A Proposed Methodology for The Development of Application-Based Formal Ontologies

Dr. Eric Little
Ontological Engineer
University at Buffalo
Center for Multisource Information Fusion (CMIF)
Center for Cognitive Science

Abstract

Formal ontologies are currently being developed for numerous applications. But the question of what constitutes a formal ontology, and how one goes about constructing such a thing, remains an open research topic. This paper proposes a general methodology for ontology construction through which it is argued that the successful design of ontology products depends upon the reciprocal relationship between abstract philosophical reasoning and application-based systems engineering. The proposed methodology offers insight into the rational, philosophically-grounded motivations for upper-level ontology construction, coupled with certain empirical, domain-specific motivations necessary for its lower-level construction and implementation. To convey this methodology, I provide a brief examination of one such current research project, a Disaster-Response Ontology (Dis-ReO) aimed at improving data fusion for disaster-response activities such as casualty servicing, ambulance routing, and structural damage assessment.

1 The Philosophical Impetus of Ontology Development

One of the largest advantages brought about by the last several decades of work in computer science and related computational fields is the ability to gather, store and transfer large amounts of data from numerous disparate sources. Humans have collected a wealth of information about the world within which they live. A large portion of that information continues to be cataloged in various kinds of information systems for instant retrieval by a few clicks of a mouse button. Given the extraordinary amount of data present to artificial system users, a problem of *information organization* has risen to the surface. Unstructured data is far less useful than data which is neatly grouped into meaningful categories. Hence, recent attention has shifted to understanding scientific principles of organizing the vast amount of information stored within data systems. One increasingly popular method for information organization is sought through the construction and implementation of *formal ontologies* (Bowman, 2001; Gruber, 1993, 1994, 1995; Guarino, 1998; Lenat & Guha, 1990, 1996; Noy & McGuinness, 2001).

Formal ontologies are logically-structured bundles of information about a given domain of existent physical entities, attributes of those entities, and the relations between them. Proper ontologies act as transparent representations, or models, of the common-sense world within which humans live (Smith, 2001c). They are transparent in the sense that they do not obfuscate or alter the items they organize. A proper ontology should mirror those organizational properties of the world's built-in structure. To understand the built-in structure of the world, consider an ordinary water molecule. It is composed of two hydrogen atoms and one oxygen atom which share both spatial as well as other physical relations to one another. In this sense, the molecular structure of the world, which an ontology of physics or chemistry can represent, exists independently of, and logically prior to, those theories that describe or model it. Factual features of the world, such as its molecular structure, predate any theoretical artifacts which serve to explain or categorize them in a rigorous empirical fashion (e.g., the Periodic Table of Elements). Chemists, professional or not, utilize references such as the Periodic Table to accurately classify those items of which all matter, such as water, is composed. However, the study of chemistry is not the study of references like *The Periodic Table*, rather the study of chemistry is the study of *the real existing elements of the world*, which are represented within the Periodic Table. The Periodic Table should be understood as a transparent, empirical tool which is used to examine factual states of affairs in the world much the same way that a visually impaired person's spectacles serve to focus their vision on physical objects. Just as the person's spectacles do not physically alter or obfuscate the objects of their perception (rather they alter the visual perspective *of* such objects), so too do ontological tools such as the Periodic Table avoid altering objects in question. A properly constructed ontology should serve as a transparent artifact for clarifying those objects within a given domain, just as spectacles do for visual stimuli.

An ontology is a conceptual tool, designed to organize those *independently existing elements that compose the fabric of reality* (Gibson, 1979; Smith, 2001b; Welty & Smith, 2001). Just as the Periodic Table has proven to be an indispensable tool for understanding the independent structure of the physical world – independent in that our conceptualization *of* the world necessarily depends on the existence of the world itself, but not vice versa – so too will formal ontologies prove to be an indispensable tool for modeling and understanding the myriad of *metaphysical* items that compose a given domain, situation or state of affairs.

2 Application-based Engineering of Ontologies

The design and implementation of ontologies is steadily growing into a burgeoning new field of applied engineering. Ontologies are being sought for various broad-based applications including: inventory/organizational purposes, user-interface solutions, classification of abstract entities, domain specification/identification, database construction, information fusion, data mining, and information querying (Gruber, 1993; Uschold & Gruninger, 1996; Gruninger and Fox, 1995; Hendler and McGuiness, 2000; Slattery, 1997). Due to the variety and complexity of these applications, ontologies are being developed in highly interdisciplinary settings, by individuals from various academic and non-academic communities, for both civilian and military purposes.

The application of ontologies to specific domains involves an application of science-minded, rigorous philosophy to those domains (Smith, 1996, 2001c, Welty & Smith, 2001). A formal ontologist is forced to wear two hats at one time. One is that of the traditional metaphysician, the other is that of a systems engineer. The former position is one which is largely based on rationalistic principles of formally structured information. The traditional metaphysician deals with information which is highly abstract and irrespective of domain considerations or any significant amount of experimental data. The latter position (i.e., that of a systems engineer) is largely an *empirical* endeavor whereby information is perceptually gathered from some specific domain of interest, and subsequently tested over and against that domain (Blanchard et al, 1998).

Acting as metaphysicians, ontologists must do the required philosophical work involved with constructing broad-based, abstract categories which capture the logico-philosophical subtleties of a sound metaphysical system. It is in this regard that the ontologist should apply their philosophical training to abstract categories such as 'enduring item,' 'dependent item,' 'independent entity,' 'part,' 'whole,' or 'relation.' The meanings of such terms must be gleaned from areas such as investigations into formal logic and the history of metaphysics in general. The theoretical underpinnings of philosophical ontology are necessary, though not sufficient, conditions for a sound ontological framework which is capable of being applied to countless domains of inquiry, since the items found therein are neither domain-specific, nor task-specific. Sound philosophical reasoning should always underlie applied ontologies in order to guarantee that the ontology in question will not be reduced to an *ad hoc* application, whereby broader applications get neglected.

Acting as systems engineers, on the other hand, ontologists must concern themselves with a plethora of concrete, material, domain-specific, or conceptually-specific items. The systems engineer is concerned with those specified *needs* and *requirements* that a given system is to fulfill (Blanchard et al, 1998). A system's needs and requirements are largely determined by the measurable use-value of the system in question. For this reason, an adequate system cannot be engineered without the system's *purpose* or *use-value* already in mind. Here it is useful to inquire about domain- or conceptually-specific information which experts in a given field are often able to provide. If, for example, one is seeking to construct an ontology for disaster-response applications, it is necessary to first understand the needs and requirements of those individuals working within disaster-response situations (e.g., what kinds of items exist in these types of domains? Who uses them and how? What kinds of socio-cultural, or political components (e.g., chain-of-command) are present?). Unlike the purely logico-philosophical items examined by metaphysicians (e.g. part-whole relations, processes), many items found within the domain of a given disaster require elucidation by some expert within that field.

For example, suppose one wanted to construct an ontology which would help to facilitate better casualty servicing by organizing information in a way that benefited individuals responsible for such services (e.g., an ontology could provide a shared lexicon of terms used to diagnose the severity of injuries, a shared protocol for ambulance routing/dispatch, or a shared protocol for hospital admittance of both walk-in and delivered casualties). The construction of such an ontology would require the following two steps: 1) the upper-most, abstract levels of the ontology would be designed in accordance with a rationally-based, formal ontology grounded upon a sound metaphysical theory; 2) the lower, more concrete levels of the ontology would be designed in accordance with the vast amount of domain-specific knowledge possessed by those disaster-response experts (e.g., FEMA or Red Cross workers) who are familiarized with the host of tangible objects, events and situations found within disaster sites. In essence, the ontologist must work the problem from two angles simultaneously. One angle is purely rational and devoid of any real content, while the other is empirical and highly content-specific. Traditional philosophical training helps the ontologist to perform step one above, but not step two. Step two requires that ontology construction be an *interdisciplinary* endeavor, since large amounts of empirical information must be gathered from experts familiarized with the given application. Disaster-response experts would, for example, be able to inform the ontologist of many significant terms, tactics, relationships, hazards, etc., all needing to be included within the formal structure of the ontology in question. Without such information, the ontology would not be fully applicable to that particular domain, since it would not address the specific *needs* of disaster-response personnel who would serve to benefit from the ontology.

Each of the two steps listed above contain numerous subsequent steps, some of which overlap. The steps depicted in this paper form a proposed methodology that seeks to capture the synergy between certain philosophical and empirical principles associated with the task of ontology construction. By capturing this important synergy, one can more easily understand the role of the ontologist as one who provides a sound conceptual product (i.e., a representational framework) capable of properly organizing large, disparate chunks of information within various application-driven domains.

The Center for Multisource Information Fusion (CMIF) is presently constructing a disaster-response ontology (Dis-ReO) through a grant from the Air Force Office of Scientific Research (AFOSR). The following will serve as both a general discussion and prescription for ontology development methodologies, as well as provide information about the specifics of the Dis-ReO itself. Using specific examples taken from the Dis-ReO should prove advantageous in understanding certain abstract conceptual issues related to ontology development in general.

3 Ontology Development Steps

A formal ontology must be able to represent the myriad complexities of states of affairs within the world. In order to do so, it is crucial that ontologies be both *consistent* as well as *comprehensive*. The consistency of an ontology rests on its formal logical structure, which means that the terms within an ontology: 1) must be used in the same manner throughout the ontology; 2) must not be conflated (assume implicit terms within them); and 3) must have the same conceptual extension and intension. The comprehensiveness of an ontology guarantees that it is of sufficient size and complexity to accurately represent all items in a given domain. The consistency of the ontology can be verified by rational means, whereas the comprehensiveness can be verified by empirical measures, once again showing that applied ontology design amounts to the fusing of both rational and empirical activities.

It is argued here that a consistent and comprehensive ontology can be designed through the following six steps:

1. Develop a sufficiently large and representative *lexicon* of terms.
2. Develop a set of metaphysically-grounded *upper-level* (abstract) categories.
3. Develop a sufficiently large set of *region-specific* (lower-level) categories.
4. Diagram formal *relations* between terms/categories.
5. Develop/find a *computational framework* capable of capturing all items in 4.
6. Develop *methodologies for evaluating* the ontology.

By following the above steps, ontologists would be able to provide their clients with the following: 1) a shared lexicon of terms which both denote and connote the wide range of items (physical and non-physical) within a given domain; 2) A formal structure capable of capturing the *relations* between those lexical items; 3) a methodology for checking the consistency and comprehensiveness of those lexical/categorical items; and 4) a sufficiently complex artificial system capable of querying information within a given domain and inferring new (and possibly more complex) relations within that domain. We shall now examine each step of the developmental cycle in order to better understand the methodology being put forth here.

3.1 Lexicon Development

The first step in ontology development should be to define that domain of items which the ontology is to capture. One can define such a domain (and its contents) by constructing an appropriately large lexicon of terms that represents everything within it that is of interest. Any domain of interest will be composed of objects, events, processes, states of affairs, attributes of objects, parts of objects, segments of processes, etc. Each item within the domain – whether physical or nonphysical, spatial or temporal – must be accurately represented in the ontology. Many items of interest within the ontology will be *compound items*, meaning that they should be thought of as wholes, unities, or aggregates, containing *simpler items* which are subordinate items. The lexical definitions of compound items will contain many subordinate terms, which themselves will need to be accurately defined.

For example, if one is interested in developing an ontology for disaster response, one will be concerned with understanding concepts such as ‘damage’ and therefore designing a lexicon that captures the *types* of damage resulting from various kinds of disasters. Damage is a relational item, meaning its very existence depends on the existence of other items (substances, processes), so the term 'damage' is necessarily a compound term. The term 'damage' refers to damage *of* some object or other, thus it exists as a property, state, or attribute of that object. Furthermore, damage is caused by some *action* which brought about the damaged state of that object.

Since the Dis-ReO is focused on earthquake disaster response, it must be capable of capturing all of the various types of damage that result from earthquakes. Examples of the types of damage associated with earthquakes include: 1) *structural* damage to buildings, bridges and dwellings; 2) *bodily* damage to various agents (e.g., civilian, military) located within affected areas in the form of injuries/casualties; 3) *psychological* damage to various agents both inside and outside the affected area that can be caused by stress or emotional trauma from the loss of loved ones, loss of property, or potential health hazards resulting from pollutants or contaminants; 4) *general* damages to personal belongings, the contents of dwellings, and the surrounding environment.

The job of constructing a sufficiently large lexicon of terms is quite labor-intensive. First, one must manually search out various representative dictionaries related to the domain of interest. Once found, some individual must manually sift through those dictionaries in order to extrapolate all of the relevant terms for that domain. This amounts to developing a rough-draft, *resource* lexicon (a la Guarino) which is a merger of other disparate sources of information. Second, once the rough-draft lexicon is compiled, each term in that lexicon must be examined in order to uncover conflated or compound terms, whose definitions contain subordinate terms which themselves require definition. For example, the definition of a term such as 'damage' will require the ontologist to delve into other subordinate definitions such as 'cause,' 'object,' or 'event.' Understanding the term 'damage' depends on understanding several other related terms associated with it. It is here that the ontologist begins to apply certain rationalistic, philosophical principles to the task of ontology construction. Sorting out the relations between subordinate terms which support compound terms is no small task, since often the subordinate terms at hand are highly abstract, and at times can provide inconsistencies when comparing various definitions. Dictionaries and glossaries that contain disaster-specific terms such as 'damage' will almost never contain the definitions of subordinate terms such as 'cause,' 'object,' or 'event' that are implicit in its definition. Moreover, different sources will sometimes define like terms in different ways. This can lead to inconsistencies when these incompatible definitions are merged together from disparate sources into a single lexicon. Therefore, the task of developing lexicons for applied ontologies must be done manually, because the task is a *semantic* one, not a *syntactic* one. Automated information systems are of little use in constructing lexicons for applied ontologies, since the task of constructing those lexicons requires the use of common-sense, semantic reasoning skills as well as philosophical analysis, things which automated systems lack. The job of compiling terms from disparate sources, and weaving them into a meaningful and representative lexicon that captures the subtleties of a given domain, is still the function of a human in the loop.

The lexicon for a formal ontology must be both consistent and comprehensive. Here again, we see the task of the ontologist as representing the merger of rationalism and empiricism. Consistency-checking is a logical function, and it is here where perhaps machines can aid ontologists in their task, since it is possible that a sufficiently complex software product could perform many of the tasks associated with checking the consistency of terms within a lexicon. The comprehensiveness of the lexicon, however, is an empirical function, in that it requires gathering input from human domain experts who are most familiar with the terms in the lexicon, their specific relations to one another, and their usage. For this reason, a formal ontology's lexicon should remain an *open item of investigation*. This means that it should always be open to revision in the form of addition or deletion of terms, based on what experts in the field recommend. Ontologies, after all, are meant to capture dynamic portions of the world. So it makes sense to assume that as the domain changes (e.g., the passing of events, the shifting of boundaries), or as someone's understanding of the domain changes (e.g., the understanding of the transmission of a disease is better understood), the lexicon that represents that domain will also change in response to it.

For Dis-ReO, it is up to disaster domain experts to say whether the Dis-ReO lexicon is sufficiently large to capture all of the items encountered within that domain. The Dis-ReO lexicon is presently 85 pages in length, only counting disaster-specific terms. It was compiled from ten separate on-line disaster dictionaries/glossaries including those produced by The Federal Emergency Management Agency (FEMA), The American Red Cross, and The Disaster-Recovery Journal (see the attached Disaster Lexicon References for an exhaustive list of these sources). The lexicon is being checked by various disaster experts familiar with earthquakes and the effects of earthquakes, the agencies that respond to earthquakes, the command and control features of those agencies, and the various kinds of resources needed to effectively respond to earthquake disaster areas. For this reason, the Dis-ReO lexicon (and thereby the entire Dis-ReO) remains open to constant scrutiny from disaster-response experts of various types.

3.2 Upper-level Category Development

The second step in the methodology for formal ontology development is to develop a set of inter-related upper-level categories in order to provide a solid metaphysical underpinning for the ontology. The metaphysical structure of the ontology is crucial to its design as well as its implementation and re-usability. The upper-level categories of a formal system provide an abstract, philosophical basis under which every specific category within the ontology will fall. The upper-level categories are not content-specific, meaning that they are not influenced by the specific items which are their members. They exert their influence downwards in such a way as to encompass all categories and specific objects which fall within their scope. Upper-level ontological categories are akin to taxonomies in biological systems which serve to organize specific organisms by creating a hierarchical tree-like structure where each higher level of the tree represents a higher level of abstraction or generality. For example, some particular animal 'Spot' is subsumed under the more general category of 'dog,' which in turn is subsumed under the category 'canine,' which in turn is subsumed under the category 'mammal,' and so on, until one reaches the upper-most biological category of 'life form' or something of the sort.

The Dis-ReO uses the Snap/Span Basic Formal Ontology (BFO) as its upper-level category scheme (IFOMIS program, Leipzig). The Snap/Span BFO is a philosophically-grounded categorical structure that is divided into two orthogonally-related ontologies. The Snap BFO is an ontological structure indexed by time instants, meaning that the items within Snap are considered independently of any of their temporal parts. There are no processes, events, actions, or the like found within the Snap BFO. The Snap BFO categorizes the world in terms of its static ontological structure, similar to taking a photographic snapshot of reality. Photos do not directly represent processes as they are unfolding in time. Rather, photos show objects and relations as they stand to one another at some very instant in time (e.g., the moment the photo is taken). A blurred object in a photo can be understood, representationally perhaps, as an object in motion, but the brunt of that interpretation is shouldered by the perceiver of the photo, meaning some layer of epistemic understanding is added to the photo itself. Similarly, the Snap BFO captures only those items and relations in the world that form static metaphysical relations to one another at some given place and time.

Conversely, the Span BFO is a videoscopic ontology indexed by temporal intervals or processes. The Span BFO, unlike its Snap counterpart, does not model static items existing as instants of time, but rather, it models the unfolding of events *over some span of time*. The Span ontology is similar to recording an event on a video camera or tape recorder, in that, rather than producing a static snapshot of items and relations, it produces a continuous stream of *dynamic* events whose very nature is temporal, rather than spatial.

The reason for dividing the BFO into two orthogonal sections, Snap and Span, is to avoid confusing objects (instants) and events (processes). For example, one can think of their hand as part of them, and one can also think of their biography as part of them. However, one's hand and one's biography are, metaphysically speaking, very different sorts of items. One's hand can stand in the same immediate part-relation to one's body at different instances in time. Plus, the relation between the hand and the body can be captured in one go, since, considered in spatial terms, it is a static relation. One's biography, however, is never a static relational item. By definition, a biography is fluid, since its essential characteristic is that it unfolds over time. It is an essentially temporal thing, and therefore, resists being captured in any instance.

The Dis-ReO is currently being developed in conjunction with the Snap/Span distinction found within the BFO. Up to the present, research into the Dis-ReO has focused on modeling Snap items alone, since Span items (i.e., processes) are dependent on substances, and therefore are relational in nature. The Dis-ReO is first and foremost concerned with accurately representing the kinds of objects, agencies, damages, losses, emergency personnel, and spatial regions found within disaster sites. That being said, since disasters are temporally unfolding *events* that take place over spans of time, and since responses to those events are also temporal events, the Dis-ReO will need to be expanded to include Span items such as processes, parts of processes, quasi-processes, spatio-temporal regions of processes, etc. However, since the Snap BFO can capture items that exist *within* processes, it is a good place to begin the initial steps for ontology development. Since the task of ontology development is a difficult and tangled one that deals with objects, attributes and processes, one must choose an appropriate place to begin. Because Span items depend on Snap items for their existence, it can be argued that Snap is the better place to begin. A Snap ontology can provide a model of those items whose permanence can serve as the basis for Span items. Thus, all of the discussion below will focus solely on the Snap Dis-ReO.

The most general category in the Snap Dis-ReO is *Enduring Item*, which is the category that includes any item existing in space and time, but which has no temporal parts (i.e., no Span items) (see Figure 1). Enduring items can be broken down into three subordinate categories: 1) *Spatial Region* (any extended area of dimension 0,1,2, or 3); 2) *Dependent Item* (any relational item such as a quality, state or attribute, which cannot stand alone as a separately existing entity. This category is synonymous with Husserl's use of the term *moment*); and 3) *Independent Item* (any maximally connected, causally unitary thing, which has a more or less rigid boundary, an identity, and whose existence is not predicated on anything else's existence. This category is synonymous with Husserl's use of the term *part* or *piece*) (Husserl, 1900-01). Each of these three categories contains several subordinate categories beneath it.

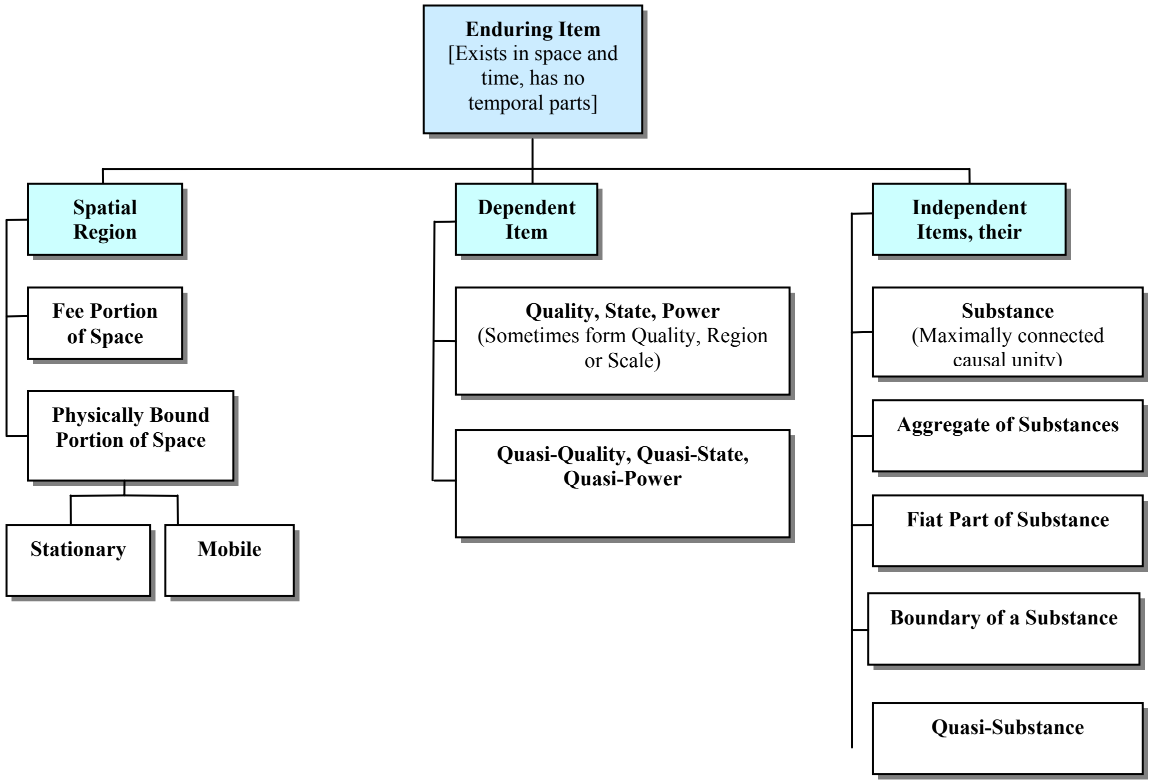

Figure 1: Snap BFO Upper-Level Categories for Dis-ReO

3.3 Domain-Specific Category Development

The domain-specific categories of an ontology can be derived from the ontology's lexicon, which, once again, points to the need for a consistent and comprehensive set of domain-specific terms. It is important to note that many of the domain-specific categories needed for the ontology may not be exhausted by the specific, alphabetically listed set of terms within the lexicon. Many terms' definitions will contain subsequent terms within them that will also need to be categorized within the ontology. However, these subordinate terms will not necessarily be explicit within the lexicon, but instead, will need to be manually fleshed out by the ontologist. For example, consider the following definition from the Dis-ReO:

> Assisting Agency
> An agency directly contributing suppression, rescue, support or service resources to another agency. (IMS/ICS) (source: Emergency Management Glossary (U.S. Steel Gary Works) (Snap Dis-ReO, 2003).

The definition above contains information about certain items, objectives, and activities related to assisting agencies, namely: suppression resources, rescue resources, support resources, and service resources, all of which may contain subordinate, more highly specific, categories within them (e.g., rescue resources will contain fire services and ambulance transportation services, all of which could be further categorized as 'active,' 'inactive,' 'available,' or 'unavailable'). Simply modeling the term 'assisting agency' within a disaster-response ontology, without modeling certain terms embedded within its definition, will not suffice in capturing the ontological structure of what assisting agencies *are*, what kinds of *members* they have, or what kinds of *services* they perform.

Domain-specific categories represent the *material* of the formal ontology, whereas the upper-level categories represent the ontology's *form*. In this manner, ontology development represents a type of Aristotelian *hylomorphism*, where it is assumed that all matter possesses some abstract form and, simultaneously, all form is existentially constituted by some matter. In this manner, rationalistic philosophical principles inform us about the form of the ontology, while data-driven empirical facts inform us about the concrete material of it.

3.4 Diagram Formal Relations Between Terms/Categories

The connection between the upper-level form and the domain-specific material within an ontology is made through the manual integration of the ontology's upper-level categories with their domain-specific counterparts. Consider once again the category 'damage' within the Dis-ReO. 'Damage' is one sub-category of the more general category 'Dependent Item,' which in turn, is a sub-category of 'Enduring Item' (see Figure 1). As previously discussed, there are various kinds of damage that are associated with various kinds of objects in a given domain. All damage is a quality (attribute) of some substance or other. Specific kinds of damages result from specific activities (earthquakes, fires, flooding) and the specific kinds of material substances one is investigating (e.g., agents, structures). The specifics of damage types results, at least in part, from the specifics of substance-types. The domain-specific level of the ontology serves to provide these kinds of specifics. The Dis-ReO recognizes various kinds of hierarchically-arranged substances, all of which can be damaged in a disaster event (see Figure 2).

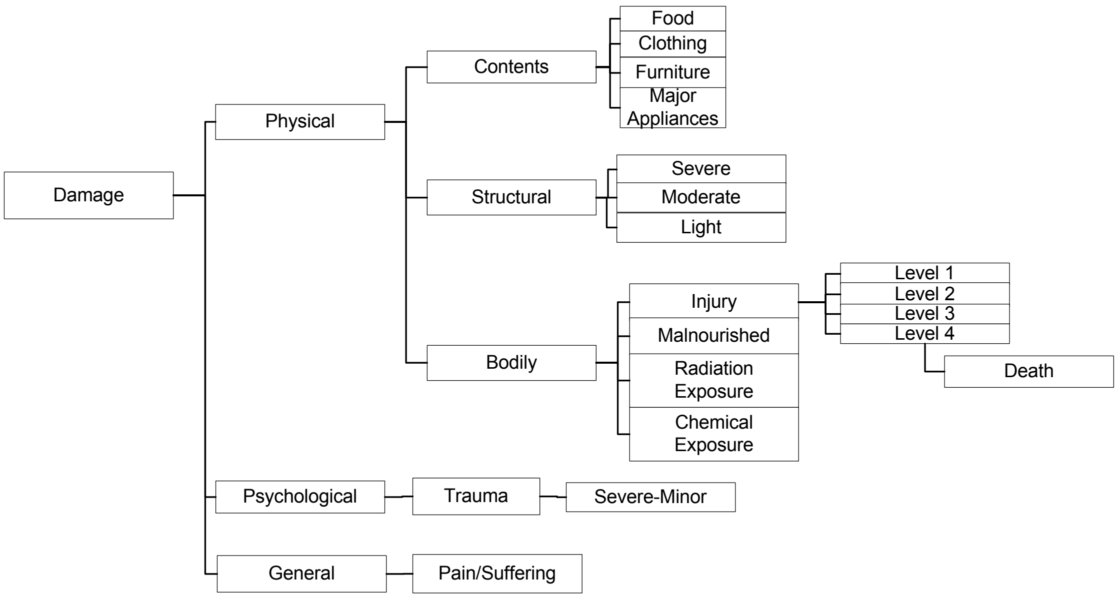

Figure 2: Domain-specific Damage Categories.

The above diagram shows the marriage of rational principles and empirical data on various interconnected levels. For example, when servicing casualties, it is important to understand the severity of their injuries. The HAZUS program used by disaster agencies to provide state estimates of injuries and casualties categorizes injuries into four specified levels, the first being minor, the fourth being mortally wounded (leading to immanent death). The Dis-ReO is able to capture that information and draw relationships between those HAZUS categories and the more abstract conceptual categories of the SnapBFO. In this way, one can see that a 'Level 3 (serious) injury' is a subcategory of 'Injury,' which is a subcategory of 'Bodily Damage,' which is a subcategory of 'Physical Damage' and so on. Thus, one important advantage to an ontology of this type is seen in its ability to map the relations between those well-understood concrete physical items (e.g., an injured person) and those perhaps-not-so-well-understood abstract nonphysical items (e.g., possessing the attribute of 'being damaged').

3.5 Computational Ontology Capture Method

In order to utilize the ontological model developed by the merger of the Upper-and Lower-level categories, it is important to find (or construct) a sufficiently complex computational language which can capture that model. While it is open to much debate as to which kind of computational tools are appropriate for this task (e.g., first-order predicate logic, description logic), there is general consensus that the language strike a balance between being sufficiently formal on the one hand, and conceptually inclusive on the other hand. A computational language that is overly formal, to the detriment of its expressive power will not be optimal. In the same way, a language that is highly expressive, but which lacks formality, will also be inadequate for the task.

Since the formal structure of the ontology is fully expressed within its diagrammatic model, the transformation from diagram to formal language is trivial, so long as the formal language is capable of capturing the subtle formal connections within the diagram. When choosing a computational language, certain factors need to be considered. These factors entail that the formal ontology is: 1) rigorous and formal; 2) interoperable; 3) user-friendly; 4) reusable; 5) customizable; and 6) capable of easily interfacing with both humans and other formal systems. Taking these factors into account, one must once again strike a balance between rational and empirical considerations. Rational considerations are those that point to the philosophical complexity and expressiveness of the formal ontology. One must be concerned with whether the ontology is capable of accurately capturing certain metaphysical relations within the world. Empirical considerations point to industry standards and computational constraints within engineering or computer science communities. In this regard, one must be concerned with whether the ontology will easily interface with the kinds of systems currently being used by members of those communities.

Due to such factors, it is important to survey the landscape of computer science and engineering, in order to develop an ontology which complies with the various kinds of systems currently under use within these subject areas. Description logics are one example of a commonly used computational tool for computer software design. Description logics have been extensively used by the AI community for the last couple of decades (Baader et al, 2003). They have been used for a plethora of applications including conceptual modeling, information integration, query mechanisms, view maintenance, software management systems, planning systems, configuration systems, and natural language understanding. Therefore, description logics possess the kind of industry standard that must be considered when building new kinds of conceptual modeling tools like formal ontologies. Description logics are knowledge representation languages tailored for expressive knowledge about concepts and concept hierarchies. They offer a balance between expressive power on one hand and computational complexity on the other, and are considered an important tool for unifying and giving a logical basis for the following items: 1) frame-based systems; 2) semantic networks; 3) KL-ONE-like languages; 4) object-oriented representations; 5) semantic data models; and 6) type systems.

The Dis-ReO is being modeled within the Protégé 2000 software tool from Stanford University's KSL laboratories (Noy & McGuinness, 2001; McGuinness et al, 2000). Protégé 2000 is a description logic tool that has been designed to be interoperable with DAML and other DARPA-funded AI projects (Hendler, et al, 2000). Since the Dis-ReO is meant to interoperate with private sector agencies, governmental agencies, and military agencies, it was important to use a software tool that has been designed with these agencies in mind. Another feature of Protégé 2000 is that it is easy to understand and use. It is a straight-forward tool designed for individuals who may not be entirely familiar with the intricate operations/structure of complex computational tools. Lastly, Protégé 2000 is shareware, so it can be both locally installed and it is highly cost effective.

3.6 Ontology Evaluation

The final step in the ontology development process is that of evaluation. It represents the most difficult step because the evaluation of the ontology amounts to an elaborate feedback mechanism, nearly identical to the form of feedback found within life-cycle models in other areas of systems engineering (see Figure 3).

An ontology's system life-cycle begins by defining the *needs* and *requirements* of that system which, along with research findings, influence the system's *conceptual design* (Blanchard, et al, 1998) . The conceptual design then serves as the ground for the system's *preliminary design*. The system's preliminary design should, in turn, provide a certain level of feedback that will re-influence and perhaps inform the conceptual design of the ontology. The preliminary design of the ontology will include those steps necessary for development of the lexicon and upper-level categories, for example. The ontology's preliminary design will then inform its *detailed design and development*. This step will include, among other things, the addition of the domain-specific concepts/categories harvested from the lexicon as well as the application of the ontology to some logic system (descriptive or other), amounting to a functional, computational prototype of the ontology. This step within the design process will, in turn, feedback once more to the overall conceptual design of the ontology. It is here where problems of fit between upper- and lower-level categories should become most apparent, since it is this stage where the ontology is first fitted to a computational system capable of expressing the relations therein. The detailed design of the ontology subsequently leads to steps involving the *production/construction* of a full-blown prototype that is housed within a software system. Production of the software's prototype leads to issues surrounding its *utilization/support* and finally its *phaseout/disposal*.

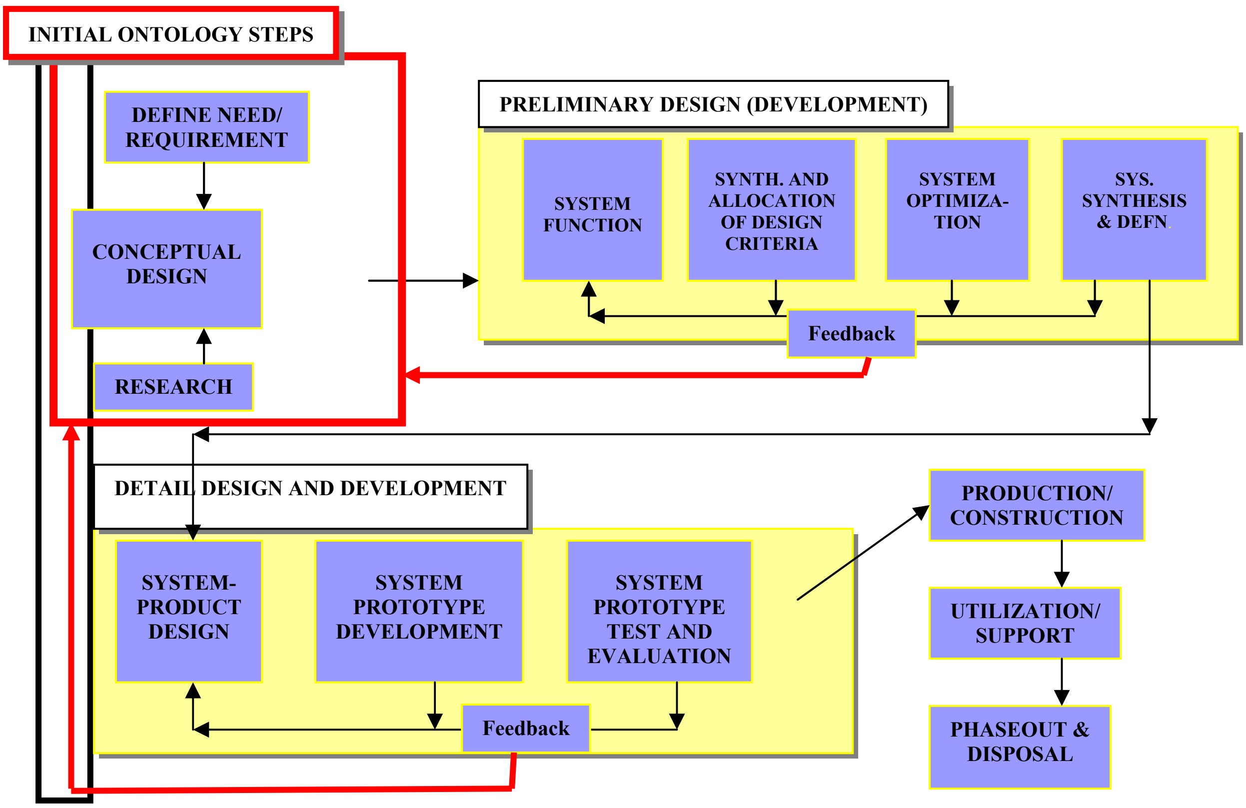

Figure 3: The System Life-Cycle for Ontology Design.

The difficulty in evaluating an ontology can hopefully be alleviated by a systematized life-cycle analysis, such as the one described above. Problems can still arise in the actual design of the ontology itself, since there are continued issues of fit between certain upper-level formal models such as the SnapBFO and certain descriptive logic systems such as Protégé 2000. However, the goal of systematics should not be understood solely in terms of the success of the prototype developed. The goal of systematics, and to a large degree, the goal of contemporary ontology design, should be focused on the methodology for its design, not the outcome. Systems engineers, after all, do not design products per se. They design systems which facilitate better product *development*. Therefore, the end result of ontology design, like that of systematics, can be conceived of as *the system itself*. We should be concerned here with a conceptual methodology first, and its product implementation second. If we can achieve a higher level of success at the conceptual and methodological level, successes at the implementation and application levels will follow suit.

4 Conclusion

It is the hope that formal ontological systems such as the Dis-ReO can aid various types of engineers and other practitioners by properly categorizing the various, disparate amounts of information contained within their disciplines. It is argued here that in order to do this job adequately, ontologies must be designed under the auspice of two complementary methodologies: rationalism and empiricism. By applying both methodologies to ontology design, ontologists will be able to: 1) better understand the function of ontology construction itself, as a philosophical exercise, and 2) better understand those empirical features of the world that an ontology categorizes. If applied correctly, the methodology argued for in this paper should be able to further marry the fields of engineering, computer science and philosophy into an overarching interdisciplinary field of *ontological engineering* capable of producing fantastic results across numerous other relevant fields.

Acknowledgements

Support for this research paper was graciously provided by The Center for Multisource Information Fusion (CMIF) at The University of Buffalo, State University of New York and The Calspan-UB Research Center (CUBRC) under the auspices of an Air Force Office of Scientific Research (AFOSR) Grant # F49620-01-1-0371. Special thanks go out to Prof. James Llinas (CMIF director) and Prof. Peter Scott (UB Dept. of Computer Science) for their support and guidance throughout the development of this project.

Academic References:

1. Baader, F., Diego, C., McGuiness D., Nardi, D., Patel-Schneider, P (eds.). (2003) *The Description Logic Handbook: Theory Implementation and Applications*, Cambridge: Cambridge Univ. Press.
2. Blanchard, B. and Fabrycky, W. (1998) *Systems Engineering and Analysis* 3rd Ed., Prentice Hall Pub.
3. Bowman, M., Lopez, Jr., A., Tecuci, G. (2001) "Ontology Development for Military Applications," Proceedings of the39th Annual ACM Southeast Conference, March, 112.
4. Casati, R. & Varzi, A. (1999) *Parts and Places*, Cambridge, MA: MIT Press.
5. Gibson, J.J. (1979) *The Ecological Approach to Visual Perception*, Boston: Houghton-Miflin.

6. Gruber, T. (1993) *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, Nicola Guarino (ed.), International Workshop on Formal Ontology, Padova, Italy.
7. Gruber, T. & Olson, G.R. (1994) *An Ontology for Engineering Mathematics*, Doyle, Torasso, Sandewall (eds.), Fourth International Conference on Principles of Knowledge Representation and Reasoning, Bonn: Morgan Kaufmann.
8. Gruber, T. (1995) "Toward principles for the design of ontologies used for knowledge sharing," in N. Guarino and R.Poli (eds.), *Formal Ontology in Conceptual Analysisand Knowledge Representation*.
9. Gruninger, M. and Fox, M. (1995) "Methodology for the design and evaluation of ontologies," in *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing* held in conjunction with IJCAI-95, Montreal, Canada.
10. Guarino, N. (1998) "Formal ontology in information systems," in N. Guarino (ed.), *Proc. Of Formal Ontology in Information Systems*, pp. 3-15, Trento, Italy, 6-8 June, Amsterdam: IOS Press.
11. Hall, D & Llinas, J. (eds.). (2001) *Handbook of Multisensor Data Fusion*, Boca Raton: CRC Press.
12. Hayes, P. (1985b) "Naive Physics I: Ontology of Liquids," in J. Hobbs and R. Moore (eds.), *Formal Theories of the Commonsense World*, Norwood, NJ: Ablex, pp. 71-108.
13. Heflin, J., Hendler, J., & Luke, S. (1999) "Coping with changing ontologies in a distributed environment," in AAAI-99 *Workshop on Ontology Management*, MIT Press.
14. Hendler, J. And McGuiness, D. (2000) "The DARPA Agent Markup Language," *IEEE Intelligent Systems*, 15, No. 6:67-73.
15. Husserl, E. (1900-01) *Logische Untersuchungen, 2 Bde*, *Husserliana, Band XIX*, Den Haag: Martinus Nijoff, 1985 ed.
16. Kokar, M.M. & Wang, J. (2002) "An example of using ontologies and symbolic information in automatic target recognition," in *Sensor Fusion: Architectures, Algorithms, and Applications VI*, pp. 40-50, SPIE.
17. Lenat, D.B. and Guha, R.V. (1990) *Building Large Knowledge-Based Systems*, Reading, MA: Addison-Wesley Pub. Inc..
18. Lenat, D. B. (1996) *Welcome to the Upper Cyc Ontology* http://www.cyc.com/overview.html.
19. Llinas, J. & Waltz, E. (1990) *Multisensor Data Fusion*, Boston: Artech House.
20. Mark, D. M., and Gould, M. D. (1992). Wayfinding as discourse: A comparison of verbal directions in English and Spanish. Multilingua, 11(3), 267-291.
21. McGuiness, D., Fikes, R., Rice, J., and Wilder, S. (2000) "An environment for merging and testing large ontologies", in *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning* (KR2000), Breckenridge, Colorado, April. Tech. report KSL-00-16, Knowledge Systems Laboratory, Stanford University.
22. Menzel, C. & Mayer, R. (1990) *IDEF 5 Ontology Description Capture Method: Concept Paper*, Research Institute for Computing and Information Systems, Univ. of Houston-Clear Lake.
23. Noy, N. and McGuinness, D. (2001) "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford University: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.

24. Simons, P. (1982) "Three Essays in Formal Ontology," in Barry Smith (ed.), *Parts and Moments: Studies in Logic and Formal Ontology*, Munich:Philosophia.
25. Simons, P. (1987) *Parts: A Study in Ontology*, Oxford: Oxford Univ. Press.
26. Slattery, N.J. (1997) "A Study of Ontology and Its Uses in Information Technology Systems," Software Engineering and Economics Conference. http://www.mitre.org/support/papers/swee/papers/slattery.
27. Smith, B. (1996) "Mereotopology: A Theory of Parts and Boundaries," *Data and Knowledge Engineering*, 20 (1996), 287–303.
28. Smith, B. (2001a) Fiat Objects, *Topoi*, 20: 2, 131–148, http://ontology.buffalo.edu/smith/articles/fiat.htm.
29. Smith, B. (2001b) "Objects and Their Environments: From Aristotle to Ecological Psychology," in Frank, A., Raper, J., and Cheylan, J.-P., *The Life and Motion of Socio-Economic Units*.(GISDATA 8). London: Taylor and Francis, 79-97.
30. Smith, B. (2001c) True Grid, in Daniel Montello (ed.), *Spatial Information Theory. Foundations of Geographic Information Science, Proceedings of COSIT 2001*, Morro Bay, California, September 2001 (Lecture Notes in Computer Science 2205), Berlin/New York: Springer, 14–27.
31. Smith, B. "Ontology and Information Science," *Stanford Encyclopedia of Philosophy*, forthcoming.
32. Uschold, M. and Gruninger, M. (1996) "Ontologies: Principles, Methods and Applications," *The Knowledge Engineering Review*, 11(2), pp. 93-136.
33. Welty, C. & Smith, B. (eds.) (2001) *Formal Ontology and Information Systems*, New York: ACM Press.

Disaster Lexicon References:

1. American Red Cross 3000 Series Directory (http://www.tallytown.com/redcross/dsp/form3001.pdf)

2. The Center for Research on the Epidemiology of Disasters (http://www.cred.be/emdat/intro.html)

3. Disaster Recovery Journal (http://www.drj.com/glossary/DRJ-Glossary.doc)

4. Disaster-Relief.Org (http://www.disasterrelief.org/Library/Dictionary/)

5. Emergency Management Glossary (U.S. Steel Gary Works) (ftp://www.emforum.org/pub/eiip/empglos.doc)

6. FEMA (http://www.fema.gov/rrr/frp/frpesf1.shtm#terms)

7. Pacific Disaster Center (http://www.pdc.org/iweb/)

8. Private and Public Businesses Inc. (FEMA Glossary) (http://www.ppbi.org/fema.htm)

9. Saint Lucia Disaster Plan: Sec. 13 (National Emergency Management Organization (NEMO)) (http://www.geocities.com/CapitolHill/Lobby/6075/Glossary.htm)

10. Suburban Emergency Management Project (SEMP) of Dupage County Ill. (http://www.ben.edu/semp/htmlpages/glossarya1.html)