Principled Pragmatism: A Guide to the Adaptation of Ideas from Philosophical Disciplines to Conceptual Modeling

David W. Embley[1], Stephen W. Liddle[2], and Deryle W. Lonsdale[3]

[1] Department of Computer Science,
[2] Information Systems Department,
[3] Department of Linguistics and English Language,
Brigham Young University, Provo, Utah 84602, U.S.A.

Abstract. The synergism among the traditional disciplines of ontology, epistemology, logic, and linguistics and their potential for enhancing conceptual-modeling applications is not fully understood. Better understanding how to adapt ideas from these disciplines should lead to improved serviceability of conceptual modeling. We argue in this position paper, however, that application objectives, rather than philosophical tenets, should guide the adaptation of ideas from these disciplines. Thus, an appropriate balance of discipline-based theory and pragmatism should temper adaptations. We evaluate the principled pragmatism we advocate by presenting several case-study examples. Each illustrates that an appropriate adaptation of ideas from the disciplines of ontology, epistemology, logic, and linguistics can significantly guide conceptual-modeling research and help build successful conceptual-modeling applications.

1 Introduction

The applicability of ontology, epistemology, logic, and linguistics to conceptual modeling seems compelling. But what role should these disciplines play in facilitating conceptual-modeling applications? To what extent should conceptual-modeling researchers adopt or adapt philosophical ideas, positions, adages, and objectives from these disciplines? Must they be purists in their adaptation, or is it appropriate to leverage fundamental ideas and objectives and let the pragmatism of the application dictate the adoption and adaptation of theoretical tenets of the various perspectives within and surrounding these disciplines?

We argue in this position paper that application-dependent pragmatism should guide the adaptation of ideas from these disciplines to the various research directions within the conceptual-modeling community. In adapting ideas from these disciplines to conceptual modeling, we espouse the adage attributed to Einstein that everything should be made "as simple as possible, but no simpler."

To establish our position, we first sketch our contextual view of ontology, epistemology, logic, linguistics, and conceptual modeling (Section 2.1). We then argue that being solution-oriented requires appropriate selective adaptation of ideas from these areas (Section 2.2). Appropriate selectivity requires tempering by two crucial, overarching considerations: the adaptation must be principled, and it must be pragmatic. From our perspective forcing purist views for adaptation may overly complicate the conceptual-modeling application in opposition to Einstein's sufficiency-with-simplicity adage. To make our views concrete, we present several case-study examples to show how the principle of practical pragmatism has served and can further serve as a guide to adapting ideas to conceptual modeling (Section 3). Then, as we conclude (Section 4), we generalize and assert that the principled pragmatism we advocate is potentially more far-reaching in its implications than to just the case-study examples we use for illustration. It provides a vision and perspective for adapting philosophical disciplines to conceptual-modeling applications. Further, it answers in part the question about the relationship among "Ontology as an Artifact, Ontology as a Philosophical Discipline, Conceptual Modeling, and Metamodeling."

2 Philosophical Disciplines and Conceptual Modeling

2.1 Contextual Overview

Ontology, as a field of philosophy, investigates problems of existence: what exists, how do we know what exists, how does an object's existence relate to universal reality, and related questions. Deciding what reality is, and which relations characterize reality, are at the core of ontological investigation. A central theme of ontology is ontological commitment, which is about having enough evidence to commit to an object's existence.

Epistemology studies knowledge and belief. It explores where knowledge comes from, how it is represented (including its structure), what its limits are, and how it can be used to refute assertions or support belief and discover truths. Important topics include how to quantify, describe, create, disseminate, and operationalize knowledge. For example, how much knowledge is necessary (or sufficient) for accomplishing a given task? How do we acquire and codify knowledge? What constitutes good evidence for justification of a belief?

Logic is about valid reasoning. It allows us to explore generalization, abstraction, and inferred relationships among objects that exist. Logic can be formalized in several different ways: proof theory, model theory, formal languages, linguistic expressions, mental representations, or graphical visualizations. Principles of correct reasoning guarantee the validity of inferences and set up systems of meaning that can be manipulated at higher levels of abstraction.

Linguistics investigates languages, either formal (e.g. logic or mathematics) or natural (i.e. human languages). Since language is crucial to communicating ideas, knowledge, beliefs, and logic, all of the above areas are of concern to linguists. So, too, are the structure, properties, contextual realities, and meaning of sounds, words, sentences, discourse, and dialog.

Conceptual modeling deals with computational representation of concepts and how to communicate these representations. Concept representation, from a linguistically-informed semantics perspective, is in part referential or denotational: symbols relate to objectively verifiable, external, real-world objects. Communication about conceptualizations is social: symbols facilitate interpersonal or intergroup communication. Conceptual modeling comprises both perspectives: most model instances are explicitly denotational or referential in their content and serve a social function by facilitating communication among stakeholders.

2.2 Principled Pragmatism

Our approach to adapting ideas from philosophical disciplines to conceptual modeling is based on principled pragmatism. The overarching epistemological principle is that our work is ontology-driven. Our pragmatism views ontologies as formal specifications of some domain that are simple enough that users can encode them without extensive experience in knowledge engineering or the truth-maintenance systems typically used for large-scale ontology development. We maintain simplicity by minimally selecting and adapting ideas from each of the disciplines in a combination that appropriately enhances our application. By extension, we assert that these same principles hold for other conceptual-modeling applications. We believe, for example, in harmony with Smith [1], that more exacting and purer ontological descriptions would enhance integration and interoperability applications.[4]

In summary, we assert our position: When adapting ideas from philosophical disciplines to conceptual modeling, we should find the right balance, being neither too dogmatic (insisting on a discipline-purist point of view) nor too dismissive (ignoring contributions other disciplines can make to conceptual-modeling applications).

3 Case Study Applications

To illustrate the principled pragmatism we advocate, we present five case-study examples. For each case study, we explain how the disciplines of ontology, epistemology, logic, and linguistics guide the theoretical conceptual-modeling underpinnings of these applications. We also show that as the applications become more complex, they draw more heavily on philosophical disciplines for

[4] Although we have engaged in a fair amount of research on integration (e.g., see [2] as a summary), we have observed (unfortunately) that other than having a formally defined conceptual-modeling language that encourages adherence to ontological principles, we do not know how to ensure that users will model in an ontologically correct way. (We thus omit integration and interoperability from our case-study examples.) We assert, however, that although applications should drive abstraction, abstraction should be proper—should maintain generally-agreed upon ontological descriptions for the application. Doing so will surely enhance integration applications since integration of conceptualizations across abstraction levels is less painful than integration across user-modeled peculiarities. Cost effectiveness in interoperability will likely drive us toward more exacting ontological descriptions, and ontology-oriented conceptual modelers can and should provide appropriate and needed guidance.

support. Thus, the relationship between philosophical disciplines and conceptual-modeling applications becomes blurred so that it is no longer clear whether the artifact produced is a conceptual model or whether it is an ontology drawing heavily on conceptualizations from epistemology, logic, and linguistics. Further, we argue: it does not matter. What matters is that the application is well-served by the artifact and that researchers are able to build serviceable applications by finding good synergistic combinations of ideas drawn from the overlapping disciplines of conceptual modeling, ontology, epistemology, logic, and linguistics.

3.1 Conceptual-Model-Based Information Extraction

The problem of information extraction is to turn raw text into searchable knowledge. It requires finding computational answers to the fundamental questions of ontology ("What exists?"), epistemology ("What is knowledge and how can it be represented?"), logic ("What facts are known or can be inferred from known facts?"), and linguistics ("What meaning do language symbols convey?").

Answering these questions computationally leads naturally to a conceptual-modeling resolution of the information-extraction problem [3, 4]. This should not be surprising since ontological conceptualizations have been the basis for formalizing information since the days of Aristotle [5], and they still provide the underlying schemas for today's fact-filled databases. Our principled pragmatic approach to building an information-extraction system thus begins with a conceptual model that defines a narrow domain of interest. For example, the conceptual-model instance in Figure 1 models the information found in typical car ads. The conceptual-model instance answers the question of "what exists" in the narrow domain of interest and is therefore ontological in nature. Next, we linguistically ground the conceptual-model instance by saying what language symbols convey meaning for each of the object sets. In Figure 2, for example, the regular expression for the external representation of *Price* recognizes "$11,995" in the car ad in Figure 3. The dictionary *CarMake.lexicon* includes an entry for "CHEVY", and thus it, along with the recognized price and the other information in the car ad, are extracted and entered into a data repository (RDF triples in our implementation) whose schema (OWL in our implementation) corresponds to the conceptual-model instance in Figure 1. The populated conceptual-model instance is a model-theoretic interpretation and thus immediately supports inference via established logic systems (the Pellet reasoner in our implementation and SPARQL queries generated from free-form user input).

As for principled pragmatism, we emphasize that although the ideas we adapt for conceptual-model-based information extraction are foundational, we do not plumb the depths of these foundational ideas. Purists might not agree, for example, that the conceptualization we propose is even an ontology. We argue, however, that since our conceptualizations not only answer "What exists?" but also explain how we know what exists in terms of linguistic clues, they are ontologies. Similarly, purists might not agree that populating a data repository involves epistemology, but because it computationally answers a fundamental epistemological question ("How is knowledge acquired?"), it is epistemological.

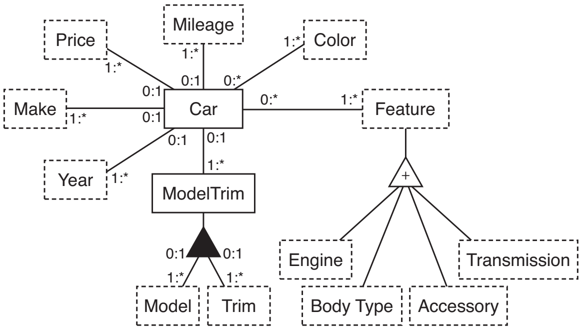


Fig. 1. Diagram for Conceptual-Model-Based Information Extraction.

```
Price
   internal representation: Integer
   external representation: \$[1-9]\d{0,2},?\d{3} | \d?\d [Gg]rand | ...
   context keywords: price|asking|obo|neg(\.|otiable)| ...
   ...
   LessThan(p1: Price, p2: Price) returns (Boolean)
   context keywords: (less than | < | under | ...)\s*{p2} | ...
   ...
Make
   ...
   external representation: CarMake.lexicon
   ...
```

Fig. 2. Linguistic Grounding of *Price* and *Make* (partial).

Our use of logic and reasoning is likely not controversial, although we must be careful not to adopt so much that our system becomes undecidable or intractable. Our use of linguistics is intentionally simplistic: our lexical grounding computationally enables the conveyance of linguistic meaning without the necessity of the traditional depth of natural-language processing systems.

3.2 The Conceptual-Modeling Language OSM

Perspectives that motivate different approaches to conceptual-model language design are varied, but it is generally true that the target domain inspires the set constructs in the modeling language. In the context of database design, the ER model is natural, but in the context of real-time systems, a language like Statecharts is more appropriate. The challenge in modeling-language design is determining the "best" set of constructs to use: enough to cover the required concepts, but not so many that the language is too difficult to master. Also, it is important to define languages carefully and precisely, preferably with an underlying mathematical formalism that supports unambiguous definitions and even the ability to translate the model directly to an executing system.

Under the guise of the principled but pragmatic approach we advocate in this position paper, we created Object-oriented Systems Modeling (OSM), an

'97 CHEVY Cavalier, Red, 5 spd, only 7,000 miles on her.
Previous owner heart broken! Asking only $11,995.
#1415 JERRY SEINER MIDVALE, 566-3800 or 566-3888

Fig. 3. Car Ad (unaltered from original).

object-oriented conceptual modeling language [6]. Figure 4 shows an example of a simple surveillance controller and illustrates how OSM describes objects (e.g. *Surveillance Controller*, *Dectector ID*, *Dectection Event*, and *Timestamp* in Figure 4), relationships among objects (e.g. *Detector Event has Timestamp*), object behavior (e.g. event detection via *@detection* and controller reaction), and interaction among objects inside and outside the controller (e.g. *user abort*). It properly describes the behavior of a surveillance controller using a mixture of natural language and formal constructs, in real-world terms. And yet this model instance can directly execute in prototype fashion and can be fully formalized in terms of OSM to become operational. Thus, OSM is a conceptual-model programming language [7] as well as a conceptual-modeling analysis language.

In designing OSM, we tried to hold to "ontological" principles, believing that having the right abstractions and expressing concepts in natural terms (not "programming" terms) leads to better models. The ideas in Bunge's ontology of physical systems [8] resonate nicely with our work on OSM. At the same time, we wove principles of epistemology and logic into the fabric of OSM. Populated OSM model instances constitute formal interpretations in model theory, and thus immediately provide a perspective on facts and knowledge (epistemology) and a basis for formal query and inference (logic). As a communication language, we used simple linguistic principles to make OSM human-readable. It is a small detail, but by convention we name object sets with spaces separating the words, not dashes or underscores as is more typical of many computer-friendly languages, and OSM provides for readable, sentence-like, relationship-set names (e.g. *Surveillance Controller has record of Detector Event* rather than more typical *detectedEvent* that is easier for a machine to parse).

3.3 Conceptualizations for Learning and Prediction

Recent initiatives by government agencies (e.g. IARPA-KDD [9]) and by academic think-tank groups (e.g. ACM-L [10]) require conceptualizations that track behavior, model what has happened and is currently happening, and analyze past and present behavior. The objectives of tracking, modeling, and analyzing include being able to predict future behavior, play out "what-if" scenarios, and warn of possible impending disasters. Although conceptual modeling can provide the foundation for storing the necessary information to support these initiatives, the lack of a unified, fact-oriented, conceptual-model formalism for temporal semantics, metamodeling, and reification leads to difficulties in reasoning about system properties and predicting future behavior from past behavior [11]. We can better serve these needs by a formalized conceptualization, called *OSM-Logic* [12], that more closely matches application demands.

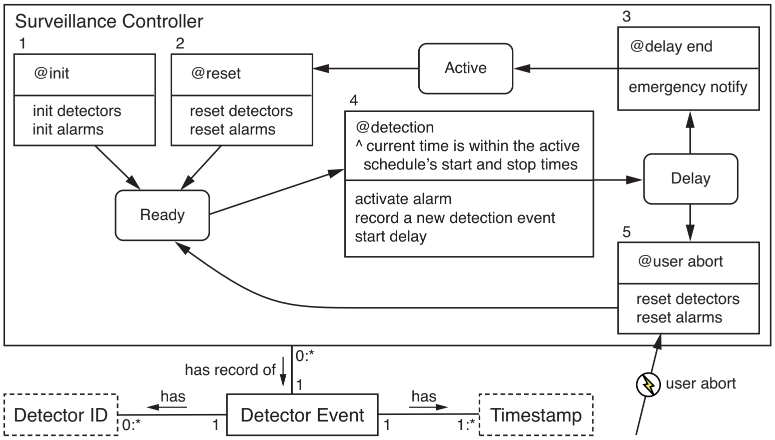


Fig. 4. OSM Diagram Illustrating Object Behavior.

OSM-Logic illustrates both the "as simple as possible" and the "no simpler" aspects of principled pragmatism. The formalism is "no simpler" because the complexity of learning-and-prediction applications demands the inclusion of real-world temporality. The inclusion of temporality, however, is "as simple as possible" because it consists only of adding a timestamp argument to each event-fact predicate and a timestamp-argument pair to each existence-fact predicate. Thus, for example, in OSM-Logic, we formalize the fact that surveillance controller s is in the *Active* state of the behavior conceptualization in Figure 4 from time t to time t' as the predicate-calculus fact $inStateActive(s, t, t')$. Also of prime importance for simplicity, the temporal predicate-calculus formalism applies uniformly to all aspects of the modeling—the metamodel instance as well as the model instance, object behavior as well as object existence, and reification for both high-level and low-level specification. And yet, it is "no simpler" than necessary for these applications, because its temporal semantics provides for a temporal history of the existence and behavior of all objects, its time-dependent metamodeling supports the tracking of model-instance evolution, and its reification properties make high-level abstractions as concrete and formal as low-level particulars, allowing users to either remove detail to survey the broader landscape or to drill down to the finest level of detail to find relevant facts and associations.

3.4 Multilingual Extraction Ontologies

Valuable local information is often available on the web, but encoded in a foreign language that non-local users do not understand. Hence the epistemological and linguistic problem: Can we create a system to allow a user to query in language L_1 for facts in a web page written in language L_2? We propose a suite of multilingual extraction ontologies as a solution to this problem [13]. We ground extraction ontologies in each language of interest, and we map both the data and the metadata via the language-specific extraction ontologies through a central, language-agnostic ontology, reifying our ontological commitment to cross-linguistic information extraction in a particular domain. Our world model can thus be language-neutral and grounded in semantic and lexical equivalencies to a practical, operational degree [14]. This allows new languages to be added by only having to provide one language-specific ontology and its mapping to the language-agnostic ontology.

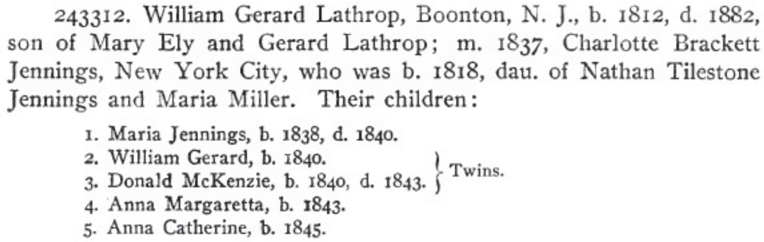
Mappings at several linguistic levels are required to assure multilingual functionality. Structural mappings (à la mappings for schema integration) associate related concepts across languages. Data-instance mappings mediate content of largely terminological nature: scalar units (i.e. fixed-scale measurements such as weight, volume, speed, and quantity); lexicons (i.e. words, phrases, and other lexical denotations); transliteration (i.e. rendering of proper names across orthographic systems); and currency conversions (i.e. mapping temporally varying indexes like monetary exchange rates). Most of this type of information is primarily referential in nature, with some element of pragmatics since commonly adopted standards and norms are implied, and these vary across cultural and geopolitical realms (e.g. use of the metric vs. imperial measurement systems). One other type of mapping, called commentary mappings, are added to the ontologies: these document cultural mismatches that may not be apparent to the monolingual information seeker, for example tipping practices in restaurants in the target culture.

3.5 Fact Extraction from Historical Documents

Many people are interested in explicit facts, as well as implied facts, found in historical documents. From the document snippet in Figure 5, taken from an online Ely Family History, facts such as *SonOf("William Gerard Lathrop", "Mary Ely", "Gerard Lathrop")* and *BornInYear("Mary Ely", "1818")* are evident, as are implied facts such as *SurnameOf("Maria Jennings", "Lathrop")* and *GrandmotherOf("Mary Ely", "Maria Jennings Lathrop")*. Images of these documents exist, and OCR engines can produce searchable text from these images.

To make explicit facts searchable, we can semantically annotate the OCRed text of the original document images using conceptual-model-based information extraction. To make implicit facts searchable, we can generate them via logic rules. Unlike the assumption of a short, data-rich text description of a single item like in a classified ad, an obituary, or a Wikipedia article, historical documents have facts about many items (e.g. many people, places, and events in the Ely family history). Therefore, computational solutions require us to delve more deeply into ontology, epistemology, logic, and linguistics.

To obtain ontological commitment, we must address the question of having sufficient evidence to declare the existence of an object. For historical documents, the appearance of names for named entities is usually sufficient, but seeing the name of a make of a car like "Ford" in "The Ford Administration, 1974–1977" does not imply the existence of a car, although seeing "Ford" in "John drove



243312. William Gerard Lathrop, Boonton, N. J., b. 1812, d. 1882, son of Mary Ely and Gerard Lathrop; m. 1837, Charlotte Brackett Jennings, New York City, who was b. 1818, dau. of Nathan Tilestone Jennings and Maria Miller. Their children:

1. Maria Jennings, b. 1838, d. 1840.
2. William Gerard, b. 1840. } Twins.
3. Donald McKenzie, b. 1840, d. 1843. }
4. Anna Margaretta, b. 1843.
5. Anna Catherine, b. 1845.

Fig. 5. Text Snippet from Page 419 of the Ely Family History.

the Ford he bought in 1925 for 20 years" does imply a car's existence. In seeking computational solutions, we apply principled pragmatism by appropriately seeking for linguistic clues denoting the existence of things but do not go so far as to delve into ethereal metaphysical arguments about existence.

Regarding epistemology and logic, Plato, and those who follow his line of thought, demand of knowledge that it be a "justified true belief" [15]. Computationally, for facts in historical documents, we "establish" truth via provenance. Users can ask for fact authentication: the system responds by returning rule chains for inferred facts grounded in fact sources with extracted facts highlighted in images of original documents.

Linguistically, our goal for fact extraction from historical documents is to enable automated reading. We simplistically define reading as being able to extract fact instances with respect to a declared ontology. As a matter of principled pragmatism, we not delve into the depths of reading cognition; however, we can learn from the observation that people typically do not read data-rich text snippets like in Figure 5 left-to-right/top-to-bottom, but rather skip around—implying that fitting facts into an ontological conceptualization likely requires best-fit heuristics for "reading" the maximal number of correct facts.

4 Concluding Remarks

We have argued in this position paper that the relationship among "Ontology as an Artifact, Ontology as a Philosophical Discipline, Conceptual Modeling, and Metamodeling" is synergistic, but should be tempered with principled pragmatism. For conceptual-modeling applications the goal is practical serviceability, not philosophical enrichment. Thus, conceptual-model researchers should draw ideas and seek guidance from these disciplines to enhance conceptual-modeling applications, but should not become distracted from computationally practical solutions by insisting that philosophical tenets should prevail.

To show how principled pragmatism works for building conceptual-modeling applications, we presented several case-study examples. Tempered by Einstein's sufficiency-with-simplicity adage, each case study illustrates how to directly leverage ideas from philosophical disciplines. Further, these case studies show that the requirements for applications to have computationally tractable solutions, itself, leads to principled pragmatism. Discipline principles can spark ideas, but decidability, tractability, and usability requirements demand practicality. Lastly, the case studies show that appropriate adaptation of ideas from philosophical disciplines have enhanced the serviceability of conceptual modeling and moreover indicate (as future work) that adaptation will likely further enhance the serviceability of conceptual modeling.

References

1. B. Smith. Ontology. In L. Floridi, editor, *Blackwell Guide to the Philosophy of Computing and Information*, pages 155–166. Oxford: Blackwell, 2003.
2. L. Xu and D.W. Embley. A composite approach to automating direct and indirect schema mappings. *Information Systems*, 31(8):697–732, December 2006.
3. D.W. Embley, D.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, 1999.
4. D.W. Embley, S.W. Liddle, and D.W. Lonsdale. Conceptual modeling foundations for a web of knowledge. In D.W. Embley and B. Thalheim, editors, *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*, chapter 15, pages 477–516. Springer, Heidelberg, Germany, 2011.
5. Aristotle. *Metaphysics.* Oxford University Press, New York, about 350BC. (1993 translation).
6. D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach.* Prentice Hall, Englewood Cliffs, New Jersey, 1992.
7. D.W. Embley, S.W. Liddle, and O. Pastor. *Conceptual-Model Programming: A Manifesto*, chapter 1, pages 3–16. Springer, Heidelberg, Germany, 2011.
8. M.A. Bunge. *Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems.* Reidel, Boston, Massachusetts, 1979.
9. Knowledge discovery and dissemination program. http://www.iarpa.gov/-solicitations_kdd.html/.
10. ACM-L-2010 workshop. http://www.cs.uta.fi/conferences/acm-l-2010/.
11. S.W. Liddle and D.W. Embley. A common core for active conceptual modeling for learning from surprises. In P.P. Chen and L.Y. Wong, editors, *Active Conceptual Modeling of Learning*, volume 4512 of *Lecture Notes in Computer Science*, pages 47–56. Springer-Verlag, Berlin, Germany, 2007.
12. S.W. Clyde, D.W. Embley, S.W. Liddle, and S.N. Woodfield. *OSM-Logic: A Fact-Oriented, Time-Dependent Formalization of Object-oriented Systems Modeling.* 2012. (submitted for publication, manuscript available at www.deg.byu.edu/papers/).
13. D.W. Embley, S.W. Liddle, D.W. Lonsdale, and Y. Tijerino. Multilingual ontologies for cross-language information extraction and semantic search. In *Proceedings of the 30th International Conference on Conceptual Modeling (ER 2011)*, Brussels, Belgium, October/November 2011. (in press).
14. S. Nirenburg, V. Raskin, and B. Onyshkevych. Apologiae ontologiae. In *Proceedings of the 1995 AAAI Spring Symposium: Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, pages 95–107, Menlo Park, California, 1995.
15. Plato. *Theaetetus.* BiblioBazaar, LLC, Charleston, South Carolina, about 360BC. (translated by Benjamin Jowett).