Formal versus Material Ontologies for Information Systems Interoperation in the Semantic Web

Robert M. Colomb

Technical Report 16/02 ISIB-CNR
Padova, Italy, November, 2002

National Research Council
Institute of Biomedical Engineering
ISIB-CNR
Corso Stati Uniti, 4
35127 Padova

Robert M. Colomb
School of Information Technology and Electrical Engineering
The University of Queensland
Queensland 4072 Australia
colomb@itee.uq.edu.au

Work performed partly while visiting LADSEB-CNR; Corso Stati Uniti, 4; Padova, Italia,

Abstract

Information systems ontology is intended to facilitate interoperability among the many applications which are now becoming available on the Internet. In particular, it is intended to facilitate the development of intelligent agents which can automate a large part of the task of a user achieving some end employing multiple autonomous applications. A large number of ontologies exist supporting specific kinds of interoperation among selected, generally mutually aware, applications. The intent of the upper ontology movement is to develop an abstract description of what there is in the world, in an application-independent form, which can be used both to help build specific ontologies and to help in finding common ground among them. This paper argues that for the purposes of information systems interoperation and the semantic web there is a distinction in upper ontologies between formal and material ontologies, based on analogies with concepts in Kant's synthetic a priori, and that formal ontologies whose focus is on how we see the world are more likely to be successfully developed in the absence of applications than are material ontologies, which attempt to catalog the world a priori.

Formal versus Material Ontologies for Information Systems Interoperation in the Semantic Web

Robert M. Colomb
School of Information Technology and Electrical Engineering
The University of Queensland[1]

Work performed while visiting LADSEB-CNR[2]

Abstract

Information systems ontology is intended to facilitate interoperability among the many applications which are now becoming available on the Internet. In particular, it is intended to facilitate the development of intelligent agents which can automate a large part of the task of a user achieving some end employing multiple autonomous applications. A large number of ontologies exist supporting specific kinds of interoperation among selected, generally mutually aware, applications. The intent of the upper ontology movement is to develop an abstract description of what there is in the world, in an application-independent form, which can be used both to help build specific ontologies and to help in finding common ground among them. This paper argues that for the purposes of information systems interoperation and the semantic web there is a distinction in upper ontologies between formal and material ontologies, based on analogies with concepts in Kant's synthetic a priori, and that formal ontologies whose focus is on how we see the world are more likely to be successfully developed in the absence of applications than are material ontologies, which attempt to catalog the world a priori.

Categories and Descriptors: C.2.4 [Distributed Systems] Distributed Applications, Distributed Databases D.2.12 [Software Engineering] Interoperability – Data Mapping; H.3.5 [Information Storage and Retrieval] Online Information Services – Data Sharing and Web-based Services; H.2.1 [Database Management] Systems

Additional Keywords and Terms: ontology, semantic heterogeneity, semantic web

Introduction

Information systems ontology is intended to facilitate interoperability among the many applications which are now becoming available on the Internet. In particular, it is intended to facilitate the development of intelligent agents which can automate a large part of the task of a user achieving some end employing multiple autonomous applications.

> A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them … an ontology is an explicit representation of a conceptualization.
>
> [Gruber 1993] p 1

It has its roots in artificial intelligence, information systems modeling and software engineering, and is making a common cause with philosophical ontology.

> …what we now refer to as philosophical ontology has sought the definitive and exhaustive classification of entities in all spheres of being … including the types of relations by which entities are tied together
>
> [Smith and Welty 2001] p 2

[1] Queensland 4072 Australia colomb@itee.uq.edu.au
[2] Corso Stati Uniti, 4; Padova, 13512 Italia

A large number of ontologies exist supporting specific kinds of interoperation among selected, generally mutually aware, applications. Many of the major ontologies predate the common use of the concept, and are called classification systems, thesauri and coding systems, among other terms. They support ontological commitment under the definition of [Gruber 1993] "We say an agent commits to an ontology if its observable actions are consistent with the definitions in the ontology" (p.2)

These established ontologies range from comprehensive library systems like the US Library of Congress or Dewey Decimal classification systems, to more narrow systems like the Standard Industrial Classification system maintained by the US Department of Labor widely used for economic statistics and the ACM (Association for Computing Machinery) Keyword List used for identifying content in publications in the information technology field. The comprehensive systems are huge, with many hundreds of thousands of entries in deep hierarchies. Some of the specific systems are also huge, for example the SNOMED system used for classifying medical treatment incidents, which has more than 150,000 terms organized into 11 semantic groups or facets. A particular incident is classified by a selection of terms from different facets, so the system is capable of generating more than 10^{40} distinct descriptions. More detail about these and other systems can be found in various sources, for example [Colomb 2002], as well as the Internet.

Since there has been so much experience constructing application specific ontologies, there is a discipline of design and engineering for them, described for example in [Rowley 1992] or [Colomb 2002]. [Gruber 1993] presents some specific guidelines.

To [Gruber 1993], an ontology is a shared terminology supporting a class of applications. Instead of designing ontologies for specific applications, the quote from [Smith and Welty 2001] above shows that the intent the upper ontology movement is to develop a description of what there is in the world, in an application-independent form, which can be used both to help build specific ontologies and to help in finding common ground among them.

Note that the intended range of applicability for upper ontologies is wider than information systems, including natural language processing and distributed artificial intelligence. The comments in this paper are intended to apply particularly to the information systems type of application, which is called by some people the Semantic Web.

The problem of interoperation among autonomous applications has been with us for many years. In the late 1980s and early 1990s it was the focus of much research in the database community under the guise of federated databases (see [Sheth and Larsen 1990], for the canonical synthesis of the problem). The research in this field tended to be to find methods to develop views so that the structural differences among the different database schemes could be resolved. However, practical federated database systems tended to fail due to the problem of semantic heterogeneity – different applications mean different things by similar terms. The following simple example shows some of the deep issues involved.

Suppose there are two systems A and B which have exactly the same conceptual model, consisting of a single entity named “enemy targets”. The semantics of the systems are the same – in case of war, the table containing the population of the entity is the list of targets at which missiles are to be fired. Suppose that the two countries owning the respective systems are enemies, so that each list of targets contains many located in the territory of the other.

Suppose now the two countries become friends, to the extent that they decide to integrate their two defense information systems. Suppose they entrust this task to an automated tool which identifies entities with the same semantics and constructs a system from the meet of the two theories. It is clear that a simple tool which identifies common terms and matches them could do this task fairly easily.

But what about the populations of the two tables? It should be clear that the two countries would be unhappy with a combined defense system which had targets, even one, which was within either country – they are after all now friends.

The problem is that although the terminology of the two systems were the same and the semantics on the surface were the same, the semantics was indexical – “enemy target” is interpreted as “target in the territory of MY enemy”. When the two countries become friends and integrate their systems, the index changes – MY is now different from the MY of either original system. In order for a semantic integration tool to be able to cope with applications like this, it would need to have a very sophisticated ability to understand and manipulate indexicial semantics, and also that the two systems would have to be described in such a way that the indexical semantics is apparent. Assuming an appropriate description making manifest the indexical semantics, it is plausible that an automated tool could integrate the two systems, although such a tool is not necessarily presently practicable. But it is unlikely that the systems would be described appropriately, since the indexical nature of the semantics is implicit in the application. To make it explicit is exactly what the human software engineers would have to do to successfully integrate the two systems. An automated tool would almost certainly fail.

Semantic heterogeneity as it affects information systems operation is examined in some depth in [Colomb 1997].

In practice, the effective demand for interoperating systems comes from groups of organizations which share considerable semantics and have some business reason to interact with each other. To achieve interoperability, the groups must first agree on the scope of their interoperation, then decide entity by entity, relationship by relationship, attribute by attribute, how their systems relate to the proposed interoperation. Very often the systems must be changed to achieve the result.

This is not a technical process, but mainly a matter of business negotiation. They will typically represent the results of their agreement using some kind of ontology. The application-specific ontologies described above are all of this kind.

The experience of federated databases is relevant to the new field of ontology. One statement of the goal of ontological research is

> An ontology for a possible world – a catalog of everything that's in the world, how it's put together, and how it works
>
> [Sowa 1984], p. 294

If this project were to succeed, then the ability to integrate existing information systems would seem to follow, since each existing system could be referred to the ontology.

But it is hard to see how the issues of semantic heterogeneity would go away, based as they are on the context-relatedness of the organizational interpretations of the concepts represented in the information systems. Below we adduce a recent contribution to the understanding of the kinds of things represented in information systems, namely John Searle's [Searle 1995] concepts of the extremely context-dependent *institutional fact* and of *background*, the large amount of poorly articulated contextual knowledge which blocks bizarre interpretations of ordinary utterances, supports context, and is a major contributor to the pervasiveness of semantic heterogeneity. We will see that it is very unlikely that one could build a useful a priori ontology of institutional facts supporting interoperation, and will recognize that nearly all the content of most information systems consists of records of institutional facts.

So the possibility exists that attempts to support information systems interoperation by building catalogs of everything are doomed to failure. But if we examine the actual upper ontologies being developed, many of them are extremely abstract. [Gruber 1993] argues that for engineering purposes, an ontology is an ontologically homogeneous collection of terms and relationships. Perhaps it makes sense to identify a subclass of term which can function independently of applications, and thus succeed in the face of strong semantic heterogeneity. The purpose of this paper is to examine the design of ontologies from this point of view.

The main philosophical underpinning of this paper is Immanuel Kant, specifically the idea of the synthetic a priori from the *Critique of Pure Reason*. Kant may be an "enemy of metaphysics" [Smith and Welty 2001], but he may turn out to be a friend of information systems upper ontology. We do not argue for Kant's entire view, but take his approach as a way of classifying upper ontologies "at the joints", as Plato advised in the *Phaedrus*, in order to obtain a clear distinction which can be used as a basis for discussion.

In the following, we first examine some of the upper ontology work, recognizing of course that the efforts are ongoing. We then look at problems of semantic heterogeneity in some more depth, taking advantage of Searle's concept of institutional fact, with the intent of casting doubt on the feasibility of the application-independent upper ontology as strongly stated in the quotations above. We then turn to what we argue is an analogous problem addressed by Kant in the *Critique of Pure Reason*, and his solution, transcendental idealism, or the study of the formal conditions under which objects in the world can be experienced by the human mind. We then proceed to show that some of the upper ontology efforts, namely formal ontologies, are strongly analogous to Kant's solution. This leads to the conclusion that even though one may be pessimistic about the feasibility of the "theory of everything" upper ontology projects, there is hope for the success of the more limited formal ontology approach for a more limited aim which recognizes the prevalence of semantic heterogeneity.

Sample of formal ontology efforts

This section examines several upper ontology efforts, without intending to be comprehensive. We look at Cyc, SUMO, OntoClean/DOLCE, GOL, the Bunge-Wand-Weber (BWW) ontology, and the top level of WordNet.

Cyc

Cyc is an enormous system, having been developed over many years.[3] To get some idea of what it looks like, we will look at the structure of a middle-level concept relevant to information systems interoperation, that of *transaction*, in Figure 1.

> **Transaction -** The collection of actions performed by two or more agents cooperating (willingly) under some agreement wherein each agent performs actions in exchange for the actions of the other(s). Note that a case of attack-and-counterattack in warfare is *not* a Transaction; nor is fortuitous cooperation without agreement (e.g. where a group of investors who, unknown to each other, all buy the same stock almost at once, thereby driving up its price). For transactions involving an exchange of user rights (to goods and/or money) between agents, see the specialization of ExchangeOfUserRights
>
> *Subtype of PurposefulAction, CooperativeEvent*
>
> **PurposefulAction** a specialization of both Action and AtLeastPartiallyMentalEvent. Each instance of PurposefulAction is an action consciously, volitionally, and purposefully done by at least one actor.
>
> *Subtype of Action, AtLeastPartiallyMentalEvent*
>
> **Action** *subtype of Event*
>
> **Event** *subtype of Situation-Temporal, IntangibleIndividual*
>
> **Situation-Temporal** *subtype of Situation, TemporalThing*
>
> **TemporalThing** *subtype of Individual*
>
> **Individual** *subtype of Thing*
>
> **Thing**

Figure 1: Some of the concept *transaction* from OpenCyc

We have included some of the comments which describe the details of the various parts of the concept. The rules implicit in the comments are represented in the system, and can be reasoned with. Cyc is explicitly intended to be a theory of everything.

SUMO

The Standard Upper Merged Ontology (SUMO) project is also intended to be a theory of everything, but at a much more abstract level than Cyc. It is much smaller. However, it goes down to the level of *transaction*, and a little below (*financial transaction*, then *buying*, *selling*). The concept is shown in Figure 2.

[3] http://www.opencyc.org/

Notice that the two definitions of *transaction* are similarly rich, but different in detail. The SUMO concept includes change of possession, while the Cyc concept requires exchange of actions under an agreement. The superclass structure is different in detail, but broadly similar.

Transaction - The subclass of ChangeOfPossession where something is exchanged for something else. *subclass of* ChangeOfPossession

ChangeOfPossession - The Class of Processes where ownership of something is transferred from one Agent to another. *subclass of* SocialInteraction

SocialInteraction - The subclass of IntentionalProcess that involves interactions between CognitiveAgents. *subclass of* IntentionalProcess

IntentionalProcess- A Process that is deliberately set in motion by a CognitiveAgent. *subclass of* Process

Process - Intuitively, the class of things that happen rather than endure. A Process is thought of as having temporal parts or stages, and so it cannot have all these parts together at one time (contrast Object). Examples include extended 'events' such as a football match or a race, events and actions of various kinds, states of motion and lifespans of Objects, which occupy the same space and time but are thought of as having stages instead of parts. The formal definition is: anything that lasts for a time but is not an Object. Note that a Process may have participants 'inside' it which are Objects, such as the players in a football match. In a 4D ontology, a Process is something whose spatiotemporal extent is thought of as dividing into temporal stages roughly perpendicular to the time-axis. *subclass of* Physical

Physical - An entity that has a location in space-time. Note that locations are themselves understood to have a location in space-time. *subclass of* Entity

Entity - The universal class of individuals. This is the root node of the ontology.

Figure 2: Concept of *transaction* from SUMO

OntoClean/DOLCE

OntoClean's top level, called DOLCE [Gangemi et al. 2002b]. is a very much smaller and much more abstract system, the top levels of which are shown in Figure 3. This ontology does not go down to the level of *transaction*, but a transaction would be an *event*. (The supertype *occurrence* is the only part of the ontology where changes can be modeled.)

- Entity
  - Abstract
    - Fact
    - Set
    - Region
      - Temporal Region
      - Physical Region
      - Non-Physical Region
  - Endurant
    - Quality
      - Temporal Quality
      - Physical Quality
      - Non-Physical Quality
    - Substantial
      - Physical Substantial
        - Aggregate
          - Amount of Matter
          - Arbitrary Collection
        - Feature
          - Relevant Part
          - Place
          - Physical Object
      - Non-Physical Substantial
  - Perdurant/Occurrence
    - Event
      - Achievement
      - Accomplishment
    - Stative
      - State
      - Process

Figure 3: OntoClean/DOLCE Top Categories

GOL

General Ontological Language (GOL) [Degen et al. 2001] is a structure-oriented very abstract ontology, whose basic concepts are shown in Figure 4. Interesting things like transactions are constructed from these components held together with relational moments. These complex structures are called *situoids*, which include situations. Dynamics of situoids are *processes*, which are resolved into *events.* So a transaction would be described as a process within a situoid.

- Entity
  - Set
    - Extension
  - Urelement
    - Individual
      - Chronoid (temporal duration)
      - Topoid (spatial region)
      - Substance
      - Moment
        - Quality
        - Relational Moment
    - Universal
      - Relational Universal

Figure 4: Class hierarchy of GOL

BWW

The Bunge-Wand-Weber ontology [Weber 1997] in its present form comes from the information systems community rather than artificial intelligence. Its key concepts are summarised in Figure 5. There are in addition a number of derived concepts expressed as predicates, often on the history of a thing. Some of these predicates (e.g. that one complex object is an input to another) require a concept of causality.

Thing
Property and Attribute
State of a Thing: at a point in time, the attributes of a thing have values.
Event: change of state in a thing.
History of a Thing: a sequence of events in a thing.
Type/Class and Subtype/Subclass
Composite thing: is composed of (made up of) things other than itself. Things in the composite are *part-of* the composite.

Figure 5: Key concepts in the BWW ontology

WordNet

Wordnet [Felbaum 1998] is a structured collection of English language terms. Some of the structures are hierarchical, leading to the top level shown in Figure 6. Although not strictly an ontology, it is widely used and cited in the ontological research community because of its richness of structure, size and availability. A search in WordNet[4] shows that a *transaction* is an *act* through the intermediate *group action*.

[4] http://www.cogsci.princeton.edu/~wn/online/

Abstraction
Act (human)
Entity
Event
Group
Phenomenon
Possession
Psychological Feature
State

Figure 6 The top level of WordNet

Semantic heterogeneity

Semantic heterogeneity is something that often occurs when we have two or more systems of concepts covering the same universe of discourse, when something expressed in one system cannot be exactly expressed in the other.

We exclude situations where something simply expressed in one system can be exactly expressed by a set of calculations in the other, what in database systems is called a *view*. For example, street addresses are expressed differently in New York, Sydney and Venice. In New York, an address is uniquely identified by street and number. In Sydney, street and number are unique only within suburb (of which there are about 1000). In Venice, addresses are identified by number within district (of which there are about 10). Streets are irrelevant. Sydney addresses could be represented in a New York-oriented system by say concatenating the suburb name with the street name. For Venice, one way the address could be represented in a New York-oriented system is by treating the district as a street name. Alternatively, if we include the street name in the information we have about the address, we could use the Sydney-oriented system and its conversion to the New York system.

In a somewhat different vein, Eskimos are said to have many words for snow. However, it is fairly likely that possibly complex English locutions could be constructed to get close enough to any one Eskimo word that we would not say that we have encountered semantic heterogeneity, which in this context would be a concept which can be expressed by the Eskimo but not by an English speaker.

In the world of medium-sized fairly concrete objects favored as examples by most writers in ontology, semantic heterogeneity can be hard to see unequivocally.

But information systems are not generally about medium-sized fairly concrete objects. Most of the content of most information systems are records of what [Searle 1995] calls *institutional facts*. We will describe the concept of institutional fact and then the related concept of background, showing a number of situations where semantic heterogeneity is present and where it may be absent. Following this, we will argue on a priori grounds that a useful application-independent ontology for systems of institutional fact is unlikely to be possible.

We have seen in the Introduction an example of semantic heterogeneity due to the indexical nature of the concept of *enemy target*. This concept is an example of an institutional fact. Searle calls concrete physical objects which exist independently of humans *brute facts*. A brute fact which has significance for a social group in a certain context is called an *institutional fact*. Searle uses the formula *X counts as Y in context C* to organize institutional facts.

Institutional facts are generally the results of speech acts. The quintessential speech act is naming. In many societies a baby gets its name by someone filling in a form (the filled-in form is a brute fact). But satisfaction of contextual rules is necessary for that brute fact to count as giving the baby its name. The person filling in the form must be authorized (generally one of the parents). The form must be handed in to the proper registry office, and must be properly processed by the staff of that office, including the making of a permanent record and the issuing of an authorized copy of the record (birth certificate).

Contemporary society requires a multitude of institutional facts to operate. In particular, the operation of most organizations consists almost exclusively of the creation of institutional facts. Buying and selling are speech acts, the results of which are institutional facts recording change of ownership. The records of buying and selling are therefore records of institutional facts. The orders given to pack and ship goods are, too. So are the orders given to purchase materials and manufacture goods. Staff being paid, students being enrolled in educational institutions and receiving qualifications, regulations being enacted and enforced, all are speech acts which create institutional facts. Information systems, including web sites, are almost exclusively concerned with records of institutional facts.

If it is difficult to see semantic heterogeneity in the world of middle-sized concrete objects, it is difficult to avoid semantic heterogeneity in the world of institutional facts. Anyone who has tried to combine independently compiled statistics, or to place a student from one educational program in the middle of another, or to evaluate complex service quotations, or to convert from one computer system to another, or to change jobs within similar industries, or to be involved in a merger, or to move from one country to another, could provide many examples.

Indeed the surprising thing is when it is possible to interoperate with two systems of institutional facts *without* semantic heterogeneity. Integrating statistics is hard, but it is easy to compare performance of companies traded on a given stock exchange. The stock exchanges require standardized financial reporting, and are able to enforce the standards. Converting from one computer system to another is hard, but it is not too difficult to convert from one web browser to another. Most of what web browsers do is mandated by standards of the World-Wide Web consortium. Evaluating complex service quotations is difficult, but if one company buys a service from another they agree on what that service consists of and what its price is. This agreement is the result of negotiation and is expressed in a contract.

Besides the concept of the complexly context-dependent institutional fact, [Searle 1995] describes another concept which makes a large contribution to semantic heterogeneity, namely that of *background*.

> The literal meaning of any sentence can only determine its truth conditions or other conditions of satisfaction against a background of capacities, dispositions, know-how, etc., which are not themselves a part of the semantic content of the sentence.
>
> [Searle 1995] p. 130

Background is a complex topic, consisting in Searle's treatment of many aspects, one of which is "dramatic category" (p. 134). A dramatic category is our expectation of the behaviour of objects in our environment, and of how various kinds of situations are supposed to develop.

A simple semantic-web type example will both illustrate the concept and its contribution to semantic heterogeneity. Consider the application of on-line search for apartments to rent, which is now fairly common in many cities throughout the world. The advertisements and descriptions of apartments in Brisbane, Australia, are much the same as those in Padua, Italy. Allowing for the difference in language (English versus Italian), and some variation in description (eg in Padua, many apartments are heated from a common plant, and therefore the rent is often quoted *plus expenses*, whereas in subtropical Brisbane apartments are generally heated if at all by space heaters so the cost of heating is not paid to the owner) the applications are very similar. So much so, that it is not difficult to imagine a software package which could be configured for either city. The two applications could therefore be running very successfully with exactly the same ontologies.

Suppose now someone comes from Brisbane to spend a year in Padua. Being used to the Brisbane application, he is delighted to find the same application running in Padua, and happily searches for an apartment in the right location, price range, and of the right type. Finding one, he goes to the agent and receives a nasty shock. Almost no apartments are available for a lease period as short as one year – the usual term is four years, although some owners will settle for two. In Brisbane, almost all leases are for six months. Suppose he does find a landlord willing to give a very short lease, he has another shock when the agent presents a bill for his commission. In Brisbane, the commission to the rental agent is always paid by the owner. In Padua, always by the renter. Nowhere in either ontology is the term of lease or who is liable for the agent's commission mentioned. These are part of the background. Residential leases and estate agents behave differently in the two environments. An automated agent would produce something of a disaster with this problem.

Of course, it would be possible to make any two such systems interoperable by identifying and representing relevant parts of the background, thereby expanding the ontology, but this is a systems development effort needing to be done by humans, not something that an agent supported by an ontology could be expected to accomplish. It could for example require changes to business practices. If there came to be a large number of medium-term visitors to Padua sufficient to make it worthwhile to investigate and represent the relevant aspects of the background, then some owners might decide to change their practices to cater to this market.

It might be objected that this very fine-grained heterogeneity is too fine for consideration when building top-level ontologies. However, the problem occurs at any level of granularity. Imagine two applications trying to interoperate with the concept *transaction*. One application uses the Cyc ontology (Figure 1), and the other the SUMO ontology (Figure 2). The SUMO ontology requires that whatever is subsumed by *transaction* involves a change of possession of something, whereas the Cyc concept does not. On the other hand, the Cyc concept requires that there be some agreement under which the agents are cooperating, whereas the SUMO ontology makes no mention of agreements in this context.

Let us look at the concept of *transaction* as we might encounter it in trying to build applications where information systems interoperate with each other. The term is used in the technology of database management systems to refer to *an atomic persistent change of state*. We may exclude this meaning on the grounds that things happening within computer systems are not relevant – we want out ontologies to describe the world.

We would pretty clearly want the concept to apply to

- the interaction with Amazon.com resulting in the placing of an order and the supply of credit card details.

This would be a transaction for Cyc, although it is hard so see where the agreement is, especially if the order originates outside the USA so that there is no overriding institutional context, but not for SUMO, since nothing has changed possession.

To bring it under the SUMO concept, we need to include

- the subsequent packing and shipping of the order, its receipt by the purchaser in good condition, and the acceptance of the credit card charge by Visa.

Further examples are:

- the interaction with Medline resulting in the placing of a query and the return of a collection of abstracts

Nothing changes possession here, so the SUMO concept is hard to apply, and the case for an agreement is harder to make, which makes it harder to apply the Cyc concept.

- the borrowing and ultimate return of a book by the University of Queensland library from the University of Sydney library (interlibrary loan), on behalf of an academic (who must also borrow and return the book from the University of Queensland library).

Here, the agreement is pretty clear, so the Cyc concept applies. However, when the transaction is completed, the whole point is that no change of possession has occurred, so again the SUMO concept is hard to apply.

Finally,

- the interaction between the 2002 Salt Lake City Winter Olympics results processing agent and the agents responsible for the maintenance of results on multiple web sites ultimately completing with the information that the medal results for ice hockey have been recorded on all sites.

It is hard to see how the agents are "performing actions in exchange for the actions of each other", since all the agents do what they are supposed to do when they are triggered, and do not model each other. Especially if all the sites are operated by the same organization, the element of agreement is very hard to argue for. So the case for applicability of the Cyc concept is weak. Also, the SUMO concept is hard to apply, since nothing ever changes possession.

On the other hand, these examples are very like each other from a computing point of view. They are all instances of successful interoperation among information systems, where a cumulation of database transactions accomplishes something in the world (creates an institutional fact in each instance). So it is reasonable for an information technology professional to want to treat them all as instances of the same concept, and to expect any ontology to permit this.

Note that all the examples, including the technical database transaction, fit under the concepts in the OntoClean/DOLCE, GOL and BWW ontologies (Figures 3, 4, and 5 respectively). These three ontologies have little if anything to say about the business aspects of the activities, concentrating on structural features and the interaction of time and state. The relevant OntoClean/DOLCE (occurrence), GOL (process situoid) and BWW (event) concepts are very much broader than the Cyc or SUMO concepts.

They also fit under the concept *transaction* in WordNet (Figure 6), if we are allowed to interpret "group" in "group action" as a group of agents rather than as a group of humans. (This is in fact questionable, as WordNet has also the concept of "causal agent" in a different branch of the ontology.) If we do accept the interpretation of "group" as not necessarily human, then fortuitously the concept in WordNet is classified in a very abstract way, so is very similar to the OntoClean/DOLCE *accomplishment*, and the eventual structure constructed for the purpose from the elements of GOL.

This example has shown some of the difficulties with application-independent upper ontologies, but is a negative argument. It is always possible to counter such an argument with the claim that the example used is unusual, and it is always possible in principle to fix the systems examined so that the problem will go away.

The following positive argument based on how institutional fact types are created will attempt to show why on reflection we would not expect to be able to make a comprehensive ontology of institutional facts.

A new type of institutional fact is created by an agreement among a possibly small group of people, sometimes as few as one. In fact, one of the reasons institutions are created is to enable a small number of people to make these kinds of decisions in particular domains, otherwise societies would not scale. Of course, the people affected in the organization have the capability to ignore, evade or subvert a decision made by the proper authorities, so there is a sort of implicit agreement among a large number of people, but these people often do not participate in the deliberations leading to the creation of the new type of institutional fact.

For example, as a result of a sudden budgetary crisis my Head of School can unilaterally decree that henceforth the School will pay for travel to conferences only on certain complex conditions, and instruct relevant administrative staff to develop forms and information systems to record the data necessary to evaluate the conditions. We now have a new institutional fact type – *travel to be funded by the School of Information Technology and Electrical Engineering at The University of Queensland*.

Note that it is very much easier to create a new institutional fact type than it is to introduce a new word into a natural language, even in a smallish group such as researchers in a given subdiscipline. For a proposed new word to become part of the language it must be found useful and adopted by relatively many people, and continually used for a relatively long period of time. New words are created frequently, but most neologisms fail to be taken up.

So institutional fact types can be easily created by limited groups of people in an enormous variety of contingencies. Why would we expect weakly constrained free inventions to conform to a small number of basic categories? There are situations where this is so.

Consider the game of chess. It has a set of rules which are constitutive, in that if you don't follow the rules you aren't playing chess. So long as you are playing chess, your choice of move is entirely free. The number of possible chess games is vast, but there is a well-established ontology for classifying them. There are a small number of openings (Ruy Lopez, King's Gambit, King's Indian, etc.). There are a small number of end games (King and pawn versus King, Rook and King versus King, etc.). There are well-established ways of evaluating middle games (relative power of forces, control of center, mobility of pieces, etc), and a small catalog of attacks (pin, discovered check, fork, etc.).

A system of free creations like chess can have such an ontology because the constituting rules are fixed outside any game. We can say that the rules are *transcendent* with respect to the games. The opposite of transcendent is *immanent*. A move in a game is immanent, because it is chosen for local reasons within that game. Suppose the rules of chess were immanent. Before the start of a game the players would agree on the rules for that game. It is hard to imagine a game like chess with immanent rules, but suppose one changed the number of dimensions of the chessboard. This could radically change the mobility of pieces in the middle game. Or could change the knight's move by doubling its size. That would change the evaluation of mobility of pieces and also the openings. Changing what is meant by winning would at least make great changes in the end games. Introduction of arbitrary moves and resurrection of pieces would eliminate the distinctions among opening, middle and end games. Making the rules immanent would destroy the possibility of the kinds of ontologies we have for the game with transcendent rules.

Social institutions are immanent, not transcendent. They constrain each other, but there is no rule that cannot be changed. My Head of School can change travel policy, but can't hire and fire as he pleases. But the university and the union can agree to change employment conditions, or they can be changed by act of Government. The European foreign exchange markets can impose constraints on governmental actions, but the introduction of a common currency can abolish the foreign exchange market. So there is no a priori transcendent system which would lead us to expect that there would be an a priori ontology for institutional facts.

We can conclude from this section that it is hard to see how ontologies created independently of applications can greatly simplify the problems of information systems interoperation/ semantic web applications requiring interoperation involving institutional facts and the background needed to interpret them. Semantic heterogeneity must be resolved before we can proceed, which implies a commitment to a class of application operating in a specific context. We can also conclude that we can't see any reason to expect such ontologies to be possible at all outside of specific contexts.

On the other hand, when we have resolved the semantic heterogeneity among a group of applications, we generally need to represent the agreement with an ontology created for the purpose. It would be very advantageous to not have to start from zero every time. The idea of having a bank of re-usable concepts is very attractive. The question becomes whether we can find aspects of the application-independent ontologies which lend themselves to re-use.

The synthetic a priori of Kant

Kant faced a problem something like our question as to the possibilities for application-independent upper ontology, namely what it is possible to know independently of experience. He developed his proposals in *The Critique of Pure Reason* [Kant 1998].

There had been a long tradition of metaphysics (sketched briefly in [Smith and Welty 2001]), in which people attempted to catalog the kinds of things in the world a priori, by reasoning. There had been the recent introduction of scientific thinking whose representation in philosophy was empiricism, a major exponent of which was David Hume. In this understanding, all knowledge comes from the external world. Kant thought that on the one hand the schools of metaphysics had no way of getting agreement one among the other, nor had produced anything of use outside the particular school. On the other, it didn't seem reasonable that nothing could be known a priori.

As shown by the choice of citation, there is some continuity between metaphysics and information systems upper ontology, at least by analogy. The analogy to empiricism is the great variety of application-specific ontologies in existence and being built. Our problem as stated at the end of the last section is to see if there are aspects of the application-independent ontologies which can be re-used in construction application-specific ones.

Kant divided possible knowledge into two classes, analytic and synthetic. Analytic knowledge is necessarily true. Today, analytic knowledge is generally thought to be limited to pure mathematics and pure logic. Synthetic knowledge is not necessarily true, but is contingent on how the world actually is, even though it could in principle have been different. Netscape could have won the browser war.

Synthetic knowledge is either a priori or a posteriori. A posteriori knowledge is knowledge derived from experience, for example the results of scientific investigation. Kant's interest was in the synthetic a priori, knowledge about the world which can be known independently of experience.

Kant looked for the synthetic a priori in aspects common to all human knowledge, that is to say on how the human mind, designed as it is, constructs the representations of things in the world which are the basis of the judgments resulting in knowledge. In other words, Kant looked to the form of knowledge rather than to content.

Content of knowledge for Kant is phenomena as distinguished from noumena. *Noumena* is the "thing in itself" which is only available to humans as *phenomena*, which are sensory representations following the laws of the synthetic a priori.

The analogy of this solution in the present problem domain is to look to ontologies which specify the form of information structures rather than the content. I argue below that three of the ontologies canvassed above are formal in this sense, namely OntoClean/DOLCE (except for the physical/non-physical distinction), GOL and BWW. The others contain formal concepts mixed in with content-related concepts.

To give more shape to this proposed solution, we will sketch Kant's synthetic a priori and draw out more detailed analogies between it and the problems of information systems interoperability. For Kant, the synthetic a priori included space, time and what are called the *categories*.

First, we look at space:

> Space is the form of all appearances of outer sense, i.e.. the subject condition of sensibility, under which alone outer intuition is possible for us[5].

Kant argued that all human perceptions of external reality take place in space, and that space was a human construct, the condition of representation, not an absolute reality. For information systems, however, human space is notoriously difficult to represent. Thus most information systems do not represent space, except in extremely rudimentary ways.

However, Kant argues that humans use space partly for identification of objects[6]. (*Object* in this section refers to phenomena, which are the only things that can be known.) Two otherwise indiscernible objects located in different places are different. So the concept of identity of objects is closely tied to the concept of space. Identity of objects is a crucial aspect of information systems interoperability, and must be designed into all representations. Without necessarily agreeing with Kant that space is a human construct, identification schemes for objects can be taken as an analog in the information systems world to Kant's a priori space.

Next, we look at time.

> Time is the form of inner sense, i.e., of the intuition of our self and our inner state[7].

Time, for Kant, is essentially the possibility of either simultaneity or succession in the perception of objects. Unlike geometric space, this elementary concept of time is crucial to information systems and interoperation. A response to a message is generated after the message is received. A payment occurs after the generation of an invoice. These concepts are so deeply imbedded in information systems that they can be taken as a priori.

Finally, we consider the categories.[8] Kant argues that there are four: quantity, quality, modality and relation.

Quantity includes unity, plurality, and totality. Kant argues that our consciousness of a coherent organization of many parts must be constructed, so the principles of construction of an organized whole from parts must be a priori. This same claim is clearly basic to information systems representations, which are constructed from fields organized into tuples organized into tables, and so on. It is also basic to institutional facts which are what the information systems are mainly about. The existence of an institutional fact is dependent on persistent representations. So the part-whole relationship can be taken as a

---

[5] I first part,first section, conclusion b.
[6] Doctrine of Elements, Pt. II, Div. I. Book II Appendix A263-4/B319-20
[7] I first part,second section, conclusion b.
[8] I second part, division I Book I Ch. I, section III.

priori, both implicit in the languages used to construct representations, and explicit in the information structures represented in the systems we build.

Quantity also includes number. Arithmetic is taken as given in information systems, so is also usefully taken as a priori.

Quality includes reality, negation and limitation. Reality produces sensation located in time. Our information systems are always about something, so there is a reality which produces the representations which the systems process and respond to. Change requires negation – a new value of a property implies the negation of the old value. Boundaries imply limitation – we have to be able to separate the objects our systems are dealing with.

The category of quality is so basic to information systems that it is hard to articulate. Negation is central to our logics, and is introduced even where it does not formally apply – database systems use negation as failure, for example. Unifying relationships which enable us to tell what parts belong to an object and which parts do not are essential to dealing with complex objects. (Guarino and Welty 2000 has a discussion of this issue as part of the OntoClean effort.)

Modality is articulated into three pairs of opposites, namely possibility – impossibility, existence – non-existence, necessity – contingency. These concepts are central to contemporary formal logic, which is taken as a priori for information systems.

Relation is more complicated. For Kant it includes the following: inherence and subsistence, causality and dependence, and community (reciprocity between agent and patient).

Kant starts from Aristotle's view that substance is the persistence of the real in time, while accidents inhere in substance. This is essentially a statement of the entity-relationship paradigm. An entity is a substance, while accidents are the values of attributes. These concepts are central to the representations used both in information systems and more generally in logic, so can be taken as a priori for information systems interoperation.

Causality is the necessary following of objects in time, while dependence is necessary preceding. Physical causality is not a central issue in information systems above the hardware level, since the objects being manipulated are institutional facts, whose relationships are intentional. A sequence of institutional facts is generated by successive decisions rather than by as it were one fact bumping into another. On the other hand, information systems abound in necessary rules (integrity constraints, programmed sequences of actions), which could be taken as causal. Dependence is central to institutional facts (if X counts as Y in context C, then Y depends on both X and C). It is also central to information systems that some objects must exist in order for others to be created. Both customer and product objects must exist before an order object can be created. Both causality and dependence can be taken as a priori.

Community is the reciprocal causal relationship of substance and accident. Community has two interpretations in the *Critique*. One is that there can be a mutually exclusive set of possible values for an attribute (accidents), so that the presence of one excludes the others (if an order has been shipped, then it is not pending). The other is that there are complex objects made up of other objects, so that the whole causes the parts and the parts the whole. Both of these interpretations are central both to institutional facts and to information systems, so can be taken as a priori.

Kant derives the categories from the faculty of *judgment*, which subsumes *objects* under *concepts*. An object is a representation of something in the world. Concepts generally derive from experience, but there can be a priori (*pure*) concepts, so that the concept of *concept* is itself a priori.

Further, Kant often uses the term *formal* when describing the uses of the synthetic a priori constituents in representation of external reality. They determine the possible forms of representations.

So by analogy from Kant's synthetic a priori, we have some plausible constituents of a formal ontology independent of material content, summarized in Figure 7.

From space – identity
From time – sequence
From quantity – representation structures, the part/whole relationship, arithmetic
From quality – negation, unity, identity of complex objects
From modality – formal logic
From community – entities and attributes, dependence, causality, mutual exclusion and complex objects

Figure 7 – Elements of formal ontology by analogy from the synthetic a priori

Kant's synthetic a priori has no special status. We have taken some aspects of it as elements of the domain of formal ontology, but there is no reason to limit ourselves to his concepts. Part of Kant's justification for distinguishing the possible structures of knowledge from the content of knowledge is that he viewed the human mind as a causal mechanism. The synthetic a priori can be seen as a sort of reverse engineered specification of the human knowing machine.

Our present theories about human cognition are much richer than Kant's. They include some universal theories, for example [Dennett 1991], [Gardenfors 2000], which more or less support Kant's specification. They also include the idea that the human mind is partly shaped by the language community in which it develops. Thus there are some concepts which are common to particular, possibly quite large, language communities but which are not universal. Arithmetic and logic, in Kant's catalog, are both of this kind.

In this spirit, we would probably want to include in an upper ontology some widely used mathematical theories such as set theory, graph theory and mereology.

We might also want to include widely used metaphors, such as is common in human-computer interaction. For example, we might want to say that a service behaves like a bank Automatic Teller Machine.

Application to the ontologies

It strikes us immediately that the GOL system (Figure 4) is very strongly analogous to Kant's system. The primitives set and ur-element are both in the category of *quantity*. Universal is what Kant would call a concept, and the faculty for generating, storing and using concepts is synthetic a priori. A chronoid (a time interval) is *time*. A topoid (a spatial region) is *space*. Substance and moment are from the category of *relation*.

The argument that GOL can be seen as a system for representing reality rather than a catalog of the contents of reality is strengthened by the observation that the main use of the elements of Figure 4 is to construct complex objects called situoids (categories of *quantity* and *community*). A particularly clear connection is seen in the comment about relations in [Degan et al. 2001], p. 38

> Relations are entities which glue together the things of the real world. Without relations the world would fall asunder into so many isolated pieces.

This is almost exactly what Kant says about the role of the faculty of the imagination in *synthesis* in the Transcendental Deduction.

OntoClean/DOLCE (Figure 3) has also aspects related to the synthetic a priori. In particular, *quality* is *accident* which inheres in an entity (*substance*). *Quality region* (not shown in the Figure) is a *concept*.

Other elements of OntoClean/DOLCE have names which suggest that they are about content. However, they are typically defined by classification from values of meta-properties, so can be thought of as formal rather than material. For example, *aggregate* suggests the category *quantity*. Its subcategories *amount of matter* and *arbitrary collection* are simply whether the aggregate is *mereologically invariant* or not (change their identity if they change their parts). The class *object* is similarly subdivided according to meta-properties including identity, whether they have spatial location or not, and mereological predicates. *Occurrences* have a time aspect, and are also classified according to mereological predicates. The only intrusion of the material universe in the OntoClean/DOLCE system is the distinction between physical and non-physical.

So, despite the names of the lower-level classes, OntoClean/DOLCE can for the most part be viewed with GOL as a formal ontology, used to classify things in the world by their possible representations in the synthetic a priori.

The BWW ontology (Figure 5) also relates closely to the synthetic a priori. Things, properties and attributes are from *community*; event and history from *time*; and type, subtype and composite thing from *community*. The causal concepts used in the derived structures are also from *community*.

The same is not true of the others. If we adopt the term *material* as the opposite of *formal*, in that *material* describes classes that are the (possibly generalized) content of particular applications, then we see for example that WordNet has many material classes. For example, consider *possession*. It has hyponyms *asset, liability, own-right, territory* and *transferred-property*. All of these are linguistic terms which designate things in the world (albeit largely institutional facts). Any attempt to define them by classification from properties would have the status of a scientific theory, and would not be a primary definition. Even more abstract top-level terms like *entity* and *event* can be seen as material by looking at the subclasses of which they are composed.

WordNet also contains formal terms. For example, under *abstraction* we find *attribute*, *measure*, *relation*, *set, space* and *time*. The question arises whether, because WordNet contains formal as well as material terms, it should be considered as partly material and partly formal.

This sort of question arises in the *Critique*. In the Introduction, Kant recognizes that the human mind is a material object, and that its parts and faculties are in the external world, so that they themselves can be known a posteriori. The a priori analysis has to do with how we know rather than what we know, so that the formal terminology can be part of a posteriori concepts as well as a priori. The latter refer to the use of the faculties to generate knowledge.

WordNet's representation and reasoning system involves navigation up and down hypernym and other hierarchies, and do not make use of the terms stored in its database. The formal terms in WordNet are therefore probably best considered as material, so that WordNet would be considered as a material ontology.

Both Cyc and SUMO contain both material and formal terms. In contrast with WordNet, both systems use their databases as part of their reasoning systems. The formal terms are therefore used both formally and materially. They should probably be considered as mixed systems.

Now we have a classification system for upper ontologies, into formal and material. Material ontologies are about the world, while formal ontologies are about the necessary a priori formal structures in which the world appears to the information system processes which interact with it. We can't see anything in the world without using these sorts of formal structures.

An objection will immediately be raised. Ontology researchers (eg [Smith and Welty 2001]) have been at pains to distinguish their work from data and knowledge representation techniques. Does the proposed distinction between formal and material ontologies not simply lump formal ontologies with knowledge representation? We need to digress into a brief look at the latter.

Computer systems interact with the world through programs. As Niklaus Wirth famously declared, a program is an algorithm plus a data structure. At the most primitive hardware level seen by programmers, the data structure of a computer is a sequence of numbered memory locations. Ultimately, whatever a computer system knows about the world is represented in groups of memory locations.

Bare memory locations are too difficult for most programmers to work with, so there has been a development of more elaborate structures to represent the data needed by the algorithms, leading to systems like slots and frames or database schemas. Note that these structures have analogies with the categories. They enable a unified representation of complex external objects, so are subject to the categories of quantity, quality and relation. If we add the algorithms used for database and logical reasoning to the kit of tools a programmer uses to write programs, we get the category of modality as well. Space, in the primitive form of identity, and time in the form of sequence are also in the programming models.

Thus knowledge representation languages express limited versions of the synthetic a priori. Every program written uses these structures to represent the reality they interact with.

The problem with knowledge representation languages at this level is not that they don't provide structures relevant to the external world. It is simply that the structures they provide are much less rich than those available via the synthetic a priori to the humans doing the programming. So there has been a continuing enrichment of the capabilities of the knowledge representation systems. For example, the ERA method introduced the substance/accident and relation paradigm from the category of *relations*.

This sort of argument is explicitly used in the exposition of GOL by [Degen et al. 2001]. They claim that existing systems are based on set theory, and are inadequate for a number of reasons. They add to the set theoretical structures available the additional structure of universal, which is not a set. Similarly, the OntoClean/DOLCE effort is based primarily on incorporating mereological concepts into what we might call the knowledge representation a priori. Efforts have been made to incorporate other rich structures into knowledge representation – for example [Colomb et al. 2001] present a case for the utility of category theory.

What we are seeing, therefore, is a steady increase in the richness and sophistication of the knowledge systems a priori, the tools and structures the programmer uses to make the information systems see the world they interact with. We can compare this with the enormously powerful tools physicists use to see the world of physical objects, through for example [Wigner 1979]. A tsunami is not a partial differential equation, even though the scientists studying it might use partial differential equations to represent it in their theories.

So yes, the present analysis lumps formal ontologies with knowledge representation systems, and argues that there are deep reasons why this should be so.

So what?

The central question for this paper is what use can be made of the distinction proposed between formal and material ontologies. We began with the problem of achieving interoperability among information systems (the semantic web), which is bedeviled by semantic heterogeneity. Because the content of the kinds of information systems we are interested in consists almost exclusively of records of institutional facts, and institutional facts are not only enormously variable but require rich background to interpret, we cast doubt on the ability of general purpose ontologies to be able to be of much use in the development of specialized ontologies for particular application domains.

Our analysis shows that the problem is mostly due to the material aspects of the general-purpose ontologies, since the material ontologies specify content. Formal ontologies specify only form, so are neutral with respect to content. This means that they perform a more limited task than material ontologies.

In the federated database literature, there are two types of semantic heterogeneity which correspond to the distinction between formal and material ontologies (see [Colomb 1997] for a detailed discussion of this point). Corresponding to formal ontologies is *structural* semantic heterogeneity, which can be resolved using more or less complex views. The more pervasive and difficult type such as illustrated in the examples in the present work is called *fundamental* semantic heterogeneity. This corresponds to material ontologies.

Formal ontologies address issues in structural heterogeneity, by providing what amount to rich abstract data types supporting powerful reasoning engines. They assist in the development of application-specific ontologies by enabling the content to be represented in these rich types, permitting the reasoning engines to work within and between interoperating applications. The material content is the responsibility of the developers of the system within their particular context. The reported successes of OntoClean, for example, are of this kind. [Guarino and Welty 2002] give an overview, while a complex natural language domain is analysed in [Welty and Guarino 2001] and WordNet in [Gangemi et al. 2002a, b]. The second includes the DOLCE extensions.

This is not to suggest that material ontologies cannot be more or less general. It would probably be useful to have a representation of the European Union intellectual property laws and regulations as a general context for developing specific ontologies involving exchange of music videos or computer software *within the European Union*. That more general ontology would not apply to exchanges operating within the United States or India. The argument of this paper is that material ontologies developed *in the absence of any application domain* are unlikely to be useful in facilitating interoperation of information systems.

Implications for reasoning with ontologies

Ontologies are put together using a number of structural relationships, the most important of which is subsumption, or the is-a relationship, which gives a progressively more specific decomposition of the most general terms. The semantics of subsumption is based on the subtype relationship.

If we take the view that the formal ontologies are like Kant's synthetic a priori, so cannot describe external reality, then it is probably not appropriate to employ them using subsumption. Just as a tsunami is not a partial differential equation, the bill of materials for an aircraft is not a part/whole relationship system.

We have argued above that formal ontologies are developments from knowledge representation systems. This view is supported in practice, if not in principle, by the fact that creators of upper ontologies often develop relationships between their ontology and one of the conceptual modeling languages such as Entity-Relationship (ER) modeling or Universal Modeling Language (UML), for example [Weber 1997] and [Evermann and Wand 2001].

Conceptual models are engineering artifacts whose elements have no necessary connection with the real world. For example, [Debenham 1989], exhibits representations of the application domain fact "the interest rate on savings accounts is 5%" in six different ways – once as an atomic data item, three times as relational schemes, and twice as rules or view definitions.

If upper ontologies are rooted in philosophical theories about what is in the world, on one hand one would expect that a given UML or ER construct could be used to model different ontological constructs (engineering design freedom). On the other hand, there are competing ontologies (eg GOL vs OntoClean/DOLCE vs BWW). How does one choose among them? If one can let a hundred flowers bloom, then it becomes harder to argue the difference between an upper ontology and a conceptual modeling language. After all, there is only one material world.

The formal structures of conceptual modeling languages are applied to concrete models as schemas rather than as types. There is no sense in which all the entities and relationships in an ER model are subtypes of *entity* and *relationship*, nor any sense in which all the classes in a UML model are subtypes of *class*.

This second example is somewhat subtle. Object-oriented programming systems like UML generally store the basic data structures and methods for a construct like *class* in an object called for example *class*. Particular classes in a given design are represented as instances of *class* in order to inherit the data structures and methods. But *class* does not have the semantics of *everything*, rather of *nothing*. What the engineer does by instantiating a class is to introduce some semantics. [Guarino and Welty 2002] argue on logical grounds that instantiation is not subsumption, even within the semantics of a given material ontology.

Our earlier sketch of the development of knowledge representation languages from computer architectures allows us an analogy which may bring the point home. A memory location in a computer together with the computer's instruction set can be seen as similarly a top class. But we don't think of the unassigned memory location as containing every number, but no number.

Another way to see that formal ontologies are not best implemented by subsumption is to look at the interoperation of multiple information systems. Consider an object in the world, say a particular aircraft A. To its manufacturer's parts management system, aircraft A appears as a bill of materials, represented as a whole/part structure. To the airline using it, aircraft A appears as a collection of seats, represented as a set/instance structure. To the manufacturer's production scheduling department, aircraft A appears as a process and a set of accomplishments (the temporal parts are the various stages of production). To a safety inspection system, aircraft A appears as an ordinary physical object together with a set of qualities.

Aircraft A is the same entity in all cases. Further, in the semantic web, these various views of aircraft A interact. Person P may wish to book a seat on an aircraft which has been recently inspected, which was manufactured in a given time period and has a particular type of avionics system. Perhaps person P was the engineer who designed the system and supervised its manufacture during the given time period, and wants to feel how it works in routine use when recently tuned according to specifications.

These diverse representations of an entity are the source of resolvable semantic heterogeneity (which can be resolved using more or less complex views – [Colomb, 1997]). Note that resolvable semantic heterogeneity even arises when the same upper ontology is used in all cases, as in the example (all upper ontology concepts come from OntoClean/DOLCE).

Therefore it would seem more appropriate to think the material ontology including *aircraft* (the set of all aircraft of which Aircraft A is a member of a subset) as represented in different formal ways in different applications. Assuming that the applications have resolved semantic heterogeneity, the different representations are convertible using views or equivalently predicates.

Having argued for a difference in kind between formal and material ontologies, the concept of a mixed ontology begins to become somewhat strange. The issue is relatively easy to see from the OntoClean/DOLCE ontology in Figure 3. Recall that this ontology is mixed because the material distinctions physical/non-physical and their subtypes are distinguished from the other elements in the taxonomy, which are formal. If the formal ontology is neutral with respect to content, then the physical/ non-physical taxonomy should appear as subclasses everywhere in the taxonomy. Besides physical and non-physical objects (eg person versus corporation) and qualities (eg weight versus name) there should be physical/ mental occurrences (eg hurricane versus sale).

In other words, the orthogonality of the formal and material sub-ontologies should be reflected in a symmetry of the taxonomy – every formal class should have physical and mental subclasses. This sort of situation is common in the development of classification systems in the information science community, where the classification system can be represented using the method of facets [Rowley 1992], [Colomb 2002]. This method can be illustrated by reference to the menu of an oriental noodle restaurant, where there might be three taxonomies: type of noodle (Hokkien, Singapore, etc.); sauce (Chinese, Japanese, Thai, etc.); and main ingredient (beef, pork, etc.). A single dish is characterized by a choice from these three taxonomies. The menu could be represented as a single taxonomy, where each noodle type has a subclass for each sauce, and further each sauce subclass of each noodle type has a subclass for each main ingredient. But representation in a single hierarchy obscures the underlying symmetry, making the system more difficult to use and to modify. It is generally more convenient to represent each of the taxonomies as a separate facet, and to classify each dish by three choices, one from each. Many major classification systems are built this way. SNOMED, used for classifying medical records, has 11 more or less orthogonal facets each of which has many thousands of classes.

Figure 8 shows the OntoClean/DOLCE system of Figure 3 represented in two facets, formal and material. A hurricane is a [physical, process] while a memory is a [non-physical, object] and weight is a [physical, quality]. The larger mixed ontologies such as Cyc or SUMO could possibly be represented in a faceted manner, thereby separating their formal and material aspects.

Formal

Entity
  Abstract
    Fact
    Set
    Region
  Endurant
    Quality
    Substantial
      Aggregate
        Amount of Matter
        Arbitrary Collection
      Feature
        Relevant Part
        Place
        Object
  Perdurant/Occurrence
    Event
      Achievement
      Accomplishment
    Stative
      State
      Process

Material

Physical
Non-Physical

Figure 8: OntoClean/DOLCE Top Categories in Faceted Representation

Taken together with our earlier argument that formal ontologies are better thought of as rich knowledge representation languages, this faceted approach to mixed ontologies makes the formal ontologies look much like types as they appear in programming languages.

Conclusions

The main claim of this paper is that attempts to define ontologies of things in the world in the absence of a particular class of applications are likely to fail, particularly when one attempts to use them to facilitate interoperation of information systems in the semantic web. This claim is supported by the argument that information systems are largely concerned with records of institutional facts. Institutional facts are very weakly determined by factors outside the institutions concerned, so that semantic heterogeneity is the norm. The possibility of an a priori ontology depends on there being strong external determiners on the content of the system. Therefore, there is no reason to suppose that a transcendent ontology is possible.

On the other hand, following the analogy of Kant's synthetic a priori there is strong support for the feasibility of application-independent formal ontologies, which specify not what there is in the world but how what there is in the world can appear to humans or information systems. We thus have a distinction among application-independent ontologies: formal, which are feasible, and material, which are not.

A practical consequence of this distinction is that when a formal ontology is used to construct a material ontology for a particular class of application, the concepts in the formal ontology should not be represented as material classes in the ontology. Mixed ontologies should be factored into formal and material facets.

Furthermore, we would not expect that the formal ontologies would help in resolving semantic heterogeneity. Their value is as libraries of richly structured and well-understood abstract data types and structural organizational principles, which make the technical aspects of ontology construction easier and more reliable.

References

COLOMB, R. M. Impact of semantic heterogeneity on federating databases. *The Computer Journal* 40, 5, (1997) 235 -244.

COLOMB, R. M. *Information Spaces: the architecture of cyberspace* Springer, London (2002)

COLOMB, R.M., Dampney, C.N.G. and Johnson, M. Category-theoretic fibration as an abstraction mechanism in information systems. *Acta Informatica*. 38 (2001) 1-44.

DEBENHAM, J.K. *Knowledge Systems Design* Prentice Hall, Englewood Cliffs (1989)

DEGAN, W., Heller, B., Herre, H. and Smith, B. GOL – a general ontological language. in Welty, C. and Smith, B. (eds) *International Conference On Formal Ontology In Information Systems (FOIS-2001)* October 17-19, 2001, Ogunquit, Maine ACM Press, New York (2001) 34-46.

DENNETT, D. C. *Consciousness Explained* Little, Brown & Co., Boston (1991)

EVERMANN, J. and Wand, Y. Towards ontologically based semantics for UML constructs. in H.S. Kunii, S. Jajodia and A. Solvberg (eds.) *Proceedings of ER 2001* Springer-Verlag, Berlin (2001) 354 – 367.

FELLBAUM, C. (ed) *WordNet – an electronic lexical database* MIT Press, Cambridge (1998)

GANGEMI A., Guarino N., Oltramari A. and Borgo S. Cleaning-up WordNet's top-level. *Proceedings of the 1st International WordNet Conference* 21-25 January (2002a)

GANGEMI A., Guarino N., Masolo, C, Oltramari, A. and Schneider, L. Sweetening ontologies with DOLCE. *13th International Conference on Knowledge Engineering and Knowledge Management EKAW02* 1-4 October 2002 - Sigüenza (Spain) to appear. (2002b)

GARDENFORS, P. *Conceptual Space: the geometry of thought* MIT Press, Cambridge (2000)

GRUBER, T.R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing* Technical Report KSL 93-04 Knowledge Systems Laboratory Stanford University (1993)

GUARINO, N. and Welty, C. Identity, unity and individuality: towards a formal toolkit for ontological analysis. in W. Horn (ed) *Proceedings of ECAI-2000: The European Conference on Artificial Intelligence* IOS Press, Amsterdam (2000)

GUARINO, N. and Welty, C. Evaluating ontological decisions with OntoClean. *Communications of the ACM* , 45, 2 (2002) 61-65.

KANT, I. *Critique of Pure Reason* Cambridge University Press, Cambridge (1998)

ROWLEY, J. *Organising Knowledge*2nd edn Gower, Aldershot (1992)

SEARLE, J. R. *The Construction of Social Reality* The Free Press, New York (1995)

SHETH, A.P. and Larsen, J. Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys: Special Issue on Heterogeneous Databases* 22, 3 (1990) 183-236.

SMITH, B. and Welty, C. Ontology: towards a new synthesis. in Welty, C. and Smith, B. (eds) *International Conference On Formal Ontology In Information Systems (FOIS-2001)* October 17-19, 2001, Ogunquit, Maine ACM Press, New York (2001).

SOWA, J.F. *Conceptual Structure: information processing in mind and machine* Addison Wesley, Reading Mass (1984)

WEBER, R. *Ontological Foundations of Information Systems* Coopers & Lybrand Accounting Research Methodology. Monograph No. 4. Melbourne (1997).

WELTY, C. and Guarino, N. Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering* 39, 1, (2001) 51-74.

WIGNER, E. “The unreasonable effectiveness of mathematics in the natural sciences”, in *Symmetries and Reflections* Ox Bow Press, Woodbridge, CT (1979)