The Role of Ontology in Semantic Integration

Chris Partridge[1 & 2]

[1] BORO Program, England
partridge@BOROProgram.org
http://www.BOROProgram.org/
[2] National Research Council, LADSEB-CNR, Italy
partridge@ladseb.pd.cnr.it
http://www.ladseb.pd.cnr.it/infor/ontology/BusinessObjectsOntology.html

Abstract. More and more enterprises are currently undertaking projects to integrate their applications. They are finding that one of the more difficult tasks facing them is determining how the data from one application matches semantically with the other applications. Currently there are few methodologies for undertaking this task – most commercial projects just rely on experience and intuition. Taking semantically heterogeneous databases as the prototypical situation, this paper describes how ontology (in the traditional metaphysical sense) can contribute to delivering a more efficient and effective process of matching by providing a framework for the analysis, and so the basis for a methodology. It delivers not only a better process for matching, but the process also gives a better result. This paper describes a couple of examples of this: how the analysis encourages a kind of generalisation that reduces complexity. Finally, it suggests that the benefits are not just restricted to individual integration projects: that the process produces models which can be used as to construct a universal reference ontology – for general use in a variety of types of projects.

1 Introduction

Integration projects come in a variety of forms. However underlying this variety of forms is a common semantic task – what can be called the 'matching of semantically heterogeneous data' or more simply 'semantic matching'. There is a reasonably clear recognition that the analysis stage of this task needs to focus on identifying the entities that the data describes – the 'real-world semantics'[1]. In theory and practice, this identification currently relies mostly on experience and intuition.

Metaphysicians suggest that experience and intuition rely upon, among other things, implicit ontological assumptions. And that we can build a much better picture if we try and understand what these assumptions are.

[1] "All these schema integration techniques require either explicitly or implicitly that (the relationship) between the real-world semantics of the classes to be integrated is known. This is a reasonable assumption in tightly-coupled approaches, but … in a federation of databases from multiple modelling contexts this may be surprisingly difficult." Vermeer, M. W. W. and Apers, P. M. G. *On the Applicability of Schema Integration Techniques to Database Interoperation*. ER 1996: 179-194.

1.1 This paper

This paper[2] suggests that in the particular case of matching of semantically heterogeneous data, the (metaphysicians') ontology is helpful: not only in clarifying applications' ontological assumptions - but also in providing a common framework across applications for analysing what entities their data describes. This acts as both an explanatory framework and a foundation for a methodology for the analysis.

The paper makes its case in five sections which:

- Clarify the context for semantic matching task that ontology is intended to help.
- Clarify the meaning of the terms used to describe the proposed approach.
- Use these terms to characterise the ontological process.
- Describes a key way in which this process can lead to better results: how the analysis encourages a kind of generalisation that reduces complexity.
- Finally, notes an important potential benefit: enabling construction of a universal reference ontology.

2 Context

This section starts by explaining why database integration has been chosen to illustrate the semantic matching task and then outlines the current perception of the main focus of this task: semantic heterogeneity. It outlines where it comes from, what it is and the basis for resolving it.

2.1 The focus on database integration

Our interest here is not in the variety of forms application integration comes in, but on how ontology helps in this common task. So it makes sense to focus on a single form that will most clearly illustrate this. Database integration is a reasonably common and straightforward form of integration, one in which the task clearly manifests itself – so this has been selected as the prototypical situation (the other forms can then be regarded as variations of this).

Simplifying slightly, the database integration design process can be regarded as[3]:

- taking the multiple databases – schemas and data as input, and
- producing as output a single unified database – schemas and data – and a mapping from the individual databases to the unified database.

The task of 'matching of semantically heterogeneous data' takes place during the analysis done in the initial stages. This determines how the elements of the multiple databases are matched with each other.

2.2 Prime source of semantic heterogeneity

What usually makes the task of matching onerous is a high level of heterogeneity between the databases. A prime reason of this heterogeneity is clearly recognised, it is what Sheth and Larson call

---

[2] This paper has been adapted from a longer technical report - LADSEB-CNR - Technical report 05/02 - *The Role of Ontology in Integrating Semantically Heterogeneous Databases*. 2002, which can be found at http://www.boroprogram.org/trap.htm.

[3] Basically the same point is made in the Introduction of Parent, C. and S. Spaccapietra (2000). *Database Integration: The Key to Data Interoperability*. (Chapter 10 of *Advances in Object Oriented Data Modeling*. M. Papazoglou and S. Spaccapietra. Cambridge, Mass., MIT Press.)

on p. 187 of [1] *design autonomy*. This is "the ability of a component DBS to choose its own design with respect to any matter". As they note, this includes "The conceptualization or semantic interpretation of the data (which greatly contributes to the problem of semantic heterogeneity)". In fact, they say: "Heterogeneity [in general] … is primarily caused by design autonomy among component DBSs."

Of course, autonomy by itself does not lead to heterogeneity. There is in principle no reason why two autonomous designers should not end up with the same design. However, in practice, autonomy allows a surprising amount of what I shall call *design diversity* to manifest itself.

2.3 Nature of semantic heterogeneity

It is important to clarify what semantic heterogeneity is. Sheth and Larson on p. 187 of [1] suggest that heterogeneity occurs "… when there is a disagreement about the meaning, interpretation or intended use of the same or related data [in different databases]." But they noted that "… this problem is poorly understood, and there is not even an agreement regarding a clear definition of the problem."

Sheth and Larson on p. 187 of [1] offer as an example two attributes with the same name - MEAL-COST. In the first case, this describes the average cost of a meal per person in a restaurant without service charge and tax. In the second case, it describes the average cost of a meal per person including service charge and tax. Despite their surface similarities these are semantically heterogeneous – where "the heterogeneity is due to differences in the definition (i.e., in the meaning)". The two attributes are closely related. This kind of close miss is characteristic of semantic heterogeneity. Ontology's role is to help unbundle the objects and make clear the relation between them.

2.4 Basis for agreement

The last example also provides an illustration of what usually makes integration hard work. Databases typically do not, by themselves, give us enough information. The two MEAL-COST attributes, in the example, do not tell us enough by themselves to determine whether they describe the same or different things. It is generally accepted that databases do not by themselves contain enough information for semantic matching[4]. The process of getting hold of the information needed to determine how the forms are related often involves securing agreement from the parties involved. The simplest way to get this information is to discuss what their databases describe with the applications' communities. As [2] note in their Introduction "…existing database schemas provide basic knowledge about the semantics of data, which may be easily enhanced … through interviews of current users and data administrators."

[4] See, for example, (Sheth and Larson 1990)'s comment that: "Typically, DBMS schemas do not provide enough semantics to interpret data consistently." As they also note: "Heterogeneity due to differences in data models also contributes to the difficulty in identification and resolution of semantic heterogeneity. It is also difficult to decouple the heterogeneity due to differences in DBMSs from those resulting from semantic heterogeneity."

Real world semantics. What one needs to know (secure agreement about) for integration, and many authors note this, is what the database representations mean, what entities they describe – their real world semantics. As [3] notes "…schema integration techniques require either explicitly or implicitly that (the relationship) between the real-world semantics of the classes to be integrated is known."[5]

Once we know what the two instances of MEAL-COST's real-world semantics are – in other words, what the database representations describe, we are in a good position to start integrating them. And it is important to keep reminding ourselves that what is being identified here is not in the database but the entities that the database is describing.

An ontological framework for analysis. What ontology provides is a framework within which the "interviews with users and data administrators" can be focused on the entities the database describes and the information this provides analysed and organised. This framework helps both to explain what is happening when one analyses the 'real-world semantics' and to suggest a systematic process for undertaking it. This paper attempts to illustrate both of these.

3 Clarifying the terms

Before we look at the details of ontological analysis, we need to clarify the use here of some basic terms: firstly, ontology and semantics.

3.1 Ontology

For the purposes of database integration, the traditional philosophical (metaphysical) notion of ontology is useful – where this is "the set of things whose existence is acknowledged by a particular theory or system of thought." [6]

This view was famously summarised by Quine, who claimed that the question ontology asks can be stated in three words 'What is there?' – and the answer in one 'everything'. Not only that, but tongue in cheek, he also said "everyone will accept this answer as true" though he admitted that there was some more work to be done as "there remains room for disagreement over cases."[7]

From the perspective of database integration, each database can be regarded as a 'theory' that acknowledges the existence of a set of objects – its ontology. Some care needs to be taken to distinguish this traditional metaphysical use of the word ontology from one that has recently developed in Computer Science. Here an ontology is regarded as a "specification of a conceptualisation" [4] and has been applied to a wide range of things, including dictionaries. This sense of the word does not give us a fine-grained enough tool for our needs: it regards a database as simply an ontology – and so it cannot make sense of talking about the ontology underlying it, let alone underlying a set of databases.

---

[5] As it also notes: "One of the central problems … is that the definition of relationships between local and imported data is far from trivial in a situation where information on the meaning of a remote schema is limited. … [I]n a federation of databases from multiple modelling contexts this may be surprisingly difficult."

[6] E. J. Lowe in the Oxford Companion to Philosophy.

[7] In W.V. Quine's *On what there is* (1948), Review of Metaphysics, Vol. II, No. 5, reprinted in *From a logical point of view* (1961).

3.2 Semantics

Along with the traditional philosophical sense of ontology there is a related notion of semantics – where this is the relationship between words (data) and the world – the things the words (data) describe[8]. This needs to be distinguished from the different, but related, sense of the word in linguistics where it means the study of meaning[9].

These notions of ontology and semantics can then be used to describe two other useful notions – that of an ontological model and semantic divergence.

3.3 Ontological model

An ontological model is a model that directly reflects the ontology. There is a simple semantics where each object in the ontology has a direct relationship with the corresponding representation in the model[10].

One of the characteristics of an ontological model is that the representations in it can be regarded as the names of the objects in the ontology – from a Fregean perspective as reference and no sense (from a Millian perspective as denotation without connotation). In [5], Ruth Barcan Marcus (explicitly following in the footsteps of Mill and Russell) calls this 'tagging'.

3.4 Semantic divergence

Semantic divergence occurs where an item in the representation does not map directly onto an object in the ontology. Semantic heterogeneity occurs when apparently similar items in two different representations have different semantics. The notion of semantic divergence and semantic heterogeneity overlap – but do not coincide. By itself, semantic divergence does not necessarily lead to semantic heterogeneity. If two databases that need to be integrated have identical semantic divergences, then they are not semantically heterogeneous, they work semantically in the same way. In practice, much of the semantic heterogeneity in databases has its sources in differing semantic divergences and most database integration projects have to deal with significant semantic divergence

Different matching strategies. The distinction between semantic heterogeneity and diversity can be used to characterise the way in which the ontological matching strategy proposed here differs from that typically adopted. Currently many integration projects view the semantic matching process as a mechanism for dealing with semantic heterogeneity – focusing on resolving the semantic differences between the databases. And they analyse these differences using 'real world semantics'. The unified database is then a combination of the homogenous and resolved heterogeneous data, both of which may or may not be semantically divergent). I call this the *heterogeneity strategy*.

The *ontological strategy* focuses on purging the semantic divergence from each of the databases. And in so doing, mapping the underlying ontology. This ontology then provides a basis for designing the "single unified database" that is the output of the integration. Another important benefit is that (unlike the heterogeneity strategy's unified database), this ontology's form is not tied to any of the

---

[8] Or as Nelson Goodman put it in the Introduction to Quine's lectures published as *Roots of Reference* – "… an important relation of words to objects – or better – of words to other objects, some of which are not words – or even better, of objects some of which are words to objects some of which are not words."

[9] "Semantics – the study of meaning" from the Concise Oxford Dictionary of Linguistics, © Oxford University Press 1997.

[10] This is called *strong reference* in Partridge's *Business Objects: Re - Engineering for re - use* (1996).

individual databases, and so should be equally applicable to other applications that cover the same ground.

For simple integrations involving a small number of fixed databases the heterogeneity strategy may be the best solution. However, for more demanding integrations it makes sense, other things being equal, to adopt the ontological strategy.

3.5 Ontological paradigm

There is one more phrase whose use here needs to be clarified: *ontological paradigm*. The term *paradigm* is used here in a sense taken from Kuhn's [6]. For him, paradigms are the frameworks of background assumptions that scientists working in a field share.

People's ontological paradigm works in a similar way. It is the background of ontological assumptions that they share. They are often formalised into a number of categories into which the rest of the things that exist fall. These categories are also sometimes known as the top ontology.

4 Ontological analysis for semantic matching

The preceding terms can now be used to characterise what ontological analysis for semantic integration is. Ontology provides a framework and suggests a process for the analysis needed for semantic matching. This process focuses on the semantics of the database, identifying semantic divergence. It aims to purge this divergence to produce an ontological model. One key aspect of this model is that it explicitly contains at its top level the categories that inform the ontological paradigm.

To give you a flavour of this kind of analysis, this section of the paper briefly describes some examples of the main forms of semantic divergence. These should reinforce the point made earlier, that semantic divergence is ubiquitous in database systems. The first form described here is one of the most persistent and pernicious: divergence at the categorical level – the level of the ontological paradigm.

4.1 Categorical semantic divergence

Databases have a number of top-level categories or metatypes that form a framework. In some texts these database's categories (entity, object, attribute etc.) are – mistakenly – presented as if they were also ontological ones. If this were the case, then an object would always be represented in the same database category in every database. If a car is represented as an entity in one database, it would also be represented as one in every other database that used the entity category.

Entity type as a design decision. In practice, this usually turns out not to be so. This is a simple point often noted, but it can be difficult to appreciate, so I quote two extracts below to ensure that it is clear. These are from data modelling textbooks and turn on the idea that it is a design decision whether to represent a thing as an entity or an attribute[11].

> "One of the difficulties in designing data entities and identifying their relationship is that it is not always clear whether a data item should be included as an attribute of an entity or constitute an entity in its own right. A simple example to illustrate this point is the data item LOC (location) which is included as an attribute in the EMPLOYEE entity. This could

[11] Part 2 of Partridge's *Business Objects: Re - Engineering for re – use* (1996) has an extended example making this particular point by showing two different designs for the same entities.

> equally be set up as an entity, in which case it would have a one-to-many relationship with EMPLOYEE, as is shown in Figure 7-6. … No hard and fast rules can be laid down concerning the identification of entities in a database." pp. 194-5 [7].
> "There is no absolute distinction between entity types and attributes. Sometimes an attribute can exist only as related to an entity type. In a different context, it can be an entity type in its own right… For example, to a car manufacture, a color is merely an attribute of one of its products; to the company that makes the paint, a color may well be an entity type." p. 26 [8].

Database categories as a mode of representation. The root of the difference is that the database categories are modes of representation – rather than modes of existence (ontological). In other words, they are categorising the representation rather than what is being represented. Though these may map directly onto the ontological categories – and in an ontological model they would – there is no reason why they should. The database category is not determined by what (category of thing) is being represented, but by how the designer chooses to represent that thing. The semantic divergence of representational categories from ontological ones provides the basis for a substantial amount of ontological analysis.

4.2 Indexicality

There are a myriad of sources of semantic divergence. I now briefly describe one more common type, indexicality. This is where the truth of an expression (representation) depends upon the conditions of its utterance. A classical example is the expression "I am here" – which is usually true, but will refer to different people and places on different occasions. This is clearly a way in which we use language (representation) and not a way in which the world is. The ontological strategy takes indexicality as a kind of semantic divergence that needs to be refined out of the database to get to the ontological model.

Indexicality becomes an issue during integration when representations are indexed to their specific applications. The amount of indexicality in applications soon becomes apparent when one tries to integrate across enterprise boundaries. Most enterprises' systems take the perspective of their particular enterprise as their given context. A common example, also often found in textbooks, is: employee modelled as a sub-type of person. Examination of how the data is used usually reveals that what is meant is that the person is an employee of the particular company that 'owns' the application.

The issues of indexicality get complicated when the 'owner' of the application is not clear. For example, an application is being used by a department, but the employer is the company of which the department is part. Identifying ownership can get even more problematic during mergers and acquisitions[12] – increasing the importance of identifying any indexicality.

5 A better analysis

This kind of ontological analysis not only offers a better way to do the existing task, it also produces a better result. This section describes an aspect of this that is particularly valuable from the perspective of the enterprise. It is that the analysis encourages a kind of generalisation that reduces complexity, leading to a simpler, more general model.

---

[12] A point pursued in Partridge, C's *LADSEB-CNR - Technical report 07/02 - STPO - Synthesis of a TOVE Persons Ontology* (forthcoming).

5.1 Encouraging generalisation that reduces complexity

Complexity is a major issue for large integration projects. As the size of the individual applications and the number of applications grow, the amount of matching required grows exponentially. Ontological analysis helps as it both prepares the ground for and suggests generalisations.

Describing generalisation. Generalisation is not often described in textbooks – so I clarify what is meant here. It can be usefully contrasted with the notion of abstraction. This has its origins in Locke [9], see for example, ii. xi. 9 and 10 and iii. iii. 6 ff. Central to the notion of abstraction is taking an idea, then, by ignoring some of its details, ending up with a more general idea.[13] More recently, computer science has taken up Locke's notion.

Generalisation works in a different way. It is similar in that it starts with a collection of types and by analysing commonalities generalises them. It is different in that it makes the original collection redundant, not only without losing any information, but also ending up with a smaller simpler collection of types[14]. Though generalisation is not mentioned much in computing textbooks, it is common practice in actual projects[15]. What is less common is an environment, like that provided by ontological analysis, which encourages generalisation and creates the opportunity for high levels of generalisation.

Outside computing generalisation is recognised as a key feature of the growth of knowledge. For example, Kuhn's [6] regards radical generalisation as a feature of scientific revolutions – where the revolutionary theory is both more general, simpler and typically contains *more* information: information that was not explicitly known at the beginning. This generation of extra information is a common feature of generalisation – within scientific theories it is often known as fruitfulness.

It is clear that semantic heterogeneity and divergence can hinder generalisation. If the commonalities of two entities are represented in a semantically different ways, their similarities are more difficult to see. And the semantic differences may create the appearance of commonality where, in reality, none exists. So ontological analysis clears the ground for generalisation, by making the characteristics of what exists clearer. But it practice it does more than this. Experience of using the ontological analysis for semantic integration shows it actually encourages, often enforces, generalisation. Within the ontological framework, integration analysis naturally leads to generalisation. And it can also, usefully engender a culture of generalisation.

6 Using integration to build a universal reference ontology

It should also be becoming clear that the application independence of ontological models makes them a prime candidate for reference models. By stripping the applications of the semantic divergences introduced to satisfy their requirements or the design judgements of the designers, a common application-independent foundation is revealed. This can act as the basis for the integration of applications (and a variety of other tasks) in the domains that it covers.

---

[13] People were unhappy with Locke's proposal from the beginning. His near contemporary Berkeley criticised the notion: see for example, paras. 6 ff. and paras. 98, 119 and 125 of his *A treatise concerning the principles of human knowledge*. More recently Frege has made some trenchant criticisms. Dummett's *Frege: Philosophy of Mathematics* (1991) has a good summary of the Fregean concerns.

[14] Partridge's *Business Objects* (1996) has an extended description of this with examples.

[15] For example, Doug Lenat, the progenitor of Cyc, in a posting to the SUO mailing list dated Thu 11/04/2002 wrote "… we continually try to police the KB and find ways to generalize and combine assertions, to REDUCE the number of assertions …".

7 Summary

The theme of this paper is that there is an important stage in the process of integration – semantic matching – that has had insufficient attention until now and can be substantially improved though the application of traditional metaphysical ontology. As the paper has been at pains to point out, this can not only help to improve the existing process but also introduce additional benefits – such as the simplifications that generalisation brings. From a wider perspective, it also enables integration projects to become the engines for the production of a universal reference ontology.

Acknowledgements

This technical report on which this paper is based was produced while on sabbatical leave at the Group of Conceptual Modelling and Knowledge Engineering at CNR-LADSEB, Padova. I would like to thank them for their support.

References

1. Sheth, A. and J. Larson, Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases. ACM Computing Surveys, 1990. Vol.22(No.3): p. 183 - 236.
2. Parent, C. and S. Spaccapietra, Database Integration: The Key to Data Interoperability, in Advances in ObjectOriented Data Modeling, S. Spaccapietra, Editor. 2000, MIT Press: Cambridge, Mass.
3. Vermeer, M.W.W. and P.M.G. Apers, On the Applicability of Schema Integration Techniques to Database Interoperation. ER 1996, 1996: p. 179-194.
4. Gruber, T.R., A translation approach to portable ontology specifications. Knowledge acquisition, 1993. Vol. 5(No. 2): p. pp. 199-220.
5. Marcus, R.B., Modalities : philosophical essays. 1993, New York: Oxford University Press. xiv, 266.
6. Kuhn, T.S., The structure of scientific revolutions. 2d ed. 1970, Chicago,: University of Chicago Press. xii, 210.
7. Brookes, C.H.P., Information systems design. 1982, Sydney: Prentice-Hall of Australia. 477.
8. Tsichritzis, D.C. and F.H. Lochovsky, Data models. Prentice-Hall software series. 1982, Englewood Cliffs, N.J.: Prentice-Hall. xiv, 381.
9. Locke, J., An essay concerning human understanding, ed. P.H. Nidditch. 1975, Oxford: Clarendon Press. liv, 867.
10. Partridge, C., LADSEB-CNR - Technical report 05/02 - The Role of Ontology in Integrating Semantically Heterogeneous Databases. 2002, The BORO Program, LADSEB CNR, Italy: Padova.
11. Partridge, C., LADSEB-CNR - Technical report 06/02 - Note: A Couple of Meta-Ontological Choices for Ontological Architectures. 2002, The BORO Program, LADSEB CNR, Italy: Padova.
12. Berkeley, G. and J. Dancy, A treatise concerning the principles of human knowledge. Oxford philosophical texts. 1998, Oxford ; New York: Oxford University Press. vi, 237.
13. Dummett, M., Frege: Philosophy of Mathematics. 1991, London: Gerald Duckworth & Company Ltd.
14. Gilbert, M., On social facts. 1992, Princeton, N.J.: Princeton University Press. x, 521.
15. Goodchild, M., et al., eds. Interoperating Geographic Information Systems. 1999, Kluwer Academic Publishers: Boston.
16. Kent, W., Data and reality : basic assumptions in data processing reconsidered. 1978, Amsterdam ; New York New York: North-Holland Pub. Co. ; sole distributors for the U.S.A. and Canada Elsevier/North-Holland. xv, 211.
17. Papazoglou, M., S. Spaccapietra, and Z. Tari, Advances in object-oriented data modeling. Cooperative information systems. 2000, Cambridge, Mass.: MIT Press. xxiv, 367.
18. Partridge, C., Business Objects: Re - Engineering for re - use. 1996, Oxford: Butterworth Heineman.

19. Partridge, C., LADSEB-CNR - Technical report 04/02 - What is Pump Facility PF101? 2002, LADSEB-CNR: Padova, Italy.
20. Partridge, C., LADSEB-CNR - Technical report 07/02 - STPO - Synthesis of a TOVE Persons Ontology. (forthcoming). 2002, LADSEB-CNR: Padova, Italy.
21. Partridge, C. and M. Stefanova, A Synthesis of State of the Art Enterprise Ontologies: Lessons Learned. 2001, The BORO Program, LADSEB CNR, Italy: Padova.
22. Quine, W.V., The roots of reference. 1973, LaSalle, Ill.,: Open Court. xii, 151.
23. Quine, W.V., From a logical point of view : 9 logico-philosophical essays. 2d ed. 1980, Cambridge, Mass.: Harvard University Press. xii, 184.
24. Searle, J.R., The construction of social reality. 1995, New York: Free Press. xiii, 241.
25. Sheth, A., Changing Focus on Interoperability in Information Systems: From system, syntax, structure to semantics., in Interoperating Geographic Information Systems, C. Kottman, Editor. 1999, Kluwer Academic Publishers: Boston.
26. Strawson, P.F., Individuals, an essay in descriptive metaphysics. 1990, London, New York: Routledge. 255.
27. Strawson, P.F., Analysis and metaphysics : an introduction to philosophy. 1992, Oxford ; New York: Oxford University Press. viii, 144.
28. Lewis, D.K., On the plurality of worlds. 1986, Oxford, UK ; New York, NY: B. Blackwell. ix, 276.