Ontology Representation : design patterns and ontologies that make sense

Hoekstra, R.J.

Publication date
2009

Chapter 4

Ontologies

> "We now begin the science of the properties of all things in general, which is called ontology. (...) One easily comprehends that it will contain nothing but all basic concepts and basic propositions of our a priory cognition in general: for if it is to consider the properties of all things, then is has as an object nothing but a thing in general, i.e. every object of thought, thus no determinate object."
>
> *M. Immanuel Kant (1782–1783)*

4.1 Introduction

In Chapter 2, the term 'ontology' was introduced as a moniker for the domain theory of an expert system (Davis et al., 1993, and Section 2.3.2). The functional approach of Levesque (1984) brought us to consider description logics languages as ideal candidates for the representation of these domain theories (Section 2.5.1), and Chapter 3 described a particular member of this language family, the Web Ontology Language, for representing knowledge on the Semantic Web.

For the sake of simplicity, we assumed that the notions of domain theory and ontology stand in direct correspondence. However, this is not the case and despite its success, the term 'ontology' has remained a rather ungainly characterisation of the things it is used to denote. A large number of academic publications revolve around a particular ontology, some ontology language, a methodology for ontology building or a discussion of different *kinds* of ontologies. An invariably significant portion of these papers include some definition of what (an) ontology *is*. Most cited in this context is the definition of Gruber (1993, 1994):

> "An ontology is an explicit specification of a conceptualisation."
>
> Gruber (1994)

The apparent convergence between different fields in AI on what an ontology *is*, does not reach very far beyond the superficiality of Gruber's definition. Arguably, taken on its own, a definition is not very helpful. In fact, there are uncountable alternative definitions of ontology, that are equally uninformative when taken out of context (cf. Section 4.4). The problem is, definitions are rather imprecise – e.g. what then is a conceptualisation? – and hide the rationale and perspective that underpin the definition. Perhaps for this reason, the definition of ontology as proposed by one researcher is often subject to heavy criticism from others who have a different background. Definitions can be widely divergent, and ontologies can range from lightweight textual descriptions of some terms to highly formal specifications of philosophical primitives.

It is not wholly inconceivable that the longstanding and still prevalent custom of including Gruber's definition in scholarly articles is a serious indication that we *still don't really know what an ontology is*. Perhaps we don't *want* to know, or at least keep up the appearance that we know what we are doing. As AI is very much an interdisciplinary field, this rather careless attitude has a detrimental effect on the overall quality of 'ontologies' produced in the field – at least when seen from the knowledge representation perspective of the preceding chapters.

4.2 Ontologies as Artefacts

McCarthy (1980) first borrowed the term 'ontology' from philosophy to refer to the things that exist in a description of (all) commonsense knowledge. The perspective of philosophy fit well with McCarthy's position that knowledge in an intelligent agent should be based on a small number of principles (see Section 2.2.1). Nonetheless, the term remained only spuriously used in AI until it was adopted by the knowledge acquisition community. And this time, it had quite a different ring to it. No longer it was used to refer to *the* theory of existence, but rather as reflection of the building blocks of a domain theory: *concepts*. Ontologies soon grew into knowledge representation artefacts in their own right.[1]

As we have seen in Chapter 2, separating problem solving knowledge from domain knowledge in knowledge based systems has proven to be a fruitful means to circumvent the interaction problem of Bylander and Chandrasekaran (1987) and improve reusability of knowledge components. Originally, this separation was not intended to exist physically inside a knowledge based system, but rather, the two types should be *modelled* separately. Because a domain model constrains that which exists for a knowledge based system and what it can reason over, it can be said to capture an ontology (Davis et al., 1993). In this

[1]In the following I will use the term *Ontology*, with a capital 'O', to denote the philosophical discipline, and *ontology* to refer to a (formal) construct reflecting some ontological commitments. The word 'ontological' in that sense means 'pertaining to existence'; an ontological *commitment* is a commitment to the existence of something, ontological *status* is some degree of certainty by which an entity is thought to exist.

view, reflected by the definitions of Gruber (1994) and Schreiber et al. (1995), an ontology is part of the *specification* of a knowledge based system:

> An *ontology* is an explicit, partial specification of a conceptualisation that is expressible as a meta-level viewpoint on a set of possible domain theories for the purpose of modular design, redesign and reuse of knowledge-intensive system components.
>
> Schreiber et al. (1995)

Initially ontologies were merely a novel term for the specification of domain knowledge in the form of documentation, schematic diagrams and textual descriptions akin to specifications in software engineering. It is this type of specification that Gruber meant. To emphasise this perspective, he referred to the specification of ontologies as *ontology engineering*.

Schreiber et al. and Gruber consider the ontology as a necessary step in the design of a system; it can be said to be *implemented* in a system. In exactly the same way that problem solving methods are abstract reusable descriptions of *reasoning*, an ontology enables reuse by providing an abstract description of some *domain*. Ontologies can be consulted when selecting a 'knowledge component' that implements some required reasoning services, or when developing a *new* system or component that re-implements that body of knowledge.

Furthermore, an ontology can help guide knowledge acquisition for a domain by providing a conceptual 'coat rack' to which new knowledge can be added. To give an example, a general expert system for medical diagnosis needs to implement (at a minimum) both the standard diagnosis PSM of Figure 2.9 and an ontology of the human physiology. A more specialised expert system could implement a more specific diagnosis PSM, e.g. a causal-dependency based approach (Bredeweg, 1994), or implement a liver disease ontology that extends the physiology ontology. Both the ontologies and the PSMs are not part of the system itself, but belong to its specification. A knowledge component that implements an ontology can be said to commit to that ontology. And different components that commit to the same ontology are more compatible than those committing to distinct ontologies.

The specification perspective on knowledge gradually grew in importance and it was soon recognised that the ontology – as rather abstract specification which is not part of the system itself – can also serve as a means to *communicate* the expertise of not just components, but of a system as a whole. It enables knowledge sharing across both systems and between systems and people (Neches et al., 1991; Uschold, 1996).

In fact, the techniques of knowledge acquisition were increasingly applied to share knowledge between different groups of people: as techniques for *knowledge management* in organisations (van Heijst et al., 1997; Schreiber et al., 2000). The notions of task decomposition and problem solving methods were very useful in the elicitation of organisational goals and business processes. Domain theories could capture the individual expert knowledge of employees, and thereby chart the distribution of expertise over a workforce. This overview was used to signal lacunae, mismatches and overlap in expertise of both persons and organisational units. The elicitation and *alignment* of domain theories as a *shared vocabulary* was deemed an especially powerful tool for improving the cohesion and co-operation within and between organisations.

Gruber (1993) takes very much the position that ontologies are essentially about *sharing* conceptualisations:

> *Ontologies* are agreements about shared conceptualisations. Shared conceptualisations include conceptual frameworks for modelling domain knowledge; content-specific protocols for communication among inter-operating agents; and agreements about the representation of particular domain theories.
>
> Gruber (1993)

Ontology is thus abstracted away from its initial purpose in knowledge acquisition, and is more about defining some commonly agreed upon vocabulary of terms for the purposes of *standardisation*. Rather than what one might initially expect, this more abstract view on ontologies puts a lot more weight on their development process and methodology. Anyone who has ever been involved in a standardisation committee will know that the interplay between different parties, with different interests can make the development of the standard a cumbersome process. It requires one to be a lot more explicit as to what the intended (or even allowed) use of an ontology is. A non-standard use of some concept definition may influence its interpretation in such a way that it no longer complies with the intended meaning of a term. If an ontology really is about the sharing of a standardised vocabulary, some rules need to be set out that prevent misuse:

> An *ontology* defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.
>
> Neches et al. (1991)

The emphasis on sharing and standardisation sparked interest in three interdependent directions: to ensure quality through a methodological foothold, to enable the physical sharing of ontologies and to facilitate (formal) specification (Neches et al., 1991). The growing importance of ontologies introduced a need for quality assurance; to safeguard extensibility and shareability by formulating a methodology that enforces general design principles and ensures a repeatable development process (see Chapter 5).

Commitment to a shared vocabulary is only possible if the vocabulary itself can be physically shared as well: ontologies should be *portable*. Portability is ensured by using a common specification language. Such a language provides a *syntactic entity* that can be manipulated, copied and referred to. The ONTOLINGUA system of (Gruber, 1993; Farquhar et al., 1997) is an online editor and library of ontologies, similar to the library of problem solving methods of e.g. Breuker (1994, 1997), that supports the storage of "ontologies that are portable over representation systems" (Gruber, 1993, p.1). Although there is no prescribed level of formality of vocabulary specifications (Uschold, 1996, p.6), direct ontology sharing between systems requires at least a structured language with clear semantics. In particular where *knowledge modelling* ontologies are concerned (van Heijst et al., 1997).

It was clear that a structured specification language for ontologies would increase their portability. However, the *purpose* of the specification turned out to play an important role in determining what a useful specification is.

Knowledge Representation Ontologies

The knowledge representation community took the specification to mean a *formal* specification that could be used to constrain valid implementations of the ontology. A formal language can be used to ensure *internal consistency* of vocabulary definitions in the ontology; it may sanction *proper extensions* – incorrect extensions of the ontology are inconsistent; and opens the door to automatic *compliance checking* of implemented knowledge based systems – does the system adequately implement the ontology?

Ontologies in ONTOLINGUA were represented using KIF (Genesereth and Fikes, 1992), a formal, highly expressive language for the explicit representation of concepts and relations. It is purely meant as an *inter lingua* and does not itself support standard inferencing. ONTOLINGUA could directly translate from and to several other representation languages, and supported interaction with dedicated knowledge editors such as Protége using the Open Knowledge Base Connectivity language (OKBC, Chaudhri et al. (1998)). OKBC was an API[2] akin to the contemporary DIG specification[3] for exchanging knowledge bases between a repository and an editor, rather than a knowledge representation language in its own right. Contrary to KIF, OKBC is a language with relatively poor expressiveness.

A language such as KIF can only be used as interchange between relatively compatible knowledge representation formalisms. If some construct from the source language is not available in the target language, or when there is divergence with respect to semantics, translation cannot occur unattended (See Section 5.4). These problems were partially alleviated by the built in *Frame Ontology* of ONTOLINGUA, a representation ontology specified in KIF that defined knowledge representation constructs commonly found in frame-based and object oriented systems. Translation between different languages was only supported through the Frame Ontology. This adoption of the frame language-style of knowledge representation for representing ontologies was perhaps an obvious step, but an influential one at that. Although Gruber presents the Frame Ontology as a mere convenience for ontology engineers over standard predicate calculus, it soon became the default paradigm for specifying AI ontologies.

The Knowledge Representation System Specification of Patel-Schneider and Swartout (1993, KRSS) took this one step further by enforcing a commitment to the frame paradigm. KRSS was developed to establish a common ground between frame-based KL-ONE like knowledge representation languages such as CLASSIC (Brachman et al., 1991) and LOOM (MacGregor and Bates, 1987). Instead of an inter lingua, KRSS is intended as *standard language* for the direct exchange of knowledge representations between such systems. Like KIF KRSS has its own Lisp-style syntax, and was based on the description logics language of e.g. Baader et al. (1991), developed to extend KL-One's formal basis for terminological knowledge representation languages (Baader et al.,

[2]API: Application Programming Interface

[3]DIG: DL Implementation Group, see `http://dl.kr.org/`.

2003, and Section 2.5.1). In this view, the DL language family was thus not just meant for terminological knowledge representation, but for *ontology representation* as well.

Ontologies specified in DL are knowledge representations and can be directly used as knowledge components. At first sight this seems to conflict with the idea that an ontology should be part of the knowledge level specification of a knowledge based system (c.f the quote of Schreiber et al. (1995) on page 67). However, the notion of an ontology as knowledge base does not necessarily imply that it should be incorporated in a knowledge based system *as is*.[4] The knowledge acquisition community was well aware of the developments with respect to description logics, and these languages were certainly not shunned by the more formal minded.

Knowledge Management Ontologies

The knowledge acquisition and knowledge management communities, on the other hand, emphasised a software engineering perspective and adopted the schematic diagrams of object-oriented modelling and design, and later the industry standard *Unified Modelling Language* (UML), to express and specify ontologies. In this view, ontologies are primarily meant for human consumption as part of the design of large scale systems. The *Common*KADS methodology used UML-like diagrams extensively for describing task decompositions, problem solving methods and ontologies alike (Schreiber et al., 2000). This approach has been very successful, as for the first time expertise within organisations could be charted and organised in an intuitive manner. The influence of knowledge management during the nineties has certainly contributed to the increasing popularity of ontologies to describe the domain knowledge of experts.

The Conceptual Modelling Language of Schreiber et al. (1994, CML) and $(ML)^2$ (van Harmelen and Balder, 1992) were proposals for structured languages that could be used for the specification of *Common*KADS models. Contrary to $(ML)^2$, CML did not have a formal semantics, but only provided a structured textual notation and a diagrammatic notation for concepts. However, as knowledge management does not require a full specification of an ontology and ontologies could well be just lists of agreed upon keywords or hierarchies of terms, these languages were only spuriously used.

An important application area for knowledge management is to help organisations deal with the enormous amount of information stored across computer systems. At the end of the nineties, ontologies started to become used to physically *index* the information within organisations. Employees were equipped with *user profiles* that expressed their area of expertise in terms of an ontology. Relevant information that matches the profile could then be brought to the attention of the employee. Because indexing documents by hand is an arduous task, data mining and natural language processing technologies were applied to perform automatic *ontology extraction* and *ontology learning*.

[4] In fact, there are several reasons why this is can be technically problematic, cf. Section 7.2

Ontology Meets the Web

Ontologies were increasingly specified using specialised ontology development tools. Knowledge acquisition tools such as Protégé (Puerta et al., 1992) were adapted for the specification and documentation of taxonomies. As a result, the language in which an ontology was expressed depended more and more on tools. Also, automatically extracted knowledge management ontologies were stored in relatively closed legacy database systems. This turned out to be a significant impediment to reuse, especially considering the growing need for information exchange between distributed information sources over the web.

Most existing initiatives to develop an interchange language for ontologies preceded the development of the web. They were based on the (then prevailing) conception of the web as a relatively slow, but huge local area network, and not the efficient, social, and uncontrollable means for human computer interaction it is today. The growing interest in ontologies sparked a renewed interest in interchange languages, in particular given the possibilities of a new, versatile syntax offered by XML. The SHOE language (Heflin et al., 1999) can be regarded in this light: a simple, frame-based syntactic interchange language for ontologies. Similar lightweight approaches are RDFS and the current SKOS. The DAML-ONT and OIL languages, on the other hand, were more directly influenced by the knowledge representation perspective on ontologies.

The DAML+OIL member submission to the W3C[5] in 2001 was in many ways a package deal that could not be scorned. It offered a full-blown knowledge representation language in the guise of a web-based ontology exchange language. Berners-Lee (1999)'s ideal of a Semantic Web was brought a significant step closer, and once OWL became a W3C recommendation in 2004 it became the de facto representation language for ontologies. However, for those primarily interested in the knowledge *management* aspect of ontologies, the resulting OWL language was somewhat like a Trojan horse: a relatively heavyweight formal language sneaked in via the back door.

Nonetheless, the more informal use of ontologies persists until today, such as in the widespread use of *folksonomies*, and – quite detached from the web – as standard vocabularies for governments and communities of practice. Many of these lightweight knowledge management ontologies are represented using the relatively inexpressive RDFS or SKOS, but a surprisingly large number are in OWL Full as well (though often by accident, Wang and Parsia (2007)).

Since McCarthy (1980) and Davis et al. (1993) borrowed the term 'ontology' from philosophy, the interpretation of the term in AI has shifted from an essential part of the specification of knowledge based systems, to standard vocabularies and full-blown terminological knowledge bases on the web, or even – as we did in Chapter 3 – any OWL file. Nonetheless, not all has been said, as philosophy certainly did not stand by idly while a centuries-old tradition was hijacked by a bunch of computer enthusiasts. The next section describes Ontology as conceived of in philosophy, and Section 4.4 discusses the main differences between the the two views.

[5] See `http://www.w3.org/Submission/2001/12/`

4.3 Ontology in Philosophy

In philosophy, the term Ontology is used in the context of the analysis of the fundamental building blocks of reality, the metaphysical study of existence by first principles: what makes that some thing exists, and how can we ascertain the existence of some thing? As Leibniz put it:

> "Ontology or the science of something and of nothing, of being and not-being, of the thing and the mode of the thing, of substance and accident."
> Gottfried W. Leibniz, in (Leibniz, 1903, p.512)

Ontology thus concerns the top-down deconstruction of reality as we perceive it: eliminate its accidental appearance and reduce it to its very bare bones. If we look at Kant's description of the 'science of ontology', we can conclude that the method adopted in philosophical ontology is to focus primarily on those things objects in the world have *in common*:

> "...ontology, the science, namely, which is concerned with the more general properties of all things."
> Immanuel Kant, in Kant (1997)

It is the commonalities (and disparities) that are the subject of ontological study, and which are used to construct a comprehensive representation of reality. Important also is that it is the study of *general* properties that *all* things have in common, and not of ad-hoc categories. It identifies elements in general which can be applied to account for differences in *particular*. Ontology operates on a meta level with respect to the things in the domain of discourse. For example, instead of studying the properties that make physical entities differ from mental entities, ontology studies what properties are by themselves. This in line with Aristotle's description of Ontology, which, in his sense, tries to answer the question "What is being?", or as Guarino (1997) rephrases it, "What are the features common to all beings?":

> Ontology is the science of being as such: unlike the other sciences, each of which investigates a class of beings and their determinations. Ontology regards "all the species of being *qua* being and the attributes which belong to it *qua* being".
> Aristotle, Metaphysics, IV, 1, from Guarino (1997)

Instead of specifying *a* vocabulary, Ontology thus tries to pinpoint *the* vocabulary used to describe things in the world; it usually adopts realism, i.e. the belief that reality exists independently of human observers. It assumes (or even requires) a direct correspondence between the elements in the ontology and entities 'out there'; and is focused at the *primitives* of being. Consequently, an ontology is to capture *directly* the domain of discourse.[6] The high level of abstraction enables a philosopher to reason a priori with respect to the elements

[6]Usually life, the universe and everything

of some ontological theory. These elements, namely, are considered primitives of human thought and reason.

Of course, the results of this study of existence needs to precipitate in some way; it is unavoidable that ontological research results in an entity embodying some 'ontology'. Formal Ontology is a branch of philosophy that addresses this issue by extending Ontology in two ways (Guarino, 1997), i.e. to:

- Capture ontological theory in a formal language, i.e. first order logic, and
- Study the *forms* and modes of being

Seen in this light, it is understandable that to McCarthy using the term 'ontology' was quite natural. His goal of an ontology as formal representation based on a fixed set of basic principles appears to almost seamlessly correspond to the meaning ascribed to the term in formal Ontology. But the knife cuts both ways: the commitment to a formal specification of an 'ontology' submits formal Ontology to the same restrictions as knowledge representation in AI: the *syntactic* form of an ontology influences the quality of an ontology as a *semantic* entity (Guarino and Giaretta, 1995).

4.3.1 Problems in Formal Ontology: Semantics and Syntax

The acknowledgement that an ontology can never be untainted by formalism and design choices is perhaps the single most prominent difference between the approaches in philosophy and AI. As discussed in Section 2.4.1, the separation of knowledge types was introduced in the first place to remediate known hurdles such as the interaction problem and the knowledge acquisition bottleneck. Although an ontology was conceived as a knowledge level specification of the domain theory of a knowledge-based system (Davis et al., 1993), it was well understood that even this trick would not shield the ontology as such from its context in knowledge based systems.

For a long time, this dependency between representation and language was deemed of no relevance for philosophy, as Ontology was expressed in the tradition of e.g. Leibniz using the "universal language of rational thought": logic.[7] Nonetheless, there even were philosophical arguments against a purely logical approach. According to Smith (1978), for a formal ontologist, even the use of first order logic to precisely and accurately define philosophical convictions poses a threat. The trade-off Smith sketches is between *overshooting*, and possibly allowing entities that have limited ontological status, and possibly *missing out* on important entities, which would diminish the ontological adequacy of a theory as a whole.

The former solution is of course regarded unacceptable by puritan realists. Namely, an ontology that commits to the existence of entities that do not exist is fundamentally flawed. On the other hand, Smith argues that early formal philosophy was caught in a 'perversion' of Occham's razor. His maxim to not multiply entities without *necessity*, was misapplied as a much stricter practice: not to add entities wherever *possible*. It is furthermore fuelled by a combination of reductionism and pragmatism as in e.g. Frege's work where philosophical

[7] Note that although the current language of choice is first order logic, Leibniz' view was primarily computational.

progress is measured "by the degree to which one can 'explain away' apparent philosophical givens in terms of less controversial entities". According to Smith, this perversion leads to a simplification of the world – the subject matter of Ontology. Not as an inevitable by-product of the use of a formal language, but rather due to the application of an overly simple *mathematical* formalisation. For, what evidence is there to expect that the logical constructs devised by Frege to explain and define mathematical theory are equally well suited to capture ontological theory?

Smith illustrates this practice by positing a school of thought by the name of '*fantology*', the idea that ontological form corresponds to one and two-placed predicates of the form $\mathsf{F}a$ and $\mathsf{R}ab$ (Smith, 2005). This view confounds the first role of formal Ontology – to capture ontological theory in a formal language – with the study of 'forms of being', by equating ontological form to logical form. Arguably this is problematic, as this conflation allows the application of common operators of logic such as the Boolean and and or, to ontological categories: ontological truth becomes equivalent to logical truth.

In this conception, the predicate F carries the meaning, whereas the subject a is a 'mere meaningless name, a matter of pure denotation' (Smith, 2005). Although nothing in logic prevents us to ascribe meaning to the subject of a predicate, the prevailing philosophical interpretation is that they refer only to individual objects. Furthermore, the predicates themselves are not ontologically neutral (Smith, 2004). For example, the relations is_narrower_than and part_of are certainly not of the same type. Where the former expresses a relation between *meanings*, the latter expresses a structural tie between *universals*.

Smith argues for a system where not the predicates, but indeed the subjects of those predicates carry meaning. The predicates themselves 'do not represent', but rather are what link together variable and constant terms. In this proposal Smith eliminates unary predicates altogether, and restricts the number of allowed relational predicates to a fixed set, containing amongst others subsumption, parthood and dependency relations. A restriction that was also advocated by Breuker and Wielinga (1987). Recall that in the 1970's semantic networks were criticised because their structure was too generic and semantically unclear (cf. Section 2.2.3). The solution was to develop languages that contained a fixed set of knowledge structuring primitives. Though given by different reasons, the proposal by Smith is in fact analogous to this solution.

Guarino (1994) takes a different approach, and proposes to limit the scope of predicates by formulating semantic constraints. These can be used to express the difference between e.g. sortals and non sortals, i.e. predicates that are substantial, e.g. whether some entity is an apple, $\mathsf{apple}(x)$, and those that express a mere characterisation, such as $\mathsf{red}(x)$.[8] This way, it is thought, a rigourous ontological foundation of the primitives in some knowledge representation language can guarantee a consistent interpretation of theories across different domains.

However, from a knowledge representation perspective, it is unclear how such a priori distinction between predicates on entities is possible. Or at least, how it is different from any other restriction posed in a knowledge base itself – and not in an ontological layer incorporated in the representation language.

[8]Guarino (1994) also introduces *rigidity*. See the discussion on the OntoClean methodology in Section 5.5.1 for a more in depth discussion of this notion.

Whether some predication of an entity meets its formal ontological requirement can only be checked against the *actual* entity *in the domain*, which is not formally specified other than by means of the predication *itself*. For instance, there is nothing 'in' the predicate red that precludes its interpretation as substantial: does $\mathsf{red}(x)$ state that x has the property of being red, or that x *is* the colour red?

In summary, the specification language used in formal Ontology turns out not to be ontologically neutral, but rather has to be used with care. This can be achieved either by having its predicates quantify over parts of the ontology, instead of individuals, or by distinguishing different ontological categories of predicates.

4.4 Two Kinds of Ontologies

As said in Section 4.2, use of the term 'ontology' by the knowledge acquisition and representation community did not go unnoticed in formal Ontology. The wider adoption of the term, until then private to a small community, sparked concern as to the place of Ontology in knowledge representation. While in knowledge acquisition, ontology construction was an important but preliminary step in knowledge based systems development, and knowledge management even posed the ability of ontology *extraction*, formal ontologists naturally saw a more prominent role.

Guarino (1994) made efforts to integrate formal Ontology with the notion of knowledge representation languages. He argues for an *ontological level* on top of Brachman's epistemological level. Where the epistemological level provides structure, the ontological level is to constrain the meaning of primitives in a knowledge representation language. Guarino criticises the neutrality of knowledge representation formalisms as regards their ontological commitment. In his view, structured representation languages such as KL-ONE cannot be "distinguished from their 'flat' first-order equivalents" without making the ontological commitments underlying their structure explicit. It should be made clear what it 'means' to interpret binary predicates as roles, and unary predicates as concepts:[9]

> At the ontological level, knowledge primitives satisfy formal meaning postulates, which restrict the interpretation of a logical theory on the basis of formal ontology, intended as a theory of *a priori distinctions.*
>
> (Guarino, 1994, p.444)

Serious efforts to reconcile ontology in AI with its older and wiser namesake were of course welcomed by the knowledge acquisition community, but quite often with a sense of bemusement. For the theoretical considerations brought to bear by the likes of Smith and Guarino seem to be of no direct *practical* relevance in the development of knowledge-based systems, let alone knowledge management. And furthermore, the naive notion of (formal) ontology as a direct reflection of reality was somewhat smirked at by a field that

[9]Ontologies that specify the commitment of a formal representation language are usually called *representation* ontologies (van Heijst et al., 1997).

had been struggling with that very same relation between representation and reality for over twenty years. On the other hand, the use of "ontology", as merely a convenient moniker of some part of a specification in knowledge acquisition methodologies was perceived as rather careless by formal ontologists for which the ontology itself is the primary goal.

The relative positions can be summarised as follows:

- Philosophy's main criticism concerned the lack of theoretical philosophical rigour underlying the *content* of ontologies in AI: domain theories are very often philosophically naive.

- AI, on the other hand, (silently) criticised philosophy's disregard of the fundamental problems in knowledge acquisition and representation, such as the interplay between language and representation, and the interaction problem (Bylander and Chandrasekaran, 1987). AI ontologies are meant to be used for *reasoning* in the context of very mundane problem solving tasks where overly theoretical conceptions are more likely to be a burden than a help.

The apparent incompatibility between principled philosophical and theoretically 'loose' AI conceptions of ontology has in fact quite often led to heated debates about a *proper* definition: a definition that would be compatible with both perspectives and one that could reconcile the positions. There have been several attempts, primarily by Guarino and Giaretta (1995); Guarino (1998) to come to such uniform definition.

One source of confusion has been that originally, both interpretations were vague as to whether an ontology is the specification *itself* or *that which is specified*. In philosophy this was characterised by disregard of the formalism, and in AI by imprecise usage of the term itself. This initial indecisiveness was settled by the well known definition of Gruber (1993, 1994) (see Section 4.1), which distinguishes the ontology, as specification, from that which it specifies, the 'conceptualisation' (Genesereth and Nilsson, 1987). Surely we can conceptualise, or understand, the world in many different ways. An ontology captures a commitment to those parts of a conceptualisation that we deem to exist in reality. This conceptualisation is a priori inaccessible: it only exists "in someone's head" (Uschold, 1996). The explicit specification of a conceptualisation is therefore just as subject to the knowledge acquisition bottleneck as other forms of knowledge representation, and it was consequently acknowledged that a conceptualisation can only be approximated:

> An *ontology* is an explicit account or representation of some part of a conceptualisation.
>
> Uschold (1996), adapted from Guarino and Giaretta (1995)

Taken in this light, Ontology, as the philosophical discipline, endeavours to approximate the a priori conceptualisation that underlies the structure of reality. However, in the context of knowledge-based systems, the notion of ontology is clearly 'disconnected' from reality: the conceptualisation being specified is that shared by one or more experts. No claim is made as to the real

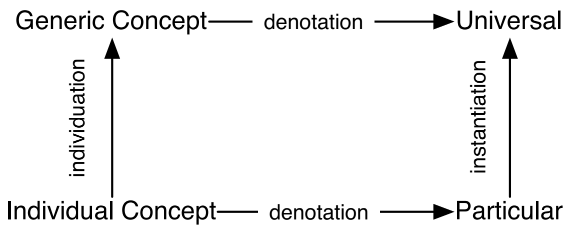

Figure 4.1: Ontological relations in Realism

existence of elements in the ontology. The ontology specifies that which a system 'knows' about:

> An (AI-) *ontology* is a theory of what entities can exist in the mind of a knowledgeable agent.
>
> Wielinga and Schreiber (1993)

In other words, the ontology prescribes what entities can be distinguished, or rather *individuated* inside a knowledge base: an ontology encompasses its *generic concepts*. Recall the relation between a knowledge representation and entities in reality in Brachman's meaning triangle of Figure 2.4.[10] Admittedly, both generic and individual concepts in a knowledge representation can be related to some entity (individual object) in reality through instantiation and denotation, respectively. It is the ontological status and strength of these relations as to which philosophy and AI differ. Firstly, in AI, denotation is a correspondence relation between a concept in a knowledge base and some entity in reality. It is generally true that an individual will only be asserted into a knowledge base given some corresponding entity, but this is not enforced. In fact, individuals are often asserted for purely practical reasons, as mere database keys. Secondly, instantiation of a generic concept by an entity is an even weaker relation in the ontological sense: the entity merely *exemplifies* the generic concept.

These are considerable weaker versions of their philosophical interpretations, especially in comparison to the position of realism. Realism holds the existence of *universals*, properties such as ''*being an apple*' that hold in multiple places, or rather are instantiated by multiple *particulars*. Using KR wording, realism essentially adopts the stance that reality contains properties (entities) denoted by generic concepts (see Figure 4.1). As AI makes no such claims, proposals for definitions of 'ontology' are often accused of adopting the opposite position of *nominalism*, which holds that universals only hold as names.

However, this is a false accusation as has been made clear by more philosophically aware AI researchers. For instance, Genesereth and Nilsson (1987) who coined the word 'conceptualisation' on which Gruber's definition is based,

[10]Keep in mind that Brachman did not distinguish symbol level and knowledge level representation.

explicitly state that no attention has been paid to the question whether the objects in a conceptualisation really exist. They do not adopt realism nor nominalism:

> "conceptualisations are our inventions, and their justification is based solely on their utility. This lack of commitment indicates the essential ontological promiscuity of AI: any conceptualisation of the world is accommodated, and we seek those that are useful for our purposes."
>
> (Genesereth and Nilsson, 1987)[11]

The keyword here is '*utility*'. Philosophical ontologies are not intended to be *used* in the operational sense of the word. They are not intended to be a part of some knowledge based system, but rather reflect a formal commitment to some philosophical theory. It raises the question as to whether the notions of philosophical and AI ontology are compatible: *can* a philosophical ontology be used in practice? This assumption is often implicit in philosophically inspired formal ontologies (Grenon, 2003; Masolo et al., 2003). But, as discussed in Chapter 6 it may not always hold (Hoekstra et al., 2007, 2008).

4.5 Discussion

The mixture of philosophical and knowledge representation perspectives involves a trade off between an ontology as *knowledge representation artefact* and as *theory of existence*. According to the first view, the quality of an ontology does not relate to philosophical adequacy, but rather to its suitability to play a role in problem solving (Genesereth and Nilsson, 1987) and knowledge reuse. The design choices inherent in the knowledge based system determine what the ontology contains, and no particular restrictions hold as to its shape or form. The main requirement is that the ontology should adequately capture the system's domain theory and be neutral with respect to its control knowledge (Chapter 2).

The philosophical perspective, on the other hand, poses additional restrictions on the *content* of ontologies. First off, it is clear that the purely philosophical view of Formal Ontology cannot easily be reconciled with the knowledge representation view because where the former sees language primitives as the primitives of existence – extending an ontology equates to extending the language – the latter restricts language primitives based on pragmatic, epistemological and computational considerations (Levesque and Brachman, 1985, and Section 2.5.1). The discrepancy between the two views is evident in e.g. Bera and Wand (2004)'s refutation of OWL as ontology language. As discussed in the preceding chapters, this is the price one has to pay for the practical application of ontologies in reasoning.

The development of ontology representation languages inspired the use of ontologies as a readily available resource for knowledge based reasoning. These *knowledge representation ontologies* are specified using their own carefully crafted representation language (OWL DL), and inference is supported by highly optimised reasoners. The ontological perspective is partly ensured by adopting the DL paradigm as it only sanctions inference that is ontologically relevant: *consistency* checking of axioms in the ontology, *classification* of concepts as belonging to more generic categories, and *instance* checking of individuals as denoting instances of some concept. This way, the semantics of the language makes the representation of an ontology correspond more directly to an explicit set of ontological commitments: the ontology cannot commit to more than what is inferred by the reasoner (cf, Davis et al. (1993) and Section 2.3.3).

Despite their importance, these are mere preconditions for the development and use of *good quality* ontologies: two OWL axioms do not make an ontology. Calling any domain theory an ontology does not do justice to the claim that underlies the adoption of the term in the first place: the ontology expresses a theory of existence. Of course, Davis et al. are entirely right in stating that a knowledge representation is "a *set of ontological commitments*" (Davis et al., 1993, p.3), but it expresses *other* commitments as well. For instance, Clancey (1983) identified causal rules as part of the domain theory of MYCIN (see Section 2.3.2). Though certainly not part of the application's control knowledge, causal rules reflect an *epistemological* rather than ontological perspective. The distinction between the two is discussed in more detail in Section 5.5.2.

Because the quality of ontologies depends on subtle, and competing requirements, their development is a delicate task that involves a large number of important decisions. Decisions that carry additional weight when considered in the light of knowledge sharing and reuse. Of course, an ontology engineer needs to decide not only which concepts and relations to include, but also the level of detail in which they are defined. Furthermore, every definition should adequately cover the intended meaning of a concept: the traditional knowledge acquisition bottleneck (Feigenbaum, 1980).[12]

This chapter presented an overview of the different conceptions regarding what 'an ontology' is. It distinguishes three views:

- *Knowledge Management Ontologies* are (structured) vocabularies developed for sharing and reuse of information within organisations. Languages suitable for representing these ontologies can be lightweight, as no (expressive) reasoning is required. Examples are RDF/RDFS, SKOS, UML or Topic Maps.

- *Knowledge Representation Ontologies* are reusable terminological knowledge representations that specify the part of a domain theory that directly reflects its ontological commitment. Languages suitable for representing these ontologies incorporate a trade-off between expressiveness and decidability, to support ontology-based reasoning services within knowledge-based applications. Examples are description logics, and most notably OWL DL.[13]

- *Formal Ontologies* are formal specifications of an ontological theory in philosophy. Languages suitable for representing these ontologies are highly expressive and involve a minimal ontological bias as regards their language primitives, such as first-order logic.

[12]The knowledge acquisition bottleneck is often misunderstood as the high threshold in effort before knowledge representation starts to pays off, and practical reasoning problems can be solved. However, in Feigenbaum's original reading it rather refers to the general difficulty of correctly extracting expert knowledge into a knowledge base, see Section 2.4.1 on page 29.

[13]Note that some methodologies, e.g. van Heijst et al. (1997) use the term 'representation ontology' to refer to an ontology that defines the primitives of a knowledge representation language. This is not what is intended here.

Although the ontology types and languages do not correspond directly, confusion may arise when one of the language paradigms is applied in the representation of a different ontology type. The following chapter outlines requirements and methodological guidelines for the construction of ontologies needed to ensure both their reusability and ontological nature. Chapter 6 describes the construction of a core ontology for the legal domain that aims to maximise these factors.