An Application of Philosophy in Software Modelling and Future Information Systems Development

Brian Henderson-Sellers[1], Cesar Gonzalez-Perez[2], and Greg Walkerden[3]

[1] Faculty of Engineering and Information Technology,
University of Technology, Sydney, Broadway, NSW, Australia
[2] Institute of Heritage Sciences (Incipit), Spanish National Research Council (CSIC)
Santiago de Compostela, Spain
[3] Dept Environment & Geography, Macquarie University, North Ryde, NSW, Australia
brian.henderson-sellers@uts.edu.au,
cesar.gonzalez-perez@incipit.csic.es, greg.walkerden@mq.edu.au

Abstract. The influence of mainstream philosophy on conceptual modelling and on modelling language development has historically been arcane or, at best, not recognized, whilst modellers might in fact implicitly espouse one particular philosophical tenet. This paper describes and discusses philosophical stances applied to conceptual modeling in order to make such influences explicit so that we, as conceptual modellers, can take the next step.

Keywords: concepts, modelling, philosophy, ontology engineering, conceptual modeling.

1 Introduction

The influence of mainstream philosophy on modelling language development, an important element in the information systems development context, has historically been minimalistic or, if even a little influential, has been a hidden source. In other words, whilst many modellers might in fact be behaving as if they held one particular philosophical tenet, or perhaps several inconsistent philosophical stances, they are often unaware of the philosophical assumptions that are thus implicit in the way they construct their modelling theories and hence unaware of consequences that may be disruptive or unhelpful in the way they model.

This paper delves back into philosophical writings to make a variety of such influences explicit in order that conceptual modellers and ontology engineers[1] can take the next step in formalizing their knowledge base. The aim of the paper is, therefore, not to espouse any particular philosophical thinking as 'correct' for conceptual modellers and ontology engineers, but to highlight the fact that every information systems researcher and developer has tacit assumptions they are often unable to articulate and to

[1] Although these belong to different communities with different histories, the relationship of these two traditions to the explication of philosophical assumptions is fundamentally similar, such that no differentiation is drawn between them in this paper.

X. Franch and P. Soffer (Eds.): CAiSE 2013 Workshops, LNBIP 148, pp. 329–340, 2013.

propose that, by taking an explicit and conscious stance, modellers will be able to make better decisions about modelling. For example, an in-depth understanding of stereotypes [1] helps a modeller to avoid the common mistake of equating them with supertypes [2].

Adopting one particular philosophical stance may entail a number of assumptions regarding the relationships between elements in a conceptual model and whether certain concepts are included or excluded. For instance, whether types exist in the real world; whether properties exist or, if not, whether they can be pragmatically useful for conceptual modelling. Opting for one stance might preclude the inclusion of an element that essentially belongs to, and is only valid in, a different philosophical view. Alternatively, one could take a more pragmatic philosophical stance, and embrace a more philosophically eclectic approach, looking for benefits for conceptual modelling not previously available - although exploration of such possibilities is a topic for future research.

With regard to concepts, for example, a Lockean [3] model asserts that we think with a mix of "simple ideas" and "complex ideas". For Locke, a "simple idea" is an idea that comes straight from experience - we have sensory experience of the world around us, and we have direct experience of an 'interior realm'. A "complex idea", by contrast, is formed by combining simple ideas, by finding relationships between them, and/or by abstracting away from them [Bk2, Ch12, S1]. All knowledge derives ultimately from experience [Bk2, Ch1, S2] -footnote1-. For Locke, the sounds of words are “marks for ideas within [one’s] own mind” [Bk3, Ch1, S1].

Each of Locke's key points has been a focus for extensive philosophical debate. For instance it has been argued: 1. There must be a knower for there to be knowledge, and the nature of the knower must influence what knowledge is, so all knowledge cannot be derived *wholly* from experience. 2. All ideas are more like "complex ideas" in Locke's terms: the simplest concepts embody rich experience, and function metaphorically. 3. It does not make sense to talk of ideas as if they existed independently of words or other symbols.

The debates about these points provide alternative points of departure for conceptual modellers. Contemporary interpretations of concepts, e.g. as discussed by [4], provide still more. We advocate sensitivity to these opportunities: a sensitivity to philosophical assumptions can bring clarity to ambiguities and contradictions in methods of software modelling and information systems development, and lead to innovations.

Although we originally sought to identify the ‘best’ philosophical underpinnings for software development and conceptual modelling, a study of the philosophical literature reveals not only differences between philosophy of the mind, philosophy of language use and philosophy of psychology – all relevant to our study – but also that no one of these philosophies can be considered ‘correct’. In other words, we can only describe and discuss these philosophical stances in the context of conceptual modelling; we can only recommend that modellers and modeling language designers should consider either (1) taking a systematic philosophical approach adopting a consistent set of assumptions, consistent from one particular philosophical viewpoint; or (2) embracing a more pragmatic approach, combining philosophical insights that are helpful for the task at hand and focusing on becoming aware of previously unrecognized constraints and possible missed opportunities.

In Section 2, we discuss some of the (historical) involvement of philosophical thinking with conceptual modelling including how varying philosophical assumptions may have unnoticed repercussions. In Section 3, we discuss some of the ideas found in the ontology engineering literature, noting the minimal overlap with philosophy whilst Section 4 presents our conclusions.

2 Philosophical Impacts on Modelling

As noted above, the impact of choosing one particular philosophical stance may exclude (or demand) certain concepts, ideas and representations. For instance, [5, Ch 5] argues for a 'logical paradigmatic' philosophical viewpoint, in which it is forbidden (incorrect) to include representations of properties within any model thus constructed. That this is counter-intuitive to accepted conceptual modelling (e.g. using UML) provides a tension between contemporary philosophy and conceptual modelling that requires future resolution (see Section 2.2).

Lakoff [6] argues from a cognitive linguistic viewpoint rather than a strictly philosophical basis that human reason is embodied symbol manipulation, noting that, through the classical theory of categories, symbols acquire meaning by virtue of their correspondence to categories in the real world. Since symbols are important for conceptual modelling languages, this notion of how symbols relate to, refer to or represent 'entities in the real world' is one of the crucial elements of the philosophical impacts on conceptual modelling.

2.1 Concepts

One area of major contention is in the notion of 'concept' itself (and 'conceptualization' and 'conceptual' – as in 'conceptual modelling'). The Stanford Encyclopedia of Philosophy [7] identifies, *inter alia*[2], three main options for discussing concepts: concepts as mental expressions, dating back to Locke [9] and Hume [10]; concepts as abilities [11]; and concepts as Fregean senses e.g. [12]. It states:

> *"Concepts are the constituents of thoughts. Consequently, they are crucial to such psychological processes as categorization, inference, memory, learning, and decision-making. This much is relatively uncontroversial. But the nature of concepts—the kind of things concepts are—and the constraints that govern a theory of concepts have been the subject of much debate."*

Smith [4], whilst decrying the over- or mis-use of 'concept', states: "In many contexts, of course, ontologists still deal with concepts, correctly, as analogous to, though more abstract than, the linguistic expressions with which they are associated." His concern centres as much around the misapplication of the notion of concept rather than the notion itself, observing that many researchers deal only with concepts rather than the entities in reality to which they correspond.

---

[2] Other possibilities are discussed in [8].

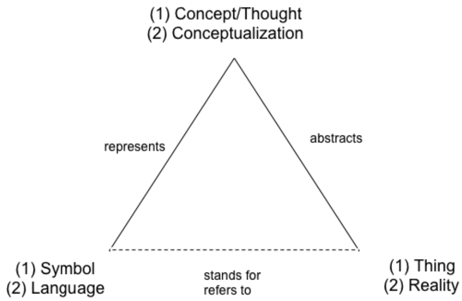

Fig. 1. Ogden and Richards' 'meaning triangle' (a.k.a. Ullmann's triangle) for (1) an individual Thing and (2) reality (or a specific reality domain such as banking or telecomms)

Further debate has emphasized two alternatives:

1. The concept strategy, as discussed above. If we wish to relate this to Fig. 1, we might argue that the three vertices of the triangle are utilized in such a way that 'concept' (the topmost vertex) effectively dominates and relegates the notion of sign or symbol to be a 'second class citizen'.
2. The sign strategy, which, whilst acknowledging mental constructs, notes that these are individualistic and that we should be most concerned about communication – via signs and symbols. In terms of Fig. 1, only the bottom part of the figure would be relevant. To date the former has been more popular in ontology engineering and the latter in UML-style software modelling.

Option 2 is favoured by Smith [4], who argues that the misuse of 'concept' has resulted from its use in the knowledge representation community and in linguistics where concepts and entities tend to become confounded. In ontological information systems, a prime reference is that of Gruber [13] who famously defined ontology as "a specification of a conceptualization", thus freezing in the notions of concept and conceptualization into the very fabric of ontology research – incorrectly according to much philosophical thinking. It should be stressed that the semantics of 'ontology' in these three cases (Smith, Bunge and Gruber) are by no means identical – another example of the challenge to conceptual modellers of employing a consistent philosophical and ontological framework.

In ontology engineering, it is too often the case, he argues [13], that the ontology being created is of social constructs, which have some verity in being labelled 'concepts', being largely human-created and intangible (see also [14]); whereas in the broader picture, it is vital that elements in the ontology have a direct correspondence to entities in the real world [15]. Option 2 also has the advantage of parsimony – a good trait in science[3]. Smith [4] thus strongly advocates replacing the emphasis on concepts and conceptualization by universals and particulars such that particulars refer to instances that exist in reality and universals signify what the corresponding instances have in common. In object-oriented conceptual modelling, the typical

[3] Usually referred to as Ockham's razor, named after William of Ockham, c 1287-1347.

approach is to relate particulars to objects – although in fact they could equally refer to classes of objects – and universals to classes (Fig. 2). This, again, offers support, this time from philosophy for the shift of emphasis in metamodelling, as suggested in [16] to a framework focussing on language use as determining the modelling architecture rather than a strict metamodelling hierarchy as advocated by the OMG (see also discussion in Section 3.2).

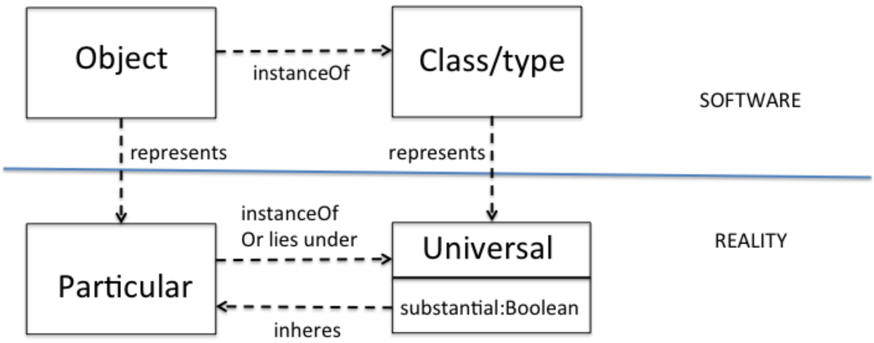

Fig. 2. A depiction of elements in the software domain and how they represent elements of reality

This provides a way to differentiate good ontologies from bad ontologies, differentiating them on the basis of whether the general terms in them do or do not relate to corresponding universals (and hence instances) in reality. Smith [4] also argues for the universal-instance model to be taken as axiomatic for ontologies and then investigates the meaning of subtyping (necessarily between universals) and mereology (linking particulars).

2.2 Properties

As well as concepts, an additional philosophical issue is that of the nature, utility and existence of properties[4], common in modelling languages like UML (see, e.g., discussion in [17]), in ontologies (e.g. the property of Bunge [18] and Wand [19]; see also [20]) and in cognitive linguistics (e.g. [4]). In philosophy, it is argued [21] that properties are crucial to analytic philosophy, although other philosophers disagree [22], arguing that properties are eschewed in contemporary philosophy.

Properties were originally identified with ‘substance’ (e.g. [3, chapter 4]), an idea that was strongly attacked by both Locke [9] and Hume [10] yet strongly supported in contemporary foundational ontologies, such as the UFO [24] and the four-category ontology of Lowe [25].

For some philosophers, this resulted in a shift from a substance paradigm to an extension paradigm e.g. [26], along with which came a shift (in philosophical thinking) from properties to logical classes and tuples e.g. [5, page 94]. However, this

[4] Properties are sometimes called qualities or characteristics e.g. [23, p105]. In the software engineering context, properties are often called attributes.

necessary realignment of properties as extensions brings its own challenges and contradictions. As an example, consider the red property of an object that is turned into a member of a class of RedThings and compare this with the representation of a 'redness' quality as a trope [27]. Consequently, we do not advocate the adoption of such philosophical traits coming from the extension paradigm into conceptual modelling; rather, we retain the notion of property since this is all-pervasive in the software and conceptual modelling and metamodelling literature to date. This is consistent with the use of the Bunge ontology by Wand and Weber [28-30]: the BWW model applied to information systems development.

3 Some Contributions of Ontological Engineering and Language Use

3.1 Ontological Engineering – A Philosophical Perspective

There is a long history (see, e.g. [31, 32]), identified in the ontological engineering literature (see, e.g., [33]), dating back to Aristotle and summarized more recently by Ogden and Richards [34], that suggests that there is a mediating role of the human mind in relating symbols to things (in reality). This leads directly to a representation such as that in Fig. 1 (with label '1'). Based on [35] but known as the Ullmann triangle [36], it illustrates the well-known 'meaning triangle' linking an object or thing in the real world to a concept or thought and then to a symbol (thus bearing some similarity with Locke's approach outlined above); in other words, that the linking of model to reality is not direct. For a given domain (part or all of reality), the set of all the individual concepts abstracted from that domain is called the *conceptualization* (Fig. 1 (label '2')) i.e. the combination of the concepts and their relationships [33]. This is the *mental model* e.g. [37, 38] – called here the *cognitive model*– see also [39] and could be said to imply and embody the ontological commitment (e.g. [40]). However, in this work, the conceptualization is considered as an *individual* mental commitment that must be shared (at some symbolic level) in order to be useful to the community; whereas in other more commonly accepted philosophies, individual mental commitments, whatever their appellation, are deemed unhelpful unless they are confirmed and agreed upon by the community who wish to share and build on these ideas i.e. conceptualization relates to the community accepted description not to the individual's.

However, whilst often used in ontological discussions (e.g. [33, 41]), we must point out that this work (of Peirce, Ogden, Nash and Ullman) is not appreciated in any of the philosophical literature and emerges more from formalising everyday usage than disciplined philosophical reflection. Nevertheless, from our pragmatic point of view, the ideas, which conflate types and instance, are nevertheless worth further investigation because the simple model of Fig. 1 may provide a starting point for a detailed mathematical description of the modelling and metamodelling domains [15].

Fig. 3 suggests a much more detailed description to tie together cognition, reality and models. The first triangle deals with the physical world, which is made of things.

Each thing in the world (such as the chair I am sitting on as I type this) may be perceived by us through an individual concept and I can utter words or use any other symbols (graphical, aural or otherwise) to refer to it. For example, the word 'chair' in the utterance "this chair is getting older" pronounced by me while I keep the mental picture of my chair active is a symbol usage. Moving to the conceptual world, things can be arranged into categories or thing types; for example, all chairs conform to (or 'fall under') a *chair* thing type, and we often use that category for abstraction purposes. Our mental picture of that category is what we call the corresponding type concept; different people may have different mental pictures of the same category (or thing type), although most of us would probably share a common thing type for *chair*. In addition, categories are also things and thus can be seen as individual concepts and described by symbol usages. The third component of the conceptual world is given by the symbols that we may use to depict the categories in it through the corresponding type concepts; for example, the word 'chair' in English, which does not represent any particular chair but the very concept of chair, is a symbol. Moving further on to the modelling world, we can conceive a model as an aggregate of symbol usages; for example, an uttered sentence (which is a model) is an aggregate of particular uttered words, and a UML model is an aggregate of particular usages of UML symbols. A language, similarly, is an aggregate of symbols and usage rules (e.g. grammar, semantics and pragmatics); English focusses on morphemes whereas UML is a collection of metamodel elements. And, finally, a conceptualization is an aggregate of concepts, i.e. a particular worldview on the physical world.

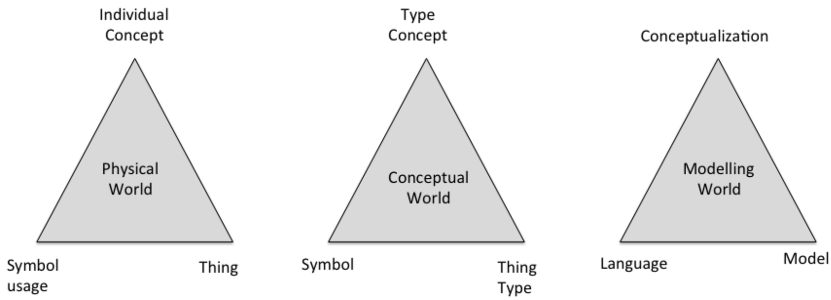

Fig. 3. Taking Fig. 1 as a base, the three triangles here explicitly differentiate the elements in the physical, conceptual and modelling worlds

The three-triangle approach to the description of meaning (Fig. 3) is, in our view, superior to the single triangle of Ogden and Richards, since it clearly differentiates between types and instances. In Fig. 1, instances and their types are conflated into a single element, which is confusing and avoids formal reasoning. For example: the English word 'chair', as found in a dictionary, and the uttered word 'chair' as part of a sentence that refers to a particular chair are definitely not the same kind of thing; the former is a symbol and the latter is a symbol usage; the former is part of a language and the latter is part of a model. In Fig. 1, this distinction goes unnoticed.

3.2 Language Use

The study of meaning in a natural language – so-called 'language use' – has a modern history of several decades (e.g. [23, 42]) and itself embodies a philosophical underpinning dating back to at least Frege [43]. The application of these ideas to conceptual modelling and information systems development is discussed in [16], where it is argued that the current multilevel architectures, such as that of the Object Management Group (OMG) in which four levels are linked by instanceOf relationships, should be replaced by an architecture based on language use in which elements in any model are not instances of classes in a metamodel but are in fact instances and/or types that conform to a modelling language definition (Fig. 4). In other words, the idea of a metamodelling stack in which the metamodel is the dominant feature (as in today's UML: [44]) is replaced by one in which the modelling language itself is the predominant feature [16]. Of course, this language may be defined by a metamodel but that definitional link is only one of several options.

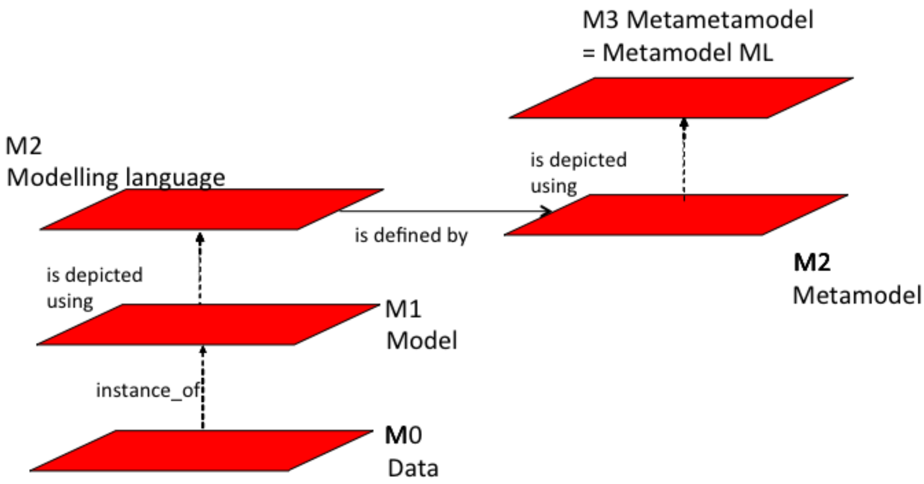

Fig. 4. Revision of the OMG multi-level architecture place the modelling language and not the metamodel at the 'M2' level in the multilevel stack (now restricted to three levels with a linkage to a second multilevel stack – here shown as having only two levels) (after [16])

Furthermore, this architecture fully supports models in the real world. This gives a correspondence relationship between particulars (instances) in the language use component of the model to real world things (Figs. 3 and 5). It is important to note that (i) this correspondence is generally ignored in conceptual modelling approaches in use today, especially those based on UML, and (ii) it introduces the need to recognize that there may be a need for a better framework. For example, in the former case, it is well-known that modellers have difficulty in identifying whether actors in a use case diagram are internal or external to the software system e.g. [45]. In the latter case, it is possible that the clumsiness of such models could be ameliorated by the use of such a theoretical framework that already exists, but outside of conceptual modelling/software engineering – for example, a philosophical underpinning between an entity (e.g. a particular or a universal) and a referent (e.g. [23, 43].

These ideas, of cognitive models (in an individual's brain) leading to a (shared) concept is well accepted in the modelling community and, indeed, in cognitive linguistics (e.g. [6, pp. 8 & 178]) and in language use/speech act theory, although Searle [23] notes that Frege's original use of the word concept confused two different definitions and usages resulting from "Frege moving in two philosophical directions which are at bottom inconsistent" [23, p98]: (a) insisting that predicates have a referent and (b) accounting for a functional distinction between referring expressions and predicate expressions. Searle argues for a solution in which the notion that predicate expressions refer is abandoned i.e. they ascribe properties they do not refer to them. Here, Searle seems to equate property with concept. Finally, we note that in the modelling area, there is a whole branch known as 'conceptual modelling' (e.g. [46, 47]), with its own eponymous conference. However, references to the Ogden and Nash/Ullmann triangle discussed above are all but absent in the philosophical research literature.

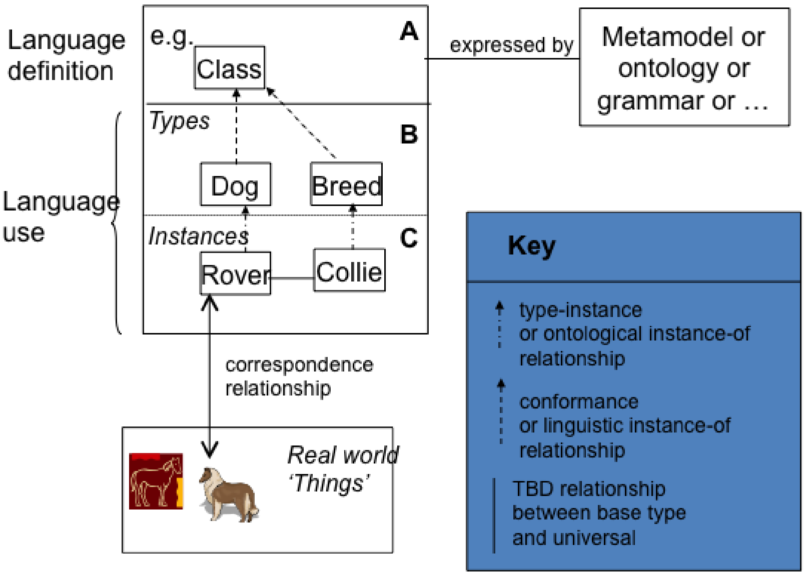

Fig. 5. Complete multilevel framework based on language use – to replace the strict metamodelling architecture of the Object Management Group when modelling in information systems development and software engineering (after [15])

3.3 Ontic vs. Epistemic Models

Conceptual models are supposed to represent things, but it is rarely clear whether the things represented by a conceptual model are the things-as-they-are or the things-as-we-know-them. A model of the things-as-they-are, also called an ontic model, aims to represent reality while being independent of the observer. A model of the things-as-we-know-them, on the other hand, aims to capture our knowledge about the reality rather than the aseptic reality itself. All but the most radical solipsists would agree that different degrees of subjectivity can be captured in a model; however, conventional conceptual modelling languages such as UML lack the necessary mechanisms to capture these nuances, and models created with them, in consequence, are oblivious to them. For example, we would certainly agree that "every person has an age", so that the property Age of class Person should not be nullable in a UML class model. However, a modeller might choose to make it nullable if there are chances that Person objects will be created without a known age. Note, however, that the first statement ("every person has an age, so that the property Age of class Person should not be nullable") is ontic in nature, whereas the second ("a modeller might choose to make it nullable if there are chances that Person objects will be created without a known age") is epistemic. The ConML modelling language [48, 49], for example, has separate *null and unknown semantics* in order to describe information that does not exist ("null", which is ontic) and information that does exist but which we ignore ("unknown", which is epistemic). Without this explicit difference, a model is confusing, since there is no way to know whether it describes an objectivized reality or a particular epistemic approach to it.

4 Summary and Conclusions

The role of conceptual modelling in information systems development has continued to increase over the last several decades. However, there has been little explicit awareness of any influences of mainstream philosophical thinking. This neglect could not only impact the cohesiveness of conceptual modelling paradigms (such as ER, OO) but also obscure potential advantages that may not be obvious since the links between a cohesive philosophical framework and the conceptual modelling approach adopted are implicit and unexamined. Identification of such links could bring benefit to the conceptual modelling community and consequently help to increase the quality of future information systems development.

In this paper, we have tried to bring to the attention of the information systems modelling community some of those arcane philosophical underpinnings. We have noted that there is no 'holy grail' i.e. there is no *one* philosophical suite of ideas that we can insist *should* be adopted. Rather, we take a more pragmatic (but still formal) approach in recommending consideration of whether or not any particular philosophy is both useful and implementable i.e. whether it is indeed understandable and usable to the benefit of the information systems community.

We have noted, in particular, philosophy-based concerns about commonly used modelling terms such as 'concepts' and 'properties/attributes', whilst noting also the value of incorporating ontological thinking and theories of language use.

Our final conclusion is more along the lines of 'caveat emptor' – modellers need to be aware of the philosophical history, albeit indecisive, that may more explicitly be incorporated into current and emerging approaches to modern conceptual modelling and information systems development.

References

1. Henderson-Sellers, B., Gonzalez-Perez, C.: Uses and abuses of the stereotype mechanism in UML 1.x and 2.0. In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (eds.) MoDELS 2006. LNCS, vol. 4199, pp. 16–26. Springer, Heidelberg (2006)
2. Atkinson, C., Kühne, T., Henderson-Sellers, B.: Systematic stereotype usage. Software and System Modelling 2(3), 153–163 (2003)

3. Locke, J.: An essay concerning human understanding, Digireads, Kansas (1690/2009)
4. Smith, B.: Beyond concepts: ontology as reality representation. In: Varzi, A., Vieu, L. (eds.) Procs. FOIS 2004, pp. 73–84. IOS Press (2004)
5. Partridge, C.: Business Objects: Re-Engineering for Re-use. Butterworth Heineman, Oxford (1996)
6. Lakoff, G.: Women, Fire, and Dangerous Things. What Categories Reveal about the Mind. Univ. Chicago Press (1987)
7. Stanford Encyclopedia of Philosophy (last major revision) (2011), `http://plato.stanford.edu/entries/concepts/` (accessed September 3, 2012)
8. Margolis, E., Laurence, S. (eds.): Concepts: Core Readings. MIT Press (1999)
9. Locke, J.: An Essay Concerning Human Understanding. Oxford Univ. Press, Oxford (1690, 1975)
10. Hume, D.: A Treatise of Human Nature. Oxford University Press, Oxford (1739, 1978)
11. Wittgenstein, L.: Philosophical Investigations. Macmillan, New York (1953)
12. Peacocke, C.: A Study of Concepts. MIT Press, Cambridge (1992)
13. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition 5, 199–220 (1993)
14. Eriksson, O., Ågerfalk, P.J.: Rethinking the meaning of identifiers in information infrastructures. J. Assoc. Info. Systems 11(8), 433–454 (2010)
15. Henderson-Sellers, B.: On the Mathematics of Modelling, Metamodelling, Ontologies and Modelling Languages, Springer Briefs in Computer Science. Springer (2012)
16. Henderson-Sellers, B., Eriksson, O., Gonzalez-Perez, C., Ågerfalk, P.J.: Ptolemaic metamodelling? The need for a paradigm shift. In: Cueva Lovelle, J.M., et al. (eds.) Progressions and Innovations in Model-Driven Software Engineering. IGI Global (2013)
17. Shan, L., Zhu, H.: Unifying the semantics of models and meta-models in the multi-layered UML meta-modelling hierarchy. Int. J. Softw. Informat. 6(2), 163–200 (2012)
18. Bunge, M.: Treatise on Basic Philosophy: Ontology I: The Furniture of the World, vol. 3. Reidel, Boston (1977)
19. Wand, Y.: Ontology as a foundation for meta-modelling and method engineering. Inf. Software Technol. 38, 281–287 (1996)
20. Opdahl, A., Henderson-Sellers, B.: A template for defining enterprise modelling constructs. Journal of Database Management 15(2), 39–73 (2004)
21. Mellor, D.H., Oliver, A. (eds.): Properties. Oxford Univ. Press (1997)
22. Partridge, C.: Personal communication (2012)
23. Searle, J.R.: Speech Acts. An Essay in the Philosophy of Language. Cambridge Univ. Press (1969)
24. Guizzardi, G., Wagner, G.: Towards Ontological Foundations for Agent Modelling Concepts Using the Unified Fundational Ontology (UFO). In: Bresciani, P., Giorgini, P., Henderson-Sellers, B., Low, G., Winikoff, M. (eds.) AOIS 2004. LNCS (LNAI), vol. 3508, pp. 110–124. Springer, Heidelberg (2005)
25. Lowe, E.J.: The Four-Category Ontology: A Metaphysical Foundation for Natural Science. Oxford Univ. Press (2006)
26. Descartes, R.: Principia philosophiae (Principles of Philosophy) (1644)
27. Guizzardi, G., Masolo, C., Borgo, S.: In defense of a trope-based ontology for conceptual modeling: An example with the foundations of attributes, weak entities and datatypes. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 112–125. Springer, Heidelberg (2006)
28. Wand, Y., Weber, R.: An ontological analysis of some fundamental information systems concepts. In: Proceedings of the Ninth International Conference on Information Systems, Minneapolis (1988)

29. Wand, Y., Weber, R.: An ontological model of an information system. IEEE Trans. Software Eng. 16(11), 1282–1292 (1990)
30. Wand, Y., Weber, R.: On the deep structure of information systems. Information Systems Journal 5, 203–223 (1995)
31. Baldinger, K.: Semantic Theory: Towards a Modern Semantics. Palgrave Macmillan (1980)
32. Sowa, J.F.: Ontology, metadata, and semiotics, Conceptual Structures: Logical, Linguistic, and Computational Issues. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS, vol. 1867, pp. 55–81. Springer, Heidelberg (2000)
33. Guizzardi, G.: Ontological foundations for structural conceptual models. CTIT PhD Thesis Series, No. 05-74, Enschede, The Netherlands (2005)
34. Ogden, C.K., Richards, I.A.: The Meaning of Meaning. Harcourt, Brace and World, New York (1923)
35. Peirce, C.S.: Collected Papers of C. S. Peirce. In: Hartshorne, C., Weiss, P., Burks, A. (eds.) 8 vols. Harvard University Press, Cambridge (1958)
36. Ullmann, S.: Semantics: An Introduction to the Science of Meaning. Basil Blackwell, Oxford (1964)
37. Jackson, M.: Some Notes on Models and Modelling. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Mylopoulos Festschrift. LNCS, vol. 5600, pp. 68–81. Springer, Heidelberg (2009)
38. Wilmont, I., Brinkkemper, S., van de Weerd, I., Hoppenbrouwers, S.: Exploring intuitive modelling behaviour. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R., et al. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 301–313. Springer, Heidelberg (2010)
39. Gonzalez-Perez, C., Henderson-Sellers, B.: Modelling software development methodologies: a conceptual foundation. J. Sys. Softw. 80(11), 1778–1796 (2007)
40. Lorenzatti, A., Abel, M., Fiorini, S.R., Bernardes, A.K., dos Santos Scherer, C.M.: Ontological primitives for visual knowledge. In: da Rocha Costa, A.C., Vicari, R.M., Tonidandel, F. (eds.) SBIA 2010. LNCS, vol. 6404, pp. 1–10. Springer, Heidelberg (2010)
41. Lorenzatti, A., Santin, C.E., Paesi da Silva, O., Abel, M.: A representation frame-work for visual knowledge. In: Vieira, R., et al. (eds.) Proceedings of Joint IV Seminar on Ontology Research in Brazil and VI International Workshop on Metamodels, Ontologies and Semantic Technologies, CEUR Workshop Procs., vol. 776, pp. 177–182 (2011)
42. Austin, J.L.: How To Do Things With Words. Oxford Univ. Press (1962)
43. Frege, G.: Über Begriff und Gegenstand (On Concept and Object). In: Vierteljahrsschrift für wissenschaftliche Philosophie, vol. XVI, pp. 192–205. Fues's Verlag (1892) (transl. Geach and Black, 1952)
44. OMG: OMG Unified Modeling Language™ (OMG UML), Infrastructure Version 2.4.1, OMG Document Number: formal/2011-08-05, 230pp (2011)
45. Lilly, L.: Use case pitfalls: top 10 problems from real projects using use cases. In: Firesmith, D., et al. (eds.) Procs. TOOLS 30, pp. 174–183. IEEE Comp. Soc. Press (1999)
46. Mylopoulos, J.: Conceptual modeling and Telos. In: Loucopoulos, P., Zicari, R. (eds.) Conceptual Modeling, Databases and CASE, ch. 4, pp. 49–68. Wiley (1992)
47. Kaschek, R., Delcambre, L. (eds.): The Evolution of Conceptual Modeling. From a Historical Perspective towards the Future of Conceptual Modeling. LNCS, vol. 6520. Springer, Heidelberg (2011)
48. Incipit: ConML Technical Specification. ConML 1.3 (2013), http://www.conml.org/Resources_TechSpec.aspx
49. Gonzalez-Perez, C.: A conceptual modelling language for the humanities and social sciences. In: Rolland, C., et al. (eds.) Sixth International Conference on Research Challenges in Information Science (RCIS 2012), pp. 396–401. IEEE Computer Society (2012)