Assessment of the Reusability of Ontologies: A Practical Example

María Angeles Capellades
Department of Electrical Engineering
The Pennsylvania State University
121 Electrical Engineering East, University Park, PA 16802
Email: mangeles@psu.edu

Abstract

Ontologies provide a common conceptualization of the world to be shared across systems. Their reusability is assumed to be one of the main advantages of their application in the development of knowledge systems. Thus, libraries of reusable ontologies have already been built and made available. This paper reports an attempt to reuse one ontology available for public use, and the practical developments associated with such attempt: the implementation of a practical ontology browser and the introduction of a reusability index. Although reusability is a goal difficult to achieve, it is concluded that, at this stage, properly assessing the reusability of existing ontologies is advisable since it complements and aids the task of building a technical ontology from scratch.

Introduction

Ontologies, defined as agreements about shared conceptualizations (Uschold & Gruninger 1996), which include conceptual frameworks for modeling domain knowledge, are growing in number and are increasingly being used in Knowledge Management and Engineering. Ontologies are usually specified in the form of definitions of a representational vocabulary, and they are designed with the purpose of sharing knowledge across systems (Guarino 1997; Uschold & Gruninger 1996).

One of the most interesting goals of the ARPA Knowledge Sharing Effort (Patil *et al.* 1992) is to define an strategy for the development of reusable ontologies. Having a bank of ready-to-use ontologies will possibly make the tedious task of developing complex ontologies more efficient. One such database of reusable ontologies is already available at the Knowledge Systems Laboratory (KSL) at Stanford University (Ontolingua.Stanford.edu). This database is being built using the Ontolingua server, which provides an on-line ontology browser, editor, and translator through the www. However, in spite of all these efforts, ontology reuse remains a difficult process that defies optimization and systematization.

Procedures for building ontologies usually underestimate the difficulty of reuse. For example, a procedure documented in the literature for the development of an ontology describes the steps involved in building an ontology as follows (Uschold & Gruninger 1996):

a. Identify the purpose of the ontology, the problem that needs to be solved and why a common vocabulary is needed to solve it.
b. Capture the ontology by:
   i. Identifying key concepts and relations in the Universe of Discourse of the problem. This is a categorization problem.
   ii. Producing precise consistent text definitions for the concepts and relations identified in the previous step.
   iii. Identify the terms that will be used to refer to such concepts and relationships.
c. Coding of the ontology using some representational language.
d. Integrate existing ontologies into the one being built.

This methodology, although simple and concise, does not address important problems in ontology reuse such as negotiation and evaluation, and does not properly place ontology reuse in the process.

Whereas the purpose of the ontology is usually clear, the key concepts and relations that should be included in it are not, which suggests that negotiation between the members of the development team must take place. In some cases, reusability can be strongly severed because developers cannot reconcile their different views of what is key and what is not. This has also been noted by O'Leary (1997) since he says "The choice of any one view, abstraction or position provides an advantage to one developer of computational agents (e.g. one ontology may be more natural or more easily developed), generally at the expense of another developer". Detailed and explicit negotiation procedures are necessary to build an ontology.

Besides negotiation procedures, evaluation procedures are also necessary to assess the completeness and consistency of the ontology being developed as well as of existing ontologies. Several evaluation methods have been reported in the literature (Gómez-Pérez 1995; Noy & Hafner 1997), but they cannot easily be automatized, and cannot be modified to give a quantitative

measure of reusability. Such a measure would be useful to quickly classify ontologies for potential reuse. However, since there is not a way to quantify either the completeness of a particular taxonomy of concepts or the adequacy of the definitions given in an ontology, finding this measure is a difficult task. The work associated to the decide-to-reuse process can be so intense at times that the development of a method by which to quickly and objectively assess reusability should be a priority in ontology research.

As it was said in the previous paragraph, reuse requires a time-consuming, in-depth analysis of the choices made in existing ontologies. This decision process involves a different kind of negotiation than the one described in a previous paragraph. In this case, the developer has to determine how, when and if to use or modify all, some, or none of the concepts in existing ontologies as opposed to define them from scratch. Thus, a thorough revision and analysis of potential candidate ontologies for reuse must be performed *prior* to any other step. Powerful and easy-to-use browsing tools would be appropriate to facilitate this process.

The work described here sprang from the design of an ontology for a multiagent system that interprets mechanical engineering drawings called MEDTooL (Capellades 1999). The main goal of the present paper is to document the problems encountered by the author when reusing ontologies and her approach to ease these problems: the design and implementation of an integrated ontology browser and the introduction of a reusability index. A brief description of known impediments to ontology reuse is given first, followed by the ontological requirements of MEDTooL, and by a recount of the problems encountered when trying to integrate other ontologies to the MEDTooL ontology. A description of how these problems were partially solved is then presented, and finally, considerations about ontology reuse and the conclusions of the paper are given.

Known Impediments to Ontology Reuse

Several impediments that hinder ontology reuse have already been investigated and reported in the literature (Noy & Hafner 1997; O'Leary 1997). These impediments are:

1. Choices made for one system might not be suitable for another, specially choices made in the capture process. These choices are affected by the *Interaction Hypothesis*.
2. As complexity increases, definitional precision decreases. The more complex a system is, the more difficult is to determine which are the key concepts involved and to find suitable definitions for them.
3. Integrating existing ontologies might be very difficult due to similarity of vocabulary, to conflicting views of the world, and to undetected internal inconsistencies, to name some.
4. Ontologies are being developed in an informal way. Since a formal methodology does not exist, it is not clear from the body of a given ontology why certain choices were made, even when there is documentation available.
5. The lack of correspondence between existent ontologies in their top-level definitions (in their choice of basic definitions and axioms) is a big obstacle for integration.

A problem found in one of the most recent examples of ontology reuse (Uschold *et al.* 1998) occurs in the translation process:

6. Due to fundamental semantic differences between the language in which the ontology was written an the language in which is intended to be used, the translation process might be either too costly or impossible to be carried out.

MEDTooL's Ontological Requirements

MEDTooL is a multi-agent system that analyzes the two-dimensional binary data of scanned mechanical engineering drawings and converts it into a DXF computer-formatted file (Capellades 1999). The agents that interact in the process of extracting visual information from the bitmap image do not necessarily represent image data and numerical measurements in the same way. These agents are independently developed, each one carrying a different task on the image, and reporting results to each other. The goal is that regardless of the internal algorithms and data structures that each of the agents implements, the agents will effectively communicate their results using all the same definitions of the terms involved. These definitions are explicitly given to the system in its ontology. In addition, the ontology will also document the commitments done in developing the system. A developer that wants to expand this system will need to review only its ontology in order to know what standards, terms, and assumptions are being used.

Given the tasks associated to MEDTooL, and given the intended purpose of its ontology, the main ontological requirements for this system were defined as follows:

a. Provide definitions and axioms for all the graphical elements present in images of mechanical engineering drawings.
b. Provide definitions for all the relations and functions that can hold between and for the elements defined in the previous step.
c. The set of definitions constructed in the previous two steps should be consistent, in the sense that no formal contradiction can be derived from the axioms or from sentences entailed by them.
d. Document the commitments made in the ontology so that expandability of the system will be properly carried out in subsequent steps.

To clarify these commitments, consider the definition of a line for MEDTooL:

> **Definition**: A `line` is a connected-component inside the two-dimensional space defined by the image with length equal to the number of points conforming it.

An agent reporting results about the lines present in the bitmap image will have to adhere to this definition, in spite that its internal representation of the line, or its treatment of the length measure, might be different from the one in the ontology. The definition above assumes that the concepts of `point`, `image`, `two-dimensional space`, `connected-component`, and `length`, have already been defined somewhere in the ontology.

In MEDTooL, a `point` is a pixel in the bitmap image and it is represented by its coordinates in the `two-dimensional space` defined by the borders of the image. The function `length` assigns to each `connected-component` in the image a numerical (integer) value. All these terms are defined within the field of image processing, and refer to particular instances of the concepts of point, line, and space that explicitly make reference to the nature of the input data of the system. If this technical ontology is made more general than this, it is possible to fall into useless definitions in the sense of their applicability to the system's tasks. In developing on ontology for MEDTooL, abstraction was delegated as less important than expressiveness.

Problems in the Incorporation of Existing Ontologies to MEDTooL's

Before starting the design of MEDTooL's ontology, a partial review of some of the ontologies made available by Stanford's KSL was done. Only ontologies from this site were considered since this is the only public browsable library available on-line. These ontologies were built using the Ontolingua editor, a very powerful tool for ontology development, designed to promote the sharing and use of ontologies (Farquhar *et al.* 1995).

From the database of ontologies available, the ones chosen for extended review were: simple-geometry, quantity-spaces, and component-assemblies. After a preliminary study in which the purpose and definitions given in each one of these ontologies were analyzed, only one was chosen as a clear candidate for reuse: component-assemblies. The others were discarded mainly because it was found that the concepts given in them were either not relevant or not suitable for MEDTooL. For example, the terms defined in simple-geometry are given for a three-dimensional space and MEDTooL operates entirely in a two-dimensional one.

Negotiation process

Component-assemblies, although well-structured and well-documented, presented several problems for its incorporation to MEDTooL. The most important one was how the intended meaning of certain terms differed from the corresponding meanings within the domain of image processing. For example, the definition of `connected-components` given in component-assemblies is as follows:

> **Definition**: `connected-components` is a binary relation between components. If `(connected-components A B)` then `A` and `B` must be components and neither can be a `subcomponent-of` the other. This relation is symmetric and irreflexive.

This definition provides a very abstract description of the relation. Transitivity is not allowed in the `subcomponent-of` relationship mentioned above, which is a strong commitment to which MEDTooL will not adhere. It is explained in the documentation that is up to the user to expand this definition appropriately by adding other parametric constraints to the definition of the relation. Since the ontology can only be augmented, reuse of this term was discarded. Instead, a new term for MEDTooL's ontology was defined: `connected-component`.

A `connected-component` in image processing is defined as an object, not a relation, as follows:

> **Definition**: A region R of an image is any subset of pixels inside the image. A region R is said to be `connected` if there is a path between any two pixels contained in R, where a path is defined by an adjacency relationship.

It was preferred to follow this formulation in the MEDTooL ontology by defining a `connected-component` as a component that has a path of pixels, the connection, associated with it describing the relation `connected` between its constituent parts. A close relative of `connected-components` in component-assemblies is `connection`:

> **Definition**: A `connection` is a `component` that represents a connection relationship between two other components. It is a reification of `connected-components`. For each pair of components in `connected-components` there exists at least one `connection` component.

This reification was done in order to facilitate the definition of different kinds of connections between pairs of components related by `connected-components`. The concept of `connection` is useful to characterize how two components are connected, therefore it was reused in MEDTooL's ontology. In particular, `connection` is used to describe the path of pixels adjoining the parts of a `connected-component`. The main differences between component-assemblies and MEDTooL's ontology are graphically shown in Figure 1.

The decision to create a new definition of `connected-component` was motivated to keep clarity, even though it meant sacrificing generality. Allowing `connected-component` to remain as a relation would

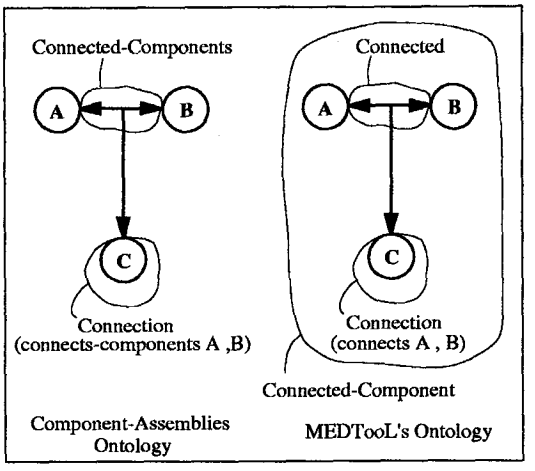

Figure 1: Differences between the definitions of `connected-components` and related terms given in the component-assemblies ontology and in the MEDTooL ontology

have had an undesirable impact on the way the algorithms and data structures in MEDTooL were going to be used.

This section of the paper exemplifies how difficult is to define key concepts inside an ontology with reusability as a constraint. The negotiation problem is not easy to solve, and more studies need to be done regarding methods for ontology design that properly handle this problem. However, it should be noted that a great deal was learned through the revision of component-assemblies and the use of the Ontolingua editor, from reuse-related issues, to ontology organization and documentation issues.

Evaluation process

In this work, verification of the consistency and completeness of the definitions and axioms in component-assemblies was not carried out. Reuse can be improved by providing evaluation tools on-line. The group at Stanford's KSL is currently working in that direction as reported in (Fikes & Farquhar 1997). To experiment with techniques for evaluating the ontology would have implied the ontology's translation and implementation into some other language for testing, and this was out of the scope of the goals of the MEDTooL project. An advantage of having used the traditional definitions of the terms in the image processing sense is that these have been tested for consistency numerous times through out the history of the field. As a result, evaluation was not a priority at that moment for the development of MEDTooL. However, it would be necessary to carry out such an evaluation process in the future as MEDTooL's ontology grows in size, and the possibility of contradictions appearing increases.

The reviewing process was time-consuming not only because it was necessary to analyze the ontologies that seemed suitable for reuse, but also the ontologies from which they were derived as well and the associated documentation. The length of this revision process is considered by the author a major problem for ontology reuse. In the present work not many ontologies were reviewed, but for complex projects needing the revision of large databases of ontologies, revision time will certainly be an issue.

Implemented Solutions

An Ontology Browser

As it is suggested by some of the problems described in the second section of this paper, (specifically problems 1, 4, and 5), the reusability of ontologies depends critically on the rapid visualization and complete assessment of the contents of an ontology. This can be achieved by the development of the proper ontology annotation and browsing tools. Recently, Fikes and Farquhar (1997) presented a vision of what tools will be necessary for ontology management in the near future. High in the hierarchy lie ontology browsers, which should be able to display inheritance information and do so efficiently even for very large ontologies. Visualization techniques might be of great use in this particular, allowing the graphical display of inherited links and other relations in a two-dimensional dynamic image. This would make the analysis of a particular ontology more efficient since humans can analyze, recognize, and recall graphical patterns better than text ones.

With these thoughts in mind, an ontology browser was added to MEDTooL's graphical user interface. The browser displays a hierarchical tree generated by a Lisp-parser that takes as input an ontology written in Lisp (in the kind of format that the Ontolingua translator gives as output). The nodes of the tree represent classes and the edges represent the inheritance relations. The browser is interactive, if a node is clicked with the mouse, the corresponding Lisp code is displayed in a separate window. The important thing is that the code can be studied without loosing view of the tree, which is an advantage based on the experience of the author. The interface is kept simple to allow the user to focus on what is important.

The Ontolingua server has the option of visualizing the hierarchy tree of an ontology, but when a class is selected the screen switches modes and the tree is not kept on it anymore. In MEDTooL, the tree and the code for different classes can be seen at the same time, each class in a separate window. In the opinion of the author, this facilitates the analysis of the ontology. The entire system is written in Java, thus it can run on different platforms. Current work on this system is guided towards the addition of editing and evaluation capabilities to the ontology browser.

The Reusability Index

As the number of public ontologies grows, it would be necessary to classify them according to some criteria. An index that can be easily calculated to measure reusability would be appropriate to make a first selection of what ontologies should be reviewed from a database for example. In this case, the author calculated a Reusability Index, defined as the number of reused lines from the code of an ontology divided by the total number of lines. Using the Ontology Browser described in the previous section, the user selects the classes he or she is planning to reuse by clicking on them in the tree while on revision mode (the Ontology menu contains selection for this mode). The program calculates the number of lines belonging to those classes and then calculates the index, which provides an estimate of the percentage of reused code. In the present work, the reusability index for component-assemblies was found to be 0.2362, meaning that 24% of the ontology was reused. The index was used in the MEDTooL project to document the effectivity of ontology reuse.

By itself the Reusability Index does not help much in selecting the ontologies. But if users of an ontology bank are required to calculate and give this number back when they evaluate candidates from the database, the index can be used to characterize and classify the potential reusability of ontologies. The average index of each ontology would be displayed, thus giving the users an idea of how good the ontology is according to the experience of previous users. The average index could also be analyzed in time to show the effects that changes in the ontology have had in its reusability.

Additional Comments

Two additional problems for ontology reuse are going to be added to the list of known problems in view of the experience gained during this work. The first one is related to the task-independence asssumption, and the second is related to the lack of formal, automated methods for ontology evaluation. After that, some comments regarding the advantages of ontology reuse are given.

Task-independence assumption

Since the ontology component-assemblies was designed with maximum-reusability in mind, the developers tried to make it the most general and abstract possible, minimizing ontological commitment. This was an impediment to reuse because some of the terms, in spite of being perfectly suited for the image processing domain, still needed editing to include the appropriate constraints. This process, although not much of an impediment in an academic research project dealing with a fairly simple ontology, takes too much time if long, complex ontologies are going to be developed. If the idea of reuse is to have off-the-shelf components, then task-independence compromises reusability. A way to improve on this is to incorporate sophisticated editing and evaluation tools in a distributed visual development environment that makes negotiation as fast and easy as possible. Generality and abstraction might not be the best policies for the development of ontologies, contrary to the current belief. It is documented in the literature that: "reusability across multiple tasks or methods ***can and should be systematically pursued*** even when modeling knowledge related to a single task or method: the more this reusability is pursued, the closer we get to the intrinsic, task-independent aspects of a given piece of reality (at least, in the commonsense perception of a human agent)" (Guarino 1997). Thus, it is often assumed that not only commonsense perception is the same among humans, but also that maximum abstraction automatically gives maximum reusability.

In regard to commonsense perception, it is been argued in cognitive science and philosophy that even though commonsense reality is one, commonsense perception is not. As Hilbert (1987) described it: "All that is necessary for the objectivity of a property is that objects have or fail to have that property independently of their interactions with perceiving subjects". It has never been stressed in any of the ontology-building methodologies to keep objectivity in this sense though. This is specially true in the capture process, where the identification of key concepts assumes that developers will somehow identify the same key concepts all together. As Smith (1995) explains: "Different persons on different occasions have different capacities to discriminate in this regard, and thus our perceptions are subject to differences of granularity, reflecting different levels of organization on the objects themselves". Negotiation procedures need to address these issues properly.

With respect to maximum abstraction giving maximum reusability, it has not been proved by the ontological community that this principle applies in all cases of ontology reuse. It is precisely from practical cases, from experience drawn from building and reusing ontologies, that it will be more clear in what occasions abstraction improves reusability and in what occasions it does not. Generality might not be as useful as specificity in the development of technical ontologies, as it was observed in the present work. More experimentation with ontology reuse might show that task-independence is more of an impediment than an advantage in most cases.

Lack of evaluation methods

There are several reported methods to compare and evaluate ontologies in the literature. For example, Noy *et al.* (1997), developed a framework for ontology comparison "as a first step in defining a set of questions and possible answers that would allow an ontology to introduce itself to a potential user". Their evaluation process was designed more as a guideline for the identification and comparison of potential ontologies for reuse from a library than as a formal method for determining the validity and completeness of a given ontology. The process needs to be performed manually, following the questions given in their framework, and then answering them by studying the ontologies under scrutiny. An-

other procedure for evaluating ontologies has been devised by Gómez-Pérez (1995). In her work, ontologies are manually tested following rules derived in part from mathematical logic. This method, as the one described before, needs to be carried out manually. In addition, it is not suitable for automation or systematization.

Both frameworks were helpful to select and analyze ontologies of interest to the MEDTooL system. However, these methodologies cannot test the internal consistency of the ontologies, and they cannot be used to quickly classify ontologies for reuse. In summary, it is necessary to devise formal evaluation methods, containing consistency tests, that can be integrated to a visual environment to facilitate the reuse and development of ontologies.

Reuse vs. From-scratch

In the present work, the decision was made of building an ontology from scratch incorporating modified versions of some of the terms found in component-assemblies, as explained in previous sections. If the entire component-assemblies ontology would had been incorporated by making the necessary augmentation, simplicity and clarity of the documentation of MEDTooL would have suffered. In view of the ontological requirements of MEDTooL, a basically-built-from-scratch ontology was implemented instead. Development time was longer than expected since the time needed for the evaluation of existing ontologies was underestimated at the beginning of the project.

In the case of the work reported in (Uschold *et al.* 1998), the perception was that reusing was cost-effective in their experiment, as in the present work. They estimated that building the ontology from scratch would have been a more lengthy process than evaluating, modifying and translating an existing one. In summary, more experimentation with ontology reuse needs to be done to reach a conclusion regarding the cost-effectiveness of the reuse effort.

Conclusion

During the development of an ontology for a multiagent system called MEDTooL, it was observed that the main problems associated with ontology reuse were: (1) the analysis and evaluation of existing ontologies is too costly in terms of development time, even when using current on-line browsing and editing tools; and (2) the lack of methods for quickly assessing the reusability and the consistency of existing ontologies make the decision process associated with reuse difficult. To help ease these problems, a browsing tool that uses common visualization algorithms was implemented, and an index to measure reusability was introduced. Although it was preferred at the end to keep clarity and simplicity along MEDTooL's ontology rather than fully integrate different existing ontologies, much was learned about reuse and ontology design by studying existing ontologies. In conclusion, it is necessary to experiment more with ontology reuse and to objectively measure its cost-effectiveness, so that convergence to a conclusion regarding the feasibility of ontology sharing can be reached.

References

Capellades, M. A. 1999. *A Multi-Agent Architecture for the Interpretation of Mechanical Engineering Drawings.* Ph.D. Dissertation, The Pennsylvania State University, University Park, Pennsylvania. (in preparation).

Farquhar, A.; Fikes, R.; Pratt, W.; and Rice, J. 1995. Collaborative ontology construction for information integration. Technical Report KSL-95-63, Stanford University Knowledge Systems Laboratory.

Fikes, R., and Farquhar, A. 1997. Distributed repositories of highly expressive reusable ontologies. Technical Report KSL-97-02, Knowledge Systems Laboratory, Stanford University.

Gómez-Pérez, A. 1995. Some ideas to evaluate ontologies. In *Proceedings of the 11th IEEE Conference on AI Applications*, 299–305.

Guarino, N. 1997. Understanding, building and using ontologies. *International Journal of Human-Computer Studies* (46):293–310.

Hilbert, D. R. 1987. Color and color perception: A study in anthropocentric realism. In *CSLI Lecture Note Series.*

Noy, N. F., and Hafner, C. D. 1997. The state of the art in ontology design. *AI Magazine* 53–74.

O'Leary, D. 1997. Impediments in the use of explicit ontologies for kbs development. *International Journal of Human-Computer Studies* 46:327–337.

Patil, R. S.; Fikes, R. E.; Patel-Schneider, P. F.; McKay, D.; Finin, T.; Gruber, T.; and Neches, R. 1992. The DARPA knowledge sharing effort: Progress report. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 777–787.

Smith, B. 1995. Formal ontology, common sense and cognitive science. *International Journal of Human-Computer Studies* (43):641–667.

Uschold, M., and Gruninger, M. 1996. Ontologies: Principles, methods and applications. *The Knowledge Engineering Review* 11(2):93–136.

Uschold, M.; Clark, P.; Healy, M.; Williamson, K.; and Woods, S. 1998. An experiment in ontology reuse. In *Proc. of the 11th Workshop on Knowledge Acquisition, Modeling and Management.*