A Faceted Approach to Building Ontologies

Rubén Prieto-Díaz
Commonwealth Information Security Center
James Madison University
prietodiaz@cisat.jmu.edu

1. Introduction

An ontology is "an explicit conceptualization of a domain of discourse, and thus provides a shared and common understanding of the domain."[Reim01] We have been producing ontologies for millennia to understand and explain our rationale and environment. From Plato's philosophical framework to modern day classification systems, ontologies are, in most cases, the product of extensive analysis and categorization.

Only recently has the process of building ontologies become a research topic of interest. Today, ontologies are built very much ad-hoc. A terminology is first developed providing a controlled vocabulary for the subject area or domain of interest, then it is organized into a taxonomy where key concepts are identified, and finally these concepts are defined and related to create an ontology.

The intent of this paper is to show that domain analysis methods can be used for building ontologies. Domain analysis aims at generic models that represent groups of similar systems within an application domain. In this sense, it deals with categorization of common objects and operations, with clear, unambiguous definitions of them and with defining their relationships.

1.1. Background

Typically, the goal of building ontologies is to create a logical framework, a philosophy, a classification, or to develop a common understanding in a discipline. The goal determines the extent and complexity of the process. Creating an ontology intended only to provide a basic understanding of a domain may require less effort than an ontology intended for supporting formal logic arguments and proofs in a domain.

Answering questions such as: Why are we building an ontology? What we want to use it for? Is the initial first step in creating an ontology.

A "skeletal" methodology for building ontologies has been proposed and tested by Uschold and King [Usch95]. This attempt to formalize the ad-hoc process consists of the following steps:

- Identify Purpose
- Build Ontology
  - Ontology capture
  - Ontology coding
  - Ontology integration
- Evaluate Ontology
- Document Ontology

There are three sub-steps in the Build Ontology process.

1- **Ontology capture** is the identification and definition of key concepts and relationships in the domain of interest and the terms that refer to such concepts.
2- **Ontology coding** deals with formalizing such definitions and relationships in some formal language.
3- **Ontology integration** deals with associating key concepts and terms in the ontology with concepts and terms of other ontologies; that is, incorporating concepts and terms from other domains.

Uschold et.al., used this approach to create an "Enterprise Ontology" [Usch98]. The TOVE (TOronto Virtual Enterprise) project from University of Toronto's Enterprise Integration Laboratory has developed and tested several ontologies for modeling enterprises[1] [Fox94]. TOVE's approach to engineering ontologies consists of four basic steps [Fox98], in essence very similar to the four steps proposed by Uschold and King above:

- Define ontology requirements.
- Define the terminology of the ontology (objects, attributes and relations).
- Specify the definitions and constraints on the terminology.
- Test the competency of the ontology.

Building ontologies is more difficult than it seems. In a special issue of Communications of the ACM on Ontology Engineering, Gruninger and Lee [Grun02] indicate that "building ontologies is difficult, time-consuming, and expensive." It involves more than

the steps in the above two approaches: it also requires consensus building. Stemming from this difficulty Holsapple and Joshi [Hols02] have proposed a collaborative approach to ontology design.

Despite the need for consensus building, the first two steps in Uschold and King's and in TOVE's approaches are essential. In this paper we argue that domain analysis provides a method and techniques for supporting these first two steps, in particular the three sub-steps in Uschold and King's "Build Ontologies" process.

This paper describes how a domain analysis method can be used for building the basis for ontologies. Section 2 relates both processes and makes the case that domain analysis can be used for building ontologies. Section 3 illustrates a step in the domain analysis method for identifying and categorizing concepts borrowed from Library Science. Section 4 describes how the faceted approach from Library Science is incorporated into the domain analysis method. Section 4 gives an overview of the method and describes a tool for automating parts of the process.

2. Domain Analysis and Ontologies

The prevailing problem of how to capture, structure, and formalize knowledge for reuse has been the focus of several research efforts in the software reuse community. Domain analysis is a process by which information used in developing software systems is identified, captured, and organized for the purpose of making it reusable [Prie90]. More specifically, domain analysis is the process of discovering and defining domain models for supporting pre-planned, systematic software reuse. Domain models support reuse of objects and operations within an architectural framework common to all systems in the application domain. A domain model can be considered as a very narrow or specialized ontology.

Systematic methods for domain analysis, such as FODA [Kang90] and DARE [Frak98], intend to formalize the ad-hoc nature of the process. We claim that the process for building a narrow ontology is almost identical to the process for building a broad domain model.

[1] http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html

If conducted ad-hoc, the domain analysis process allows knowledge of a domain to evolve naturally over time until enough experience has been accumulated and several systems have been implemented. At that point, common objects and operations can be identified and generic abstractions can be isolated and reused. As we will see, identifying objects and operations common to a family of systems, categorizing them, and abstracting their commonalities is equivalent to Uschold and King's first step in the ontology building process: ontology capture. Specifying reusable components that implement generic functions as defined by the abstractions in the domain model is equivalent to the second step in the ontology building process: ontology coding. Lastly, ontology integration is equivalent to incorporating definitions of components from other domains capable of carrying out some of the functionality in the domain being considered. The table below illustrates how little difference there is between an ontology and the product of a domain analysis: a domain model.

Feature	Domain Model	Ontology
Controlled Vocabulary	Yes	Yes
Taxonomy	Yes	Yes
Thesaurus	Yes	Yes
Abstract Concept Definitions	Informal	Formal
Semantic Relationships	Yes	Yes
Multiple Viewpoint Models	Yes	Yes
Axioms	No	Yes
Cross-Domain Associations	Implicit (Via Thesaurus)	Explicit
Formal Notation	No	Yes

Ontology capture, in particular, can be completely realized through domain analysis thus providing sufficient information and concepts to facilitate the following tasks of ontology building: ontology coding, integration and formal documentation. In other words, the ontologies produced through domain analysis include the basic concepts and relationships that make them usable and practical, and provide the basis for further refinement.

2.1. Taxonomies, Ontologies and Classification

A taxonomy is a structure of categories and classification is the act of assigning entities to categories within a taxonomy. Reimer [Reim01] defines a taxonomy as "a controlled vocabulary which is arranged in a concept hierarchy" and ontology as "a taxonomy

where the meaning of each concept is defined by specifying properties, relations to other concepts, and axioms narrowing down the interpretation."

A classification scheme in Library Science is a tool for the production of systematic order based on a controlled and structured index vocabulary. This index vocabulary is called the classification schedule and it consists of a set of names or terms representing concepts or classes, listed in systematic order, to display relationships among classes.

A classification scheme and its respective schedule then, can be considered an extended taxonomy or a reduced ontology. As an extended taxonomy it goes beyond a mere arrangement of categories since it includes relationships among categories and some brief definitions. Thesaurus-like associations provide some of the relationships. As a reduced ontology, it lacks formal definitions of concepts and axioms.

Based on this analysis and the similarity between domain analysis and building ontologies, techniques for deriving classification schemes can be used for systematically initiating the creation of ontologies.

The focus of our discussion in Section 3 is on an approach for the identification, structuring and definition of concepts and terms of a domain adopted from Library Science for creating faceted classification schemes for special collections. This approach is part of the DARE method [Frak98].

3. Building Ontologies

3.1. Classification in Library Science

A classification scheme must be able to express hierarchical relationships as well as relationships created to relate two or more concepts belonging to different hierarchies. Hierarchical relationships are based on the principle of subordination or inclusion and are typical in a taxonomy. Relationships among concepts are presented as compounded classes. For example, the compounded class "reproduction of reptiles" relates the term

"reproduction" from the class processes with the term "reptiles" from the class taxonomy.

Two types of classification schemes are used in Library Science: enumerative and faceted. The enumerative (or traditional) method postulates a universe of knowledge divided into successively narrower classes that include all possible subclasses and compounded classes. The Dewey Decimal system is a typical example of an enumerative hierarchy where all possible classes are predefined. These schemes are called enumerative because the predefined classes are listed ready-made in a classification schedule.

The faceted approach, proposed by Ranganathan in 1939 [Ran67], relies not on the breakdown of a universe of knowledge, but on building up or synthesizing from the subject statements of particular documents. Subject statements are analyzed into their component elemental classes selected from the schedule. The classifier using such a scheme expresses a compound class by assembling its elemental classes. This process is called *synthesis*. The arranged groups of elemental classes making up the scheme are called *facets*. Facets can be construed as perspectives, viewpoints, or dimensions of a particular domain.

A faceted scheme, therefore provides a controlled vocabulary in the form of terms arranged systematically by facets and a set of rules on how to combine such terms to define conceptual descriptors (i.e., categories).

3.2. Deriving Faceted Classification Schemes

Special library collections are typically classified using custom-made classification schemes. Most of these schemes are faceted and they are built using a process called *Literary Warrant* [Vick60]. Literary Warrant consists in selecting a random sample of titles from the collection to be classified, listing individual terms from the titles, grouping related terms into common classes, and organizing the common classes into a classification scheme. This process requires knowledge of the domain and of the intended use of the collection. The selected terms and their relationships can be

considered as a domain specific language used to express activities in the domain of the specialized library. Literary Warrant considers that titles capture best, in a simplified form, the concepts in a document and they are used as representative subject statements.

We illustrate this process with an example from [Buch79]. Assume we are asked to build a classification for a list of zoology related titles (i.e., books). The first step is to select a representative sample from the collection. Let us assume we select the following titles:

"Essays of the physiology of marine fauna"
"Animals of the mountains"
"Amphibious animals"
"Desert reptiles"
"Migratory Birds"
"Salt water fish"
"Mammalian Reproduction"
"Snakes of the Amazon River"
"Experimental reports on the respiration of vertebrates"
"Tropical leaf moths"

The next step is to group common terms together (i.e., conceptual clustering)

physiology, reproduction, respiration
tropical, desert, mountains, salt water
marine, amphibious
fauna, animals, vertebrates, reptiles, snakes, birds, fish, moths, mammals
essays, experimental reports

These five groups are the initial facets in our special collection of zoology books. Each group is named by the general concept it represents.

by process
by habitat
by element
by taxonomy
by literary form

These five facets are ordered by their relevance to the users of the collection and terms within each group are listed in a logical order. It is assumed in this example that the users of the collection are mainly ecologists making habitat the most relevant facet. The domain or subject area is animals/fauna and the resulting faceted classification scheme is shown below.

{by habitat}	*{by element}*	*{by taxonomy}*	*{by process}*	*{by literary form}*
land	marine	animals/fauna	physiology	essays
tropic	:	invertebrates	respiration	:
desert	:	insects	reproduction	:
mountain	amphibious	moths	:	reports
:	:	:	:	experimental
:	:	vertebrates	:	:
water		mammals		
sea		:		
river		birds		
Amazon		:		
:		reptiles		
:		snakes		
		:		
		fish		

As new titles enter the collection, new facets may be defined and new terms added to the scheme, thus extending and enriching the faceted scheme. This is the core of Literary Warrant. [Buch79] and [Vick60] present detailed tutorials.

This scheme is a structured controlled vocabulary that can be used systematically to define each title of the collection. Each title can now be reduced to a normal form of terms from each facet. To describe a title using this scheme we match by order of relevance each term in the title to the term in the scheme. For example:

The title "Essays on the physiology of marine fauna" can be represented (i.e., classified) by the following set of terms selected from the faceted scheme.

/null/marine/animals/physiology/essays/

The first entry is null because there is no term (i.e., concept) in the title that corresponds to any concept in the habitat facet. The remaining terms are selected by conceptually matching keywords in the title to facet terms in the scheme.

Similarly, the titles below are normalized (i.e., classified) following the same process.

"Animals of the mountains" → /mountain/null/animals/null/null/

"Desert reptiles" → /desert/null/reptiles/null/null/

This is the process of synthetic classification using a faceted classification scheme. The scheme provides a vocabulary and some basic rules for converting titles to a normalized set of concepts.

In summary, we have produced, from the bottom-up, a knowledge structure that can be used to generate normalized descriptors of statements that facilitate their categorization: titles sharing the same facet terms belong to same category. This can also be seen as a primitive domain language. In [Prie87] we used function descriptors, instead of book titles, to generate a faceted classification scheme for software components as part of a reuse library system. Function descriptors are, in most cases, one-liners defining a software function. In some cases a function descriptor is the title of the function. Function descriptors were found to be representative subject statements.

4. Domain Analysis Approach to Building Ontologies

The proposed domain analysis approach to building ontologies is based on a combined top-down and bottom-up method. Essentially, a high level ontology is postulated, then it is revised and validated based on a bottom-up analysis of existing domain specific documents (Figure 1). While the top-down method is highly creative, somewhat informal and manual, the bottom-up part can be made systematic, repeatable, and partially automatic. In the top-down process experts in the domain work together to identify key concepts and postulate and capture an initial high level ontology.

In the bottom-up process keywords and phrases are extracted from domain documents using standard text analysis tools. The Literary Warrant technique is then used to build a domain specific faceted classification scheme. The resulting scheme is used to group phrases into categories thus creating clusters that represent concepts in the domain.

This approach is consistent with Lakoff's view of experimental realism and the concept of "prototype-based categories" which is based on "embodied" cognitive models [Lako90]. Lakoff argues and demonstrates that ontologies are based on our perception of reality and that any effort to build ontologies must follow such perceptions rather

than built on abstract concepts imposed from rational arguments. Lakoff's position seems to be accepted by many.

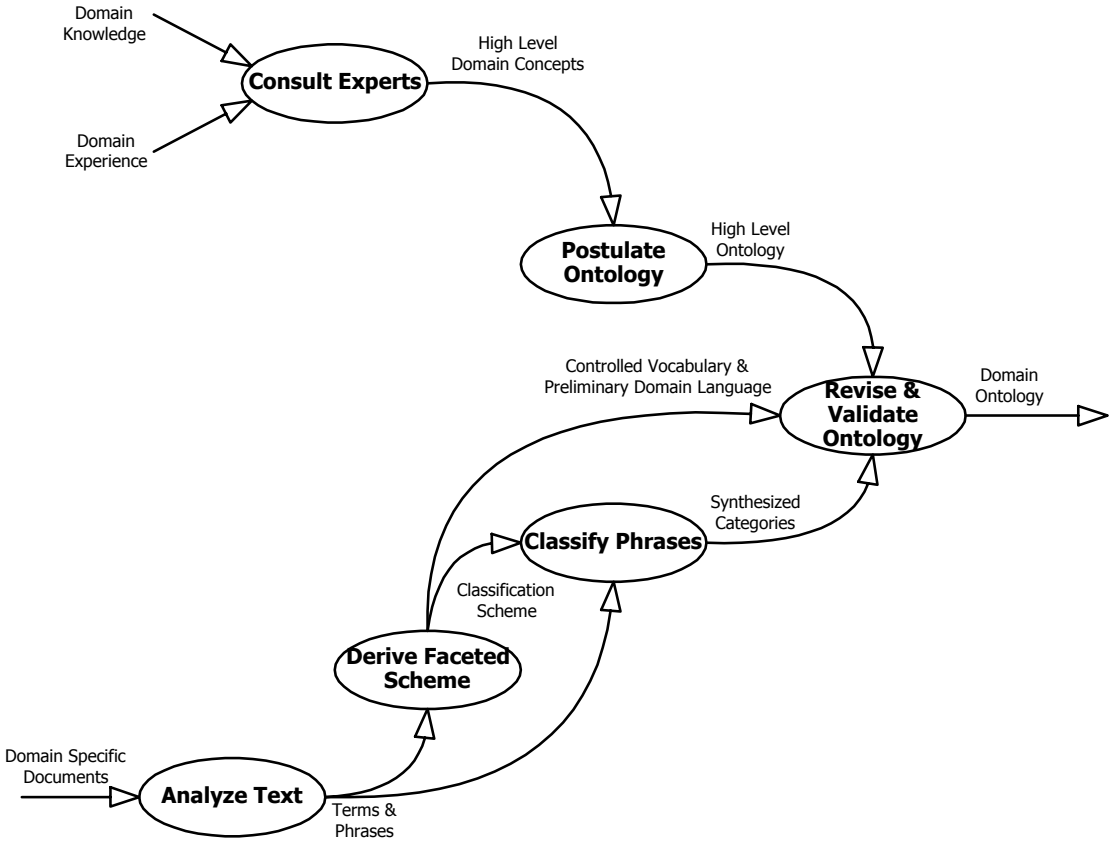

Figure 1 - A Process for Building Ontologies

Next, the synthesized clusters are compared to the postulated concepts and an iterative process is carried out where the ontology is modified and adjusted to match the bottom-up clusters. Figures 2 & 3 illustrate how this method is carried out.

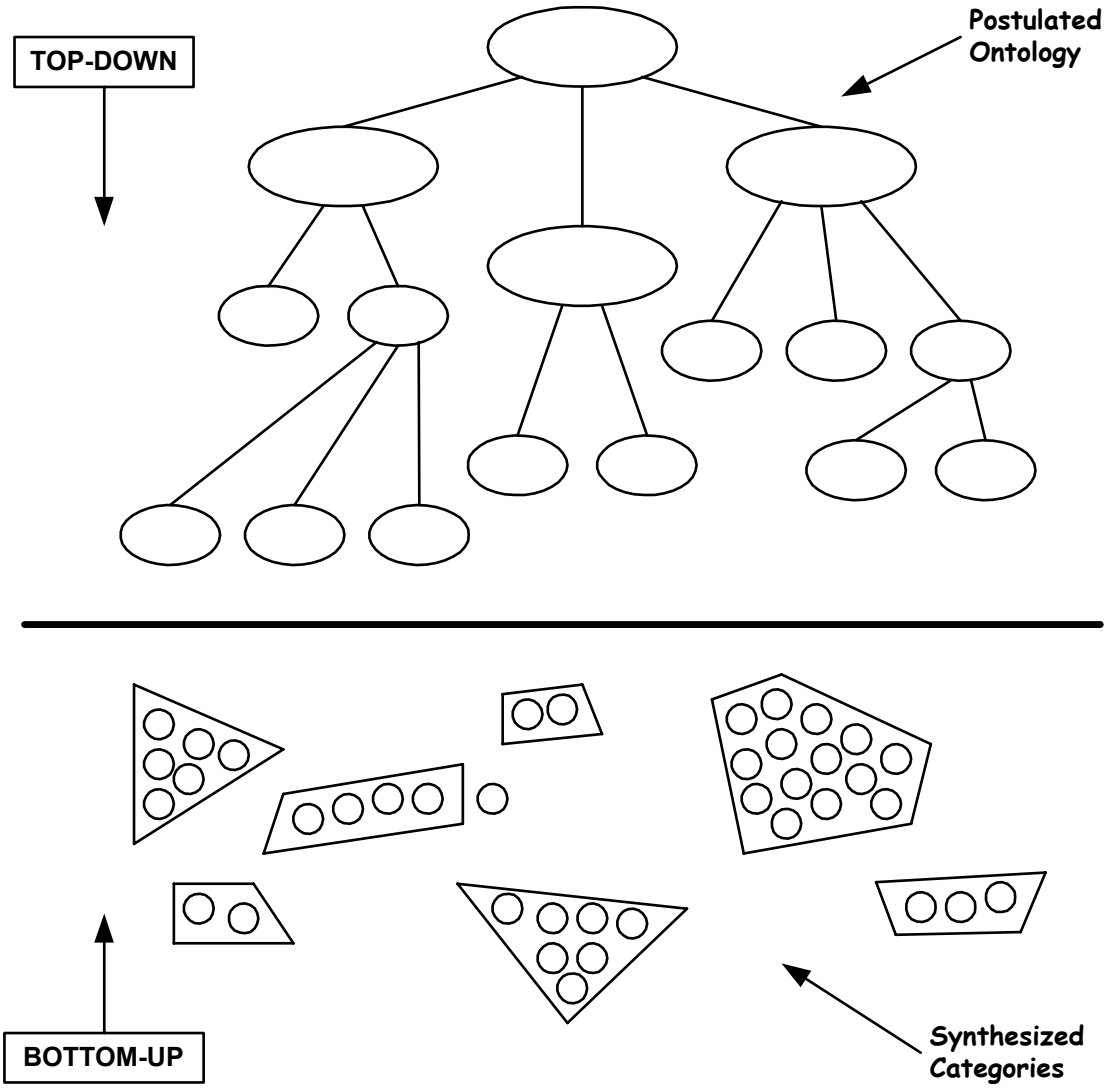

Figure 2 - Top-Down and Bottom-Up Elements of the Process

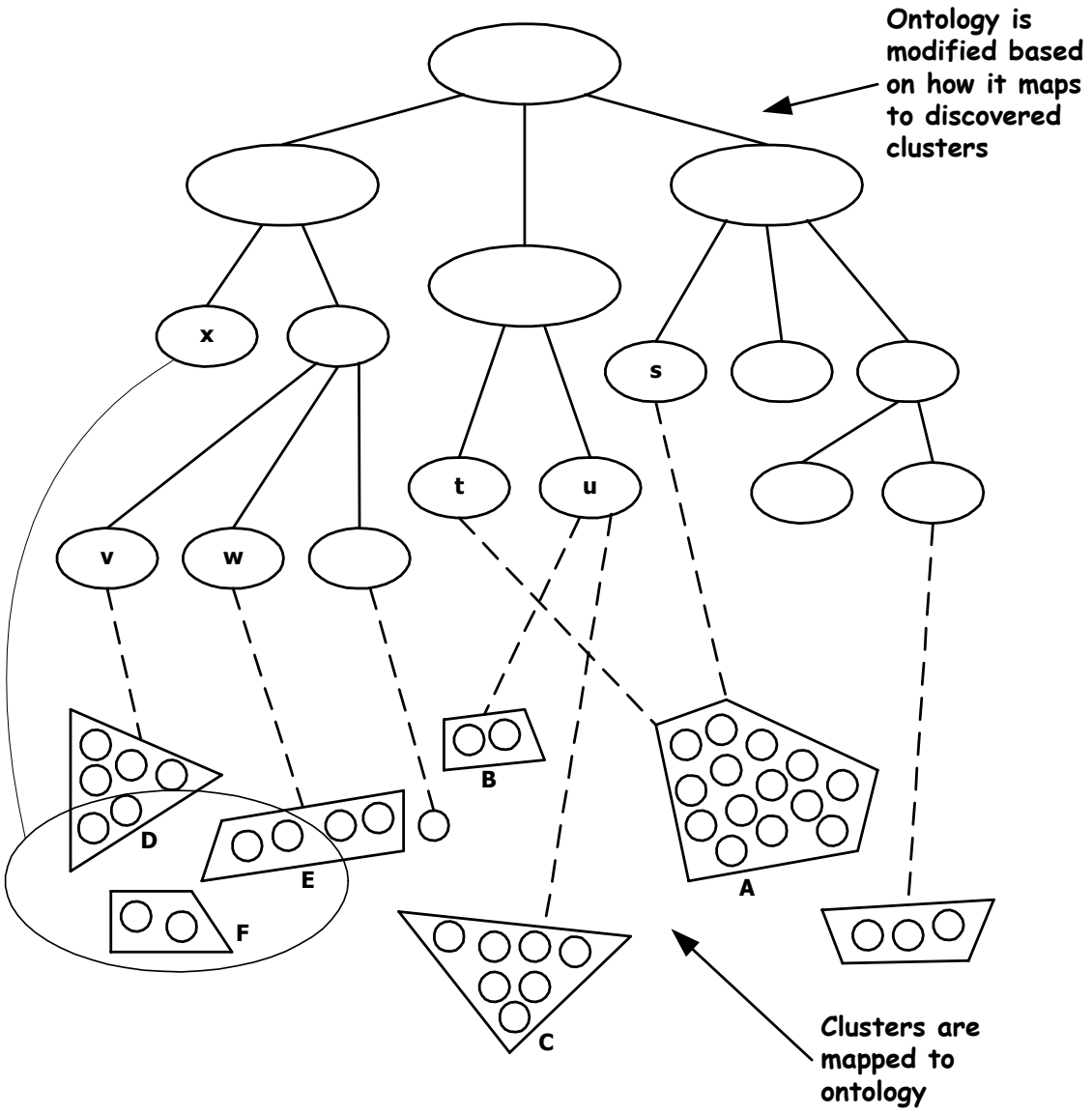

Figure 3 - Revising and Validating Initial Ontology

Mapping clusters to proposed concepts may require modifications to the ontology and the concept clusters. Some of the following situations may arise:

- Cluster **A** maps to concepts **s** and **t** (See Figure 3). In this case cluster **A** can be broken into two separate concepts, one matching **s** and another matching **t** or either **s** or **t** can be deleted from the ontology and keep only one link to **A** from either the parent of **t** or the parent of **s**.

- Clusters **B** and **C** map to single concept **u**. In this case clusters **B** and **C** can be merged to represent concept **u** or concept **u** can be partitioned into two different concepts.

- Elements from clusters **D** and **E** and cluster **F** map to concept **x**. In this case a new cluster can be created to map to concept **x** or concept **x** can be deleted and cluster **F** merged into **D** and **E**.

These examples illustrate how a postulated ontology is modified and validated by concepts extracted from domain documents.

5. Mechanizing the Bottom-Up Process

The DARE tool [Frak98] supports a semi-automatic approach for building faceted classification schemes. Keywords and phrases are extracted and filtered from domain specific documents. An interface-assisted cluster editor supports grouping similar concepts that lead to the creation of a faceted scheme and corresponding thesauri for every facet.

5.1. Analyzing Text

Text analysis is the key element for automating the bottom-up process. The goal is to extract a list of keywords and phrases that denote unique concepts in the analyzed domain. This keyword index is the main input to the cluster editor.

Automated text processing is a mature technology. The concepts, techniques and algorithms adopted by DARE are mainly from Frakes & Baeza-Yates' textbook [Frak92]. The text from a document is broken into words by a tokenizing operation (i.e., lexical analysis) and all stop words removed. The resulting keywords are placed in an internal index and used to generate stems. An inverted index is also created for traceability of keywords to their original sources. Phrases are simplified to keywords and stems.

5.2. Conceptual Clustering

DARE's clustering editor provides the user with an initial set of conceptual clusters in a graphical interface that supports drag-and-drop. The user can modify, group, and regroup clusters to define facets. At the heart of the clustering editor is an algorithm that creates lists of keyword associations and generates association nets. This algorithm computes conceptual similarity between keywords using a coefficient of similarity measure. A phrase extraction algorithm is also used. It works by measuring co-occurrence of keywords within statements as proposed by Maarek et. al. [Maar91]. In

essence, the clustering editor combines two proven syntactic techniques for generating quasi-semantic clusters.

A cluster is a group of terms and phrases whose similarities fall within a specific, user defined conceptual distance from each other. Clusters are represented by a single central term with radial terms extending outward from it. The user can manipulate term positions thus increasing or decreasing their relationships, he can bring terms from the keyword list into a cluster or remove terms from the cluster, and he can merge clusters and rearrange them.

5.3. Defining Facets

Providing an initial set of candidate clusters significantly simplifies the task of building faceted classification schemes. These clusters provide initial groups of related terms that help the builder discover facets. Facets are "discovered" when the builder feels comfortable with a cluster of related terms and gives the cluster a name that captures all the concepts within. Defining facets is an iterative process where an initial set of clusters is analyzed and regrouped several times. The shape of the cluster can also provide clues for identifying synonyms, broader terms, and narrower terms within a facet. Once a set of facets is defined, new terms from text analysis can be added to the existing facets or new clusters can be created.

The DARE method recommends building a thesaurus for each facet. Very often terms in a cluster are synonyms of each other or represent essentially the same concept. In these cases a term is selected to represent the concept. We call this selected synonym the canonical term, which is similar to Rosch's "prototype" [Lako90]. Each facet therefore includes canonical terms in its list and each term represents a concept.

The categorization effectiveness of the faceted scheme is determined in part by the number of facets and by the number of terms in each facet. On the one hand, too few facets and terms limits the number of possible categories and decreases precision but facilitates classification. On the other, too many facets and terms provides for large variety of categories and precision but makes classification difficult. In our experience, a

practical and useful faceted scheme should have 4 to 7 facets and no more than five synonyms for any one canonical term.

5.4. Classifying Phrases

The resulting faceted scheme is a structured controlled vocabulary with specific classification rules for synthesizing categories in the domain. Words in phrases are mapped to terms in facets as illustrated in the examples above. Phrases and some statements are "normalized" to ordered lists of facet terms. Phrases and terms in the same category are grouped to form clusters. These clusters are mapped to the high-level ontology as illustrated in figures 2 & 3.

Use of the DARE tool demonstrated that software component descriptions and requirements definitions required very little human interaction. Such documents are usually written in a consistent style. Use of prose as in non-technical text typically demands a higher level of human interaction in the process. More details on our experiences using DARE are reported in [Frak98].

6. Conclusion

We have described a tool-assisted method for building the basis for ontologies adopted from domain analysis. The resulting ontologies do not include formal definitions of concepts and axioms. Instead a structured controlled vocabulary is produced that define concepts informally and indirectly by example. Concepts are defined by clusters of phrases and statements extracted from a body of textual experience.

One advantage of this approach is that it is practical and useful. The ontologies built by this method may not yet be comprehensive or formal enough for some purposes but they provide sufficient information and concepts to facilitate the task of ontology coding and formal documentation.

Our experiences using DARE and applying domain analysis methods have been generally very positive and include results with immediate applications such as domain models for command and control systems and for banking services. More research,

however, is needed on how to convert or evolve domain models into complete formalized ontologies.

7. Acknowledgments

This research was partially supported by Financial Systems Architects, New York. Special thanks to Sam Redwine for his comments and multiple reviews of this manuscript.

8. References

[Buch79] Buchanan, B. *Theory of Library Classification*. Clive Bingley, London, 1979.

[Fox93] Fox, M. et.al. "A Common Sense Model of the Enterprise", *Proceedings of the 2nd Industrial Engineering Research Conference* , pp. 425-429, Norcross GA: Institute for Industrial Engineers, 1993.

[Fox98] Fox, M. et.al. "An Organisation Ontology for Enterprise Modeling", In *Simulating Organizations: Computational Models of Institutions and Groups*, M. Prietula, K. Carley & L. Gasser (Eds), Menlo Park CA: AAAI/MIT Press, pp. 131-152, 1998.

[Frak98] Frakes, W., Prieto-Díaz, R. and Fox C. DARE: domain analysis and reuse environment. In *Annals of Software Engineering*, **(5)**125-141, W. Frakes (Ed.) Baltzer Science Publishers, September 1998.

[Frak92] Frakes, W.B. and Baeza-Yates, R. (Eds.) *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Englewood Cliffs, NJ. 1992.

[Grun02] Gruninger, M. and Lee, J. "Ontology Applications and Design" Introductory article to a special issue on Ontology Engineering. Communications of the ACM 45(2):39-41, February, 2002.

[Hols02] Holsapple, C.W. and Joshi, K.D. "A Collaborative Approach to Ontology Design." Communications of the ACM 45(2):42-47, February, 2002.

[Kang90] Kang, K., et. al. *Feature-Oriented Domain Analysis (FODA) Feasibility Study.* CMU/SEI-90-TR-21. Software Engineering Institute, Pittsburg, PA, November, 1990.

[Lako90] Lakoff, G. *Women, Fire, and Dangerous Things, What Categories Reveal about the Mind.* The University of Chicago Press, 1990. (First published 1984)

[Maar91] Maarek, Y., Berry, D. and Kaiser, G. An information retrieval approach for automatically constructing software libraries. *IEEE Transactions on Software Engineering*, **17**(8):800-813, August, 1991.

[Prie87] Prieto-Díaz, R. and Freeman, P. Classifying software for reusability. *IEEE Software*, **4**(1):6-16, January 1987.

[Prie90] Prieto-Díaz, R. Domain analysis: an introduction. ACM SIGSOFT Software Engineering Notes **15**(2):47-54, April, 1990.

[Rang67] Ranganathan, S.R. *Prolegomena to Library Classification.* Asian Publishing House, Bombay, India, 1967.

[Reim01] Reimer, U. *Tutorial on Organizational Memories for Capturing, Sharing and Utilizing Knowledge*. International Conference on Enterprise Information Systems, ICEIS 2001, Setubal, Portugal, July 7-10, 2001. http://research.swisslife.ch/~reimer/OM_Tutorial/index.html

[Usch95] Uschold, M. and King, M. *Towards a Methodology for Building Ontologies.* AIAI-TR-183, University of Edinburgh, Edinburgh EH1 1HN, 1995. Presented at the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI95, Montreal. http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html

[Usch98] Uschold, M. et.al. *The Enterprise Ontology* The Knowledge Engineering Review , Vol. 13, Special Issue on Putting Ontologies to Use (eds. Mike Uschold and Austin Tate), 1998. Also available from AIAI as AIAI-TR-195 at: http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html

[Vick60] Vickery, V.C. *Faceted Classification: A Guide to Construction and Use of Special Schemes.* Aslib, 3 Belgrave Square, London, 1960.

Abstract

An ontology can be defined as a conceptualization of a domain or subject area typically captured in an abstract model of how people think about things in the domain. Humans have been producing ontologies for millennia to understand and explain our rationale and environment.

Only recently has the process of building ontologies become a research topic of interest. Today, ontologies are built very much ad-hoc. A terminology is first developed providing a controlled vocabulary for the subject area or domain of interest, then it is organized into a taxonomy where key concepts are identified, and finally these concepts are defined and related to create an ontology.

This paper describes how a domain analysis method based on faceted classification can be used for building ontologies. It relates domain analysis and ontologies, illustrates a step in the domain analysis method for identifying and categorizing concepts, and describes how this step, borrowed from Library Science, is incorporated into the domain analysis method. The paper also gives an overview of the method and describes a tool for automating parts of the process.