

# Chapter 3: A Primer on Knowledge Representation and Ontological Engineering

Pierre Grenon

## 1. Introduction

Suppose you want to understand the world, or a portion of the world – for instance, how a post office works – because you want to do something in the world (say, you want to send a letter to your grandmother) or because you want to know how the world will be if certain conditions obtain (say, you worry whether you will ever be able to send another letter to your grandmother should your neighborhood post office close), or just for the sake of understanding (simply because you are interested in post offices). The field of knowledge representation aims to provide computer support for doing precisely this sort of thing, namely, understanding the world or reality. At first glance, there appears to be no reason to expect that working with computers is any different from working without them. Computers are tools. Sometimes they perform well, but it is unlikely they will perform better than you prepare them to; they often perform worse. The point, however, is that there can be reasons for poor performance which are not due to limitations of computers. Knowledge representation can make computers better tools by being serious about representing the world.

If you take this endeavor seriously – that of understanding the world – you need some basic commitments which roughly amount to acknowledging (1) the reality of the world and (2) your own reality as part of the world. Let us call these the *basic principles of realism*. Also, it would be helpful if you have (3) a positive attitude toward your capacity to understand the world, and that you take the view that you are able to know something about reality, if even roughly or approximately. Let us call this the *realist principle of knowledge*. Combined, these three principles amount to the assertion that there is a reality which may be known (even imperfectly), and that knowing is a relation between a knower (you), and reality or a part thereof (the object of knowledge, such as a post office).

It is striking, then, that in knowledge representation there is a widespread attitude which tends to contradict this basic methodology. The present chapter is intended as a prophylactic against this attitude. Our

intention is to warn against denying, or ignoring, realist principles by highlighting some of the problems to which such attitudes may lead. As we will see, whereas the commonplace motto in knowledge representation is to represent convenient conceptualizations (i.e., representations) of the world, the basic motto for a sound knowledge representation must be that it is the world itself that ought to be represented. Perhaps this comes across as a platitude, and all the better if it does. But it is something that is too easily and too often dismissed as such, even though it should be constantly borne in mind.

I will begin with some non-technical preliminary considerations about knowledge representation and examine the role of philosophy in knowledge representation against this background along two dimensions. The first dimension concerns how philosophy itself is a tool for representing knowledge, whereas the second concerns the philosophical foundations of a methodology for knowledge representation. While presenting this second dimension, we put forward a methodological approach which adheres to the principles of realism mentioned above. I use these principles as representative of a philosophical position opposing other positions which could be called 'conceptualist'. In the same way that the methodology I propose is tied to the realist philosophical positions (that is, somewhat loosely and maybe only intuitively), the methodological approach I call 'traditional' is tied to conceptualist positions. To some extent, the views I name when discussing the traditional approach are mere reconstructions of positions, philosophical or methodological, which are only implicit in practice. These views are not established or structured doctrines. The reconstruction is needed in order to make a number of problems easier to grasp, problems arising from certain practices and outlooks which ignore or deny realist principles. Within these practices and outlooks, I select three specific methodological suggestions that embody in their own ways dreadful positions. Toward the end of the chapter, I formulate a more positive account of the methodological outlook needed for knowledge representation. Rather than aiming at firm guidelines and detailed recipes, this presentation of a realist methodology for knowledge representation aims to offer a taste of the mindset required for using and developing knowledge representation technologies in a sound way.

## *2. Preliminaries: Understanding and Representing the World*

### *2.1. Non-technical Knowledge Representation*

There is, then, the world, you, and your desire to understand the world. Fulfilling this desire is likely to involve some of the following steps, which might have to be taken recursively:

- (1) looking at the world (e.g., going to the post office and observing);
- (2) gathering facts about the world (e.g., observing some people enter with papers in their hand and leave without them, some other people behind a counter taking these papers from the former people, marking the papers and receiving money);
- (3) representing the world (e.g., taking notes of all of the above, finding some way of making perspicuous statements about what you have observed);
- (4) conjecturing the presence of sophisticated structures and the existence of other entities which may account for the facts at hand (for example: apparently the population in a post office is divided into two groups, those who pay for handing papers and those who handle the papers. Since what the latter do to the papers must be significant, perhaps there is something which makes a difference to the papers before and after the interaction, such as the stamps);
- (5) validating conjectures through inferences and experimentation (e.g., preparing a paper to send to yourself, going to the post office, verifying that there is a fee and how much it is, sending the paper, waiting and comparing the paper you receive to the paper you posted);
- (6) inferring further structures and the existence of additional entities based on valid inference patterns and the facts at hand (for example, apparently, some people deliver mail; since those you gave your mail to spend their day at a counter, there must be other people doing the delivery, and there must be a complex organization behind this).

At each step, something could go wrong which might have repercussions on the understanding you gain about the world. You might arrive at the post office while it is closed. If you do not know about hours of operation, this might lead you to infer that post offices are often or always closed. You might be overwhelmed by details or be too sensitive to the specifics of the post office you visit. If you only visit a post office in Saarbrücken, Germany, for instance, you might infer that only German is spoken in all post offices. Conversely, you might over-generalize and infer that any place with a counter is a post office. If your facts are not right or if your inferences are shaky, you might end up with quite an odd vision of the world.

One way to avoid such problems is to make sure that you get your facts straight and that you make proper inferences. This can take a great deal of care and effort, as in any science. If you are not a scientist or if you do not have the resources to spend on scientific research, you will want to ask somebody who might have the required knowledge, or to read their books. More competent people will be more helpful. The more a person knows about a domain or the more reliable her knowledge of the domain – given some criterion of reliability – the more competent she is in this domain. In increasing order of competence regarding post offices, you can ask a passerby, a post-office customer, a post-office employee, or an expert commissioned by the post-office company to design post-office regulations. Similarly, in increasing order of competence regarding the geography of Germany, you can ask a passerby on the street of another country, a passerby on a street of Berlin, a German geographer, or a geographer specialized in the geography of Germany.

These sorts of resources can improve the breadth of your factual knowledge. As your knowledge increases, it will become critical to have a way of storing it accurately and accessing it easily. It is not convenient to rely on people, because they are generally not readily available to answer your queries. Relying on paper documents might become an issue as your sources increase in size and number. You will also have difficulties finding precisely what you need in, possibly, massive amounts of irrelevant material. These are very simple and practical motivations for using technological help in storing, retrieving, and sharing knowledge.

Our concern is with several aspects of factual knowledge, namely, its quality, its efficient management, and what one can do with it. In addition to informants, one might want to turn to people with good reasoning or inferencing capabilities who can analyze one's data and extract new

knowledge from it. You might have limited knowledge and not be too sure of what else this knowledge allows you to take for granted. To a large extent, inferencing capabilities and competence, or expertise, in a domain are *prima facie* independent of one another. The sorts of inferencing relevant for us are not always very sophisticated, as we will discuss in an example below. However, they are technical and can be laborious; even in simple cases, it is a little like following an overly detailed recipe. The theory of these tools is provided by logic. In many cases, machines are very efficient at performing certain of these reasoning tasks. Here again, we find simple and practical motivations for using technological help, this time in order to manipulate, transform, and analyze the data or knowledge at hand.

## 2.2. Machine-Based and Formal Knowledge Representation

Looking to machine-based assistance to achieve this sort of goal brings one to the field of *knowledge representation*, which stands at the junction of the larger disciplines of *artificial intelligence* and *knowledge management*. Artificial intelligence, itself, is a field of computer science whose purpose is to get machines to perform tasks usually performed by human beings such as, in the present context, making inferences. For its part, knowledge management aims to make knowledge accessible, manipulable, and sharable, and may be seen as an attempt to produce efficient and re-usable tools for the understanding and manipulation of human and machine-processable knowledge.

A typical software solution for using machines to perform these sorts of tasks is an *expert system*. Expert systems consist of software dedicated to performing the tasks that a human expert would perform. An expert system contains three parts, namely, a *database*, an *inference engine*, and a *user interface*. Here, we are interested in the first two; the user interface is irrelevant, for our purposes. Databases are used to record facts about the world. An example of a medical fact is that there are streptococci in your throat, while another example is that the strand of streptococci in your throat is one whose presence in your throat is pathological. An example of a geographical fact is that Saarbrücken is west of Leipzig. Another example – maybe only a borderline geographical example – is the fact that family Gnomsfreunde in Saarbrücken harbors an impressive collection of garden gnomes in their yard. It is facts such as these that we record in databases. In the present context, databases which record such facts are

called *knowledge bases*. Nowadays, knowledge bases are not only parts of expert systems dedicated to a particular domain, there are also large multipurpose knowledge systems dedicated to multiple domains, or even developed to act as potential universal expert systems; for instance, the Cyc system (Lenat and Guha, 1989).

The inference engine is the part of the software that enables *inferencing*. Inferencing is the process of eliciting facts not recorded in the knowledge base, on the basis of two sources. These sources are (a) facts which are recorded in the knowledge base and (b) *rules of inference*. An example of a rule is, ‘if a family collects garden gnomes, this family decorates its house for Christmas’.

Suppose that we have a knowledge base which contains one fact and a rule:

Fact: The Gnomsfreunde family collects garden gnomes.

Rule: If a family collects garden gnomes, this family decorates its house for Christmas.

Running the expert system, we could infer from the fact and the rule the fact that:

The Gnomsfreunde family decorates its house for Christmas.

Using rules with an inference engine is a way of transforming the knowledge base so that it presents a finer and more explicit picture of the world than that provided by the raw set of facts which it initially contains. In a way, rules themselves contain knowledge; thus, the sum total of the knowledge in an expert system is the union of the knowledge base and the rules used by the inference engine.<sup>8</sup>

From now on, I will take the standpoint of the builder and maintainer of a knowledge base, who is often referred to as a *knowledge engineer*. The core task of the knowledge engineer is to put knowledge into computer-

---

<sup>8</sup> The rules we are dealing with are factual (it is a fact about reality that these rules obtain, and they describe reality or relevant portions thereof). There are also logical rules of transformation which have nothing to do either with the domain or the factual knowledge at hand. For instance, the deduction above uses the logical rule of *modus ponens*: *A* (premise); If *A*, then *B* (premise); Therefore, *B* (conclusion). Going more into the details would be tedious for the unfamiliar reader and mostly irrelevant for the familiar one.

processable form. Typically, this is done by imposing a formalism upon the data, thus enabling it to be stored and manipulated. At this level, we do not need to distinguish between facts and rules, for both are of interest to the knowledge engineer. Storing and handling information are tasks that belong to the data management and information retrieval part of technological knowledge management. The resultant formal representation is often used to provide support for applications, such as reasoning (inferencing) or natural language processing, which are, in turn, sometimes considered to fall within the scope of the knowledge engineer's activity. Here, I will focus only on representational issues and not address those other activities.<sup>9</sup>

There are a variety of techniques and formalisms that the knowledge engineer may use (for example, compare the different ways you and your foreign language-speaking neighbor describe or refer to the same fact). I will take, as paradigmatic,<sup>10</sup> the representation of knowledge by means of a logical formalism such as that of first-order predicate calculus. This mode of representation has the advantage of allowing, in principle, the explicit representation of the objects in the relevant domain of discourse, and a straightforward formulation of rules as statements of logical consequence.

### 3. *Ontology and Knowledge Representation*

#### 3.1. *Engineering Knowledge*

A knowledge engineer deals with bodies of knowledge which include factual data and the sort of data that are contained in rules. Very quickly, it becomes obvious that these bodies of knowledge need to be given a structure, not least for reasons of efficiency and reusability. Because the knowledge engineer is engaged in manipulating and structuring knowledge, her activity is shaped *inter alia* by philosophical assumptions which underlie her adopted methodology – a truism where any activity

---

<sup>9</sup> Actually, natural language poses a problem for knowledge representation. But, aside from the sheer difficulty of natural language processing (parsing of text via knowledge representation or generating text on the basis of formalized knowledge), it poses a problem rather indirectly and methodologically, through the more or less deliberate reliance on natural language phenomena when devising a representation of knowledge. We will come back to this when discussing what I shall later call *linguisticism*.

<sup>10</sup> See Bibel, *et al.*, 1990, for a somewhat dated, but remarkable, introduction to the field of knowledge representation and its techniques.

such as representing the world is concerned. But her activity is also directed by presuppositions inherent in the specific domain, or implicit in the structure of the framework in which the formalization of information is conducted. To see this more clearly, let us separate out the various tasks that the knowledge engineer performs.

The work of representation begins after a body of knowledge has been acquired from the preliminary stage called *knowledge acquisition* or *information gathering*. Sometimes it is the knowledge engineer herself who gathers the facts, other times she receives her data from an external source. From this point onward, the activity can be broken down into three somewhat overlapping main tasks:

- (1) feeding the knowledge base,
- (2) improving the existing framework,
- (3) formalizing the knowledge of an expert in the pertinent subject matter.

Feeding the knowledge base means recording facts, e.g., that Saarbrücken is a city in Germany or that patient Lambda consulted Doctor Mu on a particular date. Unless this is done from scratch (i.e., the knowledge base is empty), there will already be a way to represent facts which is, more or less, specific to the system used. This means, roughly, that the language for knowledge representation might already be developed to some extent and will therefore impose constraints on which facts it is possible to represent and how to represent them. In such cases, the work of the knowledge engineer is shaped by the structure of the pre-existing framework. Take, for instance, the fact that Saarbrücken is a city. If the framework countenances a type of entity to which cities belong, representing this fact might be a matter of predicating 'being a city' of Saarbrücken, yielding 'Saarbrücken is a city'. Alternatively, if the framework associates entities such as cities with something we may call a geopolitical status, we could express a relation between Saarbrücken and the object named city, yielding 'Saarbrücken *has geopolitical status* city'.

It might be the case that the framework in which the knowledge engineer is working is not completely suitable for adding some of the new facts. This could be due to a lack of vocabulary; for instance, perhaps there is no way of expressing that Mu is a doctor. It could also be because the vocabulary that the knowledge engineer used to gather facts is already used by the framework in a conflicting way. This would be the case, for

instance, if everything in the framework which deals with medical patients were geared toward the veterinary domain and that, in particular, only a limited range of non-human animals could show as patients. At this stage, there may be a need for improvement or refinement of the existing framework itself, or of the way the knowledge engineer records the facts for herself. Issues regarding lack of vocabulary are usually rather benign. Generally, they can be resolved by extending the vocabulary used in the system. So, for instance, if you have no term for speaking of a patient in the language used by the system, or if the closest term does not precisely fit the intended use when representing a particular fact, you just add the missing term. You might have to do more work in order to amend and polish the existing structure (for example, removing the inadequate vocabulary or correcting a possible ambiguity by adding a suitable generalization). More problematic, however, are issues which have to do with limitations that are intrinsic to the language of representation. This sort of problem is one which suggests that the initial language is not suitable for representing the kinds of facts in question, in which case it might be necessary simply to opt for another language.<sup>11</sup> The only other means of overcoming such limits on expressibility is to rework the facts themselves. However, such activity is less than innocent when the goal is representation of the facts.

These first two tasks constitute, in practice, the extent of the knowledge engineer's activity. The third task, formalizing the knowledge of an expert in the pertinent subject matter, can be conceived as a way of feeding the knowledge base. But this requires that the system be able to allow for the

---

<sup>11</sup> Description logics (DLs) are a family of knowledge representation languages – fragments of first-order logic with nice computational properties – which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. Where given sorts of statements are not readily expressible in a DL language, this does not mean that the language is powerless but, rather, that one will need to find some *ad hoc* way of representing the knowledge that needs to be conveyed. This can lead in turn to non-trivial alterations of the ontological resources you had in view. Trying to preserve ontological resources and the integrity of an ontology while using languages with expressivity constraints is one important research activity in ontological engineering (see, for example, Grenon, 2006). For a less technical example, consider the different degree of user-friendliness of a language containing only adjectives, nouns, and a copula for attributing the former to the latter, with that of a language containing also verbs, adverbs, and prepositions. Or, consider again the relative merits of a communication system based on drawings against those of a full-fledged language.

formal representation of the expert's knowledge, so it might involve the second task as well. The knowledge contained in a knowledge base can be made of simple facts (e.g. Saarbrücken is a city in Germany) or rule-like knowledge, such as that all the people who reside in Saarbrücken reside in Germany. For instance, the ability to record the fact that somebody resides in a geographical region is required in order to record a rule such as that all people who reside in Saarbrücken reside in Germany. This is why new vocabulary sometimes has to be introduced before knowledge can be represented accurately. It is also worth noting that aligning new vocabulary with the existing one is a way of adding more facts and rules (that is, knowledge) to the knowledge base; for instance, expressing that residing in a region involves having a dwelling in that region.

What, then, is the knowledge engineer doing when she is building a formal vocabulary suitable for the representation of facts of a certain sort? She may be doing two things: (a) trying to fit the facts to a representational schema, or (b) trying to tailor the representational schema to her intended representation of facts. Doing the latter requires that she process and analyze the facts. It requires that she try to identify the structure of each fact, as well as what entities the fact involves. This is, roughly speaking, building a theory. Theory-building along these lines typically proceeds by *generalization* (e.g., all post offices have customers and clerks) and *abstraction* (e.g., there is a kind of entity under which clerks fall, and another under which customers fall, and each of these kinds is associated with properties reflecting powers and abilities of the relevant people). Sometimes building these sorts of theories about the world, or a domain of reality of interest to the knowledge engineers, requires making use of metaphysical and ontological insights. We will see what this means now.

### 3.2. *Philosophical Ontology*

Philosophical ontology is a branch of philosophy concerned with the question of what there is. More specifically, it focuses on determining what entities exist in the world and what the categories they fall under. (See Chapter 2 of this volume.) The product of an ontological investigation is typically built around one or more backbone taxonomies or hierarchies of categories. A typical ontology would include, for example, taxonomies

with the categories of *substance* and *property* as topmost nodes.<sup>12</sup> In addition, an ontological inquiry will provide an account of the relations between entities, and of the structure of the world at a high level of generality. For instance, the ontology will represent the relation (sometimes called exemplification or inherence depending on the type of ontology at hand) between substances and properties. Philosophers then discuss whether, given a proposed category, there are actually entities which fall under it and devise different assays of the subcategories needed. Some may deny altogether the existence of a category – for example, that of substance – claiming that the entities alleged to fall under this category are, in fact, entities of another kind (for instance, instead of being substances with properties, they are merely bunches of properties, sometimes called bundles).

The picture becomes slightly more refined, however, when we consider a distinction between at least two kinds of ontological inquiry made popular by Husserl (see, for instance, his 1931). On the one hand there is *formal ontology*, which conducts analysis and produces theories of a domain-neutral sort, theories of *forms* (for example, of part-whole relations, number, and so on). On the other hand there is *material* or *regional ontology*, which is the ontology of some specific domain or material region (for example, of mind, behavior, society, and so on).

Consider the example of an ontological inquiry in the domain of post offices. This is a domain-specific inquiry, in that we are looking for the ontology of what is going on in a (typical) post office (rather than of what sorts of post offices there are). You might find that a post office has: a clerk, Mrs. Goggins; a postman, call him Pat; and a number of customers, such as Julia Pottage or PC Arthur Selby. One way of answering the ontological question (‘what is there?’) is to say that there are entities which fall under the following kinds, types, or categories: the category of clerk (Mrs. Goggins), the category of postal delivery agent (postman Pat), and the category of customers (Julia Pottage or PC Arthur Selby). This is not a very sophisticated answer – in particular, because these people are not *just* clerks, mail carriers, or post office customers – but it will do for the purposes of this simple illustration. In addition, there are a number of kinds of *activities* (things these people do) and relations between these people: Mrs. Goggins sells stamps and sorts the mail; Pat delivers the mail; PC

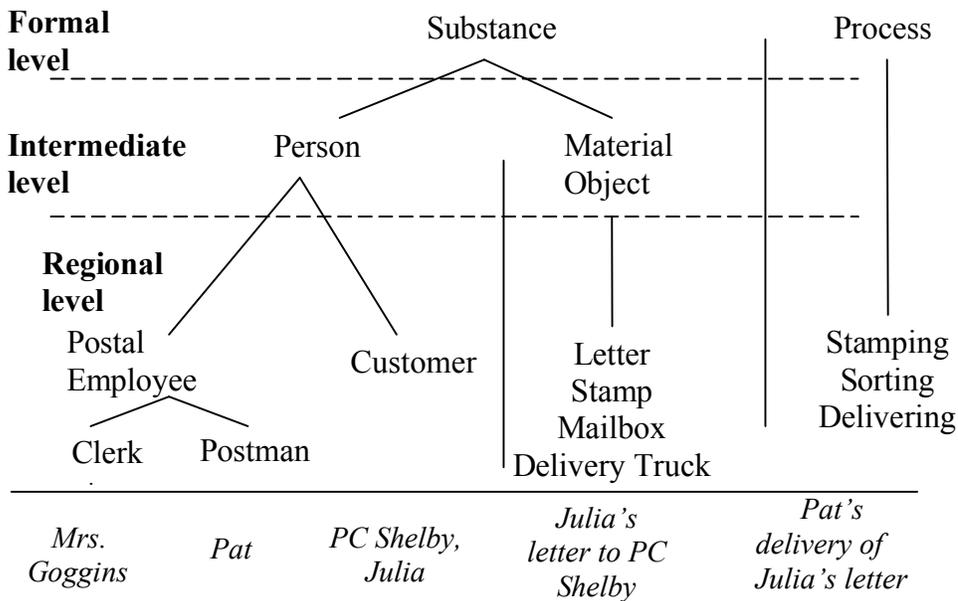
---

<sup>12</sup> ‘Substance’ is a term for the category under which those entities fall that may be characterized. ‘Property’ is the term for the category of entities which characterize substances.

Selby and Julia buy stamps, post mail and, perhaps, receive mail as well. So you may add the categories of *stamp selling*, *mail sorting*, *mail delivery*, *mail posting*, and so on to your ontology. Of course, more is needed; for instance, a category of *stamp* and a category of *mail object* with subcategories of *letter* and *package*. Examples of relations between the entities are *delivering mail to*, *selling stamps to*, *buying stamps from*, and so on.

All of these categories and relations are domain-specific; if you extend your ontology by generalization (a clerk is also a person) or by comparison (there are other domains in which some person buys some object from another person), you progressively arrive at less and less domain-specific considerations. You finally reach the level of domain unspecific (or, domain-neutral) considerations in which you have, say, categories of substance, property or quality, process or event, and relation. This constitutes roughly what Husserl calls the level of forms. The categories at this level can allegedly be applied and specialized in more restricted domains. In between the most specific level and the most general level you consider, there can be any number of intermediate levels. Intermediate levels are pertinent to more than one domain-specific level, but not to all.

Figure 1: *Ontological Levels*



This picture is presented in Figure 1: the dashed lines separate categories into levels; the continuous horizontal line separates categories from

examples of entities that fall under them, also called instances of those categories; and the lines between categories stand for subsumption (i.e., the category below is more specific than that above in the sense that all instances of the lower one are instances of the higher one, but not vice versa).

### 3.3. *Information Science Ontology*

When the knowledge engineer elaborates a theory of a domain and designs a system of categories together with the properties and relations which characterize the entities belonging to that domain, she is building an ontology. Building an ontology is, in the first place, a technical activity and not necessarily one that involves philosophical craftsmanship. But the threshold is very easily crossed. The knowledge engineer becomes an ontological engineer<sup>13</sup> as soon as she performs a philosophical analysis of the content and a shaping of the infrastructure of the knowledge representation system, in the light of metaphysical/ontological theories, or in a way that is inspired by such theories. Such situations occur, for instance, when she wonders whether the fact that Saarbrücken is west of Leipzig involves one entity and its property (Saarbrücken, being west of Leipzig) or two entities and their relation (Saarbrücken, Leipzig, being west of). It occurs when she wonders whether these entities are continuants (entities which change over time while retaining something of their identity – as common sense would have persons and maybe cities do) or whether they are more like processes or events (entities which unfold in time such as rugby games).

In information science, however, the term ‘ontology’ is used in a multiplicity of ways; a fact which, over the past decade, has generated – and continues to generate – a flood of conflicting publications on the way the word should be used. In effect, the term ‘ontology’ applies to virtually any structure resembling, to some extent, a set of terms hierarchically organized which may be put in a machine-processable format. In increasing order of sophistication, the reference of the term ‘ontology’ in information science may include:

---

<sup>13</sup> The distinction between knowledge engineering and ontological engineering essentially amounts to a division of labor. This distinction was made already by Russell and Norvig (1995) for whom it corresponds to the opposition between the domain-specific and the domain-neutral.

- (1) a set of terms (classes, categories, concepts, words),<sup>14</sup>
- (2) an axiomatic theory or a set of propositions,<sup>15</sup>
- (3) the content (conceived in a rather loose sense) of a knowledge base in general (or of some specific knowledge base such as that of the system Cyc).

A predominant view in the field of knowledge representation, reflected in the current use of the term ‘ontology’ in Semantic Web circles, is that ontological engineering is a form of modeling. On this view, an ontology would be a model or the description of a model.<sup>16</sup> Indeed, some even speak of the task of building an ontology as a matter of conceptual modeling or of conceptual representation (a conceptual system would be a model of reality). This is but a specific case of a more general trend in knowledge representation which we will now discuss.

#### 4. *Trends in Knowledge Representation*

In the Introduction, I said that the view put forward here is one according to which the basis for knowledge representation should be, not representations of reality, but reality itself. Let us call this view *realist representationalism*. On this view, when we are doing ontology, we are dealing with the things themselves, not with representations of them. While this principle may seem too obvious to need mentioning, the most widespread methodological stance in knowledge representation leads, in fact, to practices that oppose realist representationalism. This stance puts knowledge engineers at risk of committing the sorts of serious blunders which we address in the next section and elsewhere in this volume.

I will call the mainstream methodological approach to knowledge representation *pragmatist conceptualism*. According to this approach, the knowledge engineer’s main priority is to create a smoothly functioning knowledge base out of whatever conceptualization is provided to him by domain experts. This approach conceives the task of the knowledge

---

<sup>14</sup> Such sets can be more or less structured and can be anything from taxonomies to dictionaries, vocabularies, terminological systems or even thesauri as well as semantic networks such as WordNet (<http://wordnet.princeton.edu>).

<sup>15</sup> See the formal treatment of theories developed in Menzel’s *Ontology Theory* (2003)

<sup>16</sup> A model, here, is probably best understood as a representation which allows a certain degree of simulation and approximation judged adequate by the modeller for the purpose at hand.

engineer as consisting *only* in that of representing others' (the domain experts') representations, so that reality falls out of the picture almost entirely.

If a situation arises in which the expert's conceptualization does not make for the most smoothly functioning knowledge base (perhaps because the logical language in question cannot express a certain kind of statement), pragmatist conceptualism licenses the knowledge engineer to adjust the expert's conceptualization to make it expressible by means of the tools he has at hand. Thus, pragmatist conceptualism frees up the knowledge engineer to bring up any kind of objects and any theoretical construct that may prove useful for her representation. The only guidance is that such conjuring logically fulfill the practical purpose of the representation. This is emblematically endorsed by Genesereth and Nilsson (1987), who sum up their position as follows: 'no attention has been paid to the question whether the objects in one's conceptualization of the world really exist.... Conceptualizations are our inventions, and their justification is based solely on their utility' (p. 13).

But what, exactly, *is* a conceptualization? This is far from clear, even among knowledge engineers themselves. Minimally, a conceptualization involves concepts and probably also their specifications. In turn, a concept is probably a thing which carries some sense or meaning. However, if any word is polysemous, it is the word 'concept' (see Chapter 4). Information science, in particular, puts far too heavy a load upon the term 'concept'. There does not seem to be any prevailing meaning, and it is rarely used with a single, coherent meaning. For example, 'concept' might be taken to be one of the following: (1) an idea or a mental representation of objects in reality; (2) a general idea under which a multiplicity of things falls (let us call these *conceptual universals*);<sup>17</sup> (3) a Platonic<sup>18</sup> idea existing as a perfect prototype of things in the world, but itself, in some sense, exterior to the world; (4) a class, set or collection; (5) a word; (6) the meaning of a word.

These various meanings of 'concept' are often run together in more or less subtle ways.<sup>19</sup> The pervasive use of the term, and the running together

---

<sup>17</sup> Reflected in a language by the use of a (typically monadic) predicate, they allow great flexibility in representation, but it should be noted that only some correspond to any counterparts in reality. There is no universal in the world, for example, corresponding to our general concept of a unicorn.

<sup>18</sup> Here this term can be taken as a synonym for abstract and perfect model.

<sup>19</sup> This is such a routine difficulty that it can even be experienced with an ISO standard such as the terminological standard 1087-1:2000, for example (see ISO).

of its various meanings, result in further confusion about what a knowledge engineer is after – what she is trying to represent – and how she should carry out her work. The distinction between things (or real entities) and the corresponding concepts is acknowledged by the knowledge engineer, if only when endorsing the claim that conceptualizations are independent of reality. In her actual work, however, things in reality are neglected to the benefit of their conceptual proxies.

It is quite clear from the passage cited above that the sort of conceptualizations Genesereth and Nilsson have in mind are made of concepts in a non-realist, mind-dependent, sense rather than entities, such as universals or kinds, existing independently of human cognition (see Chapter 8 of this volume). To be fair, the actual nature of concepts is probably of little interest to a large number of knowledge engineers. It is credible that, the closer they are to computer science than to philosophy, the more susceptible they would be to being dismissive of what they would see as mere philosophical hair-splitting. And, to some extent, this dismissal would be understandable. It is conceivable that issues for philosophers could be non-issues for computer scientists and vice versa. But the problem here is that there are practical consequences associated with whether we adopt a methodology for knowledge representation that is inspired by realism, or by conceptualism.

One problem with pragmatist conceptualism is that, in the long run, representation that is the most *useful* is actually that which is the most accurate in relation to *reality*. Now, for computer scientists, usefulness might mean that modeling is easy or that inference is fast. There can then be architectural reasons specific to given systems which, for the sake of ease in using that system and of optimizing that system's inferencing resources, lead the knowledge engineer to adopt *ad hoc* modeling solutions and simplistic misrepresentations when using that system for representing knowledge. Giving in to such practices, however, results in idiosyncrasy (see Smith, 2006). Tailoring formal representations to suit the optimal settings of a given knowledge system might be less stable in the long run (because these settings are tied to hard coded features of the system and, thus, are dependent on the level of development of the architectural components of the system) and less reusable (because, obviously, they are *ad hoc*).

It is perfectly proper that we, as knowledge engineers, should be looking for a useful account. But it is hard to imagine what greater usefulness a knowledge representation could have than to be accurate with

respect to reality. It should be clear that, if the knowledge engineer professes pragmatic conceptualism explicitly, then there is no reason to accept her representation as anything more than daydreaming or literary fiction. A suggestion to those concerned exclusively with efficiency, then, might be that, if given tools will not perform without an *ad hoc* representation, then these tools need either to be improved (so as to allow for a representation that is more adequate to reality), or discarded. In contradistinction, aiming for consistent adequacy to reality in knowledge representation is likely to result in an improvement in the stability and usefulness of the resulting framework and, all things being equal, a broader acceptance of that framework.

Another disadvantage to pragmatist conceptualism is that it yields knowledge bases which are internally consistent, but unsuited for being linked with other knowledge bases. For example, it provides us with no principle ensuring that, for instance, bottom-up (from domain-specific to domain-neutral) and top-down (from domain-neutral to domain-specific) approaches will meet in any coherent way (suppose you want to place your post-office ontology under a more general ontology of services). Nor does it ensure us that two independently built ontologies or two independent knowledge representations of the same domain will overlap or agree about even one fact because, on this position, there are no *facts* to agree about.

For a long time, too little attention was paid to a principled resolution of the problem of unifying the many different, and often mutually incompatible, idiomatic frameworks and representations developed independently by different groups or companies. This is sometimes informally referred to as the *Tower of Babel problem*, and early attempts to solve this problem tried to devise schemas to which existing knowledge representations could relate, putting in place platforms of translations. For instance, the Knowledge Interchange Format (see: [logic.stanford.edu/kif/dpans.html](http://logic.stanford.edu/kif/dpans.html)) was first conceived of as a language to which other knowledge representation languages would be mapped, providing a central node in the net of inter-translatable languages. A sequel to this effort is the attempt under the Common Logic rubric of producing, as an ISO standard, an even more general and abstract specification of a knowledge representation language (see <http://cl.tamu.edu>). On the side of ontology, the Cyc ontology has been marketed as a potential platform for linking and comparing different ontologies. More recently, the IEEE gave its blessing to a Standard Upper Level Ontology working group (see <http://suo.ieee.org>), which aims to devise a consensus top-level ontology.

For various reasons, not all technical, there is still no ontology which is accepted as a standard in the knowledge-representation community. Instead, there are many candidates among which, for example, are OpenCyc ([www.opencyc.org](http://www.opencyc.org)) and SUMO ([www.ontologyportal.org](http://www.ontologyportal.org)), and more candidates have been springing into existence on a regular basis.

There are both pragmatic and ideological reasons explaining this state of affairs and the reluctance to work toward a common ontology. On the pragmatic side, it involves the resolution of a number of non-trivial problems (including problems of logic), and it would thus take time to come up with a decent standard. Such pragmatic considerations have sometimes been used to belittle the value of the attempt. But even if an acceptable standard ontology were successfully created, it would take more time and possibly prohibitive amounts of money to bring existing ontologies and knowledge bases up to a level of compliance. On the ideological side, what is at issue is the nearly pervasive, though often merely tacit, adherence to one or other form of conceptualism (often with constructivist or relativist leanings), which serves to render immediately suspect any candidate that is put forward as *the* standard ontology. This has the consequence that the question of the adequacy to reality of the conceptual schemas tends to be neglected. It also seems to imply that the search for an ontology of reality, rather than an ontology of its multiple representations, would bear no fruit.

The general problem of standardization has given rise to an area of research into *interoperability*, which focuses on how to manage the joint operation of distinct frameworks.<sup>20</sup> The premise is that if there is a multiplicity of conceptualizations, all should be accounted for. In effect, we end up with concurrent systems allegedly representing the same reality but, in fact, failing to do so because of presupposed *de facto* and in principle problems of interoperability. If concepts differ, what is to serve as our guide in resolving these differences? That is, if we wish to unite these competing systems together, what can serve as a *tertium quid*? The question becomes one of how to relate (or ‘fuse’ or ‘merge’) them. That such merging is a difficult problem – as illustrated, for example, by the Unified Medical Language System ([http:// umlsinfo.nlm.nih.gov/](http://umlsinfo.nlm.nih.gov/)) – reflects some of the dangers into which pragmatic conceptualism leads us.

---

<sup>20</sup> In the context of the Semantic Web, the recourse to namespaces, using a syntactic device marking the origin of each concept, helps as a bookkeeping device; but it eliminates only some of the confusion. It does not provide the needed ticket for interoperability and, probably, is only a short-term solution.

## 5. *Tendencies in Conceptualist Knowledge Representation*

There are at least three types of problematic tendencies to which pragmatist conceptualism opens the door; I will call them *linguisticism*, *algebraism*, and *subjectivism*. These tendencies echo and motivate some of the criticisms sketched above. For the sake of simplification, I present them as views in their most extreme forms. What I mean to warn against are the risks that result from making the activity of the ontologist succumb to, and crumble under, linguistic and imaginative fancies.

### 5.1. *Linguisticism*

The linguistic tendency rests on the conviction that natural language is already the best medium for representing reality (it is, after all, the most commonly used); thus, that the closer an ontology is to natural language, the more perfect it will be. Methodologically, but without reaching such an extreme, this translates into introducing alleged entities into an ontology on the sole basis that there are natural language expressions using terms which seem to refer to them. For instance, consider whether you agree that there truly are *sakes*, i.e. entities to which expressions such as ‘I did it for John’s sake’ refer. Consider also whether you believe that the terms ‘Santa Claus’, ‘Postman Pat’, ‘Quetzalcoatl the Aztec sky god’, and so on, designate real entities.

If natural language is allowed to dictate an ontological inquiry unimpeded, the ontologist risks making the radical claim that an ontology must stand in a one-to-one correspondence with the elements of natural language (or as close thereto as possible). A motto for this view could be ‘one word, one concept’. The main problem is that this results in an overwhelmingly rich system of concepts with no particular ontological credentials. There is a converse issue which is tied to the phenomenon of differences; for example, in lexical gaps, as between different natural languages.<sup>21</sup> These lead to the proclaiming of absurdities such as that there is, for example, a French ontology and an English ontology. However, the fact that your mother tongue has no ready-made term for a given entity or kind of entity does not prevent you from using or understanding a corresponding concept or talking about the entity in question; for example,

---

<sup>21</sup> Lexical gaps are visible when a language does not have a simple term for referring to certain entities. This can also be illustrated by the languages which lack technical vocabulary existing only in English.

by means of some more complex word formation. Above all, whether or not there is a term in a language does not determine whether or not there is something in reality to which the words or concepts correspond.

At best, natural language can serve as one clue for the ontologist, but it should certainly not be a *criterion* of the correctness of the end-result of her labors. Indeed, if everything in natural language corresponded to some aspect of reality, then there would seem to be no need for ontological structuring or conceptual modeling to begin with.

## 5.2. *Algebraism*

The second tendency I shall warn against can be called algebraism, which takes two forms – namely, conceptual algebraism and theoretical algebraism – depending on the objects considered: on the one hand, concepts (or general terms in a language), and on the other hand, theories. The former relates to issues of content; the latter to issues of structure.

No matter what the object considered, the common assumption behind algebraism is a view according to which a knowledge representation system ought to be maximally complete and contain all algebraically possible objects of manipulation; for example, all set-theoretically possible objects. The idea is that all combinatorial variants fall within the scope of the resultant knowledge representation system. That a total conceptualization can be viewed as a system of concepts with both an initial set and an articulating theory provides the root for the distinction between the two forms of algebraism.

On the concept variant of algebraism, each set of concepts is associated with a set of all Boolean combinations including: disjunctions, such as *dog or cat*; conjunctions, such as *red and square*; complements, such as *non-German*; and conditionals such as *if colorless then magnetic*. Further, the members are regarded as being of equal standing with the members of the initial set. Some examples of dubious objects resulting from such formal motivations are categories such as *green and number* or *horse and violin*. A related case is the postulation of a null region of space in certain mereotopologies, done to ensure that the theory behaves smoothly. Such constructs may be useful for logical purposes, but this does not guarantee them any ontological footing (see Grenon, 2003).

On the theory variant of algebraism, a theory is viewed as a set of sentences to which similar operations are applied as were applied to concepts in the above. The goal, again, is to create an end-result with a

certain kind of algebraic elegance; for example, a complete set of mutually inconsistent theories. The striving for this sort of completeness seems to have been in part responsible for the difficulties faced, for example, by the IEEE's quest for a standard upper ontology (see *IEEE* <http://suo.ieee.org>). The quest for a consensus upper merged ontology, too often, has been transformed into a process of registration of existing separate competitor ontologies.

Our dissatisfaction with algebraism arises from what we believe to be a reasonable skepticism concerning the arbitrary production of fictions for the sake of systematic neatness. We favor hunter-gatherer ontology, based on empirical evidence, rather than armchair ontology based on permutations and combinations. In practice, the latter leads to an explosion of the domain of objects (concepts or theories) which the system needs to handle, and this also leads to a problem of determining relevance and of choosing among all the theoretically possible variants. Most of all, it does not account for those natural segmentations of reality, which fall far short of algebraic neatness in domains like biology or medicine. Even if it is credible that, in some instances, different candidate theories would be retained on an equal footing, retaining an entire articulated range of combinatorially possible theories seems to be without real motivation, and to be alien to the methods of good science. Perhaps preserving the entirety of those theories which stand as alternative solutions to an as yet unsolved and critical problem might be warranted. There might be other considerations for subscribing to elements of algebraism; for instance, having to do with legacy issues (as when a system has been around long enough or used widely enough that it can be amended only with difficulty). But, even here, the combinatorial completeness demanded by the algebraist approach will not prove practically useful.

### 5.3. *Subjectivism and Relativism*

A third tendency could be named *subjectivism*. According to this view, the world is the product of a subject's conceptualization. In an extreme version, there are as many conceptualizations, and as many ontologies, as there are conceptualizing agents. For the subjectivist in knowledge representation, an ontology is nothing but a conceptualization, which may or may not be based on a consensus shared by a plurality of individuals. This raises the obvious problem of radical and permanent interpretation: how do we know that we understand one another, even when we speak a

common language? In other words, it takes the Tower of Babel as a premise.

This view is akin to relativism and to a position defended in philosophy known as perspectivism, according to which no human perspective (no conceptualization) has greater value than another; thus, no human deserves to be called an expert. From there, it is but a small step to claim that no human perspective is closer to the truth. And from there again, it is deceptively simple to issue a platitude such as ‘all perspectives are, if the purpose is only appropriately specified, equally useful’.

An assumption of relativism is clearly active in pragmatist conceptualism, and implicit in many approaches to knowledge representation. It is one of the underlying motivations for taking the problem of interoperability seriously. Relativism is also an obvious motivation for algebraism about theories. There is a link between linguisticism and relativism as well. Many are not afraid to go from differences in languages to differences in conceptualizations, and from there to differences in ontologies and, finally, to differences in the realities in which speakers of those languages have evolved.

## *6. Representation of Reality*

For the reasons discussed, a line should be drawn between external reality and whatever our systems of private or collective representation might be. The focus of ontology is not the latter, but the former. But there is nonetheless some truth behind the motivations of the pragmatist conceptualist: for one thing, we do want our representations to be useful; for another, people do actually differ and disagree in their representations of the world.

How can a knowledge representation system be made rich enough to represent epistemic and cognitive facts? The language of the Cyc system is rich in that sense, including a number of suitable constructs (see (<http://www.cyc.com/cycdoc/vocab/vocab-toc.html>)). It allows expression of the fact that an agent – for instance, a doctor – will believe that a patient is in a condition of a certain type, while simultaneously allowing the diagnosis to have only a provisional nature and allowing for a separation of the true condition of the patient from the conjectured one. This sort of approach is possible in principle because, while beliefs and other cognitive stances are real, they do not inform reality or the portions of reality that

they address or pretend to address and in which they are found, but only the cognitive domain itself.

If the representation of epistemic stances requires an accurate realist account of the reality toward which these stances are directed, where shall we find objective sources of knowledge about cognition-independent reality? At this stage three maxims suggest themselves:

- i) The primary source of knowledge is reality.
- ii) The domain being represented is a part of reality.
- iii) The expert knows the reality at hand (that is why we call him an expert).

In many domains, the best experts we have at hand are scientists (natural and social). I will take the scientist as the paradigmatic expert. But, of course, in many domains we might turn, rather, to an engineer. One remark, in relation to what has been said before, is that in a scientific domain, knowledge representation would be carried out on the basis of scientific theories, but it would not be a representation of these theories. Another point, of more importance, is that reliance on science aims, among other things, to provide ontology itself with a methodology and a goal that follows scientific standards.

A realist methodology trusts experts for matters of ontology. It asserts, for the purposes that are of prime concern in the present volume, that knowledge of reality can be obtained through an inquiry of the sort conducted via application of the scientific method (that is, 'knowledge' in the non-philosophical sense of *very well justified beliefs*; see the Introduction for this distinction). Empirical inquiry, on this view, can furnish knowledge of the world. Now, it is a fact that our ability to engage in such inquiry evolves and progresses. It must be, therefore, that our theories and our concomitant understanding of the world can be subjected to testing and revision; it is this that secures the possibility of their progress. Indeed, there is hardly any cutting-edge theory which is not the subject of contemporary debate and revision.

This means that the ontologist who is concerned with reality must make provisions for the evolution and refinement of the views underlying her work. This is the methodological doctrine of *fallibilism*, which makes room for approximations and errors in state-of-the-art knowledge. If we are looking for accuracy, we must accept the possibility of being in error, in the hope that we may follow our inquiries toward more refined

understanding in the future. We must be ready to abandon views and introduce unforeseen elements, even if this requires us to redo laborious work.

Error is not the only source of unease for the ontologist; indeterminacy is yet another issue. Sometimes empirical inquiry remains undecided and offers incompatible, but equally credible, theories. Consider, for example, the alternative between the wave and corpuscular theories of light in the 1930s. This means that, as ontologists, our methodology must also make provisions for the possibility of maintaining equally legitimate realist perspectives on reality. This will generally occur when phenomena can be accounted for by independent and contradictory stances, not only scientific, but also philosophical ones.<sup>22</sup>

It is important to bear in mind that this does not amount to the thesis that any view of reality is legitimate. Rather, it is a *realist perspectivalism* (as contrasted with *relativist* perspectivalisms of a more traditional sort) which asserts that, at least, some views of reality are legitimate (see Chapter 6). It is also a *realist adequatism*, which means that it denies the doctrine of reductionism, according to which the putative plurality of legitimate views of reality is not to be eliminated through the reduction of all such views to one central basic view; for example, the view of microphysics. To establish which views are legitimate, we must weigh them against each other and against their ability to survive critical tests when confronted with reality as, for example, in scientific experiments; including not only experiments in microphysics but also in biology, medicine, and other branches of science. Those concepts and conceptualizations which survive are then transparent to reality, to use the somewhat metaphorical expression of Smith (2003). More generally, we are concerned with those views that are *veridical under a given perspective* in relation to a particular domain. This means, among other things, that we can do the ontology of post offices without worrying about elementary particles.

## 7. Conclusion

Philosophical and ontological analysis has to be performed by the knowledge engineer in order to provide a sound basis for her knowledge representation; even when such representation is conceived as conceptual

---

<sup>22</sup> Consider the tangled issue of endurance versus perdurance in the metaphysical debate over persistence in time. See Kanzian, 2008.

modeling. More precisely, knowledge representation systems ought to be conceived as representations of *reality*, and not as systems of representations of concepts, or as mere models with no foothold in the real world of what happens and is the case. For these reasons, an ontological inquiry in its philosophical sense, one which addresses reality, must be the basic methodological principle of a sound approach to knowledge representation.

A complete knowledge representation system should be able to accommodate and articulate what may be a multiplicity of legitimate views of reality. Which alternative theories or perspectives on reality are useful for the purposes of the knowledge engineer has to be established on the basis of a realist and fallibilist methodology, and this task, which comes close to the tasks of empirical inquiry, may be the most difficult and challenging to accomplish. What matters is that the knowledge engineer should bear in mind: first, that her target is reality; second, that formal simplifications, modeling tricks, and shortcuts of various other sorts, may be detrimental to the ultimate goal of accounting for reality.

Shimon Edelman's Riddle of Representation (Edelman, 1998) reads as follows:

Q: two humans, a monkey, and a robot are looking at a piece of cheese; what is common to the representational processes in their visual systems?

A: the cheese, of course. (Meaning it ain't in the head; putting it there is just as pointless as making a sandwich with a picture of a piece of brie.)

Of course, this is a metaphor; you don't take some real entity – for example, a lump of goat cheese – and put it in a database. But this is a methodological point. The knowledge base ought to be a reproduction of reality. Knowledge representation and ontology are not, strictly speaking, representations, they are *re*-representations of reality. If the knowledge engineer were to hold fast to only one methodological proposal, then it should be this: *the world itself* should be included in a knowledge representation or ontology.