

**The CEO Project:
An Introduction**

Chris Partridge

Technical Report 07/02, LADSEB-CNR
Padova, Italy, December 2002

Questo lavoro è stato condotto nell'ambito dell'attività del gruppo di ricerca
“Modellazione concettuale e Ingegneria della Conoscenza”
del LADSEB-CNR

LADSEB-CNR
Corso Stati Uniti 4
I-35127 PADOVA (PD)
e-mail: mbox@ladseb.pd.cnr.it
fax: +39 049-829.5763
tel: +39 049-829.5702

The CEO Project: An Introduction

Chris Partridge¹ & 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:

This is, in essence, the project initiation paper for the CEO Project. Its main concern is explaining the project's aims, how it intends to achieve them and the methodological framework within which the project will work. It explains the origins, conception and motivation for the project and gives an outline of the management framework for the project, in particular the first synthesis stage. It clarifies the terms of the art and describes the nature of ontological analysis. It also characterises the requirements that shape it and the meta-ontological choices and analytic styles that underlie it. Finally, it describes the potential applications and the next steps.

Introduction

This is, in essence, the project initiation paper for the CEO (Core Enterprise Ontology) Project. Its primary purpose is to explain the project's aims and how it intends to achieve them. It also describes the methodological framework within which the project will work. It can also be used to give interested parties an introduction to the project.

The first sections start by explaining the origins, conception and motivation for the project. They then give an outline of the management framework of the project, in particular the first synthesis stage.

The next sections deal with the approach adopted. This is based upon ontology, a millennia old discipline within philosophy. But its application to enterprise computing is both innovative and radical. Hence it needs some initial explanation. This is given by clarifying the terms of the art and then describing the nature of ontological analysis. The adopted approach is characterised firstly in terms of the requirements that shape it and by the meta-ontological choices and analytic styles that underlie it.

The final sections describe the potential applications and the next steps.

Origins of the CEO Project

The CEO Project has its origins in the REV-ENG Methodology.

Origins of the REV-ENG Methodology

The REV-ENG methodology grew out of a series of legacy application re-engineering projects each of which started with the re-engineering of a business model out of the existing legacy application¹. What differentiated these projects' re-engineering approach was a focus on recovering the ontological model of the business objects that underlay the legacy applications. This was typically a demanding task as there was

¹ More of the history can be found in the Preface to Partridge (1996) *Business Objects: Re - Engineering for re - use*.

The CEO Project: An Introduction

little or no documentation, only the implemented application. Over time the approach crystallised into a systematic process, which was codified into the REV-ENG (for REVerse ENGiNEering) Methodology: this is thoroughly documented in (Partridge 1996).

Comparisons between the early projects using the REV-ENG ontological analysis revealed that a number of the same general patterns were being repeatedly unearthed – surprisingly even in quite different business areas (e.g. banking and telecommunications). Typically the specific patterns in the applications looked different because they were combinations of different sets of general patterns. It soon became clear that significant time was being wasted repeatedly re-engineering these from scratch. This indicated the potential for high levels of re-use, which was initially exploited by making the general patterns available for re-use in subsequent projects².

Experience also showed that the potential for generalising (and so simplifying) was rarely exhausted in a single re-engineering. The general patterns found in one project were found, in subsequent projects, to be combinations of even more general patterns. This indicated that there was significant scope for evolving the patterns to greater and greater levels of generality and simplicity.

The conception of the CEO Project

The CEO Project was conceived out of the realisation that there would be significant economies in the application of the REV-ENG approach if one could start with a core business model for the enterprise. Reasonable economies would come from having the lower level general patterns found in single re-engineerings. But the really significant benefits would come from the very general patterns found in heavy duty re-engineering. Hence the CEO's overall aim is not only to recover the most common patterns found in businesses into a coherent and consistent model – but also to try and evolve this to much higher levels of generality and simplicity.

Motivation for the CEO Project

The development of the CEO is motivated by the expectation that building an ontological model with patterns that have high levels of generality and simplicity will significantly enhance the benefits of using the bare REV-ENG methodology: both further reducing the costs and significantly increasing the benefits that accrue at the business requirements (more specifically the business modelling) stage of projects. Costs will come down as projects re-use the CEO's ready-made foundation instead of building from scratch. Benefits of using the REV-ENG approach will increase in a number of areas, the main ones being:

- Reducing complexity,
- Improved inter-operability.
- Increasing longevity, and
- Technology proofing,

Reducing complexity: The approach used by the CEO enables increases in the generality of the business model that lead to both significant reductions in complexity and increases in functionality (measured against the legacy system). The key lesson is that complexity is not inherent – it is apparent and much of it can be re-engineered away. The complexity of the business model is a significant contributor to the overall

² Described in pp.276-8 of Ibid..

The CEO Project: An Introduction

complexity in large business applications – and so development and maintenance costs. The complexity cost of increasing functionality is a major barrier to improvements. Reducing these has paybacks all the way through the lifetime of the application.

Improving inter-operability: The CEO model will provide a canonical picture of the business that can be used as an inter-lingua for communicating between applications. Where applications have been re-engineered using the CEO model, their inter-operability will be much simpler³.

Increasing longevity: Experience to date seems to show that underlying the apparently changing forms of business software there are a relatively stable set of general patterns. The changes in business practices are often merely different combination of these patterns. Building an application based upon these can significantly increase its longevity. From an investment perspective, this increase in the term, leads to a corresponding increase in returns.

Technology proofing: An ontology's focus is on the business rather than the application. The ontology model will represent business objects independently of the technology that is used to implement it. The prime benefit of this independence is an asset that is future proofed against technological innovation.

A management framework for the CEO Project

To help focus the CEO work, a framework for the project has been established. These three 'management' elements of the framework are described below:

- the prime goal and deliverable,
- the scope, and
- an initial project work plan.

Setting the CEO's prime goal and deliverable

Given its motivation, the CEO Project's prime goal is to exploit the ontological approach initially codified in the REV-ENG Methodology to develop a toolkit that enterprises can use to both reduce the costs and significantly increase the benefits of producing a business model. The CEO's prime deliverable, and the main element of its toolkit, will be an ontological model⁴. This will represent the objects that exist in the enterprise field in a standard way.

Setting the CEO's scope

The scope of the CEO is circumscribed at these three levels:

- ONTOLOGY,
- CORE ontology,
- core ENTERPRISE ontology.

The scope of the initial analysis work is also circumscribed, to an extent, by the scope of the applications included within the CEO's re-engineering approach.

³ For more details see Partridge (2002b) *The Role of Ontology in Integrating Semantically Heterogeneous Databases* and Partridge (2002f) *What is a customer?*.

⁴ The terms 'ontology', 'ontological model' and 'epistemology' are described later.

The CEO Project: An Introduction

The scope of an ONTOLOGY

The scope of the CEO is focused on the application's **ontology** – the business objects it refers to independently of the way it represents them. There is a substantial body of philosophical work that provides a framework for talking about business objects in this way. Good starting points are Quine's notion of ontological commitment⁵ and Armstrong's notion of truthmaker⁶. In looking at the way an application represents the business, we can ask what this representation is ontological committed to – what objects is it committed to saying exist? Similarly we can ask what are this representation's truthmakers - what objects make the application's representations true? These are the objects that it acknowledges exist (or can exist). An ontological model is a model of these. The CEO's ontological model is intended to represent the objects that exist in its field – the enterprise. The epistemological aspects of this domain – covering what and how applications may know what they do – are not within scope.

The scope of a CORE ontology

What is a 'Core' Ontology? (Breuker, Valente et al. 1997) provide this useful description – “an intermediary between the generic top and the domain ontologies that contain the *categories* that define *what a field is about*.” Where a “field is a discipline, industry or area of practice that unifies many application domains ...”. The stratification and segmentation of ontologies into fields and domains (or whatever) is a practical matter – and is guided, as Breuker et al suggest, by how much it helps in practice to provide a unifying structure. The key point is that given a 'field' (or domain) there are core 'categories' that help to “define *what [it] is about*.” This the two key characteristics of a core ontology are generality and unity.

The focus of the CEO is on the unifying categories for its selected field – the enterprise. The scope of the CEO cannot however be restricted to purely these categories. For the ontological model to make sense, it needs to be embedded in a top ontology and fleshed out with some domain elements.

The scope of a core ENTERPRISE ontology

The CEO's field is the **enterprise**, and its ontological model will focus on the major categories that 'define what this field is about'. At this initial stage of the project, making a rough intuitive guess at what these might be helps to bring the scope of the CEO into focus and a basis for organising the initial analysis work. Intuitively these three seem like the most suitable major categories:

- *Person (AKA Party)*, who can enter into a
- *Transaction*, which often include agreements which involve an
- *Asset*.

Experience with the REV-ENG methodology suggests that the recovered ontology is likely to give a radically different perspective on these categories – which could be transformed by the analysis.

⁵ See Quine (1964) *Word and object*.

⁶ See Armstrong (1997b) *A world of states of affairs*.

The CEO Project: An Introduction

Broad categories

All three categories are broad. For example the first category, Person, encompasses both people and organisations. One of its unifying characteristics is that its members can enter into transactions. Hence the name in the data modelling community for Persons – Parties – as in ‘a party to a contract’⁷.

Intertwined categories

The unified nature of the enterprise field means that the categories are closely interlinked. This involves a network of ontological dependence. Transactions (contracts) are always entered into by persons – the transaction cannot exist unless the person does. This makes them (ontologically) dependent upon Persons for their existence. Transactions also typically involve and so are dependent upon assets, in a sense in which these are not dependent upon transactions⁸. Similarly assets are owned by persons – and so dependent upon them.

But the unification is much closer, more intertwining, with the categories overlapping – for example, a company seems to be both a person that can enter into agreements and an asset that can be bought and sold. It is also, from the perspective of incorporation, seems to be a kind of contract (transaction). The completed CEO will properly account for this intertwining: both dependence and overlapping.

The scope of the analysis work

The scope of the analysis is circumscribed by its re-engineering approach. The contents of the ontologies and applications provide an input to the analysis that helps to set boundaries on the analysis. This helps to ensure that the analysis focuses on content relevant to the kinds of systems that enterprises currently deploy.

Provided the sample is reasonably large, this approach does not exclude relevant content that is not in the input sample. The analysis is looking for the general patterns that underlie the existing content. All the relevant patterns are likely to be exhibited in even a small sample, though they may be easier to extract from a reasonably large one. The ontological model will allow for these to be combined in ways that are not exemplified in the sample giving a content that provides a reasonable coverage of the domain – certainly one far exceeding the input sample.

Drawing up the initial project work plan

At this initial stage a broad project work plan has been drawn up. This envisages two major stages:

- A synthesis stage – A Synthesis of (selected) State of the Art Enterprise Ontologies (SSAEO) to produce a Base Enterprise Ontology (BEO), which will act at the foundation for the second stage.
- A development stage – A development of the BEO into the industrial strength CEO ontology.

⁷ As is done, for example, in Inmon, et al. (1997) *The data model resource book*.

⁸ Assets are not obviously dependent upon transactions, as the legal notion of inalienable assets, ones that cannot be sold, illustrates.

The CEO Project: An Introduction

The synthesis (SSAEO) stage

The initial stage has been planned in more detail than subsequent stages. It has the goal of harvesting the insights from the best of breed enterprise ontologies and their synthesis into a single coherent whole.

The initial informal review

The SSAEO started with an informal review of enterprise ontologies with two goals:

- An assessment of the state of the art, and
- A selection of the best of breed ontologies to be synthesised.

An assessment of the state of the art

The informal review found that the ‘state of the art’ is immature, in particular that:

- there are not many enterprise ontologies (though there are many resources from which these could be mined), and
- those that exist have not yet reached ‘industrial strength’ as ontologies for semantic interoperability.

This second point is one of the reasons why the SSAEO synthesis needed to be more of a full re-engineering rather than a straight-forward merge/integration⁹.

A selection of the best of breed ontologies

The review selected the following ontologies for synthesis:

- Toronto Virtual Enterprise - TOVE (Fox, Barbuceanu et al. 1996) (Fox, Chionglo et al. 1993) (TOVE:http),
- AIAI’s Enterprise Ontology - EO (Uschold, King et al. 1997) (Uschold, King et al. 1998) (EO:http),
- Cycorp’s Cyc® Knowledge Base – CYC (Lenat and Guha 1990) (CYC:http), and
- W.H. Inmon’s Data Model Resource Book - DMRB¹⁰ (Inmon, Silverston et al. 1997).

Task breakdown for the SSAEO analysis

The SSAEO involves a substantial amount of work. As such, it made sense to break it down into a smaller number of tasks. It was decided that it should be broken down along two dimensions – the selected ontologies and the core categories. The first area selected for analysis was the TOVE ontology and the Person core category – this task

⁹ And so does not fall neatly into one of the usual categories; for example, those of integration, merge and use in Gomez-Perez, et al. (1999) *Some Issues on Ontology Integration*, that assume an underlying homogeneity among the ontologies.

¹⁰ In its own terms, this is a universal data model. However, from our perspective, it is in many respects an ontology. We considered having a number of commercial data models in the sample, but found that they were very similar – so there would be no real benefit. Inmon, et al. (1997) *The data model resource book* and Hay (1996) *Data model patterns* were neck and neck as the commercial data model representative. We selected Inmon (1997) as it seemed slightly more accessible. Note, a two volume revised edition of this has appeared: Silverston (2001a) *The data model resource book 1*, Silverston (2001b) *The data model resource book 2*.

The CEO Project: An Introduction

was named Synthesis of a TOVE Person Ontology (STPO). The segmented tasks and the two dimensions are shown diagrammatically in Figure 1 below.

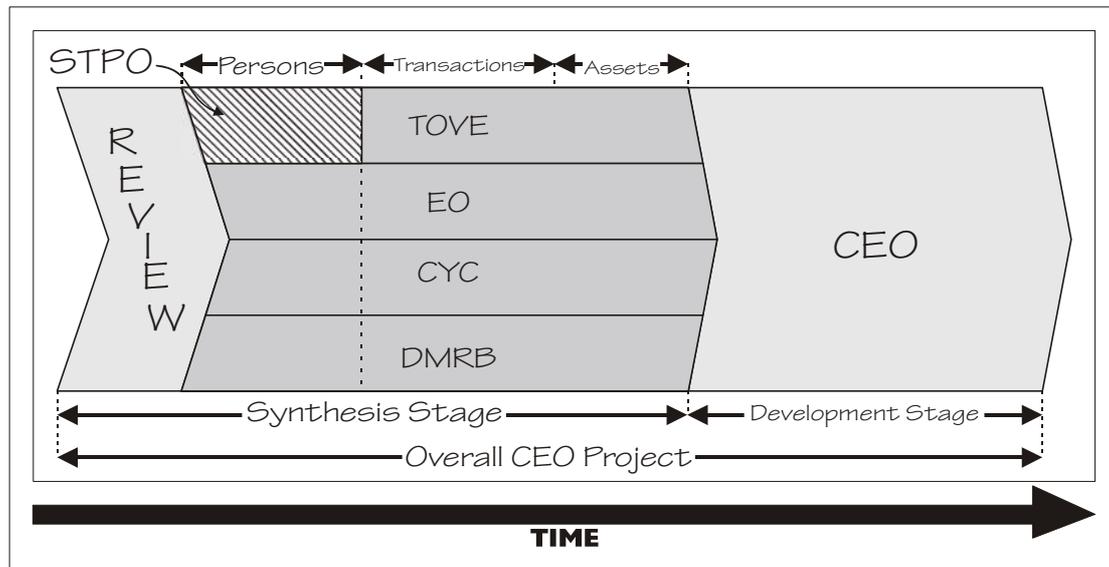


Figure 1 – The synthesis stage of the CEO project

Clarifying the terms of art

The key deliverable for the CEO is an ontological model – and this is based upon the notion of an ontology. The CEO needs to clarify what it means by these and related terms as, in the last few decades, the kinds of things that have been called ontologies has increased at least ten-fold. This clarification starts with two basic terms: ontology and semantics.

Ontology

Central to the CEO's approach is the traditional philosophical (metaphysical) notion of ontology – where this is “the set of things whose existence is acknowledged by a particular theory or system of thought.”¹¹ Here the set of things is not just restricted to simple entities, it includes every type of thing that exists: for example, it can include relations and/or states of affairs, if these are deemed to exist.

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.”¹² Quine's glib description captures the common intuitive position of many systems analysts, who unthinkingly assumed that the answer to the question “What is there – according to this application?” will be the set of things that the application represents.

Within the IT community there is no technique for identifying this ‘set of things’ apart from intuition. However, there is substantial body of philosophical work that provides techniques for analysing objects in this way. As noted earlier, good starting points are Quine's notion of ontological commitment and Armstrong's notion of truthmaker. In looking at the way a scheme represents its domain, we can ask ‘What is the

¹¹ E. J. Lowe in the Oxford Companion to Philosophy.

¹² In Quine (1948) *On what there is*, reprinted in Quine (1980) *From a logical point of view*.

The CEO Project: An Introduction

ontological commitment of this representation?’ – ‘What objects is it committed to saying exist?’ Similarly, we can ask ‘What things make the representation true?’ In this way, one can clearly differentiate between how something is represented (the representation) and what is being represented (the ontology). These can be (and often are) quite different, and different applications often have quite different representations.

Some care needs to be taken to distinguish this traditional metaphysical use of the word ‘ontology’ from one that has recently developed in some parts of Computer Science. Here an ontology is regarded as a “specification of a conceptualisation” (Gruber 1993) and has been applied to a wide range of things, including dictionaries. This sense of the word does not give a fine-grained enough tool for the CEO’s needs. For example, it regards an application as simply an ontology – and so it cannot make sense of talking about the ontology underlying it, let alone underlying a group of applications with a footprint over the same domain.

A similar point can be made about conceptual schemas, such as that described in ANSI/X3/SPARC (Tsichritzis and Klug 1978). These deal with representations of the conceptual perspective, and reflect how we conceive of the world – which is, in ways important for business modelling, not quite the same as what our conceptualisation commits to existing in the world (or what things make the conceptualisation true).

It has been recognised for a long time that metaphysical ontology has a role to play in IT. Over thirty years ago, (Mealy 1967) suggested that it was essential. (Kent 1978) makes a similar point at book length. However, it was only in the 1990’s that interest started to really grow, particularly in AI. However, work in this area has tended to be done using a revised conceptual notion of ontology – which is not suitable for the CEO’s purposes. In the sample of best of breed ontologies chosen by the CEO, the AI based ones (TOVE, EO and CYC) fall into this category. The DMRB, which (unlike the AI ontologies) is a distillation of actual practice, tries to look through the representation to the objects being represented – and, as such, takes a view reasonably consistent with metaphysical ontologies.

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) describe¹³. This needs to be distinguished from the different, but related, sense of the word in linguistics where it means the study of meaning¹⁴.

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

Ontological model

Someone who takes the metaphysical view needs to have a way to describe the ontology. At a bare minimum, they can make an inventory of the objects. As this includes relations, the result is more like a model than a mere list – so it is not

¹³ Or as Nelson Goodman put it in his Introduction to Quine (1973) *The 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.”

¹⁴ “Semantics – the study of meaning” from the Concise Oxford Dictionary of Linguistics, © Oxford University Press 1997.

The CEO Project: An Introduction

stretching the truth to call this an ontological model. For practical reasons, the model cannot name every object – and so typically restricts itself to naming types of objects and a representative sample of instances.

What characterises an ontological model is that it 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 model¹⁵.

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 (Marcus 1993), Ruth Barcan Marcus (explicitly following in the footsteps of Mill and Russell¹⁶) calls this ‘tagging’.

The distinction between an ontology and its ontological model should now be clear. However, ‘ontological model’ is a cumbersome term and it is usually clear from the context whether the ontology or its model is being referred to. So from now on, where the context can determine this, the term ontology will be used.

Semantic heterogeneity

Most applications are not ontological models. This is plain from a phenomenon commonly found in applications and much discussed in database literature – semantic heterogeneity. (Sheth and Larson 1990), on p. 187, provide a description of it. They 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 note that “... this problem is poorly understood, and there is not even an agreement regarding a clear definition of the problem.” From an ontological perspective it can be described as two semantically different representations of the same objects. Clearly where there is semantic heterogeneity both (all) of the representations cannot be ontological models.

Design autonomy and diversity

Sheth and Larson (among others) note that a prime source of semantic heterogeneity is what they call design autonomy. They describe this (on p. 187 of (Sheth and Larson 1990)) as “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 what I have called *design diversity*¹⁷ to manifest itself – where this is the actual manifestation of two different designs for the same objects. This diversity is partly the result of the different requirements of the

¹⁵ This is called *strong reference* within the REV-ENG Methodology described in Partridge (1996) *Business Objects: Re - Engineering for re - use*. See also ” Russell and Blackwell (1983) *The Collected Papers of Bertrand Russell*, Vol 8: p.176: “In a logically perfect language, there will be one word and no more for every simple object”.

¹⁶ Mill (1848) *A system of logic* and Russell (1919) *Introduction to mathematical philosophy*.

¹⁷ See Partridge (2002b) *The Role of Ontology in Integrating Semantically Heterogeneous Databases* and Partridge (2002e) *The Role of Ontology in Semantic Integration*.

The CEO Project: An Introduction

applications. But it also, partly, the result of the large amount of judgment exercised by the designers. This is reflected in the fact that different designers will typically (as a result of different judgements, different trade-offs) come up with different application designs for two similar applications. It can be quite surprising how different the designs can be¹⁸.

Semantic divergence

The notion of semantic heterogeneity is not based upon an ontological perspective. From this perspective the more relevant phenomenon is semantic divergence. This occurs where the semantic relationship between the ontology and the representation is not direct and straightforward. This is related to the notion of ontological model – as these have no semantic divergence. The kind of ontological analysis proposed for developing the CEO involves the extraction of an ontological model from applications, and this can be characterised as identifying and removing semantic divergences.

Classic example of semantic divergence

Semantic divergence is a common feature of our representations, including applications. A classical example that is often used to illustrate it is data that represents the average family as having 2.4 children. In answer to the question ‘How is this represented?’ – the answer is as a family with children. The answer to the question ‘What is being represented?’ (or what is being ontologically committed to, or what makes the representation true) is quite different. It is not, as the outward form suggests, a family – but a relationship between a set of families and the numbers of members of the sets of children they have.

A more commercially relevant example is an indexical¹⁹ representation such as a security purchase and sale. Where, for example, an organisation’s trade is represented in its application as a security sale. But the same trade is represented in the counterparty’s application as a security purchase. It is only a sale or purchase relative to a party to the trade (and their application). The underlying trade whose existence these representations commit to (is made true by) is neither a sale or purchase in itself. Technology is also a common source of semantic divergence. The technology in which an application is implemented has a strong influence on how it is represented in the implementation. A database or programming language comes with its particular forms, and the implemented representation of the business objects must fit into these. The focus on business objects independent of the application and how it is implemented removes this influence – in other words, the model is technology independent.

Experience with REV-ENG amply confirms the ubiquity of semantic divergence. Working applications are rarely straightforward ontological models – that is they have semantic divergences. Often it is the exigencies of constructing an application that

¹⁸ For example, the various chapters of Papazoglou, et al. (2000) *Advances in object-oriented data modeling* show markedly different designs for a standard car example. As its Chapter 10 Parent and Spaccapietra (2000) *Database Integration* notes there are a surprisingly wide variety of designs.

¹⁹ Indexicality is a common source of semantic divergence. It is where the truth of an expression (representation) depends 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 CEO Project: An Introduction

meets the enterprise's requirements – and then maintaining it within a budget – that give rise to them.

It is clear that the notions of semantic divergence and semantic heterogeneity overlap. What differentiates them is that semantic divergence assumes that there is a yardstick against which divergence can be measured – the underlying ontology – and so can measure this for a single application. Semantic heterogeneity merely notes differences in representation between applications. Hence, by itself, semantic divergence does not necessarily lead to semantic heterogeneity. If two applications are semantically divergent but have identical divergences, then they are not semantically heterogeneous.

However, a close examination of the literature shows that it is recognised that dealing with semantic heterogeneity (typically semantically matching heterogeneous applications) requires some knowledge of the ontology (sometimes called 'real world semantics') and so, of necessity, semantic divergence. For example, (Vermeer and Apers 1996) 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."²⁰

The REV-ENG experience is that much of the semantic heterogeneity in applications has its sources in differing semantic divergences. As the number of applications under analysis increases, the likelihood of this kind of semantic heterogeneity also increases. So, in practice, most ontological analysis projects have to deal with significant semantic divergence.

A canonical scheme

An ontological model can be seen as a canonical representation scheme. The notion of a canonical form comes from mathematics, where it is defined in terms of the general notion of a normalisation procedure, which consistently transforms objects (for example, matrices) to a canonical form. This enables one to determine whether different forms are equal relative to the normalisation procedure and its canonical form. In relational database modelling there is a well-known normalisation procedure for data that leads to a canonical form called the normal form. For computer applications, the ontological model can be seen as a semantic counterpart.

Ontological analysis as normalisation

One can see that ontological analysis is a kind of normalisation process for representations that leads to a canonical form in the shape of an ontological model. The normalisation can help to identify when the representations in different applications are of the same objects²¹ – and the ontological model is a direct representation of these objects.

²⁰ This also notes how difficult this can be: "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."

²¹ Or partially identical taking Armstrong's approach (described in, for example, Armstrong (1997b) *A world of states of affairs*) to mereology.

The CEO Project: An Introduction

Canonicity and independence

Many approaches to business modelling do not attempt to deliver either application or technology independence. This restricts the scope of the canonicity that they offer to the application and/or technology – a kind of local canonicity. By aiming for independence, the CEO will provide a more global canonicity (global relative to its domain – the enterprise).

Benefits of a canonical modelling scheme

The two main benefits of having such a scheme are firstly, that it provides a framework for re-use and generalisation and secondly, that it helps to facilitate interoperability. Re-use is dependant upon recognising where opportunities for re-use exist. A canonical scheme allows one to recognise when the same business objects are involved and so that there is a possibility for re-use. Generalisation involves recognising when two or more types of business object share common general characteristics. In a canonical scheme similar types are represented in similar ways, making similarities easier to identify – and so facilitating generalisation.

For interoperability, one needs to know when the representations in different applications refer to the same business object. A canonical scheme provides a basis for doing this by providing a framework within which the same business objects are represented in the same way in different applications' business models.

Canonical extendibility

The CEO will provide a core framework around which applications can be built, often independently, extending the CEO to meet their needs. Many of these applications will have underlying business objects in common. To support general interoperability, the ontological analysis (normalisation) process needs to work in a way that helps to ensure that the extensions are done in a consistent way: that the different extensions to the CEO independently represent the objects in the same way.

Categorical ontology

There is tradition that starts with Aristotle²² of not only ordering the types into a taxonomy but also explicitly including, at the top level, the major formal categories of entities (what can be called, more pompously, the types of existence). As a matter of principle, all the various lower level types fall under one or other of these top level headings. Following (Thomasson 1999), let's call this a categorical approach.

A number of philosophers have distinguished this categorical approach that attempts to provide an overarching structure from a more piecemeal approach that considers things on a case by case²³. They point out its advantages. For example, (Thomasson 1999) (on pp.115-6) notes a purely piecemeal ontology “can only provide a patchy view of what there is and a view that always risks arbitrariness and inconsistency.”

²² See Aristotle *The categories*.

²³ As already noted this follows Thomasson (1999) *Fiction and metaphysics* (pp.115-6). Similar distinctions are made in: Williams (1966) *Principles of empirical realism* (see p.74) – see the distinction between analytic ontology and speculative cosmology is made and Ingarden (1964) *Der Streit um die Existenz der Welt. 1. Existentialontologie* (pp.21-53) – see the distinction between ontology and metaphysics. Similar points are made in the Introduction to Hoffman and Rosenkrantz (1994) *Substance among other categories*.

The CEO Project: An Introduction

and (on p.117) “Approaching ontological decisions globally avoids the dangers of inconsistency and false parsimony that may result from piecemeal ontology.”²⁴

Computer science has picked up on the value of a categorical ontology. For example, John Sowa, in his latest book ((Sowa 2000) on p.51), states that “A choice of ontological categories is the first step in designing a database, a knowledge base or an object oriented approach.”

Core enterprise ontology

The ontology produced for the new accounting schema can be divided into a number of layers. At the top are the formal categories²⁵. Underneath this is the core enterprise ontology. A core ontology – as (Breuker, Valente et al. 1997) note – “contains the *categories* that define *what a field is about*.” Where a “field is a discipline, industry or area of practice that unifies many application domains ...”. Determining the scope of core ontology, and, in particular, the boundary between the top and core ontology, is a practical matter – and is guided, as Breuker et al suggest, by how much a candidate category helps to provide a unifying structure. The key point is that given a ‘field’ such as accounting there are core categories that help to “define *what [it] is about*.”

Epistemology

There are two reasons why it is useful to introduce the notion of epistemology here. Firstly to clarify by contrast the notion of ontology and secondly because any new applications built using the CEO will need to have an epistemology built on top of their ontology.

In philosophy, ontology and epistemology deal with two different questions, which result in two different ways of looking at and analysing the world. Ontology is concerned about what exists – whereas epistemology is concerned about what is (or can be) known by someone. For example, epistemology would attempt to explain how we can know about a particular type of thing, such as colours. Whereas ontology would be interested in what ontological type colours are.

These two different approaches are both useful when specifying a system, particularly a computer application. A system (application) will make some ontological commitment – it will assume that certain things exist. These things are its ontology, which answers the question – what exists according to the system. The ontological model will represent this.

A system will also have constraints on what it actually does (and can) know²⁶. These are described in its epistemology, which answers the question of what the system can (and must) know. In this context, an epistemology is always indexed to a knowing system. Of particular importance for operational applications is describing what it needs to know before it can do something. An epistemological model will represent this. Philosophical epistemology includes consideration of questions of belief, particularly the problems of false belief. Specifications of computer systems seem less concerned about these.

²⁴ Similar points are made in, for example, Collingwood (1940) *An essay on metaphysics* and Körner (1970) *Categorical frameworks*.

²⁵ In Partridge (1996) *Business Objects: Re - Engineering for re - use* this is called the framework level and an example of this for IT ontological analysis is given on pp. 276-8.

²⁶ In the case of a computer application this system may be a network of applications, each with its own constraints upon what it can know.

The CEO Project: An Introduction

One can regard the epistemology as looking at the world from the perspective of the system and what it knows and the ontology as standing back and describing the world that the system commits to from a perspective outside it. These two are interdependent. They deal with the same world, and mostly with the same things in that world. However, their different goals mean that they paint different perspectives of these – as the following examples show.

Examples

Let us assume, simplistically, that all humans are either male or female and that we are looking at a system that records humans' details including their gender. Then this system is ontologically committed to the existence of male and female types, which are sub-types of human and completely partition it. This is its ontology. However, we cannot guarantee that the system will always know a person's gender – so it has to deal with cases where it does not know the gender. So within the system's epistemology not all humans will be partitioned into male or female sub-types – in other words, within the epistemology the partitioning is incomplete. This gives us different, but equally valid, ways of categorising the world, illustrating how the approaches' different purposes can lead to different results.

Epistemology's purpose lines up quite neatly with one of the key requirements in specifying a computer application, clarifying what it must know and what it does not need to know. This makes documenting the epistemology an essential element of the specification of a system – though it is not usually called given such a grand name.

To see this, consider an insurance company that sells various types of policies. For its actuarial calculations, the company needs to know and so asks all its policyholders whether they are married and records the results. For its joint policyholders, it also needs to know, and so asks, whether they are married to each other and if they are, this marriage relationship is recorded. For its sole policyholders it does not need to know this information – so it does not ask for or record it.

The system is ontologically committed to the existence of persons and their married states. It is also committed to the fact that persons in married states have a marriage relationship with each other: this is what being in a married state means. In contrast, from the company's epistemic perspective, knowing someone is married does not mean knowing their spouse and marriage relationship. This is because for sole policyholders, the company can *know* that they are married, but not *know* who their spouses are and so cannot *know* their marriage relationships. Note that it may 'know' their spouses – because they are also policyholders – but not know that they are the spouses.

In current practice, the epistemic perspective plays a more prominent role in computer specifications because the current state of database technology means that the epistemology (and not the ontology) is reflected more directly in a company's database. In this example, the insurance company's database needs to be able to record persons that could be in a married state without having to record them having a marriage relationship. The fact that persons in a married state always have a marriage relationship cannot be recorded. This is why the use of the terms 'mandatory' and 'optional' for attribute and relations in database contexts are usually from an epistemic (not an ontic) perspective.

The CEO Project: An Introduction

Linking ontology to epistemology

It is important to understand how the ontology and epistemology link. One way of analysing this is to widen the scope of the ontology to include the system that is the subject of the epistemology (though this can pose some delicate problems and needs to be done carefully). Consider the first example. It may be tempting to regard the epistemological model as representing epistemic types that deal with known instances – as perforce only these are instantiated in the model. This would introduce new epistemic sub-types of human: known-male and known-female – and possibly unknown-gender.

However, it makes more sense to say that the system ‘knows’ the ontological types male and female, though it does not know all their instances – it may even know an instance of human, know it is male or female, but not know which. This captures our unreflective view of our own epistemology, which regards its male or female sub-types as ontological; in other words, as referring to all males and females not just the ones we know. Also, it avoids the possibility of an endless regress. Opting for epistemic ‘known’ sub-types would introduce the possibility of an endless regress. As it is possible for a system to know whether it knows, one would need to also introduce ‘known-known’ and ‘known-known-known’ sub-types and so on.

Under the ‘ontological types’ option, the ontology would capture the epistemology by explicitly recognising the system and its knowing relationship with the male and female sub-types and their instances. It would also recognise that only some of the gender types’ instantiation relations are known. So, for example, instances of human that are not epistemically classified by gender (in other words, whose gender is not known) would be marked in the ontology by not having a known relation between the system and the instantiation relation. This explains why the epistemic partition is incomplete – it is only representing the known instantiation relations. In this simple example, the epistemological perspective can be seen as a filtered view of the ontology – only showing what is known. This filtering led to the difference in structure.

From this brief outline it should be clear that specifications for enterprise applications schemes need both an ontology and an epistemology. Applications sometimes need to be able to record that they know someone, who has a gender, but they do not know which one. Insurance companies may need to know that if their policyholders are married that they have a married relationship with someone else – even if they do not know who the person is.

CEO sample ontologies and epistemology

All of the best of breed ontologies selected are an amalgam of ontology and epistemology. There are reasons for this. The AI based ontologies take a conceptual view of ontology and within this perspective, no distinction is made between ontology and epistemology. The data modelling based ‘ontology’ is meant to represent data that will be stored in an operational database²⁷ – which, of necessity involves epistemology. One element of the ontological analysis will be to filter out these epistemological aspects.

²⁷ This involves an element of equivocation as the model is at one time a representation of the business and another a representation of the data that, in turn, represents the business. However, this kind of equivocation is endemic in data modelling – and elsewhere. In philosophy it is often called a use mention confusion – as it confuses the use of a representation with mentioning it.

The CEO Project: An Introduction

The nature for ontological analysis

It may appear that having clarified what an ontology is, the next step is to work with the relevant experts to organise what they know about their domain: in the case of the CEO, with enterprise experts, to organise what they know about the enterprise. However, experience in building enterprise models, and more generally in specifying application requirements, shows that this is not a successful approach. Experts typically cannot articulate what it is they are working with²⁸. To understand this one must distinguish between two types of knowledge: know-how and what I shall call, know-what. Know-how is, traditionally, what experts have, otherwise they could not do their job. Know-involves the ability to articulate what the entities involved actually are.

Initially, the fact that experts do not have know-what may seem strange. But there is an example that we are all familiar with. We are experts in the use of our mother tongue. Our ability to apply subtle grammatical and syntactical principles is astonishing. We have language know-how. However, our inability, unaided, to articulate the principles that we are using is clear. We do not have know-what. (Strawson 1992)²⁹ using a similar example, points out that this shows knowing-how in no way implies knowing-what

Given this situation there is a clear need for a process to enable the articulation of the know-what. (Strawson 1992) points out that philosophical, metaphysical analysis is a traditional way to get a systematic representation of our know-how – our know-what.

An important foundation for this analysis is an understanding of why there is this gap between know-how and know-what. We develop this by looking at why it exists³⁰ for the institutional and social facts that are the subjects of most enterprise data.

Socially constructed business objects

Institutional and social facts are of a different kind than ordinary everyday physical objects, such as trees and stones. Physical objects have an existence independent of us. Whereas most institutional and social facts (which includes most business objects) are dependent upon us – and often seem to be constructed and maintained by us. This makes it even more difficult to understand why experts cannot articulate what these are.

Money is a good example of an institutional and social fact. People have used many things as money, including cowrie shells. What makes these money is that those

²⁸ There are examples of this inarticulacy in Partridge and Stefanova (2001) *A Synthesis of State of the Art Enterprise Ontologies* and Partridge (2002d) *STPO - A Synthesis of a TOVE Persons Ontology (forthcoming)* and Partridge (2002a) *What is pump facility PF101?*.

²⁹ Strawson (on pp. 5-7 of [Strawson 1992]) makes a similar point using the example of the first Castilian grammar being presented to Queen Isabella of Castile (in the Ninth Century). She asked what use it was, because “in a sense [Castilians] knew it already. ... though in a sense they knew the grammar ... , there was another sense in which they did not know it.” He draws “ the general moral that being able to do something ... is very different from being able to say how it is done; and that it by no means implies the latter.” Noting that “In contrast with the ease and accuracy of our use are the stuttering and blundering which characterise our first attempts to describe and explain our use.” Interestingly, he goes on to suggest that “the philosopher labours to produce a systematic account of the general conceptual structure of which our daily practice shows us to have a tacit and unconscious mastery.”

³⁰ That it exists is acknowledged - see, for example, Searle (1995) *The construction of social reality* and Gilbert (1992) *On social facts*.

The CEO Project: An Introduction

people accept them as such. Cowrie shells are certainly not intrinsically money, independently of the humans that use them as such. Similar things are true of languages and social institutions, such as marriage. The same is not true (at least not in the same way) of trees and stones. (Searle 1995) analyses this difference and describes these people-dependent objects as *socially constructed* and calls them *human institutions*.

Furthermore, the rules that characterise human institutions (to use Searle's name) work in a different way from the rules that characterise physical objects. Physical objects have rules (laws) that govern their behaviour, but cannot be said to know the rules. Stones do not have to learn the rules of gravity before they fall – nor can they decide, once they know the rules, that they do not want to follow them. Whereas people have to learn the rules that govern their human institutions. This can be quite arduous, as, for example, when someone has to learn a new language. It is also possible (in many cases quite easy) to 'disobey' the rules. For example, fluent language speakers can and do choose to make deliberate grammatical mistakes.

Most people are comfortable with notion that the conceptual structures of business and law that underlie the enterprise are socially constructed artefacts. However, with this notion often comes an assumption that is not so well warranted which is relevant to our topic: that these artefacts are solely the result of people following rules (even constituted by the rule following). This would imply that following the rules only involves consulting the rules in their heads. Ontological analysis would then involve examining these rules to develop a more precise picture – for example, by interviewing experts or training them in introspection. Though there are elements of truth in this assumption, I shall argue that it is mistaken in the case of analysing a core ontology and that a different method of analysis is more appropriate.

Following rules

The assumption seems, on the face of it, reasonable. As already noted, language is an archetypal example of a human institution. Consider someone who is learning a new language. When they try to speak, they have to laboriously consult the rules that they have been taught and are conscious of trying to follow them. Things are less clear for children learning their mother tongue. However, this can be explained, as Chomsky does in his account of Universal Grammar (Chomsky 1975). He reckons a child is able to learn grammar because he or she is already innately in possession of the rules of a universal grammar, though these are unconscious.

Closer examination of specific cases shows that most rule following is unconscious. When someone has learnt a language properly, they are no longer conscious of consulting and following its rules. Similarly, practicing business people and lawyers are typically not conscious of the rules they are following. The examination also reveals deeper problems with the rule-following account – it does not seem to fit the obvious facts. As Searle points out (on p. 127):

“the structure of human institutions is a structure of constitutive rules ... the people who are participating in the institutions are typically not conscious of these rules; often they have false beliefs about the nature of the institution, and even the very people who created the institution may be unaware of its structure.”³¹

³¹ Searle (1995) *The construction of social reality*, p. 127. Gilbert (1992) *On social facts* makes a similar point.

The CEO Project: An Introduction

Even when the beliefs are true, they are often inadequate by themselves for their purpose. As every system designer knows, experts typically cannot articulate rules to a sufficient level of formality and precision. Even though they have no problem in actually undertaking the tasks precisely enough.

If we know these rules and are following them, it seems strange that we can so regularly have false beliefs about them. Particularly when this seems to have no correlation with our ability to follow them correctly. It also seems strange that we cannot articulate the rules to a level of accuracy that we must know to be able to follow them properly.

The problem is in the assumption that we are always following rules. Searle articulates³² the issue as a question about the causal role of the rules, which neatly distinguishes between the two extremes in the ways in which rules operate. Are there rules in our heads that are the cause of us following the rules – in other words, are the rules representations which we consult and follow? Or, at the other extreme, do these rule representations have no direct causal role – merely providing a description of the actions we take? In other words, we do not ‘follow’ the rule representations.

Neither extreme seems to fit all the evidence. As noted earlier, human institutions are clearly not completely governed by rules in the way physical objects are. But, on the other hand, neither are they completely subject to forms of rule following. A number of philosophers³³ (including Searle) have suggested that our more conscious rule following is grounded in natural propensities that operate at the level of neurophysiological (non-intentional, non-representational) processes. This implies that the closer the rules are to the foundations, the less rule following is involved.

Nature of ontological analysis

Irrespective of the chosen explanation, these facts have a clear implication for the process of analysis involved in building a core ontological model. Given the level of false and inaccurate knowledge there is of the rules that experts are capable of articulating, and that, in many cases, they are incapable of articulating the rules, it does not make sense to base an analysis methodology on the presumption that they have a sufficiently complete and accurate knowledge of the rules. While interviewing experts may play a part in the analysis, it will never give a complete picture. And experts’ claims will need to be examined in the light of what actually happens, what people and organisations actually do.

CEO’s requirements for a core ontology

The CEO’s requirements will guide its choice of ontology and its use. Because it makes the relevancy of the requirements clearer, we consider the specific CEO requirements first, and then set them in context, by looking at the general requirements for a core ontology.

³² Gilbert (1992) *On social facts* p.127-8.

³³ Wittgenstein (1953) *Philosophical investigations* introduced the question of how we ‘follow’ rules and discussed natural dispositions. Kripke (1982) *Wittgenstein on rules and private language* revived the discussion more recently and it is now a lively topic. See also, for example, this point in Wright (1987) *Realism, meaning, and truth*, p.28 “... the path to understanding exploits certain natural propensities which we have, propensities to react and judge in particular ways. The concepts which we ‘exhibit’ by what we count as correct, or incorrect, use of a term need not be salient to a witness who is, if I may so put it, merely rational ...”.

The CEO Project: An Introduction

Specific requirements for the CEO

Currently most of people's understanding of the enterprise is at the level of particular domains (such as banking or retail). Their understanding of these domains is complex and this is reflected in the computer applications that they build to support it. There is not really any general, integrated, homogeneous understanding across these domains. And this is a significant barrier to making across the board improvements.

A core enterprise ontology has a basic requirement for at least some level of general understanding as it is intended to capture the general categories that unify its field – categories that are wider than the individual domains within the field. This requirement needs to be met if the CEO is to qualify as a core ontology.

The starting point for delivering this is the relevant generalisation of existing understanding – existing categories. The goal is to subsume the various categories of domain specific understanding under more general categories of understanding that transcend the domains – without any loss of information³⁴. This constitutes a requirement for the CEO to make the significant increases in relevant generalisation.

To meet its goal of making significant improvements, the CEO project has set itself the more stringent requirement of developing a *simple* integrated homogeneous understanding – one that is sufficiently simple that it will be feasible for people to grasp it as a whole. It is anticipated that this will lead to not only significant improvements in people's understanding of the enterprise, but also much simpler computer applications (so ones that are easier to develop and maintain).

It is not just generalisation and simplification that are important – understanding is as well. In most projects, this understanding is developed in a relatively informal, indirect way. The CEO project, with its ontological approach, focuses more directly on this. It has a requirement to offer a reasonable explanation that can act as a basis for a (sufficiently) common understanding. Along with the generalisation and simplification requirements this constitutes a requirement to provide *significant* improvements in our understanding.

Finally there is a requirement to formally and precisely specify the ontology. The specification (the ontological model) needs to be formal enough to form the basis for a computer specification. It needs to be precise enough to enable the automation of the relevant kinds of processes.

Summary of specific requirements for the deliverables

These requirements can be summarised as follows:

- A basic requirement to capture (in the CEO's ontological model) the general categories that unify the enterprise field – categories that are wider than the individual domains within the field.
- A requirement to make the significant increases in relevant generalisation.
- A more stringent requirement to develop a sufficiently simple understanding of these categories that it will be feasible for people to grasp their unifying structure as a whole.

³⁴ In Partridge (1996) *Business Objects: Re - Engineering for re - use* this is called compacting and involves the use of faceted generalisation – which is briefly described later.

The CEO Project: An Introduction

- A requirement that the ontology should also offer a reasonable explanation of the unifying structure that can act as a basis for a (sufficiently) common understanding.
- A requirement to specify the CEO's ontological model in an appropriately formal way, so that it can form the basis for a computer specification.
- A requirement to precisify its formal ontological model in ways relevant to enterprises – that is, ways that enable automation of the relevant kinds of processes.

Requirements for the use of the CEO's deliverables

These specific requirements relate to qualities of the CEO's deliverables rather than qualities of its use and so can be checked by inspection. However, the overall goal of the CEO project is to develop a toolkit that enterprises use successfully. So a key requirement is that it can be used successfully – and the best check for this is to actually apply it.

General core ontology requirements

These specific requirements for the CEO's deliverables can usefully be put into a general context. As just noted, the CEO will be judged by its utility: in particular, the value of the domain ontologies and applications that it helps to build. Engineering and science are assessed in a similar way and have identified a number of closely inter-related values³⁵ that characterise successful technologies and theories. It is a requirement that these should also characterise the CEO and its applications.

The engineering-based values are:

- Teachability
- Consistent applicability

These engineering values characterise use, whereas, in general, the scientific values characterise the general qualities of the theory/model. As such, for the CEO's purposes, they need to be qualified: the qualified values are:

- Relevant precision and sufficient formality
- Sufficient simplicity and relevant generality
- Appropriate unity and explanation
- Relevant fruitfulness
- Relevant repeatability – re-usability

Here the relevancy, sufficiency and appropriateness qualifications are relative to the uses of the CEO.

Most of these values apply at two levels. Firstly, they apply directly to the CEO: it needs to exhibit them. Secondly they apply to the domain ontologies developed using the CEO: they also should exhibit the values.

Most people will have some general understanding of what these values are. However, there are a few points worth touching upon to clarify details.

³⁵ Most of these values are not particularly new. For example, Aristotle *The metaphysics* [981a-b] characterises *teche* (technology) with a similar list – see also the discussion of *teche* in pp.95-6 Nussbaum (2001) *The fragility of goodness*.

The CEO Project: An Introduction

Teachability

The value of the CEO – and the domain models (developed using the CEO) – will be strongly determined by whether they are consistently teachable / learnable. If teaching a model is not easy enough, if learning it is a difficult, mysterious and unpredictable process, this makes it difficult to deploy and so less valuable.

Consistent applicability

The models also need to be applicable. The domain models need to provide a relevant basis for tasks such as specifying the requirements for a computer application or the semantic integration of a range of applications. The CEO model needs to provide a relevant basis for developing domain models that can do this.

Furthermore, both types of model need to be consistently applicable. It defeats the purpose of the model, if two trained practitioners working independently on similar tasks consistently produce markedly different results. Similarly, a model of a domain developed in one enterprise by one practitioner should be consistently applicable in another enterprise in the same domain. The model of a domain should not depend upon who develops it or where it is developed. This makes the models eminently re-usable – and also facilitates inter-operability.

Though this consistency requirement is not often proposed in software engineering³⁶, it is commonplace in traditional engineering and scientific processes. Engineering processes and scientific experiments are expected to be repeatable. Engineers and scientists in Australia expect to be able to repeat a process or experiment developed in England.

Relevant precision and sufficient formality

It is useful to make a distinction between what I shall call formality and precision: between the formality with which the representation is expressed and the precision with which it refers³⁷. This makes formality a property of the representation and precision a property of the relation between the representation and the represented. There is a one-way dependence between the two. It is possible (but not useful) to have a very formal representation that is not particularly precise. However, it is more difficult to increase the precision of a representation without also needing to increase the level of formality.

Formality is important here as it is needed if the ontological model is to be translated into an adequate computer specification. The formality needs to be sufficient for this translation. It is worth noting that formalisation typically leads to a change in meaning – as, for example, it eliminates ambiguity (as Quine notes for a formalisation into logic³⁸).

³⁶ For example, it is common in database design to expect two designers to produce very different ways of representing the same information – this is known as design diversity Parent and Spaccapietra (2000) *Database Integration*.

³⁷ This is a technical use of the terms – the ordinary language meaning is not ‘precise’. For example, Russell (1923) *Vagueness* p. 153 uses precise to mean what I have called formal, saying a belief ‘is accurate when it is both precise and true’ allowing for the possibility of a false precise belief – in my terms, an imprecise formal representation.

³⁸ Quine (1964) *Word and object* pp. 159-60, where he calls this formalisation, regimentation. In §33 - *Aims and Claims of Regimentation*, explains the consequences of regimentation “... If we paraphrase a sentence to resolve ambiguity, what we seek is not a synonymous sentence, but one that is more informative by dint of resisting some alternative interpretations. Typically, indeed the paraphrasing of a

The CEO Project: An Introduction

Most of ordinary language is manifestly imprecise, and a reasonable position is that all representation is inherently imprecise³⁹. This makes a goal of exactitude impossible: as Aristotle said “it is the mark of an educated man to look for precision in each class of things just so far as the nature of the subject admits”⁴⁰. Furthermore the cost of exactitude may exceed any benefits. So the goal is to represent objects to the relevant level of precision. This is a well-established notion in engineering, where the tolerance (the degree of precision) required is often specified.

A key driver for increases in precision is increases in generality, which introduce objects that span larger numbers of domains. In small (specific) domains, there can be quite a high degree of tolerance (low precision) – but as the scope grows, the tolerances need to become smaller (the precision higher). This can be illustrated using inference.

Since Frege, there has been a growing recognition that imprecision has an impact on logical inference. Starting with imprecise premises, each inference tends to significantly dilute precision⁴¹. As Duhem pointed out⁴² one needs inferences where the approximate truth of the premises is sufficient to guarantee the approximate truth of the conclusion. This suggests an engineering approach that recognises the dimensions along which ‘precise’ inference is likely to be needed (the relevant dimensions), and focuses on getting the required level of precision for a representation along these dimensions. Given this, one can see that enlarging the scope of an application is likely to enlarge the number of different dimensions along which representations have a required level of precision – and so likely to lead to increased their required level of precision.

An ontology captures the increased levels of formality and precision in a public and explicit specification – the ontological model. In our existing paper and ink technology based culture, the profound knowledge of a domain is in a private, intuitive form – the know-how in experts’ heads. As has already been noted, building an ontological model involves the construction of a public and explicit explanation of know-what corresponding to this know-how.

sentence S of ordinary language into logical symbols will issue in substantial divergences. Often the result S' will be less ambiguous than S, often it will have truth values under circumstances which S has none (cf. §§ 37 f.), and often it will even provide explicit references where S uses indicator words (cf. § 47).”

³⁹ As taken, for example, by Charles Peirce and Bertrand Russell. For a brief overview of their position on this see Williamson (1996) *Vagueness*.

⁴⁰ Introductory quotation headed ‘A Warning’ in Armstrong (1997b) *A world of states of affairs*.

⁴¹ Dummett points out quite clearly on pp. 50-1 of Dummett (1991) *The Logical Basis of Metaphysics*, in a section called ‘Degeneration of Probabilities’. “Hence it is sufficient, for mathematical purposes, that a principle of inference should guarantee that truth is transmitted from premises to conclusion. Outside mathematics, we have a motive to demand more, if we can get it. ... Most of our beliefs are perforce based upon grounds that fall short of being conclusive, but a form of inference guaranteed to preserve truth is not, in general, guaranteed to preserve degree of probability. ... The ‘ideal’ subject starting from beliefs whose probability is close to 1, will end up with beliefs negligibly greater than 0; the man of common sense, initially adopting beliefs with a much weaker evidential basis, but reasoning from them only to a meagre extent, will finish with far fewer beliefs than he. That is why scientific conclusions arrived at by long chains of impeccable reasoning almost always prove, when a direct test becomes possible, to be wrong. ... In practical life, truth is valued chiefly as a guide to action; and then the principal remedy for the degeneration of probability in the course of inferential reasoning is to employ it sparingly.”

⁴² As noted in Black (1937) *Vagueness*.

The CEO Project: An Introduction

Sufficient simplicity and relevant generality

The CEO's goal is a core ontology model that contains the unifying categories for its field (in other words, is relevantly general) and is sufficiently simple that a single person can grasp it.

Part of the process will be taking the general terms from the existing languages and identifying the general entities that they are committed to. The problem is that in fields, such as the enterprise, there is a plethora of these languages. Different domains naturally develop their own dialects, and enterprises their own idiolects. And there is often little real commitment to, and so little work done on, developing and using a common language.

There is some benefit to be derived from a harmonisation of the existing terms in the various dialects and idiolects. This will unearth any generality that already exists, sometimes implicitly, across the various domains. However, there is no reason to expect any of the individual domains to have developed terms (and an associated ontology) that cover the whole enterprise field.

Furthermore, there is no reason to expect harmonisation to increase simplicity – it is more likely to increase complexity. To increase simplicity, there needs to be a level of revision that goes beyond harmonisation. As David Lewis notes (on pp.133-5 (Lewis 1986)) “trying to improve the unity and economy of our total theory” involves “two things that somewhat conflict”. Firstly, “to *improve* that theory, that is to change it”, and secondly, “to improve *that* theory, that is to leave it recognisably the same theory we had before”. As he notes, the first of these can, and does, lead to improvements that “correct” common sense. As he says, this may have some costs but these “must be set against the gains” within the overall scheme of things. The REV-ENG experience confirms that significant gains in simplicity tend to come at a cost of ‘correcting common sense’.

It may seem counter-intuitive to expect a revised model to be both more general and simpler. This goes against the grain of most application building experience. But it is a trait of successful scientific theories. The classic scientific example is Einstein's well-known equation ‘ $E=mc^2$ ’. This replaced pages of complex equations with a single line, and the ‘improved’ theory involved a new perspective that ‘corrected’ common sense.

So, as Lewis points out, to meet the CEO goal of increasing both simplicity and generality, the ontological analysis will involve more than a mere harmonisation of the existing conceptualisations. It will need to make the kind of revisions that are exemplified in scientific revolutions, where well-accepted common sense needs to be corrected.

Appropriate unity and explanation

Unity and explanation are more global than local features of the model. In a unified ontology, the individual local general patterns form part of a common global framework. The two are linked, as providing a unifying framework is also one aspect of giving an explanation. Another aspect is describing the causes (in the sense of Aristotle's four causes) – answers to the question ‘Why?’ in terms of ‘Because ...’.

Experience has shown that unity and explanation are useful in two main ways. Firstly, they make a model easier to comprehend. Secondly, they appear to be good indicators of fruitfulness. One apparent difficulty is that it is hard to give an exact measure for

The CEO Project: An Introduction

the level of unity or explanation, however this is no real problem as people find it easy enough to recognise this intuitively.

While the formal ontological model will explain to an extent what the objects are, it is not sufficient for human understanding. Typically it needs to be supplemented and supported with a certain amount of informal explanation.

Relevant fruitfulness

Fruitfulness is often cited as the key property of a successful theory. History shows that successful scientific theories are typically fruitful, having a plethora of major applications that were not considered by their developers. One explanation (put metaphorically) that goes back to Plato is the notion of carving nature at its joints. More recently, and more prosaically, Hilary Putnam talked about cookie cutters and lumpy cookie dough. The underlying idea is that the theory captures somehow the real structure of nature.

This helps to explain another important feature of fruitfulness, its support for semantic extensibility. Where a fruitful model needs to be extended, the extensions tend to fit naturally. The extended system retains its simple general structure.

From the enterprise perspective, fruitfulness is a valuable property. Business change is a feature of modern enterprises. New business products are regularly introduced (for example, a new synthetic instrument in the financial sector), requiring support for them to be developed. Yet these 'new' products are often not fundamentally different from the existing range, rather new combinations of existing elements. A fruitful ontological model – one that carves the enterprise at its joints – will already have these elements as components and cater for the new product as a new combination of these. And if the ontological model is translated into the enterprise's systems, they will support it.

Developing the CEO to requirements

These requirements, specific and general, are intended to shape the development of the CEO. The sample of best of breed ontologies will be harmonised and revised to produce the kind of general, simple, explanatory and fruitful model required. However, there is an important pre-requisite to this.

Choosing an ontological paradigm

The CEO is intended to be a categorical ontology, and as such requires a set of top categories into which the rest of the things that exist fall. This helps to organise what kinds of things can exist – and how they can exist. As such, these sets are ontological paradigms – the term 'paradigm' is used here in the sense (Kuhn 1970) coined: a scheme that fixes a particular world view.

Philosophical analysis has revealed that there are a number of possible ontological paradigms that one can adopt. It has also revealed that ordinary everyday conceptions of the world, such as those encountered in the enterprise, are neutral with respect to these paradigms. These conceptions do not seem to be developed on the basis of any particular paradigm and, with suitable revisions, they can be aligned with any of them – they are victims of a kind of ontological relativity⁴³.

⁴³ This term is taken from Quine (1969) *Ontological relativity, and other essays*.

The CEO Project: An Introduction

This ontological relativity can be explained as arising from making different meta-ontological, metaphysical choices⁴⁴. These choices dictate the top level categories into which the rest of the things that exist fall. They are also essential for making the analysis work consistently, ensuring that it does not lead to different results in different times and places – enabling a kind of semantic normalisation process.

Within a paper and ink culture, most communities have no practical need to understand the meta-ontological choices and consciously fix on an ontological paradigm – regimenting their classifications to fit under its top categories. This is reflected in the difficulty they have in characterising an adequate set of top-level categories. However, with the advent of computer technology it makes sense to build applications upon ontological models and so these choices and their related paradigms become relevant – and working practices will need to adapt to this.

This lack of awareness of meta-ontological choices is reflected in the sample of ontologies selected. None of them has a well-formulated top ontology. This is yet another reflection of the immaturity of work done in this area.

The Business Object Paradigm

The CEO needs to have an ontological paradigm and so has selected one based upon the substantial REV-ENG experience with ontological analysis. This is called the business object paradigm and is described in detail in (Partridge 1996). This is to be regarded as a starter top ontology that will be honed and refined in the light of the CEO analysis. The details of the ontology are to be found in Appendix A.

It will be helpful here to give some insight into the nature of the paradigm by characterising it in terms of:

- the metaphysical choices it embodies, and
- the styles of analysis it encourages.

As often happens at this very general level these two are inter-related – this will become clearer in the exposition below.

General characterisations

Before looking at the choices and styles specific to the business object paradigm, it is useful to examine a couple of characterisations that apply reasonably generally. One is a meta-ontological preference that informs the choices and the other is a style that applies to most ontological analysis.

A general meta-ontological preference for unifying entities

The general requirement for simplicity leads to this meta-ontological preference. A number of the meta-ontological alternatives involve a choice between multiplying or unifying entities. From a simplicity perspective, the concerns are wider than just inflating the number of entities, there is also a requirement to explain the relation between the multiplied entities. This is not an issue for the unifying option, as they have been unified into a single entity.

Hence the general requirement for simplicity therefore leads to a preference, other things being equal, for a unifying option. This can be seen as a variety of ontological parsimony, much like Ockham's razor.

⁴⁴ See Partridge (2002c) *Note: A Couple of Meta-Ontological Choices for Ontological Architectures* and also Chap. 1 – *Meta-ontology* of Van Inwagen (2001) *Ontology, identity, and modality*

The CEO Project: An Introduction

A general ontological style

Identity is of central importance to ontology. It is extremely useful for generating questions that help us to see what something is. Typically, the question will ask whether and why two names or descriptions refer to the same or different entities, or more generally, under what conditions two names or instances of a type refer to the same object. Usually, the answer depends upon which metaphysical alternative is selected.

Associated with identity is mereology; which looks at the whole-part relation. This can be seen as partial identity (Armstrong 1997b): a part of something is partially identical to its whole. Mereology generates similarly useful questions about the partial identity of objects.

The specific metaphysical (meta-ontological) choices

The metaphysical (meta-ontological) choices help to determine the overall structure (architecture) of the ontology. Typically, these choices commit us to particular ontological categories. These choices influence one another: making one choice has architectural implications for the other choices. This means that a major concern in constructing an ontology is making these meta-ontological choices in a co-ordinated way, committing to a (reasonably) coherent set of ontological categories.

The meta-ontological choices

We now look at three of the meta-ontological choices that are particularly relevant to ontological analysis:

- Minimal categoricity
- Perdurantism, and
- Extensionalism,

All three are motivated by concerns for ontological parsimony.

Minimal categoricity

As part of a general preference for simplicity and ontological parsimony, the business object paradigm only commits to a minimal number of necessary categories for its objects. There are three main category simplifications that it adopts:

- Naturalism,
- Materialism, and
- Unifying space-time and matter

And a natural extension of these three simplifications:

- Mereological extensionalism.

Naturalism

The first simplification is what (Armstrong 1997b) calls naturalism: “It is the contention that the world, the totality of entities, is nothing more than the spacetime system.” This helps to enforce a useful rigour to the analysis, as one has to identify the objects in space-time, and cannot have recourse to abstract objects – except as place-holders.

The CEO Project: An Introduction

Materialism

A second simplification-unification is a single category for objects existing in naturalistic space-time – the material category. This obviates the need to have disjoint categories for form and matter. So, using the traditional example, a statue and the clay it is made of both belong to the same single material category. Materialism and identity extensionalism (see below) together imply that where descriptions of a statue and a piece of clay pick out the same spatio-temporal extension, they refer to the same object.

Unifying space-time and matter

A third simplification is the unification of space-time and matter. Under this scheme, space-time regions and their occupants belong to the same category. This and extensionalism (see below) imply that descriptions of a physical object and the space-time region it (completely) occupies refer to the same object⁴⁵. This unification allows for unfilled space-time, regions that are unoccupied by matter.

In a dualist scheme there would be two disjoint ontological categories – space-time and matter. In this scheme, immaterial regions of space-time can be co-located with physical objects. For example, there would be an immaterial region called the Earth and the planet Earth that (always) occupies it. In the unified scheme, these are two ways of describing the same object. Within this scheme, regions, such as Earth, are ‘immaterial’ in a different sense – in the sense that it is immaterial what occupies them or, indeed, whether they are occupied.

Mereological extensionalism

A natural extension of the last three category simplifications is mereological extensionalism (see (Simons 1987) for more details), which is adopted. This is the principle that the sum of parts can only be a single whole. This makes sense where there is only a single main category, and so the whole and parts must both belong to it,

Unifying the past, present and future

There is a simplification relating to time, which is similar to but not quite, a category simplification. Physical objects, such as trees, stones and people, persist as individuals through time, despite changing. There is a meta-ontological choice to be made about how this is dealt with. A choice between regarding these bodies as changelessly extended in time (perduring through time) or changing as they endure through time⁴⁶. The business object paradigm chooses to treat them as perduring through time – this is called a perdurantist position in contrast to the opposing endurantist position. From the perdurantist perspective, changes over time are regarded as different temporal parts of the object as having different properties.

This naturally leads to a position that deflates tense distinctions. Endurantists tend to argue that there is a difference between the future, present and past me – even with

⁴⁵ See Note 10 on p.76 of Lewis (1986) *On the plurality of worlds*, where quoting arguments from Nerlich (1994) *The shape of space*, he makes the same choice.

⁴⁶ The terms ‘endurantist’ and ‘perdurantist’ are taken from David Lewis’s book *On the plurality of worlds*. (1986), where ‘persist’ is intended to be neutral with regard to the ‘endure’ and ‘perdure’ interpretations. For a more extended discussion of this choice see Partridge (2002c) *Note: A Couple of Meta-Ontological Choices for Ontological Architectures*.

The CEO Project: An Introduction

respect to the same time. So, the future-me tomorrow is different from the present-me when tomorrow arrives and the past-me after tomorrow. The perdurantist has no need for such a distinction – nor the problem of explaining how these ‘me’s are different but the same.

Identity extensionalism – categorical formal identity criteria

(Quine 1969) sloganised his position as ‘no entity without identity’. A straightforward strategy for implementing this adopted by the CEO (and Quine) is to provide the top categories with criteria of identity for their instances.

Extensionalism is a way of characterising the formal identity criteria for the categories – in terms of their extension. The alternative is intentionalism, where two different objects can have the same extension, but be differentiated by their intension. The formal identity criteria for intension has then to be formulated. This is discussed for types and elements below.

Extensional identity criteria for types

The extension of a type is the collection of objects that are instances of it. A simple view of this only considers the actual objects that are instances of the type – and maybe even additionally restricts actual to now – that is, the present instances. Basing identity on this leads to problems, the classical illustration of which is the story of Plato’s definition of humans as featherless bipeds – on the basis that the two types had the same ‘actual’ extension. Whereupon, Diogenes is said to have plucked a chicken and said “Here is Plato's human”. A more sophisticated view takes the extension of a type to include all possible instances⁴⁷. This is sufficiently fine grained to avoid these kinds of issues. It also provides a reasonably robust mechanism for determining identity, and partial identity. Where ‘two’ types have the same instances, the same extension, they are identical. Where two types have some instances in common, they are partially identical.

An intensional view is even finer grained than this. It would recognise two (ontological) types with different ‘meanings’, but the same extension. For example, it would regard equilateral triangle and equiangular triangles as different (ontological) types – even though they have exactly the same extensions. This intensionalist policy has a number of problems. It has a multiplier effect, increasing complexity. One would need not only more types, but also a need for structures /explanations to ‘manage’ the relations between types with the same possible extensions. For example, explaining why equilateral triangle and equiangular triangles necessarily have the same extension. This additional complexity does not seem to bring any apparent overall benefit. Of course, there are differences in meaning between the terms ‘equilateral triangle’ and ‘equiangular triangle’ – but these can be explained in semantical⁴⁸ terms – without burdening the ontology.

Extensional identity criteria for elements

Within the minimal categorical structure of the business object paradigm, all elements have a spatio-temporal extension. This provides us with an identity criterion. If

⁴⁷ Though this, of course, raises ontological questions about the nature of these possible instances’ existence. See Lewis (1986) *On the plurality of worlds* for one position on this.

⁴⁸ Semantical in the philosophical sense – the relationship between words and objects. One place where this kind of semantic explanation is provided is Bealer (1982) *Quality and concept*.

The CEO Project: An Introduction

elements have the same spatio-temporal extension, then they are the same. In less technical jargon, if two things are always in the same place at the same time, then they are the same. As (Locke 1690) pointed out⁴⁹ some time ago, if two things have different beginnings (or endings) they cannot be the same thing.

A classic example is the two names 'Morning Star' and the 'Evening Star'. Ancient astronomers at first thought these were two different planets. However as their observations became better, they realised that these were in the same places at the same times - that they were one thing, the planet Venus.

Styles of analysis

It has been found useful to characterise the history and practice of science in terms of styles of thinking (this is done, for example, in (Crombie 1994) and (Hacking 2002)⁵⁰). A couple of examples of the kind of style that Crombie identifies are:

- ordering of variety by comparison and taxonomy
- the deployment of experiment both to control postulation and to explore by observation and measurement.

One reason for focusing on styles is that they offer a more stable way of characterising the scientific enterprise than scientific theories. Clearly Crombie's 'ordering of variety' style has persisted through a number of changes of theory.

There are a number of styles that characterise the kind of ontological analysis needed for the CEO. These have, like scientific styles, emerged from successful practice rather than theory. The meta-ontological choices turn out to have a big influence on the styles of analysis – in particular, extensionalism has led to a style called here extensional analysis. Experience has also revealed some styles successfully inherited from the general scientific store. Examples are the two styles noted above: called here categorical taxonomy and empirical investigation. These three styles and another called facet generalisation are described below.

Categorical taxonomy

An important part of building the ontological model is unearthing the underlying taxonomy. As noted earlier, this has been a standard approach since Aristotle. It involves identifying the major categories and organising the various entities into hierarchies under them. As a matter of principle, every entity has to fall into one or other of the categories. And care must be taken to ensure that it is placed into the correct category – that its ontological type is accurately identified.

⁴⁹ Book II, Chapter xxvii, 1 – XXVII – Of identity and diversity – "... When we see any thing to be in any place in any instant of time, we are sure, (be it what it will) that it is that very thing, and not another, which at that same time exists in another place, how like and indistinguishable soever it may be in all other respects: ... [O]ne thing cannot have two beginnings of Existence, nor two things one beginning, it being impossible for two things of the same kind, to be or exist in the same instant, in the very same place; or one and the same thing in different places. That therefore that had one beginning is the same thing, and that which had a different beginning in time and place from that, is not the same but divers."

⁵⁰ See, in particular, Ch.12 "*Style*" for historians and philosophers, where this analysis is referred to as styles of reasoning.

The CEO Project: An Introduction

Faceted generalisation

Facet generalisation is a key style delivering both simplification and generalisation. Aristotle's hierarchical analysis, and much taxonomic analysis since has used what might be called abstraction generalisation. This groups together lower level types under a single more general type – which is characterised by the common qualities of its sub-types. In this approach, there is a measure of generalisation, but the lower level types remain indispensable as they deal with the more specific characteristics.

In faceted generalisation, the qualities of the lower level types are analysed for general patterns with the goal of characterising all of them under more general types – known as the lower level type's facets. Typically a smaller, simpler group of these higher level types/facets fully characterises the lower level types, giving a simpler, more general picture, and rendering them, from one perspective, redundant. (Partridge 1996) calls this compacting and has several worked examples illustrating how it works.

Extensional analysis

As a result of the meta-ontological choice of extensionalism, the extensional analysis of objects corresponds to an analysis of identity both partial and total. This turns out to be a particularly fruitful style. At the level of elements, the analysis maps out their extension in relation to other elements, showing inclusion, overlap and disjointness. A useful technique has been developed for modelling the relationships between spatio-temporal extensions of elements called space-time maps⁵¹ - and it is anticipated this will be used extensively in the CEO's analysis. At the level of types, the extensional analysis corresponds to a taxonomic analysis: for example, type inclusion (partial identity) corresponds to the taxonomic sub-type relation⁵².

Empirical investigation

The CEO ontological analysis has a strong empirical bent. It typically starts with a close look at individual elements. As new general patterns are discovered, they are tested against their instances.

Even the top categorical levels are subject to empirical tests. An important aspect of the meta-ontological choices is that they, typically, cannot be verified or falsified empirically. One can think of these as ways in which the world is that, unlike empirical scientific claims, cannot be directly tested. Or, from another perspective, as making choices about how we organise what we know about the world. This may seem to suggest that we can make the choices without considering the way the world is. But this is not so. The results of the choices are judged by how well they actually organise the world – on the basis of the values described earlier.

The core ontology's general patterns can be tested empirically. Experience has shown that the re-engineering of existing large applications provides a good mechanism for doing this. The data in the application system is a useful source of 'observations', which can be used to check the 'theory' embedded in the general pattern. Running the data against a proposed general pattern often shows up faults that are not easily found by human inspection. The large volume of data available makes it likely that it will

⁵¹ For more on these see pp.179-80 of Partridge (1996) *Business Objects: Re - Engineering for re - use*, which has an explanation and examples.

⁵² The extensional perspective on the type taxonomy is described in Lewis (1991) *Parts of classes*.

The CEO Project: An Introduction

contain the degenerate and pathological situations that are often forgotten in the development of general patterns.

Interlinked styles and choices

As noted at the beginning of the section, the styles and choices are interlinked and mutually supporting. For example, the general meta-ontological concern with identity is linked to the metaphysical choice of extensionalism, which is linked to the style extensional analysis. These also bear a close relation to the style categorical taxonomy as, given the metaphysical choice of identity extensionalism, its ordering can be seen as delineating the mereology of the types involved – a sub-type is partially identical to its super-type as it shares some members.

Potential applications

The CEO has a range of applications – the most obvious being in the area of semantic interoperability and consistency.

The business models embedded in enterprise applications typically represent the domain of the application from the point of view of that part of the enterprise using the application. The CEO provides a toolkit for developing reference models that integrate business models from a variety of applications within and across enterprises.

The CEO can also be used to help ensure the semantic consistency of business models during their development. Consider, for example, a large modular development, where the team working on each module develops its own domain ontological model. Then the CEO can be used to ensure that the business semantics of the different modules are developed in a way that is semantically consistent – and so easily interoperable. Similarly, if two teams are working independently on applications in the same or related domains – if they use the CEO as the foundation of their domain models, this will guarantee a high level of semantic consistency – and so straightforward interoperability.

Once an ontological model has been established for an enterprise domain, as it is application and technology independent, and so can be used in many ways. It can be (re-)used directly in the specification of business requirements for new applications and the re-development of existing ones. Using the model will ensure high levels of semantic consistency between the application and other applications using the same ontological model.

For existing (legacy) applications, it can be used in the specifications for exchanging information with other systems. It could, for example, provide a common business language for the middleware integrating applications and the people using the middleware. Using it helps to ensure semantic consistency – and so interoperability.

The CEO also has a part to play during deployment of applications. It can help to ensure operational data is semantically consistent. For example, that different people at different times using different systems will classify the same entities in the same way – and re-identify the entities as the same when they re-encounter them. This consistent classification and re-identification is essential to interoperability.

The next steps

The paper describes the context within which the CEO work will be undertaken. The next step is start the synthesis stage, in particular, the Synthesis of a TOVE Person Ontology (STPO) task.

The CEO Project: An Introduction

Acknowledgements

This report was produced while on sabbatical leave at the Group of Conceptual Modeling and Knowledge Engineering at CNR-LADSEB, Padova. I would like to thank them for their support. I would also like to thank Milena Stefanova, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Bob Colomb for the numerous fruitful discussions we had on topics related to enterprise ontologies.

The CEO Project: An Introduction

Appendix A: The CEO Starter Top Ontology

The final CEO deliverables will include a top ontology as a foundation for the core ontology. To kickstart the synthesis stage the following starter top ontology has been developed – based upon that used for the REV-ENG analysis⁵³. As this is only a starter ontology, it is not intended to be completely rigorous or final – merely adequate to support the initial synthesis. In the process of developing the CEO, it is anticipated that it will evolve.

The basic elements of the framework

The most basic categories within the ontology are types and elements⁵⁴. These are connected by the formal relation, *instance_of*. An example of a type would be ‘dog’ and an example of an element would be ‘Fido’. An example of an *instance_of* relation would be ‘Fido is an *instance_of* a dog’.

Elements and types are disjoint and an element is typically an *instance_of* a type but cannot have instances, whereas types will typically have a number of instances. This gives rise to a classical characterisation of types as objects that can have instances, and elements as objects that cannot – but can be instances of other things.

This does not exhaust the list of categories. Entities such as relations and states of affairs need to be accommodated. However, the CEO is not yet committing to a particular way of treating these⁵⁵ – its decision will be dictated by the practical needs of the core ontology. When the final CEO is issued, it will contain a reasonably complete top ontology – including all the relevant categories of existence for the objects in the core enterprise ontology.

Identity conditions

As noted earlier, the CEO is following a strategy of having criteria of identity for the top categories. These have already been given for the two basic categories – in the section on identity extensionalism. The identity for the *instance_of* relation is dependent upon its two relata and the role they play in the relation: in other words, if two descriptions refer to *instance_of* relations with the same relata in the same roles – they refer to the same relation.

Types, elements and instances of

It is good practice to explicitly introduce the top categories into the ontology and also to show their relationship with their instances. So the top ontology includes the categories Type and Element and a formal relation of *instance_of* between them. All objects that are types are *instances_of* the category Type and all objects that are elements are *instances_of* the category Element.

⁵³ Which is described in Part 4 of Partridge (1996) *Business Objects: Re - Engineering for re - use*.

⁵⁴ Types are also sometimes called universals properties or classes. Elements are also sometimes called particulars. The CEO has chosen the names, Type and Element because they seem to come with the least ontological baggage.

⁵⁵ Partridge (1996) *Business Objects: Re - Engineering for re - use* pp. 159-60 (Ch.7 §6) offers a logician’s answer to the nature of relations.

The CEO Project: An Introduction

Some relevant structural options

Within this framework there are a couple of key structural options:

Higher order types, whether:

- only elements are allowed to be instances_of types, or
- types (instances_of Types) can be as well.

Multiple instantiation (also known as multiple classification), whether:

- an instance_of a type can only be an instance_of one type, or
- is allowed to be an instance_of more than one type.

The CEO selects the second option in both cases for the simple practical reason that it needs to make use of these more ‘expressive’ structures. People typically make use of these, as the following examples illustrate.

Types as instances of types

For example, people say that the rose’s colour is an instance_of red, and red (a type) is an instance_of colour. Or George is (an instance_of) an accountant, and accountant (a type) is (an instance_of) a profession. Colour and profession, whose instances are types, are often called higher-order types.

One way of describing this is to introduce a higher-level category ‘Object’ – which is the union of Type and Element – and regard the relational category instance_of as relating instances_of Type and Object.

Instances of multiple types

Similarly, we typically classify things under multiple types⁵⁶ – a ram is classified as both ‘Sheep’ and ‘Male’. Furthermore, enforcing a classification under a single type – such as Ram – rather than multiple types adds unnecessary complexity with no apparent consequential benefits⁵⁷. An obvious related point, but worth noting, is that types typically have a number of instances.

Other not-so-relevant structural options

There are some other structural options for types that (given the CEO’s practical goals) do not need to be exercised, but are worth being aware of.

Are Type and Element types?

A naïve commonsense view would be that Type, Element and instance_of are types – as they have instances. It might seem that they are all instances of some meta-Type. However, this raises the question of whether this meta-Type is also a type – and the possibility of an endless regress. A practical solution may seem to be to regard them as simply instances_of Type – making Type is an instance_of itself. However, unless this is formulated carefully, we are faced with Russell’s paradox. The CEO elects to leave the matter open for now, as it seems to be a technical and theoretical matter, with no practical consequences from the CEO perspective.

⁵⁶ Many computing programming languages do not have this expressiveness – they only allow ‘single classification’ – for more see Ibid. pp. 81-2 – Ch.4 §4.3.1.

⁵⁷ This difficulty is discussed in more detail in Ibid. pp. 81-2 – Ch.4 §4.3.1.

The CEO Project: An Introduction

Two other structural options

Two other not-so-relevant structural options are:

- Whether to insist that all the instances of a type must be either exclusively types or elements, or allow mixtures? Under the first option, types would have to be either element types (that only have elements as instances) or type types (that only have types as instances). It is hard to make a principled case for a restriction like this – and so for now the CEO allows the second option.
- Whether there are any restrictions on what collections of elements or types can be the full set of instances of a type? In particular, if there were no restrictions, then there could be a type corresponding to every collection of entities? This is a more difficult question to answer – in particular, to identify what restriction one might want to apply⁵⁸. The CEO applies no restriction at this stage.

It is neither clear how to deal with these restrictions simply nor anticipated that there is a pressing practical need to do so. Hence the CEO does not attempt to deal with them at this stage. Of course, if the analysis should reveal that it is practically useful to take a firm position on any of these, the CEO will do so.

The Framework for the Synthesis

As well as these basic categories of existence, there is a general ontological framework based upon three structural hierarchies into which objects must fit: these are:

- Typonomy⁵⁹,
- Taxonomy,
- Partonomy (or mereonomy).

Each of these is built from a basic formal relation: Typonomy is generated by the instance_of relation, Taxonomy is generated by the sub-type_of relation and Partonomy is generated by the part_of relation. Hence the alternative names for them:

- Type-instance hierarchy,
- Super-sub-type hierarchy, and
- Whole-part hierarchy.

It turns out to be good analytic practice to ensure every object has a place within these hierarchies. A good strategy for this is to determine the main ontological categories that occupy the highest level of each of these hierarchies. This makes it simpler to consider the structure from the top down.

Typonomy and the instance-type relation

Instances instantiate types – instances are in an instance-type (or instance_of) relation with types. For example, my pet Fido is an instance_of the type dogs. Typonomy is the hierarchy created by this instance-type relation. At the base of the hierarchy are

⁵⁸ This is, in essence, a difficult problem in philosophy that David Lewis has characterised in terms of sparse and abundant properties (restricted and unrestricted types). See Lewis (1997) *New Work for a Theory of Universals* and Armstrong (1997a) *Properties* in Mellor and Oliver (1997) *Properties*.

⁵⁹ There is no obvious or established name for a type-instance hierarchy (unlike taxonomy and partonomy). Typonomy's main merit is that it is less of a mouthful than 'instanceconomy' or 'memberonomy'.

The CEO Project: An Introduction

elements that cannot be instantiated, and at the higher levels types that can. At the top of the hierarchy are the categories *Type*⁶⁰ and *Element*.

As a general rule, the ontological model should show how every object fits into the typonomy. The two roots of the hierarchy are the basic categories *Type* and *Element*, and, as a basic first step, the model needs to show, for every object, which category it is an instantiation of. As far as possible, it also needs to show all the other instance_of relations – this typically yields a rich hierarchy.

Obviously, it is impossible to show all the instances_of most types – as there are typically an infinite number. So the most common use of typonomy is to characterise types by providing well-known, prototypical examples of their instances. Another important use is to provide examples that illustrate pathological cases. Taking advantage of the fact that we can draw on a common experience, often proper names of the instances are adequate for a common recognition (for example, ‘George Bush, the President of the United States’ would be easily recognisable). Specifying the prototypical instances helps to guarantee the absence of radical differences of interpretation – by ensuring that different people include some of the same instances under the ‘same’ type – and so cover at least some of the same territory.

Taxonomy and the super-sub-type relation

When faced with a specific object, people naturally see it as an instance_of a type. They also consider both the more general types of which it is a sub-type_of as well as the more specific ones that it has as sub-types. So, for example, we see something and recognise it as a dog. We remember that dog is a sub-type_of animal. We also probably look for a more specific type to characterise the dog – saying, for example, that it is an Alsatian. In other words, that Alsatian is a sub-type_of dog. This sub-type_of relation creates a hierarchy of types called a taxonomy – ordered by the sub-type_of (or super-sub-type) relation.

In line with its general strategy, the CEO assumes that this sub-type_of relation is extensional. In other words, saying that type A is a sub-type_of type B means that all possible instances_of type B are also instances_of of type A. And vice versa, if all possible instances_of type B are also instantiations of type A, A is a sub-type_of B. For example, ‘dog is a sub-type_of animal’ means every possible instance_of a dog is also an instance_of of an animal. And if every possible instance_of a dog is also an instance_of of an animal, then dog is a sub-type_of animal. As with the instance_of relation, the identity of the sub-type_of relation is dependent upon its two relata and their roles.

In general, every type should appear in the taxonomy. This example, and most taxonomies, only deal with ‘first-order’ types – types that only have elements as instances. These ‘first-order’ types should appear in the part of taxonomy for types of elements which has at its top the category *Element*. Of course, it is possible to have higher-order types and each of these will have a top object. The parts of the taxonomy for the various orders may overlap where they contain types that have instances of different orders.

⁶⁰ As noted earlier, Russell’s paradox raises a moot theoretical point about whether *Type* should be considered as a type, and what this means. Should, for example, *Type* be an instance_of itself or *Element* an instance_of *Type*. From a practical point of view, this problem can be left to the theoreticians. For simplicity’s sake, the CEO does not consider *Type* or *Element* to be instances_of *Type*.

The CEO Project: An Introduction

For the ontological analysis, it is important to extend the non-categorical part of the taxonomy both upward and downwards. Extending upwards will identify the very general types that can be used to harmonise most, if not all, models. Extending downwards should introduce sufficiently common specific types, that people have no difficulty in supplying very similar prototypical instances. This enables the taxonomy to draw on their common experience, giving a degree of confidence that different people will end up with at least similar interpretations.

Partonomy and the whole-part relation

Wholes are composed of parts – the whole has a whole-part (or part_of) relation with the part. Partonomy, in the first instance, is the hierarchy created by the whole-part relations between elements: for example, Fido's foot (an element) is part_of his leg (another element) and Fido's leg is part_of his body. The part_of relation is transitive, so Fido's foot is part_of his body. The top of this hierarchy would be the universe – the full extent of space-time.

However general whole-part relations between types of wholes and parts are more useful, and the partonomy normally deals in these. These are the result of generalising to the types that the elements instantiate – in this example, we would generalise to 'dog's bodies typically have legs as parts – and the legs typically have feet as parts'.

The reason for picking out partonomy is its important explanatory role. Explaining what something is often involves talking about how it is composed, about its parts – for example, we describe a dog in terms of legs, tail, ears, etc.

The partonomy is related to the taxonomy of relations. There is a general part_of relation of which the whole-part relation between animals' body parts is a sub-type and the whole-part relation between dogs' body parts is sub-type of that.

Generalising Taxonomy and Partonomy

As a number of people have noticed, these last two hierarchy building relations can be generalised. One can look at sub-type_of over types and part_of over elements as just specific types of a more general mereological sub-part_of relation⁶¹. David Armstrong⁶² explains this as both relations being partial identity – and mereology as the study of partial identity. So a wheel is part_of a car because it is partially identical to the car. Similarly the type human is a sub-type_of the type animal because it is partially identical to it.

As Armstrong notes this naturally leads from consideration of part_of to related mereological relations such as overlapping. If being a part_of (and sub-part_of) is a kind of partial identity so is overlapping⁶³. This is a useful insight. In practice, both taxonomies and partonomies need to take account of overlapping and disjointness. It is extremely useful to be able to say that a car's wheel is disjoint from the dashboard – and that dogs are not cats. And Armstrong's explanation explains why overlapping and sub-part_of seem intuitively to be dealing with the same kind of relation.

⁶¹ This is, for example, the main subject of Lewis (1991) *Parts of classes* and also examined in detail Partridge (1996) *Business Objects: Re - Engineering for re - use*, see pp. 223-4 (Ch.10 §5.1), where it is called the sub-part relation.

⁶² In various places in Armstrong (1997b) *A world of states of affairs*, e.g. pp.17-18. The suggestion is supported in Partridge (1996) *Business Objects: Re - Engineering for re - use*.

⁶³ Where this generalised mereological overlapping can apply to both elements and types.

The CEO Project: An Introduction

The generalisation is useful in revealing the underlying categories. However from an analysis methodology's point of view, the distinction between part_of and sub-type_of is crucial.

The ontological heterarchy

These hierarchies overlap. For example, the type dog appears in the examples for all three hierarchies. They also complement each other. For example, the typonomy's proto-typical instances help to illustrate the types in the taxonomy and partonomy. This overlapping and complementing helps to unify the overall ontology. The final ontology should be a network of connected hierarchies – a kind of heterarchy.

Building the ontological heterarchy

This starter top ontology provides the framework within which the CEO's analysis will build its ontological heterarchy. As noted at the beginning, it is a starter ontology and it will be refined in the light of the needs of the emerging core ontology.

The CEO Project: An Introduction

References

- Aristotle The categories.
- Aristotle The metaphysics.
- Armstrong, D. M. (1997a). Properties. Properties. D. H. Mellor and A. Oliver. Oxford ; New York, Oxford University Press: 160-172.
- Armstrong, D. M. (1997b). A world of states of affairs. Cambridge ; New York, Cambridge University Press.
- Bealer, G. (1982). Quality and concept. Oxford New York, Clarendon Press ; Oxford University Press.
- Black, M. (1937). "Vagueness: an exercise in logical analysis." Philosophy of Science(4): 427-55.
- Breuker, J., A. Valente and R. G. F. Winkels (1997). "Legal Ontologies: A Functional View." Proceedings of the First International Workshop on Legal Ontologies (LEGONT'97).
- Chomsky, N. (1975). Reflections on language. New York, Pantheon Books.
- Collingwood, R. G. (1940). An essay on metaphysics. Oxford, Eng., Clarendon Press.
- Crombie, A. C. (1994). Styles of scientific thinking in the European tradition. London, Duckworth.
- CYC:http "CYC - <http://www.cyc.com/publications.html>."
- Dummett, M. (1991). The Logical Basis of Metaphysics. London, Gerald Duckworth & Company Ltd.
- EO:http "EO - <http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html>."
- Fox, M. S., M. Barbuceanu and M. Gruninger (1996). "An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour." Computers in Industry **Vol. 29**: pp. 123-134.
- Fox, M. S., J. Chionglo and F. Fadel (1993). A Common-Sense Model of the Enterprise. Proceedings of the Industrial Engineering Research Conference.
- Gilbert, M. (1992). On social facts. Princeton, N.J., Princeton University Press.
- Gomez-Perez, A., J. P. Martins and H. Sofia Pinto (1999). Some Issues on Ontology Integration. IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden.
- Gruber, T. R. (1993). "A translation approach to portable ontology specifications." Knowledge acquisition **Vol. 5**(No. 2): pp. 199-220 (doi:10.1006/knac.1993.1008).
- Hacking, I. (2002). Historical ontology. Cambridge, Mass., Harvard University Press.
- Hay, D. C. (1996). Data model patterns : conventions of thought. New York, Dorset House Pub.
- Hoffman, J. and G. S. Rosenkrantz (1994). Substance among other categories. Cambridge ; New York, Cambridge University Press.
- Ingarden, R. (1964). Der Streit um die Existenz der Welt. 1. Existentialontologie. Tübingen, M. Niemeyer.
- Inmon, W. H., L. Silverston and K. Graziano (1997). The data model resource book : a library of logical data models and data warehouse designs. New York, Wiley.
- Kent, W. (1978). Data and reality : basic assumptions in data processing reconsidered. Amsterdam ; New York New York, North-Holland Pub. Co. ; sole distributors for the U.S.A. and Canada Elsevier/North-Holland.
- Körner, S. (1970). Categorical frameworks. Oxford,, Blackwell.

The CEO Project: An Introduction

- Kripke, S. A. (1982). Wittgenstein on rules and private language : an elementary exposition. Cambridge, Mass., Harvard University Press.
- Kuhn, T. S. (1970). The structure of scientific revolutions. Chicago,, University of Chicago Press.
- Lenat, D. B. and R. V. Guha (1990). Building large knowledge-based systems : representation and inference in the Cyc project. Reading, Mass., Addison-Wesley Pub. Co.
- Lewis, D. K. (1986). On the plurality of worlds. Oxford, UK ; New York, NY, B. Blackwell.
- Lewis, D. K. (1991). Parts of classes. Oxford, UK ; Cambridge, Mass., B. Blackwell.
- Lewis, D. K. (1997). New Work for a Theory of Universals. Properties. D. H. Mellor and A. Oliver. Oxford ; New York, Oxford University Press: 188-227.
- Locke, J. (1690). An essay concerning human understanding. London, T. Basset.
- Marcus, R. B. (1993). Modalities : philosophical essays. New York, Oxford University Press.
- Mealy, G. H. (1967). "Another Look at Data." Proceeding of AFIPS 1967 Fall Joint Computer Conference Vol. 31.
- Mellor, D. H. and A. Oliver (1997). Properties. Oxford ; New York, Oxford University Press.
- Mill, J. S. (1848). A system of logic, ratiocinative and inductive; being a connected view of the principles of evidence and the methods of scientific investigation. New York,, Harper & Brothers.
- Nerlich, G. (1994). The shape of space. Cambridge ; New York, Cambridge University Press.
- Nussbaum, M. C. (2001). The fragility of goodness : luck and ethics in Greek tragedy and philosophy. Cambridge, U.K. ; New York, Cambridge University Press.
- Papazoglou, M., S. Spaccapietra and Z. Tari (2000). Advances in object-oriented data modeling. Cambridge, Mass., MIT Press.
- Parent, C. and S. Spaccapietra (2000). Database Integration: The Key to Data Interoperability. Advances in ObjectOriented Data Modeling. S. Spaccapietra. Cambridge, Mass., MIT Press.
- Partridge, C. (1996). Business Objects: Re - Engineering for re - use. Oxford, Butterworth Heineman.
- Partridge, C. (2002a). LADSEB-CNR - Technical report 04/02 - What is pump facility PF101? Padova, The BORO Program, LADSEB CNR, Italy.
- Partridge, C. (2002b). LADSEB-CNR - Technical report 05/02 - The Role of Ontology in Integrating Semantically Heterogeneous Databases. Padova, The BORO Program, LADSEB CNR, Italy.
- Partridge, C. (2002c). LADSEB-CNR - Technical report 06/02 - Note: A Couple of Meta-Ontological Choices for Ontological Architectures. Padova, The BORO Program, LADSEB CNR, Italy.
- Partridge, C. (2002d). LADSEB-CNR - Technical report 08/02 - STPO - A Synthesis of a TOVE Persons Ontology (forthcoming). Padova, The BORO Program, LADSEB CNR, Italy.
- Partridge, C. (2002e). The Role of Ontology in Semantic Integration. Second International Workshop on Semantics of Enterprise Integration at OOPSLA 2002, Seattle.
- Partridge, C. (2002f). What is a customer? The beginnings of a reference ontology for customer. 11th OOPSLA Workshop on behavioral semantics, Seattle, Washington, Northeastern.

The CEO Project: An Introduction

- Partridge, C. and M. Stefanova (2001). A Synthesis of State of the Art Enterprise Ontologies: Lessons Learned. Open Enterprise Solutions: Systems, Experiences, and Organizations (OES-SEO 2001). A. D'Atri, A. Solvberg and L. Willcocks. Rome, Luiss Edizioni, Centro di Ricerca sui Sistemi Informativi: 130-133.
- Quine, W. V. (1948). "On what there is." Review of Metaphysics II(5).
- Quine, W. V. (1964). Word and object. Cambridge, MIT Press.
- Quine, W. V. (1969). Ontological relativity, and other essays. New York,, Columbia University Press.
- Quine, W. V. (1973). The roots of reference. LaSalle, Ill., Open Court.
- Quine, W. V. (1980). From a logical point of view : 9 logico-philosophical essays. Cambridge, Mass., Harvard University Press.
- Russell, B. (1919). Introduction to mathematical philosophy. London,, G. Allen and Unwin.
- Russell, B. (1923). "Vagueness." Australasian Journal of Philosophy and Psychology 1: 84-92.
- Russell, B. and K. Blackwell (1983). The Collected Papers of Bertrand Russell. London ; Boston, G. Allen & Unwin.
- Searle, J. R. (1995). The construction of social reality. New York, Free Press.
- Sheth, A. and J. Larson (1990). "Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases." ACM Computing Surveys 22(3): 183 - 236.
- Silverston, L. (2001a). The data model resource book 1. New York, John Wiley.
- Silverston, L. (2001b). The data model resource book 2. New York, John Wiley.
- Simons, P. M. (1987). Parts : a study in ontology. Oxford New York, Clarendon Press ; Oxford University Press.
- Sowa, J. F. (2000). Knowledge representation : logical, philosophical, and computational foundations. Pacific Grove, Brooks/Cole.
- Strawson, P. F. (1992). Analysis and metaphysics : an introduction to philosophy. Oxford ; New York, Oxford University Press.
- Thomasson, A. L. (1999). Fiction and metaphysics. Cambridge, U.K. ; New York, Cambridge University Press.
- TOVE:[http "TOVE - http://www.eil.utoronto.ca/tove/."](http://www.eil.utoronto.ca/tove/)
- Tsichritzis, D. and A. Klug (1978). The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management., AFIPS Press.
- Uschold, M., M. King, S. Moralee and Y. Zorgios (1997). The Enterprise Ontology, AIAI, The University of Edinburgh.
- Uschold, M., M. King, S. Moralee and Y. Zorgios (1998). "The Enterprise Ontology." The Knowledge Engineering Review Vol. 13.
- Van Inwagen, P. (2001). Ontology, identity, and modality : essays in metaphysics. Cambridge, U.K. ; New York, Cambridge University Press.
- Vermeer, M. W. W. and P. M. G. Apers (1996). "On the Applicability of Schema Integration Techniques to Database Interoperation." ER 1996: 179-194.
- Williams, D. C. (1966). Principles of empirical realism; philosophical essays. Springfield, Ill., C.C. Thomas.
- Williamson, T. (1996). Vagueness. New York, Routledge.
- Wittgenstein, L. (1953). Philosophical investigations. Oxford,, B. Blackwell.
- Wright, C. (1987). Realism, meaning, and truth. Oxford, UK ; New York, NY, B. Blackwell.