Chapter XIII

Some Applications of a Unified Foundational Ontology in Business Modeling

Giancarlo Guizzardi, University of Twente, The Netherlands

Gerd Wagner, Eindhoven University of Technology, The Netherlands

Abstract

Foundational ontologies provide the basic concepts upon which any domain-specific ontology is built. This chapter presents a new foundational ontology, UFO, and shows how it can be used as a guideline in business modeling and for evaluating business modeling methods. UFO is derived from a synthesis of two other foundational ontologies, GFO/GOL and OntoClean/DOLCE. While their main areas of application are natural sciences and linguistics/cognitive engineering, respectively, the main purpose of UFO is to provide a foundation for conceptual modeling, including business modeling.

Introduction

A *foundational ontology*, sometimes also called “upper level ontology”, defines a range of top-level domain-independent ontological categories, which form a general foundation for more elaborated domain-specific ontologies. A well-known example of a foundational ontology is the *Bunge-Wand-Weber (BWW)* ontology proposed by Wand and Weber in a series of articles (e.g., Wand & Weber, 1990, 1995) on the basis of the original metaphysical theory developed by Bunge (1977, 1979).

As has been shown in a large number of recent works (e.g., Green & Rosemann, 2000; Evermann & Wand, 2001; Guizzardi, Herre, & Wagner, 2002a, b; Opdahl & Henderson-Sellers, 2002), foundational ontologies can be used to evaluate conceptual modeling languages and to develop guidelines for their use. Business modeling can be viewed as the main application domain of conceptual modeling languages and methods. In the *model-driven architecture* approach of the Object Management Group (OMG), a business model is called a “computation-independent model” because it must not be expressed in terms of IT concepts, but solely in terms of business language. The business domain, since it contains so many different kinds of things, poses many challenges to foundational ontologies.

A unified foundational ontology represents a synthesis of a selection of foundational ontologies. Our main goal in making such a synthesis is to obtain a foundational ontology that is tailored towards applications in conceptual modeling. For this purpose we have to capture the ontological categories underlying natural language and human cognition that are also reflected in conceptual modeling languages such as ER diagrams or UML class diagrams. In Gangemi, Guarino, Masalo, Oltramari, and Schneider, (2002) this approach is called “descriptive ontology” as opposed to “prescriptive ontology”, which claims to be “realistic” and robust against the state of the art in scientific knowledge.

For UFO 0.2, the second[1] (still experimental) version of our unified foundational ontology (UFO), we combine the following two ontologies: 1) the general formal ontology (GFO), which is underlying the general ontological language (GOL) developed by the OntoMed research group at the University of Leipzig, Germany; (see *www.ontomed.de* and Degen, Heller, Herre, & Smith, 2001); 2) the OntoClean ontology (Welty & Guarino, 2001) and the descriptive ontology for linguistic and cognitive engineering (DOLCE), developed by the ISTC-CNR-LOA research group in Italy, as part of WonderWeb Project (see *http://wonderweb.semanticweb.org/*).

Existing foundational ontologies, notably SUO, OntoClean-DOLCE, GFO-GOL, and even BWW, all have severe limitations in their ability to capture the basic concepts of conceptual modeling languages. For instance,

1. SUO, OntoClean-DOLCE, and BWW do not make a clear distinction between entities and sets, which is needed to capture the characteristic difference between *entity type* and *datatype*.
2. SUO, GFO-GOL, and BWW do not include an ontology of entity type categories, which is needed to capture the categories of *role types, phase types,* and *mixin types*.
3. SUO, GFO-GOL, and BWW do not pay much attention to the sphere of intentional and social things with the core category of *agents*, which is needed to capture the characteristics of business processes.

UFO does not have these (and some other) shortcomings of SUO, OntoClean-DOLCE, GFO-GOL, and BWW. Our choice to use OntoClean-DOLCE and GFO-GOL as its basis rests on the fact that these two ontologies offer more constructs that are relevant to conceptual modeling than the other foundational ontologies. Specifically, OntoClean-DOLCE include an ontology of entity type categories and an account of agents, while GFO-GOL includes the fundamental distinction between entities and sets.

We have obtained our synthesis by: 1) selecting categories from the union of both category sets; 2) renaming certain terms in order to create a more "natural" language; and 3) adding some additional categories based on relevance for conceptual modeling according to our experience.

Using the acronyms "BWW", "owl", "UML", "ISO", and "BSBR", we also make references to BWW, the Web ontology language OWL (W3C, 2004), the Unified Modeling Language (UML), the terminology standard ISO1087-1:2000 (ISO, 2000), and to the *Business Rules Team submission* to the OMG Business Semantics for Business Rules RFP (Chapin, Hall, Ross, Morgan, & Baisley, 2004). For making a distinction between terms used differently in different vocabularies, we use the XML namespace prefix syntax and write, for example, "BWW:thing" and "owl:Thing" for distinguishing between the concepts termed "thing" in BWW and in OWL.

We present UFO 0.2 both as a MOF/UML model (OMG, 2004) and as a vocabulary in semi-structured English, similar to the *BSBR Structured English* of Chapin et al. (2004). MOF/UML is a fragment of the UML class modeling language that is recommended by the OMG as a language for defining modeling languages; in other words, MOF/UML is a *meta-modeling language*. There are two reasons for using MOF/UML for defining a foundational ontology: first, it allows to express it graphically in the form of a UML class diagram; second, it facilitates the communication of the foundational ontology by making it accessible to the large (and still growing) language community of people familiar with the UML.

An alternative, and more flexible, mode of expression for defining a modeling language such as UFO consists of using semi-structured English to specify the vocabulary of the modeling language. Our UFO vocabulary has three kinds of entries marked up with different font styles:

- term : A term in this font style denotes being of a type and is used to refer to things of that type; for example, the term individual in the phrase "individual that is wholly present whenever it is present" stands for a thing of type "individual" (i.e., it stands for an individual).
- *name* : This is a name of an individual or a type; when abc is a type term referring to things of that type, *abc* is a name referring to the type itself.
- *term1 relationship phrase term2* : This is a name of a binary relationship type.

A vocabulary entry may contain, additionally,

- "Corresponding terms" (or "corresponding relationship type expressions"): terms (or relationship type expressions) that are roughly equivalent.
- Examples.
- Constraints: logical statements that have to hold in any given ontology based on UFO.

When there is a primary source for a definition, we append it in brackets, like in "...[based on GFO]".

UFO is divided into three incrementally layered *compliance sets*: 1) UFO-A defines the core of UFO, excluding terms related to perdurants and terms related to the spheres of intentional and social things; 2) UFO-B defines, as an increment to UFO-A, terms related to perdurants; and 3) UFO-C defines, as an increment to UFO-B, terms related to the spheres of intentional and social things, including linguistic things.

This division reflects a certain stratification of our "world". It also reflects different degrees of scientific consensus: there is more consensus about the ontology of endurants than about the ontology of perdurants, and there is more consensus about the ontology of perdurants than about the ontology of intentional and social things.

We hope that this division into different compliance sets will facilitate both the further evolution of UFO and the adoption of UFO in business modeling and ontology engineering. In the next section, we present UFO-A 0.2, while UFO-

B 0.2 and UFO-C 0.2 are presented in the two subsequent sections, respectively. The next section illustrates how UFO can be used to evaluate some business modeling methods, and the final section concluding the chapter.

UFO-A: The Core of the Unified Foundational Ontology

Things, Sets, Entities, Individuals, and Types

We first present the upper part of UFO-A 0.2 as a MOF/UML model in Figure 1. Notice the fundamental distinction made between *sets* and *entities* as things that are not sets (called "urelements" in GFO).

In structured English, the upper part of UFO 0.2 can be introduced as follows.

- **thing** : This is anything perceivable or conceivable [ISO:object]. *Corresponding terms*: GFO:entity; DOLCE:entity, owl:Thing; BSBR:thing.
- **set**: This is a thing that has other things as members (in the sense of set theory).
- ***thing is member of set*** : This is the name of a formal relationship type that is irreflexive, asymmetric and intransitive.
- ***member*** : This is the role name that refers to the first argument of the *thing is member of* set relationship type.
- ***set is subset of set*** : This is the name of a formal relationship type that is reflexive, asymmetric and transitive. *Constraint*: For all t:thing; s_1, s_2 : set — if t is member of s_1 and s_1 is subset of s_2, then t is member of s_2
- **entity** :This is a thing that is not a set; neither the set-theoretic membership relation nor the subset relation can unfold the internal structure of an entity [GFO:urelement].
- **entity type** : This is an entity that has an extension (being a set of entities that are instances of it) and an intension, which includes an applicability criterion for determining if an entity is an instance of it; and which is captured by means of an axiomatic specification, that is, a set of axioms that may involve a number of other entity types representing its essential features. An entity type is a space-time independent pattern of features, which can be realized in a number of different individuals [based on

GFO:universal]. *Corresponding terms*: UML:class; DOLCE:universal; owl:Class; BSBR: "generic thing".

- ***entity is instance of entity type***: This is the name of a formal relationship type (called *classification*).
- ***instance***: This is a role name that refers to the first argument of the *entity is instance of entity type* relationship type.
- ***set is extension of entity type***: This is the name of a formal relationship type. *Constraint:* For all *o*:entity, *t*:entity type, *s*:set — if *o* is instance of *t* and *s* is extension of *t*, then *o* is member of *s*.
- ***extension***: This is a role name that refers to the first argument of the *set is extension of entity type* relationship type.
- ***entity type is subtype of entity type***: This is the name of a formal relationship type that is irreflexive, asymmetric and transitive (also called *generalization*). *Constraint:* For all t_1, t_2 : entity type; s_1, s_2 : set — if t_1 is subtype of t_2 and s_1 is extension of t_1 and s_2 is extension of t_2, then s_1 is subset of s_2.
- ***subtype***: This is a role name that refers to the first argument of the *entity type is subtype of entity type* relationship type.
- **individual**: This is an entity that is not an entity type. An entity type that classifies individuals is called individual type. *Corresponding terms*: GFO:individual; DOLCE: particular.
- ***thing is part of individual***: This is name of a formal relationship type that is reflexive, asymmetric and transitive (also called *aggregation*).
- ***part***: This is a role name that refers to the first argument of the *thing is part of individual* relationship type.
- ***entity type is classification type of entity type***: This is the name of a formal relationship type where the first argument is a higher-order entity type whose instances form a subtype partition of the second argument (also called *higher-order classification*). *Examples*: BiologicalSpecies is classification type of Animal; PassengerAircraftType is classification type of PassengerAircraft. *Constraint*: For all t_1, t_2, t_3: entity type — if t_3 is classification type of t_1 and t_2 is instance of t_3, then t_2 is subtype of t_1.
- ***classification type***: This is a role name that refers to the first argument of the *entity type is classification type of entity type* relationship type. *Corresponding names*: GFO: "higher-order universal"; BSBR: "categorization type"; UML: powertype.
- ***entity type is classified by entity type***: This is the name of a formal relationship type that is the inverse of the *entity type is classification type*

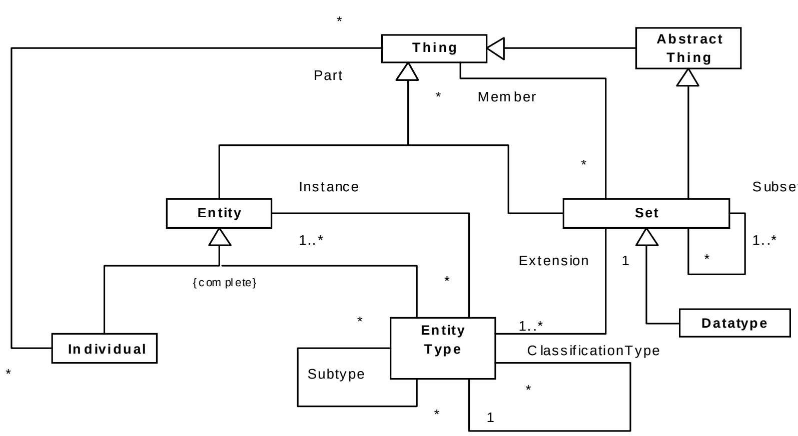
of <u>*entity type*</u> <u>relationship type</u>. *Corresponding relationship type expressions*: BSBR: “type has categorization-scheme”.

Different Kinds of Types

In UFO, we make a fundamental distinction between *datatypes*, which are sets, and *entity types*, which are not sets, but whose extensions are sets. Based on Wiggins (2001), van Leeuwen (1991), Gupta (1980), and Hirsch (1982), we distinguish between several different kinds of entity types, as shown in Figure 2. These distinctions are elaborated in Guizzardi, Wagner, and van Sinderen (2004a), in which we present a philosophically and psychologically well-founded theory of types for conceptual modeling. In Guizzardi, Wagner, Guarino, and van Sinderen (2004b), this theory is used to propose: 1) a profile for UML whose elements represent finer-grained distinctions between different kinds of types and 2) a set of constraints defining the admissible relations between these elements. One should refer to Guizzardi et al. (2004a, 2004b) for: a) an in-depth discussion of the theory underlying these categories as well as the constraints on their relations; b) a formal characterization of the profile; and c) the application of the profile to propose an ontological design pattern that addresses a recurrent problem in the practice of conceptual modeling.

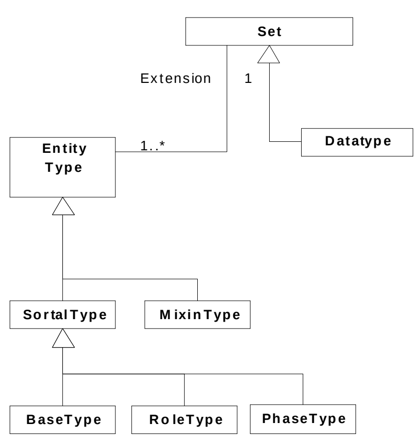
In structured English, the different kinds of types are defined as follows.

Figure 1. The upper part of UFO-A 0.2 as a MOF/UML model



- **datatype**: This is a set whose members are data values. *Examples*: Integer; String.
- **sortal type**: This is an entity type that carries a criterion for determining the individuation, persistence, and identity of its instances. An identity criterion supports the judgment whether two instances are the same. Every instance in a conceptual model must have an identity and, hence, must be an instance of **sortal type**.
- **base type**: This is a sortal type that is rigid (all its instances are necessarily its instances) and that supplies an identity criterion for its instances [OntoClean:type]. *Examples*: Mountain; Person. *Corresponding terms*: BWW: "natural kind".
- **phase type**: This is a sortal type that is anti-rigid (its instances could possibly also not be instances of it without loosing their identity) and that is an element of a subtype partition of a base type [OntoClean:"phased sortal"]. *Examples*: Town and Metropolis are phase subtypes of City; Baby, Teenager, and Adult are phase subtypes of Person.
- **role type**: This is a sortal type that is anti-rigid and for which there is a relationship type such that it is the subtype of a base type formed by all instances participating in the relationship type [OntoClean:role]. *Examples*: DestinationCity as role subtype of City; Student as role subtype of Person.
- **mixin type**: This is an entity type that is not a sortal type and can be partitioned into disjoint subtypes, which are sortal types (typically role

Figure 2. Different kinds of types in UFO-A 0.2



types) with different identity criteria. Since a mixin is a non-sortal it cannot have direct instances [OntoClean:non-sortal]. *Examples*: Object; Part; Customer; Product

- **relationship type**: This is a type whose instances are (material or formal) relationships .

Notice that role types and phase types cannot supply an identity criterion for their instances. For this reason, they must be derived from suitable base type from which they inherit their identity criterion.

The theory of types, which is part of UFO-A, provides a foundation for a number of modeling primitives that, albeit often used, are commonly defined in an ad hoc manner in the practice of conceptual modeling. In particular, this theory can be considered as an extension of the BWW account of types. In Evermann and Wand (2001), it is proposed that a UML class should be used to represent a BWW: “natural kind” (i.e., it should be equivalent to the “functional schema” of a BWW: “natural kind”). As discussed in Guizzardi et al. (2004a), the concept of a natural kind corresponds to the UFO concept of a base type, that is, a natural kind is a rigid entity type that provides an identity criterion for its instances. It has been argued, however, (e.g., Welty & Guarino, 2001; Gupta, 1980; Wiggins, 2001; van Leeuwen, 1991; Guizzardi et al., 2004a, 2004b), that, in addition to this concept, several other type concepts are needed in descriptive ontologies and in conceptual modeling.

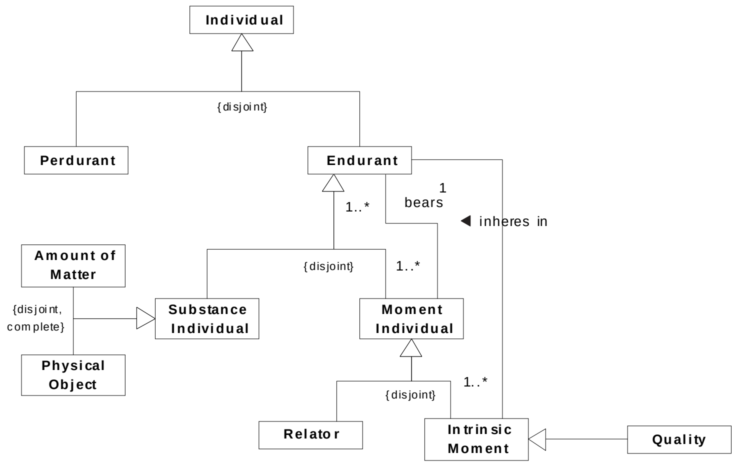
Different Kinds of Individuals

We distinguish between a number of different kinds of individuals, as shown in Figure 3. The fundamental distinction between *endurants* and *perdurants* corresponds to the colloquial distinction between “objects” and “processes”.

In structured English, the different kinds of individuals considered in UFO are explained as follows.

- **endurant**: This is an individual that is wholly present whenever it is present, that is, it does not have temporal parts, and that persists in time while keeping its identity [DOLCE]. *Examples*: a house; a person; the moon; a hole; the redness of a certain apple; an amount of sand. *Corresponding terms*: GFO:3D-individual.
- **perdurant**: This is an individual that is composed of temporal parts; whenever a perdurant is present, it is not the case that all its temporal parts

Figure 3. Different kinds of individuals in UFO-A 0.2



are present [DOLCE]. *Examples*: a storm; a heart attack; a conversation; the Second World War; a business process.

- **substance individual**: This is an endurant that consists of matter (i.e., is "tangible" or concrete), possesses spatio-temporal properties, and can exist by itself; that is, it does not existentially depend on other endurants, except possibly on some of its parts) [based on GFO:substance]. *Examples*: a house; a person; the moon; an amount of sand. *Corresponding terms*: BWW:thing
- **moment individual**: This is an endurant that cannot exist by itself; that is, it depends on other endurants, which are not among its parts [based on GFO:moment]. *Examples*: the redness of a certain apple; a belief of George Bush; a flight connection between two cities.
- *endurant bears moment individual*: This is the name of a formal relationship type [based on GFO: "substance bears moment"].
- **physical object**: This is a substance individual that satisfies a condition of unity and for which certain parts can change without affecting its identity. *Examples*: a house; a person; the moon.
- **amount of matter**: This is a substance individual that does not satisfy a condition of unity; typically referred to by means of mass nouns. An amount of matter is *mereologically invariant*, that is, it cannot change any of its parts without changing its identity [DOLCE]. *Examples*: a liter of water; a piece of gold; a pile of sand.

- **intrinsic moment**: This is a moment individual that is existentially dependent on one single individual. *Examples*: the redness of a certain apple; a belief of George Bush.
- ***intrinsic moment inheres in endurant***: This is the name of a formal relationship type [GFO].
- **quality**: This is an intrinsic moment that inheres in exactly one endurant and can be mapped to a value (DOLCE:quale) in a *quality dimension* (Gärdenfors, 2000). *Corresponding terms*: GFO:quality; DOLCE:quality; BWW: "intrinsic property". *Examples*: the color (height, weight) of a physical object; an electric charge. *Constraint*: For all e_1, e_2 : endurant; q:quality — if q inheres in e_1 and q inheres in e_2, then e_1 is equal to e_2. *Examples*: the redness of a certain apple.
- **relator**: This is a moment individual that is existentially dependent on more than one individual. Relators provide the basis for material relationships (Guizzardi, Herre, & Wagner, 2002b) [GFO:relator]. *Corresponding terms*: BWW: "mutual property", UML:link, owl:. *Examples*: a particular employment (Susan is employed by IBM); a particular flight connection (LH403 flies from Berlin to Munich).

The notion of relators is supported in several works in the philosophical literature (see, e.g., Smith & Mulligan, 1983, 1986). The concept of relators plays an important role in:

1. distinguishing material relationship types, such as "person *is married to* person" and "person *studies at* university", from formal relationship types, such as "number *is greater than* number" and "day is *part-of* month");
2. answering questions of the sort: What does it mean to say that John is married to Mary? Why is it true to say that Bill works for Company X but not for Company Y?

Putting all UFO-A terms and relationship-type expressions together in one UML/MOF diagram results in Figure 9 (see Appendix A).

Some Applications of UFO-A to Business Modeling Problems

Modeling Customers

Most business information systems include a "business object class" Customer for representing the customers of the business. In Figure 4, the role type *Customer* is defined as a supertype of *Person* and *Corporation*. This model is deemed ontologically incorrect for two reasons: first, not all persons are customers, that is, it is not the case that the extension of *Person* is necessarily included in the extension of *Customer*. Moreover, an instance of *Person* is not necessarily (in the modal sense) a *Customer*. Both arguments are also valid for *Organization*. In a series of papers (e.g., Steimann, 2000), Steimann discusses the difficulties in specifying supertypes for roles that can be filled by instances of disjoint types.[2] As a conclusion, he claims that the solution to this problem lies in separating the hierarchies of role type and base type (named natural type in the article) — a solution, which strongly impacts the meta-model of all major conceptual modeling languages. By using the theory of types underlying UFO-A we can show that this claim is not warranted and we are able to propose a *design pattern* that can be used as an ontologically correct solution to this recurrent problem (Guizzardi et al, 2004b).

In this example, Customer has in its extension individuals that obey different identity criteria, that is, it is not the case that there is a single identity criterion, which applies both for Persons and Corporations. Customer is hence a mixin type (a non-sortal). Since every instance in the model must have an identity, thus, every instance of Customer must be an instance of one of its subtypes (forming a partition) that carries an identity criterion. For example, we can define the sortals PersonalCustomer and CorporateCustomer as subtypes of Customer (Figure 5). These sortals, in turn, carry the (incompatible) identity criteria supplied by the base types Person and Corporation, respectively.

Product Modeling

In many business information systems, both individual products and product types have to be represented. In a prototypical case, the product individual type, whose instances are identified with the help of serial numbers, is classified by the corresponding product model type, which is a second order classification type, whose instances are subtypes of the product individual type. Figure 6 shows this situation for the case of cars and car models.

Figure 4. An ontologically incorrect model of the entity type Customer

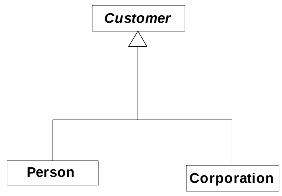


Figure 5. An ontologically-correct model of the Customer entity type according to UFO 0.2

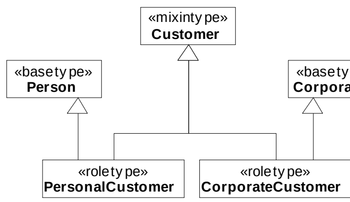
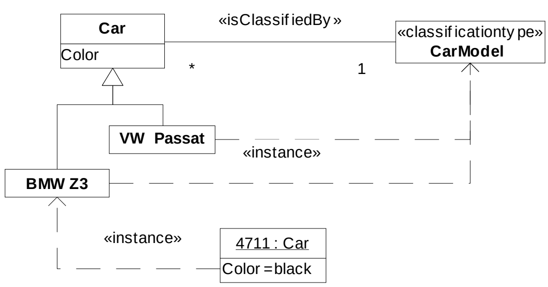


Figure 6. UML product modeling with UFO-based stereotypes



In a proposal for the ontological foundations of the *Resource-Event-Agent (REA)* model (Geert & McCarthy, 2000, p. 13), the authors argue about the importance of the distinction between individual types and classification types accounted here:

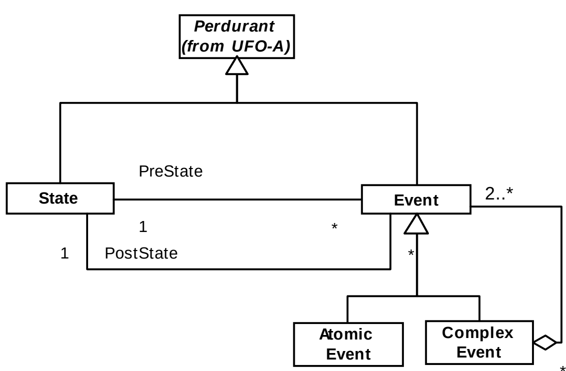
Economic Resources like (especially) inventory have an instance/ type definition problem that must be solved in the REA ontology (or in any information system)… cars in an automobile dealership would be modeled with instances (a car with a given engine#) …with classes of cars (1975 Corvette) as type-images.

UFO-B: The Ontology of Perdurants

A complete treatment of an ontology of perdurants requires an ontology of *temporal entities* (GFO:chronoids) (Degen et al., 2001). In this section, instead, we restrict our attention to the most basic perdurant categories for defining UFO-B 0.2 as a foundation for defining some intentional and social entities later. In the sequel we discuss the following basic kinds of perdurants shown in Figure 7: (atomic and complex) events and states.

- **state**: This is a perdurant that is *homeomeric*, that is, each of its temporal parts belongs to the same state type as the whole [based on DOLCE].
- **event**: This is a perdurant that is related to exactly two states (its pre-state and its post-state). An event is related to the states before and after it has happened.
- **atomic event**: This is an event that happens instantaneously, that is, an event without duration, relative to an underlying time granularity [based on BWW:event and GFO:change]. *Examples*: an explosion; a message reception.

Figure 7. The perdurant categories of UFO-B 0.2



- **complex event**: This is an event that is composed of other events by means of event composition operators. *Examples*: a parallel occurrence of two explosions; an absence of a message reception (within some time window); a storm; a heart attack; a football game; a conversation; a birthday party; the Second World War; a Web shop purchase.
- **process**: This is a complex event that is a sequence of two or more (possibly parallel occurrences of) atomic events. *Examples*: a storm; a heart attack; a football game; a conversation; a birthday party; the second World War; a Web shop purchase.
- ***state is pre-state of event***: This is a name of a formal relationship type.
- ***state is post-state of event***: This is the name of a formal relationship type.

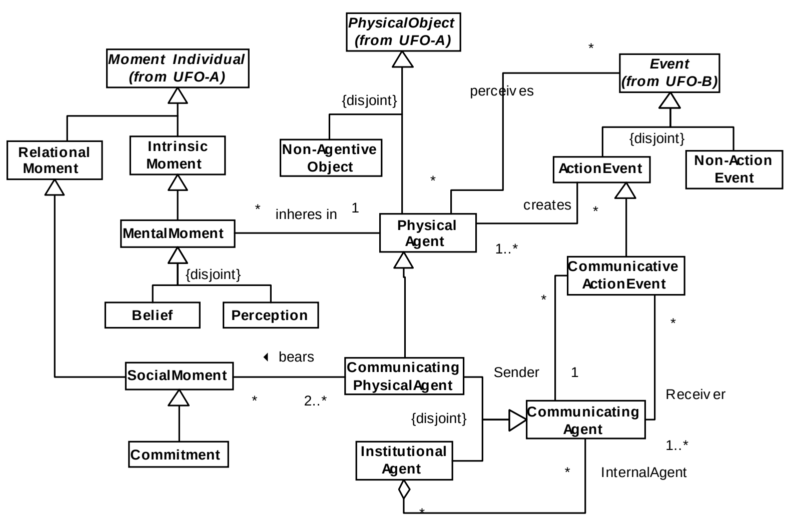
UFO-C: The Ontology of Intentional, Social, and Linguistic Things

The "objective" perdurant categories *(atomic and complex) event* and *state* defined in UFO-B are essential concepts for process modeling, but they are not sufficient for *business process* modeling, where intentional and social concepts such as *action*, *activity*, and *communication* are needed. The following account of intentional and social things is at an early stage of development and therefore rather incomplete. Nevertheless, we think that it gives an impression of the range of ontological categories that is needed to explain business process modeling.

- **physical agent**: This is a physical object that creates action events affecting other physical objects, that perceives events, possibly created by other physical agents, and to which we can ascribe a mental state. *Examples*: a dog; a human; a robot.
- **action event**: This is an event that is created through the action of a physical agent.
- **non-action event**: This is an event that is not created through an action of a physical agent.
- ***physical agent creates action event***: This is the name of a formal relationship type.
- ***physical agent perceives event***: This is the name of a formal relationship type.

- **non-agentive object**: This is a physical object that is not a physical agent. *Examples*: a chair; a mountain .
- **mental moment**: This is an intrinsic moment that is existentially dependent on a particular agent, being an inseparable part of its mental state. *Examples*: a thought; a perception; a belief; a desire; an individual goal. *Constraint*: For all *mm* : mental moment; *e*:endurant — if *mm* inheres in *e*, then *e* is physical agent.
- **communicating physical agent**: This is a physical agent that communicates with other communicating physical agents. *Examples*: a dog; a human; a communication-enabled robot.
- **institutional agent**: This is an institutional fact (Searle, 1995) that is an aggregate consisting of communicating agents (its *internal agents*), which share a collective mental state, and that acts, perceives and communicates through them. *Examples*: a business unit; a voluntary association.
- **agent**: This is an endurant that is either a physical agent or an institutional agent.
- **communicating agent**: This is an agent that communicates with other communicating agents.
- **social moment**: This is a moment individual that is existentially dependent on more than one communicating agent. *Examples*: a commitment; a joint intention.

Figure 8. The categories of the UFO-C 0.2 agent ontology



Agents may interact with their inanimate environment, or they may interact with each other, which involves some form of communication; in the latter case, we speak of *social interaction*.

We consider a business process as a special kind of a social interaction process. Unlike physical or chemical processes, social interaction processes are based on communication acts that may create commitments and are governed by norms. We distinguish between an interaction process type and an interaction process individual, while in the literature the term business process is used ambiguously both at the type and at the instance level.

- **interaction process**: This is a process that includes at least one perception event and one action event perceived and performed by agents that participate in it. *Examples*: someone turning on the light in the office when it becomes dark outside; a football game; a conversation; a birthday party; the second World War; a Web shop purchase.
- **social interaction process**: This is an interaction process that includes at least one communicative action event. *Examples*: a football game; a conversation; a birthday party; the second World War; a Web shop purchase.
- **business process**: This is a social interaction process that occurs in the context of a business system and serves a purpose of that system. *Examples*: a football game; a Web shop purchase.

Using UFO to Evaluate Business Modeling Methods

In the following subsections, we briefly present some preliminary results in order to exemplify how UFO can be used to evaluate business modeling methods.

The Enterprise Ontology

The *enterprise ontology*, which was developed in a project led by the AI Applications Institute at the University of Edinburgh (see Uschold, King, Moralee, & Zorgios, 1998). Based on a simple upper-level ontology ("meta-ontology") consisting of the three modeling concepts *entity*, *relationship*, and *actor*, it provides definitions for nearly 100 terms, both in natural language and in the formalism of *Ontolingua*.

For simplicity, the distinction between an individual and an entity type is avoided. An agent (called *actor)* is defined as a special entity that can play an actor role in certain relationships (such as in *performs Activity*, *has Capability*, etc.).

There is no independent concept of an event in the enterprise ontology: events are defined as "a kind of activity". Remarkably, the authors consider also events that take place as a result of natural necessity (such as "water flowing down a hill") as activities of "inanimate actors" (such as gravity).

The following points highlight some shortcomings of the enterprise ontology: 1) For conceptual modeling, it is essential to distinguish between individuals and entity types; 2) It seems to be questionable to view natural forces that cause certain events to happen, such as gravity, as actors/agents; in UFO agents have a mental state and are able to act (create action events), perceive and possibly to communicate; and 3) Events must not be subsumed under activities. Rather, they should be first-class citizens of the meta-model. Unlike events, activities are always associated with an agent (their performer).

The Eriksson-Penker Business Extensions: Subsuming Agents Under Resources

In Eriksson and Penker (1999), an approach to business modeling with UML based on four primary concepts is proposed: *resources*, *processes*, *goals*, and *rules*. In this proposal, there is no specific treatment of agents. They are subsumed, together with "material", "products", and "information" under the concept of *resources*. This unfortunate subsumption of human agents under the traditional "resource" metaphor, which is common in many business modeling methods, prevents a proper treatment of many important agent-related concepts (such as commitments, authorization, communication and interaction).

The REA (Resource-Event-Agent) Model

The REA framework, whose ontological foundations are defined in Geert and McCarthy (2000), is based on a notion of an "economic exchange". An economic exchange comprises a pair of economic events: an inflow and an outflow event. Economic agents participate in economic events and resources are affected (e.g., produced, used, acquired) by these events. In UFO, an economic event is a type of complex action event and resource is a type of substance individual (resources can be physical objects or amounts of matter). In REA, an individual is an (economic) agent by virtue of its participation in an economic event, while in UFO an agent is an individual to which we can ascribe a mental state.

Despite considering both individuals and entity types, the authors do not elaborate on the different sorts of entity types, which are necessary for conceptual enterprise modeling.

An example of lack of ontological clarity is found when the authors mix the notions of event and commitments. For instance, in Figure 5, commitment and economic event are collapsed in one single type-image. Additionally, the relationships *partner* and *reserves* (defined to hold between agent/commitment and resource/commitment, respectively) are considered as subtypes of *participation* and *stock-flow* (defined between agent/economic event and resource/ economic event). In our framework, whilst an economic event is a complex action event, a commitment is a social moment. Examples of other types of social moment defined in REA are *accountability*, *responsibility*, *assignment*, and *custody*.

Despite recognizing the importance of part-whole relations in the enterprise domain (for example to model the relation between a resource and its parts), the treatment offered is insufficient. The authors only briefly mention a relation of composition that, together with other relations such as *substitutes* (meaning that a resource can substitute another), is subsumed under the relation of *linkage* between resources. No axiomatization for composition is provided. In a companion paper (Guizzardi, Herre, & Wagner, 2002b), we provide a formal characterization for parthood and discuss different types of this relation, which are important for conceptual modeling.

Conclusions

The unified foundational ontology UFO 0.2 presented in this chapter should be viewed as an attempt to assemble a foundational ontology for conceptual modeling on the basis of other, already well established and philosophically justified foundational ontologies. We have stratified UFO into three ontological layers in order to distinguish its core, UFO-A, from the perdurant extension layer UFO-B and from the agent extension layer UFO-C. Although there is not much consensus yet in the literature regarding the ontology of agents, such an ontology is needed for building the foundation of conceptual business process modeling. UFO-C 0.2 is a first attempt to construct these foundations. We hope that we can validate and further improve it by investigating its applicability to business modeling problems.

References

Bunge, M. (1977). *Treatise on Basic Philosophy. Vol. 3. Ontology I. The Furniture of the World*. New York: Reidel.

Bunge, M. (1979). *Treatise on Basic Philosophy. Vol. 4. Ontology II. A World of Systems*. New York: Reidel.

Chapin, D., Hall, J., Ross, R., Morgan, T, & Baisley, D. (2004). *Business semantics of business rules (BSBR*). Initial submission to OMG BEI RFP br/2003-06-03, 12 January 2004. Available from *http://www.omg.org/cgi-bin/doc?bei/04-01-04*.

Degen, W., Heller, B., Herre, H., & Smith, B. (2001). GOL: Towards an axiomatized upper level ontology. In B. Smith & N. Guarino (Eds.), *Proceedings of FOIS'01*, Ogunquit, Maine, October 2001. ACM Press. (pp. 34-46).

Eriksson, H. E. & Penker, M. (1999). *Business Modeling with UML: Business Patterns at Work*. New York: John Wiley & Sons.

Evermann, J. & Wand, Y. (2001). Towards ontologically based semantics for UML constructs. In H. S. Kunii, S. Jajodia, & A. Solvberg (Eds.), *Proceedings of ER 2001*, (pp. 354-367). Berlin: Springer-Verlag.

Gangemi, A., Guarino N., Masolo C., Oltramari, A., & Schneider L. (2002). Sweetening ontologies with DOLCE. *Proceedings of EKAW 2002*, Siguenza, Spain.

Gärdenfors, P. (2000). *Conceptual Spaces: The Geometry of Thought*. Boston: MIT Press. (pp. 166-181).

Geert, G. & McCarthy, W. E. (2000). The ontological foundation of REA enterprise information systems. Retrieved from *http://www.msu.edu/user/mccarth4/rea-ontology*.

Green, P. F. & Rosemann, M. (2000). Integrated process modelling: An ontological evaluation. *Information Systems, 25*(2), 73-87.

Green, P. F. & Rosemann, M. (2002). Usefulness of the BWW ontological models as a "core" theory of information systems. *Proceedings Information Systems Foundations: Building the Theoretical Base*, Canberra, 2002, (pp. 147-164).

Guizzardi, G., Herre, H., & Wagner G. (2002a). On the general ontological foundations of conceptual modeling. Lecture notes in *Computer Science, Vol. 2503*. Berlin: Springer-Verlag.

Guizzardi, G., Herre, H., & Wagner, G. (2002b), Towards ontological foundations for UML conceptual models. Lecture notes in *Computer Science, Vol. 2519*. Berlin: Springer-Verlag.

Guizzardi, G., Wagner, G., & van Sinderen, M. (2004a). A formal theory of conceptual modeling universals. *Proceedings of the International Workshop on Philosophy and Informatics (WSPI)*, Cologne, Germany (pp. 73-80).

Guizzardi, G., Wagner G., Guarino, N., & van Sinderen, M. (2004b). An ontologically well-founded profile for UML conceptual models. Lecture notes in *Artificial Intelligence, Vol. 3084*. Berlin: Springer-Verlag.

Gupta, A. (1980). *The Logic of Common Nouns: An Investigation in Quantified Modal Logic*. New Haven, CT: Yale University Press.

Hirsch, E. (1982). *The Concept of Identity*. New York, Oxford: Oxford University Press.

ISO. (2000). ISO 1087-1. Terminology work — Vocabulary — Part 1: Theory and application. Retrieved from *http//:webstore.ansi.org*.

Object Management Group (OMG). (2003). Meta object facility (MOF) 2.0 core specification, version 2.0. Retrieved from *http://www.omg.org/docs/ptc/03-10-04.pdf*.

Opdahl, A. L. & Henderson-Sellers, B. (2002). Ontological evaluation of the UML using the Bunge-Wand-Weber model. *Software and Systems Journal*, *1*(1), 43-67.

Searle, J. R. (1995). *The Construction of Social Reality*. New York: Free Press.

Smith, B. & Mulligan, K. (1986). A relational theory of the act. *Topoi, 5*(2), 115-130.

Smith, B. & Mulligan, K. (1983). Framework for formal ontology. *Topoi, 3*, 73-85.

Steimann, F. (2000). On the representation of roles in object-oriented and conceptual modeling. *Data & Knowledge Engineering*, *35*(1), 83-106.

van Belle, J. P. (1999). Moving towards generic enterprise information models: From Pacioli to CyC. *Proceedings of the 10th Australasian Conference on Information Systems*, Wellington. (pp. 1084-1094).

van Leeuwen, J. (1991). Individuals and sortal concepts: An essay in logical descriptive metaphysics. PhD thesis, University of Amsterdam.

World Wide Web Committee (W3C). (2004). Web ontology language (OWL). Retrieved from *http://www.w3.org/2001/sw/WebOnt*.

Wand, Y. & Weber, R. (1990). Mario Bunge’s ontology as a formal foundation for information systems concepts. In P. Weingartner & G. J. W. Dorn (Eds.), *Studies on Mario Bunge’s treatise.* Atlanta: Rodopi.

Wand, Y. & Weber, R. (1995). On the deep structure of information systems. *Information Systems Journal, 5,* 203-223.

Welty, C & Guarino, N. (2001). Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering*, *39*(1), 51-74.

Wiggins, D. (2001). *Sameness and Substance Renewed*. Cambridge, MA: Cambridge University Press.

Endnotes

[1] UFO 0.2 differs from UFO 0.1, which has been presented at the EMOI-INTEROP Workshop at CAiSE'04, by adding the categories of datatype, process, and business process.

[2] This problem is also mentioned in (van Belle, 1999, p. 1089): "How would one model the customer entity conceptually? The Customer as a supertype of Organisation and Person? The Customer as a subtype of Organisation and Person? The Customer as a relationship between or Organisation and (Organization or Person)?"

Appendix

Appendix A.

Figure 9. UFO-A 0.2 as a MOF/UML model