, hence *kind*.	Connexion act
Module	A learning unit with attached elements.	*Kind* (see regulation).	Predication act
Element	Smallest learning unit like a course or a seminar.	*Kind* (see regulation).	Predication act
Degree_Regulation	Mediates regulations and degrees.	Derived from an material relationship, hence *relator*.	Predication act
Variant_Regulation	Mediates regulations and variants.	*Relator* (see degree regulation).	Predication act
Sub_Phase	Mediates sub phase and upper phase.	*Relator* (see degree regulation).	Predication act
Module_Phase	Mediates modules and phases.	*Relator* (see degree regulation).	Predication act
Element_Module	Mediates elements and modules.	*Relator* (see degree regulation).	Predication act
regulation for degree	Specialises regulation: Regulation that regulates a degree.	External specialisation condition, hence *role*.	Inclusion act
regulation for variant	Specialises regulation: Regulation that regulates a variant.	*Role* (see regulation for degree).	Inclusion act
phase with module	Specialises phase: A phase with an attached module.	*Role* (see regulation for degree).	Inclusion act
upper phase	Specialises phase: A phase with attached sub phases.	*Role* (see regulation for degree).	Inclusion act
sub-phase	Specialises phase: A phase that acts as a sub phase.	*Role* (see regulation for degree).	Inclusion act

Table 2: Our UFO terms with synopsis to Ortner's “Begriffskalkül” (TermCal)

5.5 Conceptual Structural View

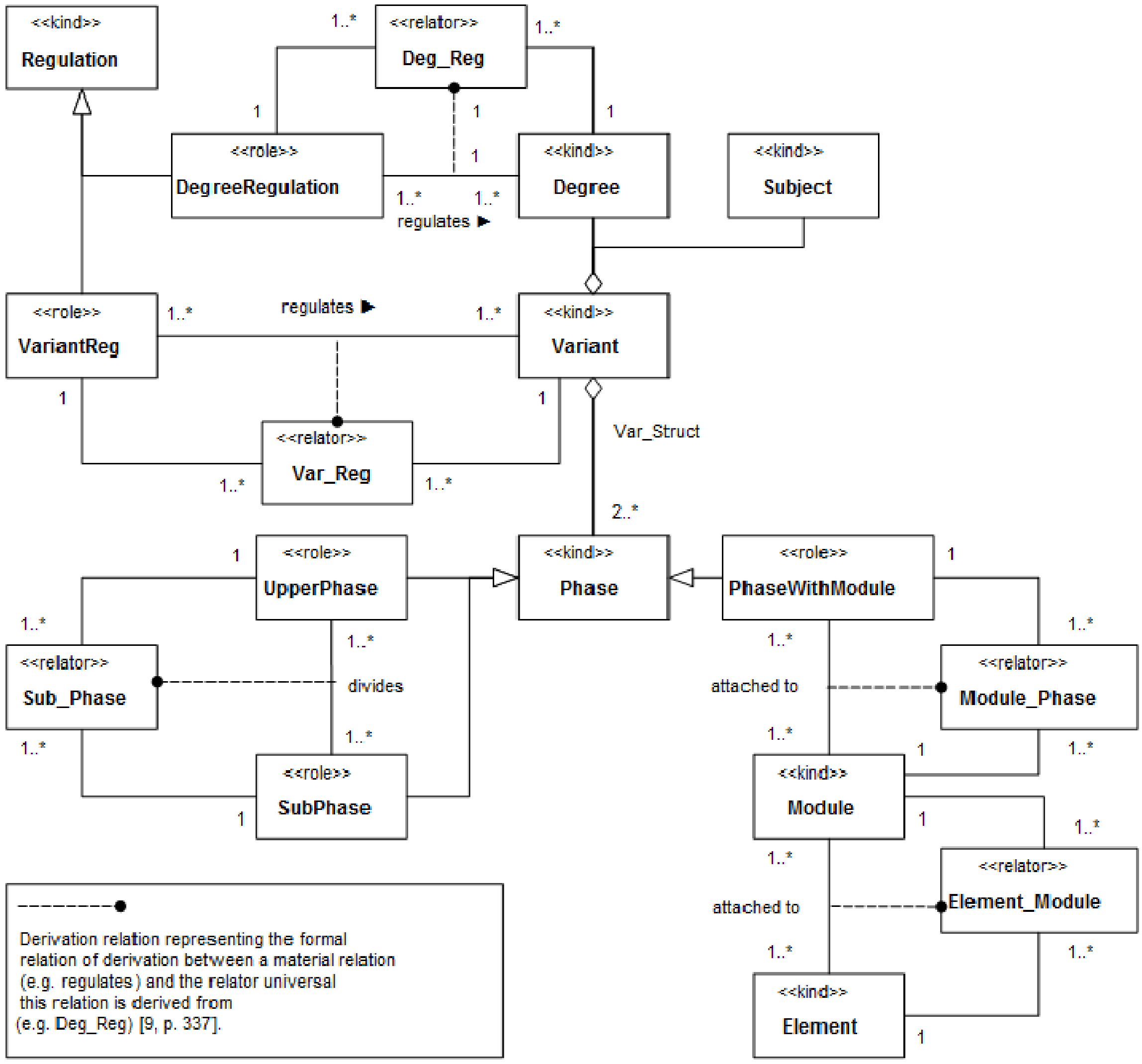

Figure 2: Structural view built with OntoUML

5.6 Object View

We show found entities in an object view: Terms mapped to UFO-kinds are represented as relations, UFO-roles materialise as role names. Further we get a new structural element: Variant_Struct, denoting the whole-part relation between Variant and Phases. The given attributes aren't considered complete – e.g. we drop attributes that contain long textual descriptions. A reference model can never contain all relevant attributes for every conceivable implementation, they vary more often from university to university than the far more stable structural view. The values of enumeration types are adapted to our validation example.

We use the following notation for our relational model: '#' represents keys, in addition, primary keys are underlined. Alternate keys are marked by '#a'. A foreign key is a not underlined attribute with '#'. [en] is an *enumeration* attribute type.

```
Regulation(id#, year, name, valid_from, valid_to, reType[en])
  with reType = {APSO, FPSO}
```

```
Degree(id#, degree[en], studyCourse[en], cp, periodOfStudy)
  with degree = {Bachelor of Science(B. Sc.), Bachelor of Arts (B.A.), ...}
       studyCourse = {Diplom, Master, 1-Fach-Bachelor, 2-Fach-Bachelor, ...}
```

```
Degree_Regulation(deg.id#, reg.id#)

Subject(subjNo#, name, abbr)

Variant(id#, name, variant[en], subjNo#, deg.id#, cp)
  with variant = {Vollstudium, Kernfach, Nebenfach, Lehramt, ...}

Variant_Regulation(var.id#, reg.id#)

Var_Struct(id#, var.id#, phase.id#, binding[en], cp)
           with binding = {Pflicht (P), Wahlpflicht (WP), Wahl (W)}

Phase(id#, name, abbr, cp)

Sub_Phase(upperPhase.id#, subPhase.id#, binding[en])

Module_Phase(id#, modNo#, phaseWithMod.id#, binding[en], cp, Wkl)

Module(modNo#, name, abbr, cp, level[en], cycle[en], lang[en])

Element_Module(modNo#, element.id#, binding[en], cp)

Element(id#, name, elemType[en], cycle[en], lang[en])
```

5.7 Empirical Validation

Table 3 shows some sample data of the course "Bachelor Mathematik" at the TU Munich running CAMPUSonline. We gained the structure from its representation in this system, which can be found on the university's homepage [35].

Regulation	*id#,*	*year,*	*name,*	*val_from,*	*val_to,*	*reType[en]*
	60	2011	Allgem. Prüfungs- und Studienordnung für Bachelor- und Masterstudiengänge an der TUM	01.04.11		APSO
	70	2012	Satzung zur Änderung der allgem. Prüfungs- und Studienordnung für Bachelor- und Masterstudiengänge an der TUM	01.10.12		APSO
	80	2007	Fachprüfungs- und Fachstudienordnung für den Bachelorstudiengang Mathematik	01.10.07		FPSO
	90	2009	Satzung zur Änderung der Fachprüfungs- und Fachstudienordnung für den Bachelorstudiengang Mathematik	01.04.09		FPSO
	100	2009	Zweite Satzung zur Änderung der Fachprüfungs- und Fachstudienordnung für den Bachelorstudiengang Mathematik	01.10.09		FPSO

Degree	*id#,*	*degree[en],*	*stCourse[en],*	*cp,*	*periodOfStudy*
	100	B.Sc.	Bachelor	180	6

Var	*id#,*	*name,*	*subjNo#,*	*deg.id#,*	*cp*
	100	Bachelor Mathematik Vollzeit	10	100	180

Var_Str	*id#,*	*var.id#,*	*phase.id#,*	*bind[en],*	*cp*
	100	100	10	P	12
	150	100	20	P	52
	200	100	30	P	68

Var_Reg	*var.id#,*	*varReg.id#,*
	100	80
	100	90
	100	100

Deg_Reg	*deg.id#,*	*degReg.id#*
	100	60
	100	70

Phase	*id#,*	*name,*	*cp*
	10	Bachelor's Thesis	12
	20	Pflichtmodule Mathematik	52
	30	Wahlmodule Mathematik	68
	60	A 1.1 Basis	36
	70	A 1.2 Propädeutika	16
	80	A 1,3 Aufbaumodule Reine Mathematik	19
	90	A 1.4 Aufbaumodule Angewandte Mathematik	19
	100	A 1.5 Vertiefungsmodule Mathematik	9

Subject	*subjNo#,*	*name,*	*abbr*
	10	Mathematik	MA

Sub_Phase	*upperPhase.id#,*	*subPhase.id#,*	*bind[en]*
	20	60	P
	20	70	P
	30	80	P
	30	90	P
	30	100	P

Mod_Phase	*id#,*	*modNo#,*	*phWiMod.id#,*	*bind[en],*	*cp,*	*wkl*
	50	MA6012	10	P	12	360
	150	MA1001	60	P	9	270
	250	MA1002	60	P	9	270
	350	MA1101	60	P	9	270
	450	MA1102	60	P	9	270
	550	MA1501	70	P	4	120
	650	MA1302	70	P	4	120
	750	MA1401	70	P	4	120
	850	MA1902	70	P	4	120

Module	*modNo#,*	*name,*	*cp,*	*level[en],*	*cycle[en],*	*lang[en]*
	MA6012	Bachelor's Thesis	12	Bachelor	WS/SS	de/en
	MA1001	Analysis 1	9	Bachelor	WS	de
	MA1002	Analysis 2	9	Bachelor	SS	de
	MA1101	Lineare Algebra 1	9	Bachelor	WS	de
	MA1102	Lineare Algebra 2	9	Bachelor	SS	de
	MA1501	Einf. Diskrete Mathematik	4	Bachelor	WS	de
	MA1302	Einf. Numerik	4	Bachelor	WS	de
	MA1401	Einf. Wahrscheinlichkeitstheorie	4	Bachelor	WS	de
	MA1902	Einf. Math. Modellbildung	4	Bachelor	SS	de

Elem_Mod	*modNo#,*	*elem.id#,*	*bind[en],*	*cp*
	MA6012	10	P	12
	MA1001	20	P	9
	MA1001	30	P	0
	MA1001	40	P	0
	MA1001	50	P	0

Element	*id#,*	*name,*	*elType[en]*	*cycle[en],*	*lang[en]*
	10	Bachelor's Thesis	Prüfung	WS/SS	de/en
	20	Analysis 1	Prüfung	WS	de
	30	Analysis 1 (MA1001)	Angebot	WS	de
	40	Tutorübungen Analysis 1	Angebot	WS	de
	50	Übungen zur Analysis 1	Angebot	WS	de

Table 3: Sample validation

6 Conclusion

Our language-based method for conceptual data modelling applies the approaches of [37; 39 and 3]. We normalise NL in a terminology, capturing the UoD semantics in clearly and unambiguously defined modelling terms. Our structural model is built by means of an ontological well-founded ML, OntoUML.

As a foundational ontology, UFO is a theory about the structure of the real world. By linking NL terms to UFO constructs we map the UoD semantics, captured in NL terms, to ontological constructs. This facilitates interpreting a NL's term as a component of the real world constitution with specific properties. We provide this mapping by comparing the meta-properties of terms resulting from Ortner's “Begriffskalkül” with those of UFO-constructs. The complete interpretation and clear representation mapping of UFO-constructs to ML constructs in the sense of [39, p. 221f] leads directly to an unambiguous structural model following the grammatical rules of OntoUML (see [11, chap. 8]). OntoUML restricts the UML2 ML in the same way, as the normalisation of NL terms restricts the (common-) speech in the UoD.

We applied our approach in the domain of campus-management systems to create an ontologically founded reference model as a necessary prerequisite of sustainable (standard-)software. Such a model provides a base for further communication seeming to be very urgent in the ongoing 'Bologna Process'. As an important element of this process, a language-based reference model should be helpful for the improvement of today's pilot installations. Our first validations intensified the impression that the *actual* 'language landscape' seems to be more diverging than converging.

References

1 Bick, M.; Grechenig, T.; Spitta, T. (2010): Campus-Management-Systeme. In: *Pietsch, W.; Krams, B.* (ed): Vom Projekt zum Produkt. Workshop Aachen, Lecture Notes in Informatics 178, Koellen, Bonn: 61-78.

2 Bode, A.; Borgeest, R. (2010): Informationsmanagement in Hochschulen, Springer, Berlin et al.

3 Castro, L.; Baião, F.; Guizzardi, G. (2011): A Semantic Oriented Method for Conceptual Data Modeling in OntoUML Based on Linguistic Concepts. In: Jeusfeld, M.; Delcambre, L.; Ling, T.-W. (ed.): Conceptual Modelling, LNCS 6998, Springer, Berlin et al.: 486-494.

4 Chomsky, N. (2002): Syntactic Structures. Mouton de Gruyter, Berlin et al.

5 Dixon, R. M. W. (2005): A Semantic Approach to English Grammar. Oxford University Press.

6 Ernst & Young (2011): Campus-Management zwischen Hochschulautonomie und Bologna-Reform. Studie.
http://www.ey.com/Publication/vwLUAssets (24.5.2013).

7 Fettke, P.; Loos, P. (2004): Referenzmodellierungsforschung. In: Wirtschaftsinformatik, 46(5): 331-340.

8 Greenspan, Sol Jaffe (1984): Requirements Modeling: Knowledge Representation Approach to Software Requirements Definition. Doctoral Disseration.

9 Guarino, N.: Formal Ontology and Informations Systems. In: Guarino, N. (ed.) (1998): Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy: 3-15.

10 Guarino, N.; Oberle, D.; Staab, S. (2009): What Is an Ontology. In: : 1-17.

11 Guizzardi, G. (2005): Ontological foundations for structural conceptual models. CTIT Ph.D.-thesis series No. 05-74, Center Telematics and Information Technology, University of Twente.

12 Guizzardi, G.; de Almeida Falbo, R.; Guizzardi, R. S. S. (2008): Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. In:

Lencastre, M.; Falcão e Cunha, J.; Valecillo, A. (ed.): 11th Iberoamerican Conference on Software Engineering (CibSE'2008), Recife, Brasil: 127-140.

13 Guizzardi, G.; Pinheiro das Gracas, A.; Guizzardi, R. S. S. (2011): Design Patterns and Inductive Modeling Rules to Support the Construction of Ontologically Well-Founded Conceptual Models in OntoUML. In: Salinesi, C.; Pastor, O. (ed.): Advanced Information Systems Engineering Workshops, LNBIP vol. 83, Springer, Berlin et al.: 402-423.

14 Guizzardi, G.; Wagner, G.: Tutorial: Conceptual simulation modeling with Onto-UML. In: Laroque, C. et al. (eds.): Proceedings of the 2012 Winter Simulation Confernce (WSC'12), Berlin, Germany, December 9-12, 2012. IEEE: 52-66.

15 Hesse, W; Mayr, H. C. (2008): Modellierung in der Softwaretechnik: Eine Bestandsaufnahme. Informatik-Spektrum 31(5): 377-393.

16 Hochschulrektorenkonferenz: http://www.hochschulkompass.de/hochschulen/download.html, abgerufen 23.05.2013.

17 Kauermann, G.; Küchenhoff, H. (2011): Stichproben. Springer, Heidelberg et al.

18 Kop, C.; Mayr, H.C. (1998): Conceptual Predesign: Bridging the Gap between Requirements and Conceptual Design. In: Proceedings of the 3th IEEE International Conference on Requirements Engineering (ICRE'98), Colorado Springs, CO, U.S.A, April 6-10, 1998. Los Alamitos, CA: IEEE Computer Society, 90-98.

19 Lorenzen, P. (1974): Konstruktive Wissenschaftstheorie. Suhrkamp-Taschenbuch Wissenschaft. Frankfurt am Main: Suhrkamp.

20 Masolo, C.; Borgo, S.; Gangemi, A.; Guarino, N.; Oltramari, A. (2003): WonderWeb deliverable D18. Ontology Library (final). http://www.loa.istc.cnr.it/Papers/D18.pdf, abgerufen 27.8.2013.

21 Mylopoulus, J.: Conceptual Modelling and Telos. In: Loucopoulos, P.; Zicari, R. (ed.) (1992): Conceptual Modeling, Databases, and CASE: An Integrated View of Information Systems Development, Wiley, New York: 49-68.

22 Ortner, E. (1985): Semantische Modellierung – Datenbankentwurf auf der Ebene der Benutzer. Informatik Spektrum 8(1): 20-28.

23 Ortner, E.; Söllner, B. (1989): Semantische Datenmodellierung nach der Objekttypenmethode. Informatik Spektrum 12(1): 82-92.

24 Rubin, E.; Wand, Y. (2007): A Framework Supporting the Utilization of Domain Knowledge Embedded in Software. In: Grundy, John u.a. [eds]: Tutorials, posters, panels and industrial constributions at the 26th International Conference on Conceptual Modeling – ER 2007, Auckland, New Zealand, December 2007. Vol. 83. Darlinghurst, Australia: Australian Computer Society: 85-90.

25 Plattner, H. (1981): Systems R/SAP – Real Time System. In: Goos, G. (ed.): Werkzeuge der Programmiertechnik. GI-Working Conf., March 1981, IFB 43, Springer: 244-260.

26 Searle, J. R. (2006): Social Ontology: Some Basic Principles. In: Anthropological Theory 6(1): 12-29.

27 Santos Jr., P. S.; Almeida, J. P. A.; Guizzardi, G. (2010): An Ontology-Based Semantic Foundation for Organizational Structure Modelling in the ARIS Method. In: Workshop Proc. 14th IEEE International Enterprise, EDOCW '10: 272-282.

28 Schillbach, H.; Schönbrunn, K.; Strahringer, S. (2009): Off-the-Shelf Applications in Higher Education: A Survey on Systems Deployed in Germany. In: Abramowicz, W.; Flejter, D. (ed.): BIS 2009 Workshop, LNBIP 37, Springer, Berlin et al.: 242-253.

29 Seidewitz, E. (2003): What models mean. IEEE Software, 20(5): 26-32.

30 Spitta, T.; Mordau, J. (1995): Entwicklung und Ergebnisse eines allgemeingültigen Fachkonzeptes für die Prüfungsverwaltung an Hochschulen. In: Schäfer, J.P.; Grauer, M. (ed.): Universitätsverwaltung und Wirtschaftsinformatik. Proceedings, Siegen: 128-147. http://pub.uni-bielefeld.de/publication/2669014

31 Spitta, T. (1996): Wiederverwendbare Attribute als Ordnungsfaktor der Unternehmensdaten – Konzept und empirische Analyse. In: Ortner, E.; Schienmann, B.; Thoma, H. (Ed.): Natürlichsprachlicher Entwurf von Informationssystemen, Workshop Emisa, Tutzing, Universitätsverlag Konstanz: 79-93.

32 Spitta, T. (1997): Standardsoftware zur Verwaltung und Führung von Fakultäten. Invited Talk GI `97. Published: Univ. Bielefeld, Fakultät für Wirtschaftswissenschaften, Diskussion Paper 354. http://pub.uni-bielefeld.de/publication/2669574 (15.06.2014)

33 Staab, S.; Studer, R. (ed.) (2009): Handbook on Ontologies. Springer, Dordrecht et al.

34 TU Munich 1 (2008): Hintergrundinformationen zum Projekt CM@TUM vom 19.02.2008 https://portal.mytum.de/iuk/cm/dokumente/00.allgemein, abgerufen 02.05.2013.

35 TU Munich 2 (2013): Bachelorstudium Mathematik. Studienplan für das Studienjahr 2013. https://campus.tum.de/tumonline/wbstpcs.showSpoTree?pStpStpNr=343&pStpKnotenNr=25328, abgerufen am 24.07.2013

36 vom Brocke, J. (2002): Referenzmodellierung – Gestaltung und Verteilung von Konstruktionsprozessen. PhD Thesis, University of Muenster.

37 Wedekind, H. (1979): Eine Methodologie zur Konstruktion des Konzeptionelles Schemas. In: Niedereichholz, J. (ed.): Datenbanktechnologie: Einsatz großer, verteilter und intelligenter Datenbanken; Workshop Bad Nauheim, Stuttgart: 65-79.

38 Wand, Y.; Weber, R. (1990): An Ontological Model of an Information System. IEEE Transactions on Software Engineering 16(11): 1282-1292.

39 Wand, Y.; Weber, R. (1993): On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. Information Systems Journal 3(4): 217-237.

40 Wand, Y.; Weber, R. (2002): Research Commentary: Information Systems and Conceptual Modelling. Information Systems Research 13(4): 363-376.