Towards Ontological Foundations for Agent Modeling Concepts using UFO

Giancarlo Guizzardi[1] and Gerd Wagner[2]

[1] Centre for Telematics and Information Technology, Univ. of Twente
Enschede, The Netherlands
guizzard@cs.utwente.nl
[2] Brandenburg Univ. of Technology at Cottbus, Computer Science Department
Cottbus, Germany
G.Wagner@tu-cottbus.de

Abstract: Foundational ontologies provide the basic concepts upon which any domain-specific ontology is built. This paper presents a new foundational ontology, UFO, and shows how it can be used as a foundation of agent concepts and for evaluating agent-oriented 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 the natural sciences and linguistics/cognitive engineering, respectively, the main purpose of UFO is to provide a foundation for conceptual modeling, including agent-oriented modeling.

1 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, 1989; 1990; 1995) on the basis of the original metaphysical theory developed by Bunge (1977; 1979).

As has been shown in a number of recent works (e.g., Wand, Storey & Weber, 1999; 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. Agent-based conceptual modeling can be viewed as an extension of more traditional conceptual modeling approaches by the explicit consideration of intentional entities. The position defended here is that agent modeling languages should also be based in a foundational ontology that accounts for both the concepts underlying basic conceptual modeling constructs, and their extension in terms of intentional entities.

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, which are also reflected in conceptual modelling languages such as ER diagrams or UML class diagrams. In (Gangemi et al, 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 and Guarino, 2001) and the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) developed by the ISTC-CNR-LOA research group in Trento, Italy, as part of WonderWeb Project; see http://wonderweb.semanticweb.org/.

Our choice is based on personal familiarity and preferences and not on an evaluation of all alternatives. Nonetheless, in previous attempts, GFO has been proven insightful in providing a principled foundation for analyzing and extending conceptual modeling and ontology representation languages and constructs (Guizzardi, Herre & Wagner, 2002a-b; Loebe, 2003).

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 and corresponding theories,

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[2], 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 et al, 2004). For making a distinction between terms used differently in different vocabularies, we use the XML namespace prefix syntax and write, e.g., “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, 2003) 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

[1] UFO 0.2 differs from UFO 0.1, which has been presented at the AOIS Workshop at CAiSE’04, by adding the categories of datatype, process and business process.
[2] http://www.w3.org/TR/owl-semantics/

using MOF/UML for defining a foundational ontology: first, it allows expressing 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; e.g., 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* : 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 predicate term2 : an expression that denotes being of a relationship type and that is used to refer to relationships of that type.

A vocabulary entry may contain, additionally,

- 'Corresponding terms' (or 'corresponding relationship type expressions'): terms (or relationship type expressions) that are roughly equivalent;
- Examples; and
- Constraints: logical statements that have to hold in any given ontology based on UFO 0.2.

When there is a primary source for a definition, we append it in brackets, like [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 conceptual 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 sections 3 and 4, respectively.

2 UFO-A 0.1 – the core of A Unified foundational Ontology

2.1 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).

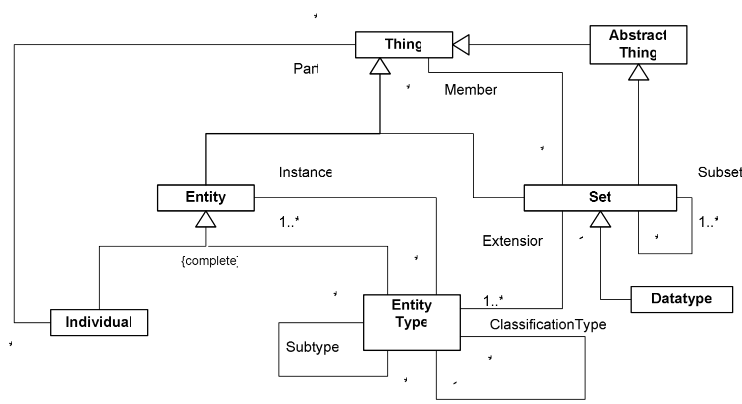


Figure 1: The upper part of UFO 0.2 as a MOF/UML model.

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

thing: anything perceivable or conceivable [ISO:object]. *Corresponding terms*: GFO:entity; DOLCE:entity, owl:Thing; BSBR:thing

set : thing that has other things as members (in the sense of set theory)

thing is member of set : name of a formal relationship type that is irreflexive, asymmetric and intransitive

member : role name that refers to the first argument of the *thing is member of set* relationship type

set is subset of set : 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: 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 : entity that has an extension (being a set of entitys that are instances of it) and an intension, which includes an applicability criterion[3] for determining if an entity is an instance of it; and which is captured by means of an axiomatic specification, i.e., 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 : name of a formal relationship type (called *classification*)

instance : role name that refers to the first argument of the *entity is instance of entity type* relationship type

set is extension of entity type : name of a formal relationship type. *Constraint:* For all e:entity, t:entity type, s:set – if e is instance of t and s is extension of t, then e is member of s.

extension : 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 : 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 : role name that refers to the first argument of the *entity type is subtype of entity type* relationship type

individual : 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 : name of a formal relationship type that is reflexive, asymmetric and transitive (also called *aggregation*). For a fuller treatment of part-whole relations in which we consider both modality and context-sensitivity, one should refer to (Guizzardi & Herre & Wagner, 2002b).

part : 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 : 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.

---

[3] The notion of applicability criterion (or principle of application) and its role in conceptual modeling are discussed comprehensively in (Guizzardi et al, 2004).

classification type : 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 : name of a formal relationship type that is the inverse of the *entity type is classification type of entity type* relationship type. *Corresponding relationship type expressions*: BSBR:”type has categorization-scheme”.

2.2 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; Hirsch, 1982), we distinguish between several different kinds of entity types, as shown in figure 2. These distinctions are elaborated in (Guizzardi, Wagner & van Sinderen, 2004), in which we present a philosophically and psychologically well-founded theory of types for conceptual modeling. In (Guizzardi et al, 2004), this theory is used to propose: (i) a profile for UML whose elements represent finer-grained distinctions between different kinds of types; (ii) a set of constraints defining the admissible relations between these elements. One should refer to (Guizzardi, Wagner & van Sinderen, 2004; Guizzardi et al, 2004) 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; (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.

datatype : set whose members are data values. In UFO, a datatype is a set-theoretic representation of a conceptual space and the constraints imposed by its geometrical structure (see Guizzardi & Wagner & Herre, 2004). *Examples*: *Color* domain composed of *hue*, *saturation* and *brightness* subdomains; *Weight* and *Mass* domains as linear orders homomorphic to the half-line of non-negative numbers .

sortal type : entity type that carries a criterion for determining the individuation, persistence and identity[4] 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 : 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”.

[4]For a deeper discussion on the notion of individuation, persistence and identity criteria and its role in conceptual modeling one should refer to (Guizzardi et al, 2004).

phase type : 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 : 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 : entity type that is not a sortal type and can be partitioned into disjoint subtypes which are sortal types (typically role 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 : 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.

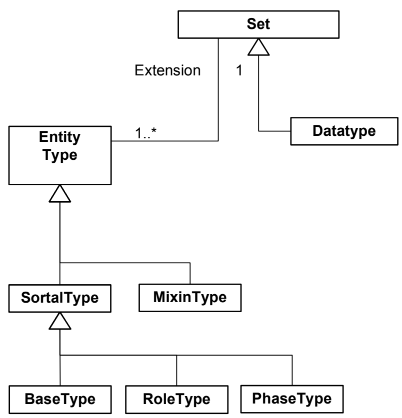


Figure 2: Different kinds of types in UFO-A 0.2.

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 (e.g. kind, phase or state, role, mixin). In particular, this theory can be considered as an elaboration in the way types are accounted for in the BWW approach. In one of the BWW papers (Evermann & Wand, 2001), it is proposed that a UML class should be used to represent a BWW-natural kind (it should be equivalent to functional schema of a BWW-natural kind). A natural kind is in the same ontological footing as what is named here a Base type, i.e., it is a rigid type that provides an identity criterion for its instances. As demonstrated in several works in the literature (Welty & Guarino, 2001; Gupta, 1980; Wiggins, 2001;

van Leeuwen, 1991; Guizzardi et al, 2004), this kind of type construct constitutes only one of the sorts which are necessary to represent the phenomena available in cognition and language. In other words, a conceptual modeling construct representing a base type is only one of a set of modeling constructs which should be available to the conceptual modeler.

2.3 Different Kinds of Individuals

We distinguish a number of different kinds of individuals, as shown in figure 3.

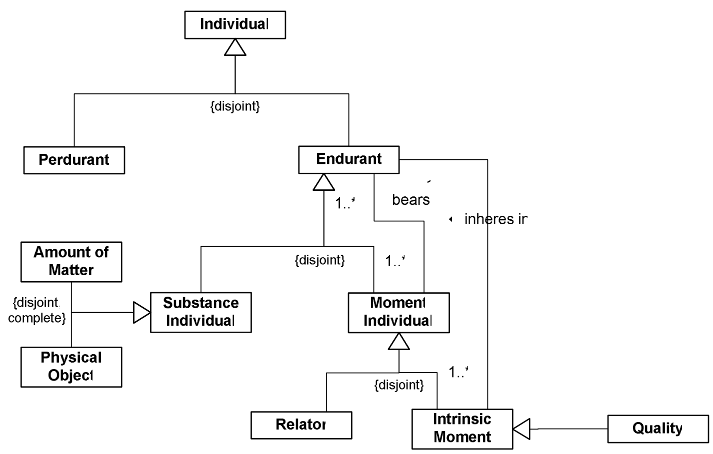


Figure 3: Different kinds of individuals in UFO-A 0.2.

In structured English, these different kinds of individuals are explained as follows.

endurant : individual that is wholly present whenever it is present, i.e. it does not have temporal parts. An endurant is something which persists in time while keeping its identity. Examples are a house, a person, the moon, a hole, the redness of an apple and an amount of sand. [DOLCE]

Corresponding terms: GFO:3D-individual

perdurant : individual that is composed of temporal parts; whenever a perdurant is present, it is not the case that all its temporal parts are present. The distinction between endurants and perdurants can be understood in terms of the intuitive distinction between "objects" (things, entities) and "processes"(events). Examples of perdurants are a race, a conversation, the Second World War and a business process [DOLCE]

substance individual : 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 : 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 Noam Chomsky; a flight connection between two cities.

endurant bears moment individual: designated relationship [based on GFO:"substance bears moment"]

physical object : 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 : substance individual that does not satisfy a condition of unity; typically referred to by means of mass nouns. Amounts of matter are, in general, mereologically invariant, i.e., they cannot change any of their parts without changing their identity [DOLCE]. *Examples*: a liter of water; a piece of gold; a pile of sand.

intrinsic moment : moment individual that is existentially dependent on one single individual

intrinsic moment inheres in endurant : designated relationship [GFO]

quality : intrinsic moment that inheres in exactly one endurant and can be mapped to a value (quale) in a quality dimension (Guizzardi & Wagner & Herre, 2004). *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.

relational moment: moment individual that is existentially dependent on more than one individual. Relational moments provide a foundation for the construction of material relationships between individuals (Guizzardi & Wagner & Herre, 2004). The category of relational moments in UFO is based on the concept of a [GFO:Relator]. The notion of relators is supported in several works in the philosophical literature (Smith & Mulligan, 1986; Smith & Mulligan, 1986) and, the position advocated here is that, they play an important role in: (i) distinguishing material relations such as *'being married to'* and *'studies at'* from their formal counterparts (e.g. 5 *is greater than* 3, this day is *part-of* this month); (ii) answering questions of the sort: what does it mean say that John is married to Mary? Why is it true to say that Bill works for Company X but not for Company Y? *Corresponding terms*: BWW:"mutual property". *Examples*: a particular employment (Susan is employed by IBM); a particular flight connection (LH403 flies from Berlin to Munich); a kiss; a handshake.

Putting all UFO-A terms and relationship type expressions together in one UML/MOF diagram results in figure 8 in APPENDIX A.

2.4 An Application of UFO-A 0.2 to Agent-Oriented Modeling

Modeling Agent Roles

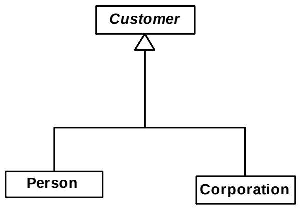
In figure 4, the role type *Customer* is defined as a supertype of *Person* and *Corporation*. This model is deemed ontologically incorrect for two reasons: first, not all persons are customers, i.e. it is not the case that the extension of *Person* is necessarily included in the extension of *Customer*. Moreover, an instance of *Person* is not necessarily a *Customer*. Both arguments are also valid for *Organization*. In a series of papers (Steimann, 2000a-b), Steimann discusses the difficulties in specifying supertypes for Roles that can be filled by instances of disjoint types[5]. As a conclusion, he claims that the solution to this problem lies in the separation of role type and base type (named natural type in the article) hierarchies; a solution which would strongly impact the metamodel of all major conceptual modeling language. 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, 2004).

In this example, Customer has in its extension individuals that obey different identity criteria, i.e., it is not the case that there is a single identity criteria which applies both for Persons and Corporations. Customer is hence a mixin type (a non-sortal) and, by definition, cannot supply an identity criterion for its instances. 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 PrivateCustomer 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.

In summary, in many modeling problems, we have to model agent types that are role mixin types, which implies that

1. there is a disjoint partition into subtypes, and
2. these subtypes are role types, that is they are subtypes of appropriate base types.

[5] This problem is also mentioned in (van Belle, 1999): "*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)?.*"



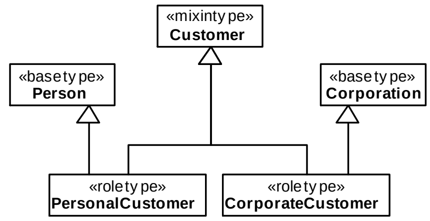


Figure 4: An ontologically incorrect model of roles; **Figure 5:** An ontologically correct version of (Figure 4) according to UFO 0.2.

3 UFO-B 0.2 – Perdurants

A complete treatment of an ontology of perdurants requires a an ontology of *temporal entities* (GFO:chronoids) (Degen, Heller, Herre & Smith, 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 in section 4. In the sequel we discuss the following basic kinds of perdurants shown in figure 6: (atomic and complex) events and states.

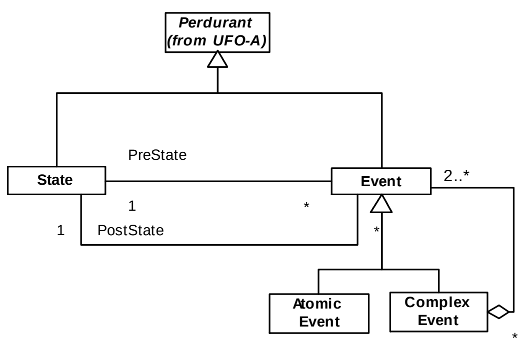


Figure 6: The perdurant categories of UFO-B 0.2

state : perdurant that is homeomeric (each temporal part of it is again a state) [based on DOLCE]

event : 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 : event that happens instantaneously, i.e. an event without duration, relative to an underlying time granularity [based on BWW:event and GFO:change]. *Examples*: an explosion; a message reception.

complex event : 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.

state* is pre-state of *event : name of a formal relationship type

state* is post-state of *event : name of a formal relationship type

4 UFO-C 0.2 – Intentional, Social and Linguistic things

The 'objective' perdurant categories *event*, *process* 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 : 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 : event that is created through the action of a physical agent

non-action event : event that is not created through an action of a physical agent

physical agent creates action event: designated relationship

physical agent perceives event: designated relationship

non-agentive object : physical object that is not a physical agent

Examples: a chair; a mountain

mental moment : 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 : physical agent that communicates with other communicating physical agents

Examples: a dog; a human; a communication-enabled robot

institutional agent : 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 : endurant that is either a physical agent or an institutional agent

communicating agent : agent that communicates with other communicating agents

social moment : relational individual that is existentially dependent on more than one communicating agent

Examples: a commitment; a joint intention

The above categories are also defined in the MOF/UML model of figure 7.

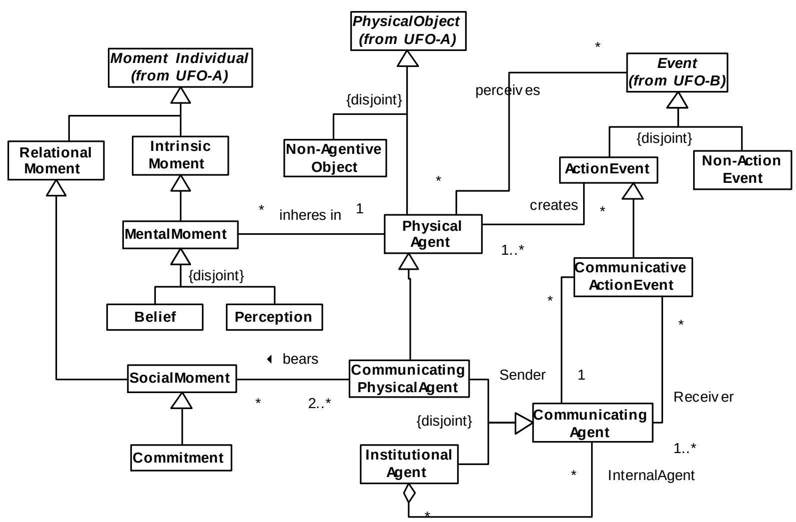


Figure 7: The categories of the UFO-C 0.2 agent ontology.

Agents may interact with their inanimate environment, or they may interact with each other involving 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 : 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 : 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 : 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.

6 Conclusions

The unified foundational ontology UFO is stratified 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 not only as a basis of agent-oriented modeling but also of 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 agent-oriented modeling problems.

Acknowledgement: Giancarlo Guizzardi's work on this paper is funded by the Freeband A-MUSE project. Freeband (http://www.freeband.nl) is sponsored by the Dutch government under contract BSIK 03025.

References

1. Bunge, M. (1977). Treatise on Basic Philosophy. Vol. 3. Ontology I. The Furniture of the World. D. Reidel Publishing, New York.
2. Bunge, M. (1979). Treatise on Basic Philosophy. Vol. 4. Ontology II. A World of Systems. D. Reidel Publishing, New York.
3. Chapin et al (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.
4. Degen, W., Heller, B., Herre, H. & Smith, B. (2001). GOL: Towards an axiomatized upper level ontology. In Smith, B. & Guarino, N. (eds.), Proceedings of FOIS'01, Ogunquit, Maine, USA, October 2001. ACM Press.
5. Evermann, J. & Wand Y. (2001). Towards ontologically based semantics for UML constructs. In Kunii, H.S., Jajodia, S. & Solvberg, A. (eds.), Proceedings of ER 2001, pages 354–367, Springer-Verlag.
6. Gangemi, A., Guarino N., Masolo C., Oltramari, A. & Schneider L. (2002). Sweetening Ontologies with DOLCE. Proceedings of EKAW 2002, Siguenza, Spain.
7. Gärdenfors, P. Conceptual Spaces (2000): the Geometry of Thought. MIT Press, USA, 2000.

8. Green, P.F. & Rosemann, M. (2000). Integrated Process Modelling: An Ontological Evaluation. Information Systems 25 (2), 73-87.
9. Green, P.F. & Rosemann, M. (2002). Usefulness of the BWW Ontological Models as a "Core" Theory of Information Systems. In Proceedings Information Systems Foundations: Building the Theoretical Base, Canberra, 2002, 147-164.
10. Guizzardi, G., Herre, H. & Wagner G. (2002a). On the General Ontological Foundations of Conceptual Modeling. In Proceedings of 21th International Conference on Conceptual Modeling (ER 2002). Springer-Verlag, Berlin, Lecture Notes in Computer Science.
11. Guizzardi, G., Herre, H., Wagner G. (2002b): Towards Ontological Foundations for UML Conceptual Models. In Proceedings of 1st International Conference on Ontologies Databases and Applications of Semantics (ODBASE 2002), Springer-Verlag, Berlin, Lecture Notes in Computer Science.
12. Guizzardi, G., Wagner G., Guarino, N.; van Sinderen, M. (2004): An Ontologically well-Founded Profile for UML Conceptual Models, In Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CaiSE). Springer-Verlag, Berlin, Lecture Notes in Computer Science.
13. Guizzardi, G., Wagner G., Herre, H. (2004): On the Foundations of UML as an Ontology Representation Language. In Proceedings of 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW), Springer-Verlag , Berlin, Lecture Notes in Computer Science.
14. Guizzardi, G., Wagner G., van Sinderen, M. (2004): A Formal Theory of Conceptual Modeling Universals, In Proceedings of the International Workshop on Philosophy and Informatics (WSPI), Germany.
15. Gupta, A. (1980). The Logic of Common Nouns: an investigation in quantified modal logic, Yale University Press, New Haven.
16. Hirsch, E. (1982). The Concept of Identity. Oxford University Press, New York, Oxford.
17. ISO (2000). ISO 1087-1 Terminology work - Vocabulary - Part 1: Theory and application. Copies of all ISO standards can be purchased from ANSI, 25 West 43rd Street, New York, NY 10036, (212) 642-4980, info@ansi.org, or http:webstore.ansi.org".
18. Leeuwen, J. van (1991). Individuals and sortal concepts: an essay in logical descriptive metaphysics, PhD Thesis, University of Amsterdam.
19. Loebe, F. (2003): An Analysis of Roles: Towards Ontology-Based Modelling, Diploma Thesis, Institute for Medical Informatics, Statistics and Epidemiology (IMISE), University of Leipzig.
20. 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.
21. OMG, Object Management Group (2003). Meta Object Facility (MOF) 2.0 Core Specification, version 2.0, http://www.omg.org/docs/ptc/03-10-04.pdf
22. Searle, J.R. (1995). *The Construction of Social Reality*. Free Press, New York.
23. Smith, B.; Mulligan, K. (1983): Framework for Formal Ontology, Topoi (3), 73-85.
24. Smith, B.; Mulligan, K. (1986): A Relational Theory of the Act, Topoi (5/2), 115-30.
25. Steimann, F. (2000a): On the representation of roles in object-oriented and conceptual modeling, Data & Knowledge Engineering 35:1, 83–106.
26. Steimann, F. (2000b): A radical revision of UML's role concept, in: A Evans, S Kent, B Selic (eds) UML 2000 Proceedings of the 3rd International Conference (Springer, 2000) 194–209.
27. 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.

28. Wand, Y. & Weber, R. (1989). An ontological evaluation of systems analysis and design methods. In Falkenberg, E.D. & Lindgreen, P. (eds.), *Information System Concepts: An In-depth Analysis*, North-Holland.
29. Wand, Y. & Weber, R. (1990). Mario Bunge's Ontology as a formal foundation for information systems concepts. In Weingartner, P. & Dorn, G.J.W. (eds.), *Studies on Mario Bunge's Treatise*, Rodopi, Atlanta.
30. Wand, Y. & Weber, R. (1995). On the deep structure of information systems. Information Systems Journal 5, 203-223.
31. Wand, Y., Storey, V.C. & Weber, R. (1999). An ontological analysis of the relationship construct in conceptual modeling. ACM Transactions on Database Systems, 24(4):494–528.
32. Welty, C. & Guarino, N (2001). Supporting Ontological Analysis of Taxonomic Relationships. *Data and Knowledge Engineering*, 39(1), 51-74.
33. Wiggins, D. (2001). Sameness and Substance Renewed, Cambridge University Press.

APPENDIX A

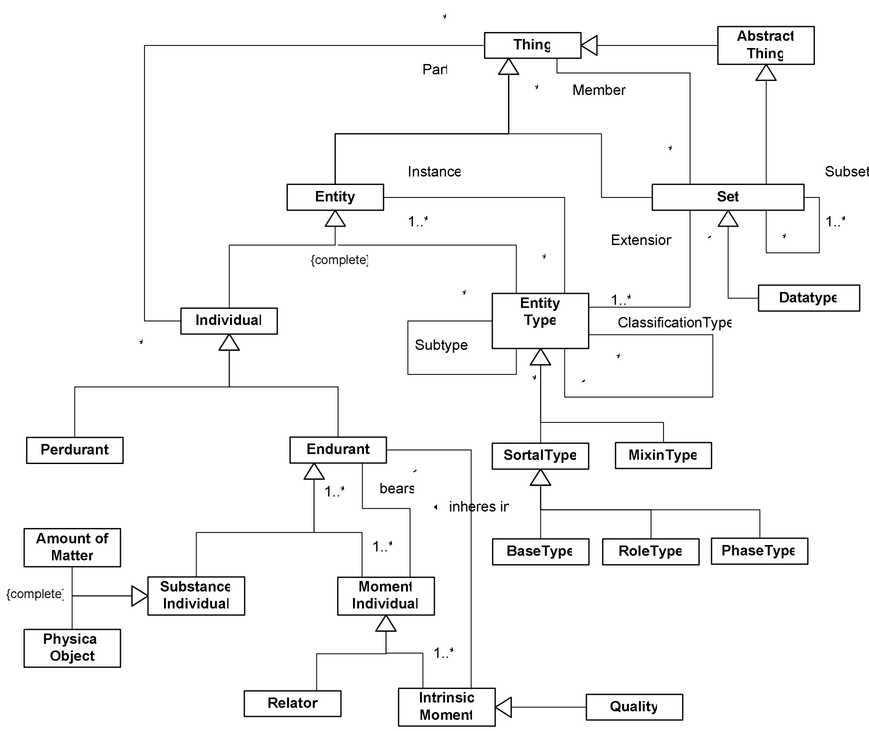


Figure 8: UFO-A 0.2 as a MOF/UML model