Software and Systems Modeling manuscript No.
(will be inserted by the editor)

A UML and OWL Description of Bunge's Upper-Level Ontology Model

Joerg Evermann[1]

Victoria University of Wellington
e-mail: `jevermann@mcs.vuw.ac.nz`

The date of receipt and acceptance will be inserted by the editor

Abstract A prominent high-level ontology is that proposed by Mario Bunge. While it has been extensively used for research in IS analysis and conceptual modelling, it has not been employed in the more formal settings of semantic web research. We claim that its specification in natural language is the key inhibitor to its wider use. Consequently, this paper offers a description of this ontology in open, standardized knowledge representation formats. The ontology is described both in UML and OWL in order to address needs of both semantic web and conceptual modelling communities.

1 Introduction

Ontologies play an increasingly important role in information systems (IS) research and practice. Two different understandings of the term "ontology" have evolved. Research in the areas of ontology-driven information systems (ODIS) [1] and the semantic web [2,3] uses the term "ontology" without necessarily implying a firm commitment to the existence of a particular set of entities in reality [4–6]. In this research area, ontologies are descriptions of shared conceptualizations of application domains [7], they are constructed as needed [8], and engineered to fit a particular problem [9–11,4]. There, domain specific ontologies can be integrated by referring to concepts in upper-level ontologies. The upper-level ontologies in this research area, such as SUMO [12] and Cyc [13–15], are specified in formal knowledge representation languages, such as KIF, DAML, or OWL. The current standard ontology language OWL [16] is based on the formalism of description logics [17–19], which allows reasoning and inference. Specifically, the OWL-DL subset is restricted to the $\mathcal{SHOIQ}$ description logic to guarantee efficient reasoning [20].

In contrast, in conceptual modelling research [21–54] , the term ”ontology” is used in its original philosophical sense, understood as meta-physics or the philosophy of existence [55]; an ontology is a fundamental, domain-independent philosophical position, a commitment to the existence of certain entities in external reality. Research in this area has drawn primarily on the ontological work by Mario Bunge [56,57]. In contrast to upper-level ontologies for the semantic web and ODIS, and other philosophically-based ontology such as GFO/GOL [39–42] and Dolce [43], this ontology has not been specified in a formal ontology description language, but is defined in natural language with some formal description in set theoretic terms.

1.1 Problem Statement

Both areas, semantic web applications and conceptual modelling research, can benefit from the availability of a well-developed and well-known high-level ontology, such as that of Bunge. However, the current representation form hinders the application and wide-spread use of Bunge’s ontology in both the semantic web and the conceptual modelling research areas. To overcome this limitation, this paper presents a new representation of Bunge’s ontology in a standardized and open format that can benefit both communities. The objective of the paper is to develop a UML model and an OWL model of this ontology.

It is not the aim of the paper to argue for or against the adoption of a particular ontology. The focus of this paper on Bunge’s ontology does not dispute the validity of other ontologies or imply the superiority of Bunge’s ontology. Instead, the argument is one about the usefulness of this research to the community of researchers and practitioners using this ontology:

- Bunge’s ontology has been widely used in the IS research literature to compare and evaluate conceptual modelling languages, clarify the notion of data quality, and inform object-oriented modelling principles [21–38, 44–54,58–64] . This research has been using informal argumentation even though the potential for more rigorous and formal discussion has been recognized [48,65].
- Other upper-level ontologies in IS, such as SUMO [12], Cyc [13–15], GFO/GOL [39–42] and Dolce [43] are already expressed using modern ontology or knowledge description languages such as DAML, KIF or OWL. Bunge’s ontology is not. Given the research interest in this ontology, especially in conceptual modelling, a UML and OWL description is useful.

Note that while UML is not a formal language in the sense that fixpoint, model-theoretic or operational semantics are defined for it, it is less ambiguous than a natural language representation.

1.2 Expected Benefits

In conceptual modelling, UML is the de-facto standard description language. Providing a description of Bunge's ontology in UML makes it available to be used with established modelling, model conversion, model exchange, and model repository tools[1]. The availability of these tools for use with Bunge's ontology can in turn promote the use and adoption of this ontology. Furthermore, availability of a UML description of Bunge's ontology allows research on conceptual modelling languages to employ meta-model based comparisons to other ontologies or models [48, 65] and supports the derivation of modelling guidelines and rules [31].

The OWL Web Ontology Language [16] is an accepted standard to express semantic web ontologies, either in its abstract syntax form, or in an XML-based exchange syntax form. A large number of domain ontologies have been developed in OWL and DAML, a precursor to OWL[2]. The availability of upper-level ontologies in a semantic web language can help with the integration of disparate domain ontologies in the semantic web context. For example, two large-scale domain ontologies, TOVE [11] and the AIAI Enterprise Ontology [66], describe the same domain, albeit based on different foundations. Formally relating concepts of both to concepts in Bunge's ontology enables interoperability. For example, "primitive action" in TOVE may be equivalent to an "event" in Bunge's ontology, and "activity" in the AIAI Enterprise Ontology may also be equivalent to an "event" in Bunge's ontology. An ontology reasoner can exploit these equivalences and derive inter-ontology inferences based on the combined knowledge base of TOVE and the AIAI Enterprise Ontology.

Moreover, an OWL description makes Bunge's ontology available to be used with established ontology tools and technologies, e.g. for ontology alignment, reasoning, editing and ontology repositories [2, 67, 68]. In turn, the availability of these tools can promote the use and adoption of Bunge's ontology, as tool availability is an important quality aspect [69, 70].

Finally, the availability of Bunge's ontology in an ontology description language as well as a conceptual modelling language brings us closer towards the goal of ontology driven IS development [1]: Domain ontologies could be transformed to domain conceptual models, which can then, as part of an MDA (model driven architecture) process, be transformed to code. Existing ontologies may be harnessed as conceptual models. In turn, existing conceptual models may be leveraged for the development of domain ontologies.

The remainder of this paper proceeds as follows. Section 2 offers a discussion of related work. Our solution approach is presented in Section 3, followed by a description of developing a UML model of Bunge's ontology in Section 4. Subsequently, Section 5 presents a brief description of a UML to OWL translation approach and its application to the UML model of

---

[1] e.g. IBM Rational Rose, Visual-Paradigm, Poseidon-UML, MagicDraw

[2] `http://www.daml.org/ontologies/`, `http://knowledgeweb.semanticweb.org/o2i/ontology_repository.php`

Bunge's ontology. The paper discusses the challenges and limitations of this work (Section 6) before concluding (Section 7).

2 Related Work

An extended Entity-Relationship (eER) model of Bunge's ontology had been developed previously [48], but with several limitations.

1. The model in [48] is not reported to have been intersubjectively validated.
2. The eER language dialect chosen is not widely known or used.
3. OWL and UML are the current de-facto standards for ontology description and conceptual modelling, respectively. A description of the model as an ER model makes it difficult to e.g. compare this model to models of UML or ebXML, both of which are specified as UML models. Comparisons to such software design and process modelling languages have been shown to be theoretically important and useful [23,27,35,71].
4. Most importantly from a pragmatic perspective, the model developed in [48] is not available in a format that allows further use: The CASE tool with which it is generated uses a proprietary, binary format.
5. The model described in [48] is based on a limited scope derivative of Bunge's ontology, while the present effort is based directly on Bunge's work [56,57].

The research by Bera and colleagues brings together Bunge's ontology and OWL [21,22]. They view OWL as an ontology, rather than an ontology language, and assign real-world meaning to its constructs by mapping them to concepts of Bunge's ontology. They propose modelling rules and add concepts to OWL such as event, and state. Adding such concepts to OWL is not problematic when OWL is viewed as an ontology, rather than as a formal description logic that requires formal semantics and inference rules for the new concepts [19,18,20]. In contrast, the present work views OWL as the logical formalism it is intended to be, a means for describing ontologies. Consequently, no real-world meaning is required for OWL, only for the ontologies described in OWL.

Under the auspices of the Object Management Group (OMG), which coordinates modelling and meta-modelling standards such as Corba and UML, a working group on ontologies has been formed. The product of this working group is the Ontology Definition Metamodel Specification [72]. This is a specification of the OWL ontology language in terms of the MOF (meta-object facility), in essence a definition of OWL in UML. It is *not* a specification of a particular ontology in UML and OWL, as developed in this paper.

Work on other ontologies and ontology languages has led to profiles for UML to guide and support conceptual modellers. For example, Guizzardi and colleagues [73] have developed a profile for UML based on GOL [74], and Djuric [75] has proposed a UML profile based on description logics. The proposal in Sect. 4.3 develops an analogous UML profile based on Bunge's ontology.

Finally, the close relationship between OWL and UML has been expressed in [76], where the ability to reason with UML class diagrams is explored by translating UML class diagrams to OWL. Sect. 5 makes use of this work. Such a translation is also implemented by the UML storage backend for the Protege ontology editor [3].

3 Solution Approach

As formal ontology and conceptual modelling research can both benefit from the same ontology, albeit using different representation languages, there are three alternative approaches:

1. Developing and maintaining separate UML and OWL models of Bunge's ontology has the advantage that both models can be developed to make full use of the expressiveness of each language. However, the modelling would require twice the initial effort, the models would not be identical and may not even be consistent, and increasing effort must be expended to maintain the models.
2. Developing and maintaining an OWL model and automatically deriving a UML model reduces modelling and maintenance effort. On the other hand, the modelling capabilities of UML cannot be fully exploited, as the model would be limited to the expressiveness of OWL.
3. Developing and maintaining a UML model and automatically derive an OWL model has the same benefits and drawbacks as the second alternative.

In this paper we have chosen the last option for the following reasons.

- We aim to guarantee a single model in both languages. This, and the associated effort, rules out the first approach.
- While each language (UML and OWL) offers language constructs not available in the other (Sect. 5), the UML constructs that cannot easily be mapped to OWL constructs ([72,76]) are not critical to describing Bunge's ontology[4].
- UML has been successfully used not only for conceptual modelling but also for ontology modelling [75,76,78].
- Pragmatically, UML is more widely used than OWL and has the more mature modelling tool support.

---

[3] `http://protege.cim3.net/cgi-bin/wiki.pl?UMLBackendMapping` (last access on Sept. 25th, 2007)

[4] While UML class diagrams can be supplemented with the object constraint language OCL (Object Constraint Language) [77], this would prohibit the use of the ontology in the semantic web context, as OCL is full first order logic, so that efficient reasoning abilities cannot be guaranteed.

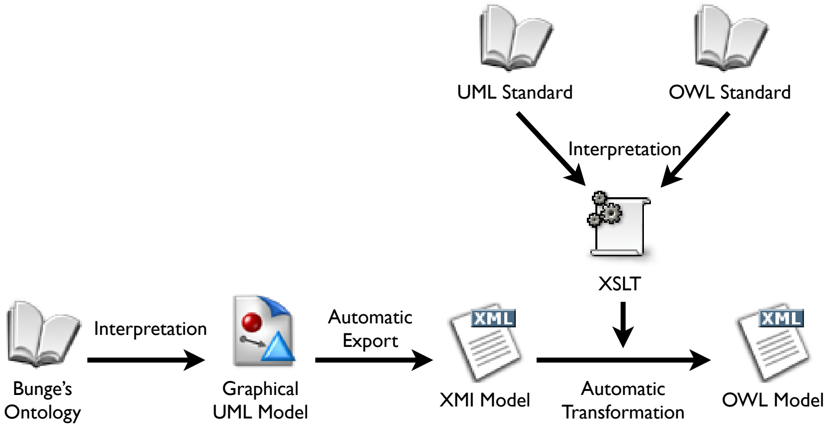


Fig. 1 From Bunge's writings to an OWL model

- Bunge's ontology is mainly used in conceptual modelling research where UML is the prominent description language.

The diagram in Fig. 1 summarizes the chosen approach. An interpretation of Bunge's ontological writings [56, 57] leads to a graphical UML model (Sect. 4), which is exported in the standardized XMI (XML Model Interchange) format. From the UML and OWL standards, and based partially on existing research, an XSL transformation is developed and applied to the XMI model, yielding an OWL description of Bunge's ontology (Sect. 5). Only one model (the UML model) of Bunge's ontology needs to be created and maintained.

4 Development of a UML Model of Bunge's Ontology

The translation of a text or description from one language or representation form to another requires its interpretation and understanding. Consequently, we choose a hermeneutic approach for the development of the UML model of Bunge's ontology. Hermeneutics is an interpretation technique to identify the meaning of a text. It was pioneered by Gadamer [79] and Ricoeur [80], and is well accepted in IS research (e.g. [81–84]).

Hermeneutics is a dialectic process, using a cycle that iterates over repeated interpretations of a text before different backgrounds of understanding. Originally this hermeneutic cycle was conceived of as a process that relates the meaning of parts of a text to the meaning of the whole text, and vice versa. More generally, every time the reader interprets a text in light of a different understanding, be it caused by a different sense of the holistic meaning or otherwise, the reader in turn gains a different understanding of the text. This change in the understanding of the reader consequently changes the reader's interpretation of subsequent readings of the text [79, 80]. Interpretation of a text by a reader is complete when the reader's interpretation does not change from previous readings of a text. In terms of [79], this is the point of "fusion of horizons".

In the case of this research, knowing that Bunge's work [56,57] needs to be represented in UML or OWL, will (consciously or not) invoke a different "sense" or "reading" of the text in the interpreter, as for example, when comparing it to Kant's work on categories. The first iteration over the hermeneutic cycle begins with a "naive" reading of Bunge's work, shaped only by the knowledge that a representation in UML is the aim of the reading, and knowledge of the modelling capabilities of UML. Representing this "naive" understanding of Bunge's writings in UML provides a different background to the next reading of the text. For example, this next reading may emphasize particularly unclear passages, or passages that appeared inconsistent when naively expressed in UML. The interpretation of Bunge's writings on ontology was achieved by such an iterative process of developing a UML or OWL model based on the detailed and critical reading of [56,57].

A rigorous and systematic procedure was adopted for each iteration, as follows: During each iteration all definitions and postulates in [56,57] were examined as to whether they defined concepts within the scope of the study. If this was the case, the concepts were included in the model as UML classes. In this case, further critical reading of related corollaries and plain text explanations in [56,57] was carried out to define the relationships of the new concept with others. For every concept and relationship added in this way, the remainder of Bunge's writings were re-examined to cross-validate and clarify the relationships within the text. Once all definitions in [56,57] were examined in this way, another iteration of the hermeneutic cycle was begun.

The iterations continued until the understanding of the ontology, and the UML model based on that understanding, was not changed during subsequent readings. The UML model underwent a total of five iterations, corresponding to five critical readings of the text. Of these, the second model iteration was done in OWL, rather than UML, in order to gain a better understanding of potential differences and tradeoffs between UML and OWL modelling.

Scoping of the translation was determined by examining the previous body of work that employs Bunge's ontology [21–38,44–54,58–64]. That body of work shows consensus on a certain subset of Bunge's writings as relevant for research and practice in IS. Because the present research effort is aimed at delivering a useful result for those researchers and practitioners, the scope of the present work is restricted accordingly. Consequently, we omit topics such as the structure of space and time, possibility, and human and social systems, as they have not been employed in previous work.

4.1 Validation

The resulting model was inter-subjectively validated by three independent researchers in the semantic web and conceptual modelling community. All have published on Bunge's ontology and can be considered topic experts [21, 22, 24, 65]. The independent validation, detailed in the following paragraphs, highlighted different issues with the overall model.

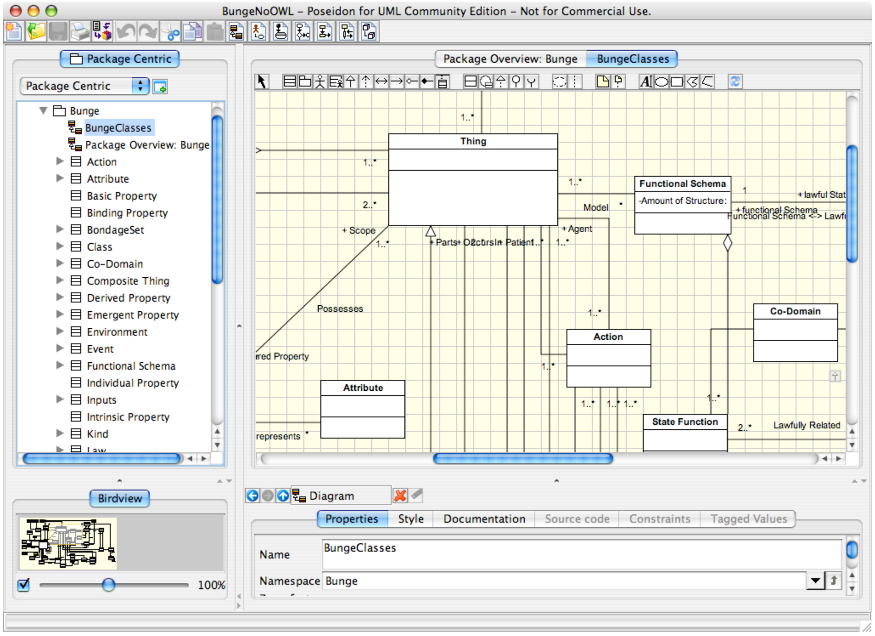
- The validation effort included the maintainer of the eER model. The extent of this validation was a full day in person discussion of both models, and a subsequent conference call, in addition to weekly email exchanges. The focus of the discussions was on the modelling of states, attributes, and state functions, because the largest differences were found there. As a result, both the UML model and the eER model, which had not originally been inter-subjectively validated, underwent substantial revisions[5]. However, as the eER was obtained primarily from the comparatively brief description in [25], rather than the full source [56, 57], some differences remain (Tables 2 and 3).
- Subsequent discussions with a second expert were based on the model resulting from the validation process with the first expert. This validation was conducted via multiple telephone discussions and email exchanges, and highlighted two main issues of omission in the UML model. This expert reported approximately 8 hours of time spent on validating the model against Bunge's work [56, 57] over a two week period. The focus was on the relationships between the identified concepts and their multiplicities. The concept of interaction was introduced to the model as a result of this validation. Minor clarification issues, dealing e.g. with synonymous terminology, were also raised and addressed in the model. A subsequently repeated validation by this expert of the revised model yielded agreement on the entire model.
- The validation with a third expert, based also on the result of the validation process with the first expert, revealed inconsistencies in the modelling of co-domains and their relationship to state spaces and state functions. Two telephone discussions with this expert led to clarifications and minor changes in this area of the model. A minor error in the multiplicity of the association between events and states was corrected as a result of this validation. Overall, this validation process revealed no significant problem areas in the model.

A complete exposition of the developed model is beyond the scope of this paper, as Bunge's writings consist of two book volumes[6]. Figure 2 shows the UML model in the Poseidon UML CASE tool. The complete model contains

---

[5] The current version of the eER model is 5.1, last revised 15 Sept 2005.

[6] The complete model is available electronically at `http://www.mcs.vuw.ac.nz/~jevermann/Bunge/` and print copies are available from the author.



Name BungeClasses

Namespace Bunge

Fig. 2 Bunge ontology model in the Poseidon UML tool

65 classes, 62 generalizations, and 79 associations[7]. The following table lists all the concepts in the model and their definitions in [56,57]. The table is not intended as a discussion of the ontology, but instead is intended to offer an indication of rigor and completeness of this research.

Table 1: Definitions and postulates used from [56,57]

Section, definition or postulate in [56]	**Construct**
Chapter 1	Individual
Definition 1.1	Composite individual
Definition 1.1	Simple individual
Definition 1.2	Individual part of relation
Definition 1.3, Postulate 1.2	World individual
Postulate 1.1, Corollary 1.1	Null individual
Postulate 1.5	Individual superposition
Postulate 1.5, Definition 1.10	Individual juxtaposition
Definitions 2.2, 2.5, 2.17, Chapter 2, Sections 1.1 and 3	Property
Postulate 2.1	Property in general

Continued on next page

---

[7] This compares to 76 entity types in the current eER model. However, of these, more than 25 are redefined relationship types, which are relationship types as well as entity types, similar to the concept of a UML association class.

From previous page

Section, definition or postulate in [56]	**Construct**
Postulate 2.1	Individual property (property in particular)
Definition 2.6, Definition 3.14	Class
Postulate 2.8	Basic property
Postulate 2.8, Corollary 2.1	Complex property
Definition 2.15	Property weight
Postulate 2.2, Chapter 2, Section 2.2	Mutual property
Postulate 2.2, Chapter 2, Section 2.2	Intrinsic property
Chapter 3, Section 5.1, pg. 101f	Binding mutual property
Chapter 3, Section 5.1, pg. 101f	Non-binding mutual property
Definition 2.16	Emergent property (gestalt property)
Definition 2.16	Resultant property (hereditary property)
Chapter 2, Section 1.2 (esp. pg. 60)	Attribute
Definition 2.4	Property incompatibility
Definitions 2.7, 2.9, 2.10, Postulate 2.7	Property precedence as law[8]
Definition 3.1	Thing
Definitions 3.1, 2.5, 2.17	Thing possesses properties
Definitions 3.3, 3.4, Postulate 3.2	Thing juxtaposition
Definition 3.4	Composite thing, composition, and part-of relation
Definition 3.4	Basic thing
Postulate 3.2	Null thing
Postulate 3.3, Corollary 3.2	World
Definition 3.6, also p. 125	Domain
Definition 3.6	Co-domain
Definition 3.6	State function
Definition 3.6	Functional schema (model, model thing)
Definition 3.7	Amount of structure
Definition 3.9	State (associated with functional schema)
Definition 3.9	Value of total state function (defines state)
Definition 3.9	Function value (assumed value of state function)

Continued on next page

---

[8] Bunge uses the terms law and law statement synonymous in Chapter 2. A second definition of law statement in Chapter 3 (Definition 3.10) is of a different form than the one in Chapter 2, but with the same content. In the present model, we distinguish law from law statement and propose that a law statement expresses or describes a law.

From previous page

Section, definition or postulate in [56]	**Construct**
Definitions 3.10, 2.7, Criterion 2.2	Law statement (restricts state functions)
Definition 3.11	Lawful state
Definition 3.11	Lawful state space (defined by law statements)
Chapter 3, pg. 133f	Conceivable state space[9]
Definition 3.17	Kind
Definitions 3.21, 3.22, 3.25	Natural kind (species)[10].
Chapter 3, Section 3.5	Basic law
Chapter 3, Section 3.5	Derived law[11]
Definition 3.25	Natural genus
Definition 5.2	Change (w.r.t. a lawful state space)
Definition 5.3	Qualitative change
Definition 5.3	Quantitative change
Definition 5.4, Principle 5.2	Event (ordered pair of states)
Definition 5.4	Event space
Definition 5.4	Composition of two events
Definition 5.6	Composite event (complex event)
Definition 5.8	Lawful transformation
Definition 5.9, Principle 5.3	Functional change (lawful event)
Definition 5.9	Lawful event space
Definition 5.10	Composite functional event
Definition 5.11, Rule 6.1	Reference frame
Definition 5.11	Coordinatization
Definition 5.14	Relative rate of change
Definition 5.14	Relative extent of change
Definition 5.15	Global rate of change
Definition 5.15	Global extent of change
Definition 5.24, Definition 5.6	Process
Definition 5.26, Postulate 5.9	Process predecessor and successor
Definition 5.27	History
Definitions 5.29, 5.31, Postulate 5.10, Criterion 5.1	Action
Definitions 5.30, 5.31	Interaction
Definitions 5.32, 5.33	Bond
Definition 5.33, [57] 1.5	Bondage (internal A-Structure)

Continued on next page

---

[9] The conceivable state space was modelled even though it is not formally defined in [56]. This allows us to introduce the idea that state spaces are spanned by the co-domains of state functions of functional schemata. The lawful state space is a conceivable state space that is constrained by law statements.

[10] Note that natural kinds are defined by laws. As laws relate properties, members of natural kinds also possess certain properties. These are called idiosyncratic properties (Definition 3.22).

[11] Note that while the distinction between basic and derived laws "is logically, methodologically and ontologically crucial" [56], it is not formally defined in [56].

From previous page

Section, definition or postulate in [56]	**Construct**
Definition 5.35, [57] 1.1, Corollaries 5.14, 5.15	System
[57] Definition 1.2, Theorem 1.1	A-Composition
[57] Definition 1.2	A-Structure
[57] Definition 1.2	A-Environment
[57] Definitions 1.6, 1.7	Sub-system (nesting structure)
[57] Definition 1.8	Level
[57] Definition 1.8	Level structure
[57] Definition 1.10	Input
[57] Definition 1.10	Output

Table 2 shows the elements in the current eER model that are not contained in the UML model of this research. Except for internal and external events, and stable and unstable states, these are acknowledged by the eER model maintainer not to be part of the ontology as described in [56]. Internal and external events, stable and unstable states are not formally defined in [56] and therefore not included in the UML model.

Table 3 shows the elements in the current UML model which are not contained in the eER model. As shown in Tab. 1, they are formally defined in [56]. This suggests that the present UML model may be a more complete representation of Bunge's ontology than the eER model.

- Internal event
- External event
- Stable state
- Unstable state
- Time instant
- History
- Natural law
- Human law
- Well-defined event
- Poorly-defined event
- Known state
- Predictable state
- Transformation law
- Conceivable event space
- Corrective action
- Value change
- Stability condition

Table 2 Elements in the eER model not in the UML model

- Individual
- Individual juxtaposition
- Individual superposition
- World (individual)
- Composite individual
- Simple individual
- Null individual
- Natural kind
- Natural genus
- Thing juxtaposition
- Null thing
- Property precedence
- Derived law
- Basic law
- Functional schema
- State function
- Domain
- Co-domain
- Qualitative change
- Quantitative change
- Reference frame and coordinatization
- Composite event
- Functional change
- Process
- Bondage

Table 3 Elements in the UML model not in the eER model

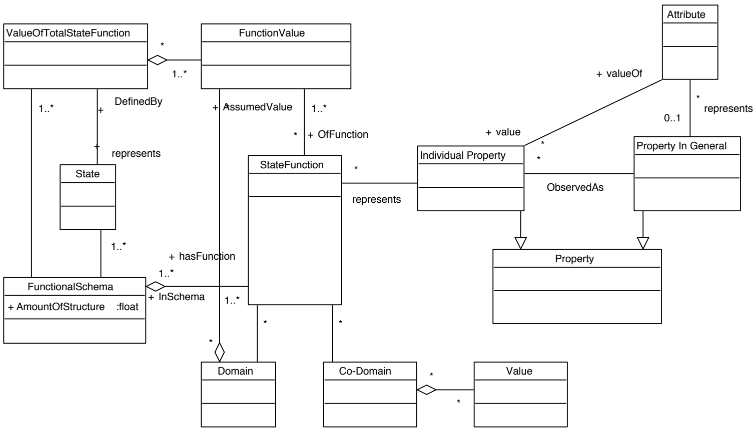


Fig. 3 Excerpt from the UML model of Bunge's ontology

4.2 Excerpt of UML Model

Figure 3 is an excerpt of the full model, showing states and related concepts to offer an impression of the resulting UML model[12]. This subsection briefly examines the derivation of the model excerpt in Figure 3 from [56].

Bunge introduces a functional schema in [56, Def. 3.6] as "a certain nonempty set M together with a finite sequence F of non-propositional functions on M, each of which represents a property". Therefore, we have modelled FunctionalSchema as an aggregate of StateFunction. Because the set is described as nonempty, the lower multiplicity is "1". While Bunge makes no mention of whether a state function can be part of multiple

---

[12] A note on the specialization of associations in the model: While the specialization of associations may not be widely used, a related technique, subsetting of association-end properties, is widely used by the OMG itself and applied to most associations in the definition of the UML meta-model [85]. The specification suggests that these can be used interchangeably: "In the case of associations, subsetting ends, ... , correlates positively with specialization the association" [85, pg. 37], except in a special case, which does not apply in this research: "This view falls down because it ignores the case of classifiers which, for whatever reason, denote the empty set." [85, pg. 37]. The paper therefore uses specialization as a clearer alternative to subsetting.

schemata, we make the least restrictive assumption that the aggregation is a shared one[13], with multiplicity "1..*" on the FunctionalSchema end.

"Any substantial property in general is representable as a predicate (or propositional function)" [56, Post. 2.1, pg. 63]. Therefore, the non-propositional functions of the functional schema represent individual properties. "We shall call them [the functions of a functional schema] state functions" [56, pg. 125]. We represent the relationship between state functions and individual properties by means of an association between StateFunction and IndividualProperty in Fig. 3. Bunge only suggests that "each of which [the state functions] represents a property" [56, Def. 3.6]. Hence, the multiplicity at the StateFunction end of the association is not restricted and the multiplicity at the IndividualProperty end is "1".

Propositional functions that represent properties in general [56, Post. 2.1, pg. 63] are called attributes: "Such propositional functions will be called attributes" [56, pg. 62]. Therefore, we associate Attribute and PropertyInGeneral in Fig. 3. The multiplicities on the Attribute end of the association are not restricted, so that it is possible that a property in general can be represented by multiple attributes, or not be represented at all. This is because "the representation of properties by attributes is a function $\rho : P \rightarrow 2^A$ that assigns each property p a collection $\rho(p) \in 2^A$ of attributes", [56, pg. 60]. Because "there are attributes with no ontic correlate" [56, pg. 60], the multiplicity on the PropertyInGeneral end is "0..1".

Bunge states that "any individual substantial property, ..., is representable as the value of an attribute." [56, pg. 63, Post. 2.1]. Consequently, we have associated IndividualProperty and Attribute. Bunge does not mention whether attributes can represent multiple individual properties. We make the least restrictive assumption and model a multiplicity of "*" on the IndividualProperty end of the association.

Because every functional schema possesses a unique total state function [56, Def. 3.9], we do not introduce a separate concept. Instead, the FunctionalSchema is associated with ValueOfTotalStateFunction. Because the total state function may take on different values (e.g. at different times), the multiplicity at the ValueOfTotalStateFunction end is "*".

The total state function is defined as the set of all state functions of a functional schema [56, Def. 3.9] and "its value is said to *represent* the state of" [56, pg. 127, emphasis added]. Consequently, we have modelled ValueOfTotalStateFunction as a shared aggregation of multiple FunctionValue. Because Bunge does not indicate whether a state function value can be part of multiple values of the total state function, we make the least restrictive assumption and assign a multiplicity of "*" to the ValueOfTotalStateFunction end.

---

[13] As the precise semantics of shared aggregations are not defined in [85], we characterize them in the terminology of [86] as shareable and separable.

From the terminology "its value is said to represent the state" [56, pg. 127] we assume an association between State and ValueOfTotalStateFunction with multiplicities of "1" at both ends.

Because a state is defined only in the context of a functional schema ([56, Def. 3.9]) the association between State and FunctionalSchema is modelled with a "1" multiplicity at the FunctionalSchema end and a "1..*" multiplicity at the State end.

A functional schema "consists of ... a list F ... of functions with ... unspecified co-domains V_i" [56, pg. 125]. Therefore, we have modelled the Co-Domain as an aggregation of Value.

The amount of structure of a functional schema is defined in [56, Def. 3.7] as a function of the rank and number of functions in a functional schema [56, pg. 122] and is represented as the attribute AmountOfStructure of FunctionalSchema.

4.3 A UML Profile for Bunge's Ontology

An immediate extension of the UML model is the definition of an ontological UML profile to annotate models, thereby providing for an ontological interpretation of the model elements and also offering guidance to the modeller [31]. The UML profile for Bunge's ontology presented here is analogous to the work by Djuric [75] for a UML profile for OWL and Guizzardi [73] for a UML profile for GOL.

UML 2.0 features a substantially revised definition of "lightweight" extension mechanisms. Profiles are specializations of packages and stereotypes are specialization of classes. Consequently, the graphical depiction of profiles and stereotypes is identical to that of packages and classes (Fig. 4 (a)). This UML revision makes it possible that a model that was developed on the M1 level (model-level) can be re-used as a model on the M2 level to define profiles. In fact, the OMG suggests that "a profile must therefore be defined as an interchangeable UML model" [85, pg. 633]. This is accomplished syntactically by all classes being specialized to stereotypes and all packages being specialized to profiles. The specialization preserves generalization and association relationships, as well as features. This simple syntactical step is a powerful way to define and create profiles based on existing (M1) models. As required by the OMG [85, Section 18.1.2], the use of the profile is strictly an addition to the UML 2.0 meta-model and introduces only constraining, but not contradicting, semantics.

Specifically, this new profile definition mechanism is used here as a simple but powerful way to make use of the extensive research work mapping UML constructs to elements of Bunge's ontology [23,28,29,31,35,60]. We illustrate using a brief excerpt from the developed UML model of Bunge's ontology. Packages in that model become profiles (a specialization of metaclass 'package') and classes become stereotypes (a specialization of metaclass 'class'). The existing mappings from UML to Bunge's ontology are used to identify those UML meta-classes that are extended by the stereotypes. For example, the mapping in [35][14] [15]:

- Natural kind ↔ UML-class
- Property ↔ UML-structural feature
- Intrinsic property ↔ UML-property (attribute)

leads to the following extensions:

- Stereotype ”NaturalKind” extends meta-class ”Class”
- Stereotype ”Property” extends meta-class ”StructuralFeature”
- Stereotype ”InstrinsicProperty” extends meta-class ”Property”

Other ontological concepts in the UML model similarly become stereotypes, with the existing research that maps UML to concepts in Bunge's ontology serving as the foundation for these extensions. An excerpt of such a profile is shown in Fig. 4 (a)[16].

The profile defined in this way can then be applied to adorn model elements on the M1 level. For example, Fig. 4 (b) shows a (M1) model of some domain that indicates that ”Supplier” is a natural kind and ”Name” and ”Number” are intrinsic properties.

Application of this profile requires the satisfaction of the constraints defined in the profile, such as multiplicities of associations between stereotypes. However, these profile constraints are constraints in the originally developed UML model of Bunge's ontology. For example, there exists a constraint that a natural kind is defined by two or more common properties among members of the kind. Applying the resulting profile consequently requires that whenever a class is stereotyped as ”NaturalKind”, two or more of its properties must be stereotyped as ”IntrinsicProperty”.

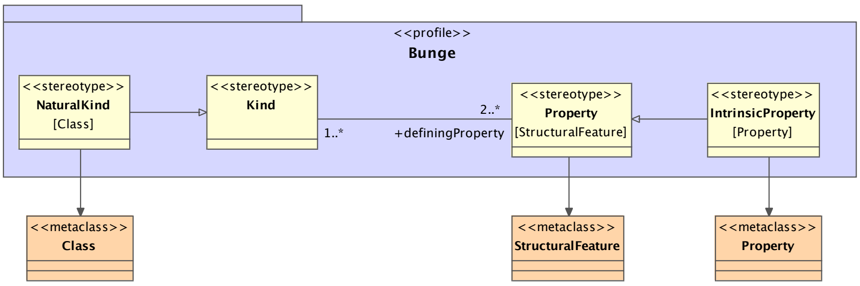
Modelling constraints such as these can be automatically enforced, due the fact that the ontological model is available to a CASE tool as a profile. Such ontology-derived modelling support has been demonstrated to be beneficial to IS development projects [60,61]. Previous work has derived ontological modelling rules based on Bunge's ontology [28,29], and has proposed to describe these in the form of a UML profile [31]. However, while a method to automatically derive modelling constraints from an ontology was proposed, the natural language description of Bunge's ontology prevented its demonstration [31].

---

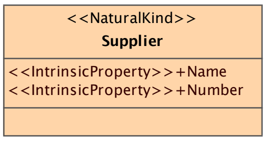
[14] Previous research on mappings from UML to Bunge's ontology [23,28,29,31, 35,60] used UML 1.x. In UML 2.0 the definitions of attributes, properties, structural features and associations have changed significantly, so that we can only present a rough sketch without revisiting and updating the existing work, which is beyond the scope of this paper. The excerpt presented here is for illustration of the principle only.

[15] For a different mapping see earlier work in [28]. There, a UML class is mapped to a functional schema and a UML attribute is mapped to a state function.

[16] The association with the filled arrow notation is the symbol for an Extension in UML [85].



(a) Excerpt from the UML model of Bunge's ontology, used as a profile



(b) Application of the profile

Fig. 4 Using the UML model of Bunge's ontology as a profile

5 Towards a Formal Ontology

This section briefly describes the UML to OWL translation and its application to the UML model of Bunge's ontology. It begins by examining previous research.

A representation of UML in description logics has been developed in [76] to explore reasoning on class diagrams. However, this work does not consider specifics such as navigability of associations and n-ary associations, nor does it consider the inverse representation of OWL in UML.

The UML storage backend plug-ins for the Protege ontology editor also allows a translation from UML to OWL and vice versa, but is more limited than what is presented here[17]. Specifically, association classes, n-ary associations, generalization of associations, association end navigability and changeability are not supported for UML import to Protege.

The approach in [75] relies on an extension of UML called the Ontology UML Profile (OUP) which allows annotation of UML models with tagged values and stereotypes defined in this profile. This allows, e.g. describing a UML class to be a union of two other classes. This approach is limited in its applicability to newly created UML models using this stereotype. Existing UML models are excluded, as they are not annotated.

---

[17] `http://protege.cim3.net/cgi-bin/wiki.pl?UMLBackendMapping` (last access on Sept. 25th, 2007)

Finally, as part of the ODM specification [72], the OMG also proposes an informative, rather than normative, mapping from UML to OWL and vice versa.

5.1 XMI to OWL Translation with XSLT

The translation between UML and OWL is implemented as a set of XSLT 1.0 style sheets (XML Style-Sheet Language Transformation) that take an XMI (XML Model Interchange) description of the UML model as input. The inverse translation is also implemented in a separate XSLT. The XMI 1.2 description (of UML 1.5) contains 5113 lines of XML code. The XSLT transform contains 283 lines of code in 5 templates. Run with the open source Saxon 8.4 XSLT processor[18] it creates an output file of 2761 lines of OWL XML code.

Table 4 shows the correspondences between concepts of each language. This table is partially based on existing work presented in [72, 76], and agrees with the correspondences between UML and OWL established there. The mappings are therefore not validated further.

The most important conclusion to be drawn from this table is that there are more OWL constructs that have no UML equivalent, than UML constructs without OWL equivalent. This supports the decision to develop Bunge's ontology in UML, rather than OWL, in order to guarantee translatability to the other language.

Table 4: Comparison of UML and OWL constructs

UML	**OWL**	**Note**
Class	Class	
Attribute	DatatypeProperty	1
Association	ObjectProperty	2
AssociationClass	Class and ObjectProperties	3
Generalization between Classes	subClassOf	
Generalization between Associations	subPropertyOf	
Multiplicities	Cardinalities	
(Generalization)	UnionOf	4
(Generalization)	IntersectionOf	5
	ComplementOf	
AssociationEnd isNavigable	InverseOf	6
	TransitiveProperty	
	SymmetricProperty	
(Objects)	OneOf (class definition by enumeration)	7

Continued on next page

---

[18] `http://saxon.sourceforge.net/`

From previous page

UML	**OWL**	**Note**
AssociationEnd participant	ObjectProperty value constraint "allValuesFrom"	
	ObjectProperty value constraint "someValuesFrom"	
AssociationEnd changeability	ObjectProperty value constraint "hasValue"	
Attribute typedFeature	DatatypeProperty value constraint "allValuesFrom"	
	DatatypeProperty value constraint "someValuesFrom"	
Attribute changeability and initial value	DatatypeProperty constraint "hasValue"	
Mutual generalization	EquivalentClass	8
	DisjointWith	9
Mutual generalization	EquivalentProperty	10
(Multiplicity constraint)	FunctionalProperty	
	SameAs	
	DifferentFrom	11
	AllDifferent	11
Aggreagtion kind		12

Note 1 UML attributes can be of a data type that is a class in the model, thus having the same semantics as a uni-directionally navigable association. If that is the case, the attribute should instead be modelled as an association *when using this translation.*

Note 2 UML associations can involve three or more participant classes, while object properties in OWL represent binary predicates. In contrast to [76] where these associations are omitted, but in agreement with [72], we translate n-ary associations to OWL classes. This is an accepted approximation, as it is difficult to correctly represent the multiplicities from the syntactic information only [87–89].

Note 3 In agreement with [72, 76] association classes are not represented as associations (of which they are specializations), but as classes (of they are also specializations) that are connected by binary associations to the classes that participate in the association class.

Note 4 The OWL union of two classes is approximated in UML as two generalizations: $A = B \cup C \rightarrow B \subseteq A \wedge C \subseteq A$. Further, these two generalizations are part of the same generalization set which is covering. This approximation rests on the interpretation of generalization as subsets, not as feature inheritance.

Note 5 The OWL intersection of two classes is approximated in UML as two generalization: $A = B \cap C \rightarrow A \subseteq B \wedge A \subseteq C$. This approximation rests on the interpretation of generalization as subsets, not as feature inheritance.

Note 6 Object properties in OWL are uni-directional while associations in UML are bi-directional, barring explicit navigability constraints. Hence, only when the inverse of an OWL object property exists, does the corresponding association become navigable in both directions. In turn, if an association is navigable in both ways, two OWL properties are modelled, inverse of each other, in agreement with [72].

Note 7 Class definition by enumeration in OWL cannot be expressed in UML, but the specified OWL instances can be modelled as UML objects of the corresponding UML classes. UML provides the Enumeration and EnumerationLiteral meta-classes. Enumeration is a subset of the meta-class DataType, "a type whose instances are identified only by their value" [85, pg. 57]. Accordingly, an EnumerationLiteral does not represent objects but data values [85, pg. 64]. Enumeration and EnumerationLiteral can therefore not be used to define classes by enumeration of instance objects.

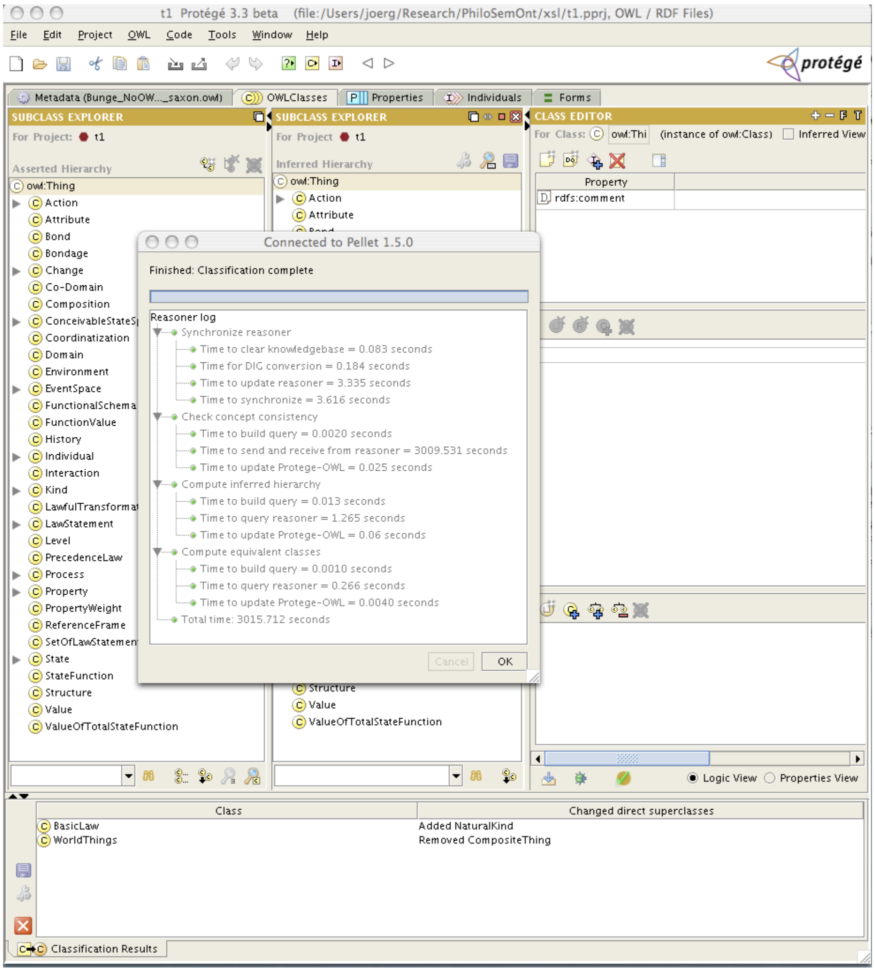
Note 8 To express OWL class equivalence in UML, we use mutual generalization: $A \equiv B \rightarrow A \subseteq B \wedge B \subseteq A$. This approximation rests on the interpretation of generalization as subsets, not as feature inheritance, and agrees with [72]. Of note, while the OMG ODM specification [72] endorses this expression, the OMG UML specification [85] in contrast requires that generalization relationships be acyclic.

Note 9 Disjointness of classes in OWL does not need an explicit modelling construct in UML, as classes by default are interpreted as disjoint [76].

Note 10 To express OWL object property equivalence in UML, we use mutual generalization of associations, analogous to OWL class equivalence, in agreement with [72]. See also Note 8 above.

Note 11 It is not necessary to express the fact that an instance is distinct from another, as this is the default assumption in UML and conceptual modelling [76].

Note 12 UML defines two kinds of aggregation types for associations (shared and composite). Characteristics of aggregation are described in terms of dynamics, e.g. instance creation or deletion [86]. However, as OWL is limited to static representations of domains, these constructs offer no additional semantics to regular associations *for purposes of translation to OWL* [76].



Connected to Pellet 1.5.0

Finished: Classification complete

```
Reasoner log
▼ Synchronize reasoner
    Time to clear knowledgebase = 0.083 seconds
    Time for DIG conversion = 0.184 seconds
    Time to update reasoner = 3.335 seconds
    Time to synchronize = 3.616 seconds
▼ Check concept consistency
    Time to build query = 0.0020 seconds
    Time to send and receive from reasoner = 3009.531 seconds
    Time to update Protege-OWL = 0.025 seconds
▼ Compute inferred hierarchy
    Time to build query = 0.013 seconds
    Time to query reasoner = 1.265 seconds
    Time to update Protege-OWL = 0.06 seconds
▼ Compute equivalent classes
    Time to build query = 0.0010 seconds
    Time to query reasoner = 0.266 seconds
    Time to update Protege-OWL = 0.0040 seconds
  Total time: 3015.712 seconds
```

Cancel OK

Class	Changed direct superclasses
BasicLaw	Added NaturalKind
WorldThings	Removed CompositeThing

Fig. 5 Bunge Ontology in Protege, using the Pellet reasoner for consistency checking and concept hierarchy inference

5.2 Using the OWL Ontology

The generated OWL code was validated and confirmed to be OWL-DL compliant[19]. It can be imported into and maintained with the Protege ontology editor[20]. Figure 5 shows the Bunge ontology in Protege 3.3, interfaced with the Pellet reasoner.

At this stage, the Bunge ontology becomes usable for formal reasoning. The Bunge ontology is describable in the description logic $\mathcal{ALCHIN}$, i.e. attributive language $\mathcal{AL}$ plus role hierarchies $\mathcal{H}$, inverse or symmetric roles $\mathcal{I}$, and number restrictions $\mathcal{N}$. In other terminology, this is the

---

[19] `http://www.mygrid.org.uk/OWL/Validator`

[20] `http:://protege.stanford.edu`

description logic $\mathcal{SHIN}$, a subset of $\mathcal{SHIQ}$, where $\mathcal{S} = \mathcal{ALC}$ and $\mathcal{Q}$ (quantifiers) subsumes $\mathcal{N}$ (number restrictions) [18–20]. Using the Pellet open source reasoner[21] via the DIG (Description Logics Implementation Group) interface from Protege, the Bunge ontology was checked for consistency, i.e. concept satisfiability. All concepts were found to be satisfiable. Further, the reasoner was used to infer the concept hierarchy, i.e. find the most specific super-concept for any concept. No changes to the explicit concept hierarchy were found. These results lend further confidence to the validity of the UML to OWL translation proposed above.

As the Bunge ontology is a TBox (terminology) only, limited practical reasoning can be performed. Concrete applications require instances of the concepts, i.e. an ABox (assertions), to be useful. Besides being used to reason over instances, another possible use for the formal Bunge ontology is in the integration of multiple domain ontologies, as indicated in Sect. 1.2. Different domain ontologies can specialize the same Bunge concepts and can then be used for integrated reasoning across multiple application domains. For example, medical diagnosis ontologies could be integrated with drug effects and treatment ontologies.

6 Discussion, Challenges, and Future Research

Information loss is inevitable when translating an ontology description from a specification in natural language and set theory, which is equivalent to full first order logic, to the restricted languages of UML, used in conceptual modelling, and OWL, used on the semantic web. This is an inevitable drawback in the absence of more expressive standards for semantic web ontologies.

On the UML side, one can make use of the Object Constraint Language OCL, which is capable of expressing first order logic. However, while OCL is well defined, little to no tool support exists. For example, most UML modelling tools simply store OCL constraints as text, and the few existing OCL parsers and compilers (e.g. [90]) are not well integrated. Consequently, while use of OCL would allow a better representation of Bunge's ontology, the ontology would not be more accessible to researchers or practitioners. It would also lead to two very distinct models, one in OWL and another, much more expressive one, in UML/OCL. In the interest of developing a single model, we have therefore decided against the use of OCL.

As the brief discussion in Section 5 shows, most of the difficulties translating between UML and OWL arise when trying to translate OWL constructs which have no direct counterpart in UML, e.g. EquivalentClass. In these cases, approximations have been outlined for a translation from OWL to UML. The work presented in this paper requires a translation only from UML to OWL. Consequently, the validity of the OWL model of Bunge's ontology is not affected by these approximations.

---

[21] `http://pellet.owldl.com/download`

7 Conclusion

In summary, we have developed a validated model of Bunge's ontology in two important representation formats, UML and OWL, making this widely-used ontology more accessible to researchers in both the conceptual modelling and semantic web community.

In the future, we plan to use this representation of Bunge's ontology as a foundation for formal and rigorous research. Specifically, three important applications are currently being pursued. (1) The use of the UML model in schema integration to e.g. compare Bunge's ontology with enterprise models, such as ebXML. (2) The use of the UML model as a profile to generate ontological modelling constraints. (3) The use of the OWL representation in ontology integration in the semantic web context.

The intention of this research is to begin an ongoing community process. Through the use of open formats and open tools we hope to ensure this effort will be supported by contributions from the research community. The ongoing evolution of the ontology representation by the community is supported by a web site with discussion forums[22]. The research community is invited to participate in a joint effort of creating an accepted consensus model of Bunge's ontology in useful and widely-used representation formats.

References

1. Guarino, N.: Formal ontology and information systems. In Guarino, N., ed.: Formal Ontology in Information Systems. IOS Press, Amsterdam, NL (1998) 3–15
2. Fensel, D., Hendler, J., Lieberman, H., Wahlster, W., eds.: Spinning the Semantic Web - Bringing the World Wide Web to its Full Potential. The MIT Press, Cambridge, MA (2003)
3. Kim, H.: Predicting how ontologies for the semantic web will evolve. Communications of the ACM **45** (2002) 48–54
4. Uschold, M., Gruninger, M.: Ontologies: Principles, methods and applications. Knowledge Engineering Review **11** (1996) 93–155
5. Noy, N.F., Hafner, C.D.: The state of the art in ontology design: A survey and comparative review. AI Magazine **18** (1997) 53–74
6. Smith, B., Welty, C.: Ontology: Towards a new synthesis. In: Proceedings of the Second International Conference on Formal Ontology and Information Systems, FOIS'01, October 17-19, Qgunquit, Maine. (2001) iii–ix
7. Gruber, T.R.: A translation approach to portable ontology specification. Knowledge Acquisition **5** (1993) 199–220
8. Pinto, H.S., Martins, J.P.: Ontologies: How can they be built? Knowledge and Information Systems **6** (2004) 441–464
9. Gruninger, M., Lee, J.: Ontology applications and design. Communications of the ACM **45** (2002) 39–41
10. Holsaple, C.W., Joshi, K.: A collaborative approach to ontology design. Communications of the ACM **45** (2002) 42–47

---

[22] `http://www.mcs.vuw.ac.nz/~jevermann/Bunge/`

11. Fox, M., Grüninger, M.: Ontologies for enterprise integration. In: Proceedings of the Second International Conference on Cooperative Information Systems. (1994) 82–89
12. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of the 2nd International Conference on Formal Ontologies in Information Systems FOIS, Ogunquit, Maine 2001. (2001) 2–9
13. Guha, R., Lenat, D.B.: Enabling agents to work together. Commmunications of the ACM **37** (1994) 127–142
14. Lenat, D.B., Guha, R.V., Pittman, K., Pratt, D., Shepherd, M.: CYC: Toward programs with common sense. Communications of the ACM **33** (1990) 30–49
15. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM **38** (1995) 33–38
16. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuiness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language – W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-ref. (2004)
17. Baader, F., Calvanese, D., McGuiness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Cambridge University Press, Cambridge, UK (2003)
18. Horrocks, I., Patel-Schneider, P.F.: Reducing owl entailment to description logic satisfiability. Journal of Web Semantics **1** (2004) 345–357
19. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. J. of Web Semantics **1** (2003) 7–26
20. Horrocks, I., Sattler, U.: A tableau decision procedure for $\mathcal{SHOIQ}$. Journal of Automated Reasoning **39** (2007) 249–276
21. Bera, P., Wand, Y.: Analyzing OWL using a philosophy-based ontology. In: Proceedings of the 2004 Conference on Formal Ontologies in Information Systems FOIS, Torino, Italy. (2004)
22. Bera, P., Krasnoperova, A., Wand, Y.: Improving real-world semantics in OWL. Working paper, The Sauder School of Business at UBC (2005)
23. Dussart, A., Aubert, B.A., Patry, M.: An evaluation of inter-organizational workflow modeling formalisms. Journal of Database Management **15** (2004) 74–104
24. Fettke, P., Loos, P.: Ontological evaluation of reference models using the Bunge-Wand-Weber model. In: Proceedings of the 2003 Americas Conference on Information Systems, August 4-6, Tampa, FL. (2003)
25. Green, P., Rosemann, M.: Integrated process modelling: An ontological analysis. Information Systems **25** (2000) 73–87
26. Green, P., Rosemann, M.: Ontological analysis of integrated process models: Testing hypotheses. The Australian Journal of Information Systems **9** (2001) 30–38
27. Green, P., Rosemann, M., Indulska, M.: Ontological evaluation of enterprise systems interoperability using ebXML. IEEE Transactions on Data and Knowledge Engineering **17** (2005) 713–725
28. Evermann, J., Wand, Y.: Towards ontologically based semantics for UML constructs. In Kunii, H., Jajodia, S., Solvberg, A., eds.: Proceedings of ER 2001. (2001) 354–367
29. Evermann, J., Wand, Y.: An ontological examination of object interaction in conceptual modeling. In: Proceedings of the Workshop on Information Technologies and Systems WITS'01, New Orleans, December 15-16, 2001. (2001) 91–96

30. Evermann, J.: The association construct in conceptual modelling - an analysis using the Bunge ontological model. In Pastor, O., e Cunha, J.F., eds.: Proceedings of the 17th International Conference on Advanced Information Systems Engineering, CAiSE 2005, Porto, Portugal, Springer Verlag (2005) 33–47
31. Evermann, J., Wand, Y.: Towards formalizing domain modeling semantics in language syntax. IEEE Transactions on Software Engineering **31** (2005) 21–37
32. Opdahl, A., Henderson-Sellers, B.: Evaluating and improving OO modelling languages using the BWW-model. In: Proceedings of the Information Systems Foundation Workshop 1999. (1999) www.comp.mq.edu.au/isf99/Opdahl.htm.
33. Opdahl, A., Henderson-Sellers, B., Barbier, F.: An ontological evaluation of the OML metamodel. In Falkenberg, E., Lyytinen, K., eds.: Information System Concepts: An Integrated Discipline Emerging. IFIP/Kluwer, Netherlands (1999)
34. Opdahl, A.L., Henderson-Sellers, B.: Grounding the OML metamodel in ontology. The Journal of Systems and Software **57** (2001) 119–143
35. Opdahl, A., Henderson-Sellers, B.: Ontological evaluation of the UML using the Bunge-Wand-Weber model. Software and Systems Modeling **1** (2002) 43–67
36. Wand, Y., Weber, R.: An ontological evaluation of systems analysis and design methods. In Falkenberg, E., Lingreen, P., eds.: Information System Concepts: An In-Depth Analysis. Elsevier Science Publishers B.V., North-Holland (1989)
37. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Journal of Information Systems (1993) 217–237
38. Wand, Y., Storey, V.C., Weber, R.: An ontological analysis of the relationship construct in conceptual modeling. ACM Transactions on Database Systems **24** (1999) 494–528
39. Guizzardi, G., Herre, H., Wagner, G.: On the general ontological foundations of conceptual modeling. In: Proceedings of the 21st International Conference on Conceptual Modeling (ER-2002), Tampere, Finland, Springer-Verlag (2002)
40. Guizzardi, G., Wagner, G., van Sinderen, M.: A formal theory of conceptual modeling universals. In: Proceedings of the Workshop on Philosophy and Informatics (WSPI), Cologne, Germany. (2004)
41. Heller, B., Herre, H.: Ontological categories in GOL. Axiomathes **14** (2004) 57–76
42. Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., Michalek, H.: General formal ontology (GFO): A foundational ontology integrating objects and processes. Part 1: Basic principles (version 1.0). Technical report, University of Leipzig, Research Group Ontologies in Medicine (Onto-Med) (2006)
43. Masolo, C., Borgo, S., Gangemi, A., Guariono, N., Oltramari, A.: Ontology library, WonderWeb deliverable D18. Technical report, University of Trento Laboratory for Applied Ontology (2003)
44. Green, P.: Use of information systems analysis and design (ISAD) grammars in combination in upper CASE tools - an ontological evaluation. In: Proceedings of the 2nd CAiSE/IFIP8.1 International Workshop on the Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'97). Barcelon, Spain. (1997) 1–12

45. Green, P., Rosemann, M.: Applying ontologies to business and systems modelling techniques and perspectives: Lessons learned. Journal of Database Management **15** (2004) 105–117
46. Herrera, S.I., Pallioto, D., Tkachuk, G., Luna, P.A.: Ontological modelling of information systems from Bunge's contributions. In: Proceedings of the PHISE workshop, Porto, Portugal. (2005) 571–582
47. Opdahl, A.L., Henderson-Sellers, B.: A template for defining enterprise modelling constructs. Journal of Database Management **15** (2004) 39–73
48. Rosemann, M., Green, P.: Developing a meta-model for the Bunge-Wand-Weber ontological constructs. Information Systems **27** (2002) 75–91
49. Soffer, P., Golany, B., Dori, D., Wand, Y.: Modeling off-the-shelf information systems requirements: An ontological approach. Requirements Engineering **6** (2001) 663–679
50. Soffer, P., Wand, Y.: Goal-driven analysis of process model validity. In: Proceedings of the 16th International Conference on Advanced Information Systems Engineering, CAiSE 2004, Riga, Latvia. (2004) 521–535
51. Wand, Y., Weber, R.: Mario Bunge's ontology as a formal foundation for information systems concepts. In Weingartner, P., Dorn, G., eds.: Studies on Mario Bunge's Treatise. Rodopi, Atlanta (1990)
52. Wand, Y., Wang, R.: Anchoring data quality dimensions in ontological foundations. Communications of the ACM **39** (1995) 86–95
53. Wand, Y., Monarchi, D., Parsons, J., Woo, C.: Theoretical foundations for conceptual modelling in information systems development. Decision Support Systems **15** (1995) 285–304
54. Weber, R.: Ontological Foundations of Information Systems. Coopers & Lybrand, Melbourne, Australia (1997)
55. Angeles, P.: Dictionary of Philosophy. Harper Perennial, New York, NY (1981)
56. Bunge, M.A.: Ontology I: The Furniture of the World. Volume 3 of Treatise On Basic Philosophy. D. Reidel Publishing Company, Dordrecht, Holland (1977)
57. Bunge, M.A.: Ontology II: A World of Systems. Volume 4 of Treatise On Basic Philosophy. D. Reidel Publishing Company, Dordrecht, Holland (1979)
58. Bodart, F., Patel, A., Sim, M., Weber, R.: Should optional properties be used in conceptual modelling? A theory and three empirical tests. Information Systems Research **12** (2001) 384–405
59. Cockcroft, S., Rowles, S.: Ontological evaluation of health models: Some early findings. In: Proceedings of the 7th Pacific Asia Conference on Information Systems, 10-13 July, Adelaide, Australia. (2003) 611–625
60. Evermann, J., Wand, Y.: Ontology based object-oriented domain modelling: Fundamental concepts. Requirements Engineering (2005) 146–160
61. Evermann, J., Wand, Y.: Ontological modelling rules for UML: An empirical assessment. The Journal of Computer Information Systems **46** (2006)
62. Gemino, A.: Empirical Comparisons of Systems Analysis Modeling Techniques. PhD thesis, University of British Columbia, Canada (1999)
63. Gemino, A., Wand, Y.: Complexity and clarity in conceptual modelling: Comparison of mandatory and optional properties. Data and Knowledge Engineering **55** (2005) 301–326
64. Weber, R., Zhang, Y.: An analytical evaluation of NIAM's grammar for conceptual schema diagrams. Information Systems Journal **6** (1996) 147–170

65. Davies, I., Green, P., Milton, S., Rosemann, M.: Using meta models for the comparison of ontologies. In: Proceedings of the EMMSAD Workshop, Velden, Austria. (2003)
66. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. (1997) http://www.aiai.ed.uk/project/pub/documents/1998/98-ker-ent-ontology.ps.
67. Gomez-Perez, A., Fernandez-Lopez, M., Corcho, O., eds.: Ontological Engineering. Springer Verlag, London et al. (2004)
68. Staab, S., Studer, R., eds.: Handbook on Ontologies. Springer Verlag, Berlin et al. (2004)
69. Su, X., Ilebrekke, L.: A comparative study of ontology languages and tools. In: Proceedings of the 10th International Conference on Advances in Information Systems Engineering (CAiSE), Toronto, ON. (2002)
70. Su, X., Ilebrekke, L.: Using a semiotic framework for a comparative study of ontology languages and tools. In: Information Modeling Methods and Methodologies. IDEA Group Publishing (2004)
71. Recker, J., Mendling, J.: On the translation between BPMN and BPEL: Conceptual mismatch between process modelling languages. In: Proceedings of the EMMSAD workshop, Luxembourg. (2006) 521–532
72. Object Management Group: Ontology Definition Meta-Model. (2006) Final adopted specification, Document ptc/06-10-11.
73. Guizzardi, G., Wagner, G., Guarino, N., van Sinderen, M.: An ontologically well-founded profile for UML conceptual models. In: Proceedings of the 16th International Conference on Advances in Information Systems Engineering (CAiSE), Latvia, Springer-Verlag (2004)
74. Heller, B., Herre, H.: Formal ontology and principles of GOL. Technical report, University of Leipzig, Research Group Ontologies in Medicine (Onto-Med) (2003)
75. Duric, D.: MDA-based ontology infrastructure. International Journal on Computer Science and Information systems **1** (2004) 91–116
76. Berardi, D., Calvanese, D., De Giacomo, G.: Reasoning on UML class diagrams. Artificial Intelligence **168** (2005) 70–118
77. Warmer, J., Kleppe, A.: The Object Constraint Language: Precise Modelling with UML. Addison-Wesley (1999)
78. Cranefield, S., Purvis, M.: UML as an ontology modelling language. In: Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence. (1999) 46–53
79. Gadamer, H.G.: Philosophical Hermeneutics. University of California Press (1976)
80. Ricoeur, P.: Interpretation Theory: Discourse and the Surplus of Meaning. The Texas Christian University Press, Fort Worth, TX (1976)
81. Boland, R.: Phenomenology: A preferred approach to research in information systems. In Mumford, E., Hirschheim, R., Fitzgerald, G., Wood-Harper, T., eds.: Research Methods in Information Systems. Elsevier (1985)
82. Chalmers, M.: Hermeneutics, information, and representation. European Journal of Information Systems **13** (2004) 210–220
83. Myers, M.: Dialectical hermeneutics: A theoretical framework for the implementation of information systems. Information Systems Journal **5** (1995) 51–70

84. Prasad, A.: The contest over meaning: Hermeneutics as an interpretive methodology for understanding texts. Organizational Research Methods **5** (2002) 12–33
85. Object Management Group: Unified Modeling Language: Superstructure. (2005) Document formal/05-07-04.
86. Henderson-Sellers, B., Barbier, F.: Black and white diamonds. In: Proceedings of the 1999 Conference on the Unified Modelling Language UML 99. (1999) 550–565
87. Flynn, D., Knight, D., Laender, A.: Multiple relationships: An analysis of their semantics and their modelling. In Falkenberg, E.D., Hesse, W., Olive, A., eds.: Information System Concepts: Towards a consolidation of views. IFIP/Chapman & Hall, London (1995)
88. Jones, T.H., Song, I.Y.: Binary equivalents of ternary relationships in entity-relationship modeling: A logical decomposition approach. Journal of Database Management **11** (2000) 12–19
89. Genova, G., Llorens, J., Martinez, P.: The meaning of multiplicity of n-ary associations in UML. Software and Systems Modelling **1** (2002) 86–97
90. Demuth, B.: The Dresden OCL toolkit and its role in information systems development. In: 13th International Conference on Information Systems Development: Methods and Tools, Theory and Practice Conference, Advances in Theory, Practice, and Education (ISD'94), Vilnius, Lithuania. (2004)