Ontologies, Reuse and Domain Analysis

Dean Allemang

TopQuadrant, Inc.,
dallemang@topquadrant.com,
WWW home page: http://www.topquadrant.com/

Abstract. Ontology engineering is rapidly becoming an important discipline for the development of the semantic web. For many ontology applications, standard systems analysis methodology is an appropriate way to approach the systematic design of the ontology. For other applications, however, a domain analysis approach is more suited. This paper examines the sorts of applications that could benefit from a domain analysis approach, and provides suggestions about how to incorporate domain analysis methods into ontology engineering discipline.

1 Introduction

The World Wide Web (WWW) has opened the door to information sharing possiblities that were unheard of even a few years before its inception. Along with these new opportunities has come new ideas in how to integrate data of all sorts - at an enterprise level, as well as at the global level of the WWW. The availability of truly colossal amounts of data has fueled a push toward utilization of search and cataloguing technology that has made great headway in organizing all this information. However, it has also made clear the limitations of string matching, syntactic means for searching through data. From this awareness, technologies that now go by the name of "Ontologies" promise a new paradigm of semantic information sharing. The grandest vision of this is the Semantic Web, in which the enormous body of data available on the web will be organized in a way that allows it to be indexed by its meaning, not just by its form.

Along the way to this ideal, many applications of ontologies in more modest settings have been proposed, and many of them have been deployed. However, there is no single unifying value proposition for ontology applications. Here are a few distinct value propositions that are common at this time:

– *Organize portals.* An ontology can be used by content authors to mark up their content in a way that will make it easier to find materials in a semantic portal.
– *Semantic integration.* Multiple legacy data sources encode information according to different schemas, DTDs, etc. Ontologies provide a means for representing the meaning of the information so that integration can proceed automatically.

- *Collaborative knowledge gathering.* Several authors use a single ontology to specify their knowledge in a uniform way. The result is a meaningful collection of semantically tagged knowledge.
- *Conceptual search.* Keyword searches based on incidence frequency, bayesian statistics and information theory are useful for indexing large, unstructured information spaces. More focused, structured corpuses can be searched more accurately by matching concepts found in the documents.

For many applications, the ontology plays a role in a larger system that can be designed much like any other complex information system, beginning with a context of use and requirements analysis. However, for many applications in the categories above, the ontology plays the role of mediator for reusable knowledge assets. In such a situation, the value added by the ontology is only as good as its ability to organize materials that have not yet been encountered. For example, in many enterprise integration contexts, the value promised by the use of an ontology is that new data sources can be integrated into a data (or application) federation without having to make major changes to the federation structure. In collaboration contexts, and even in many portal contexts, the ontology must be able to help content authors create and/or index new information in such a way that users in a variety of work contexts can find it and use it in a cost-effective way. In all these cases, there is a requirement on the design of the ontology that it be able, in some sense, to anticipate the data organizational needs of contexts that have not yet been determined.

Domain Analysis for Software Reuse [1] is the engineering discipline of designing repositories of reusable software. In such a setting, reuable components are constructed in such a way that they will be useful in new contexts that have not yet been examined or specified. These assets must be organized in such a way that designers who are creating new systems can find reusable assets that are applicable to the new work context, and it must be possible to adapt these assets to the new context with less effort than it would have taken to have built the desired functionality without the reusable components.

The centerpiece of such a reuse project is the Domain Model, which plays a role very similar to that played by the ontology in the work settings outlined above. While it is certainly possible in many cases to use conventional systems design methodologies to build ontologies, many techiques that have been developed in Domain Analysis can also be applied. In this paper, we will investigate how some of the central tenets of domain modeling can be applied to ontology engineering.

There are a number of well-known problems in ontology design that become apparent to anyone who tries to build an ontology of any complexity, or even to anyone who examines an ontology in some detail. These problems, outlined clearly in [2] and studied more formally in [3], and include such issues as the decision to represent some entity as a concept or as an individual, the decision to represent a class of individuals as a concept, or simply as a restriction on some property of those individuals. While these decisions appear to be quite detailed and technical, they can make an important difference in the comprehensibility and usability of an ontology.

A very powerful methodological tool for managing these sorts of issues is the idea of a set of *competency questions* for an ontology. This is a set of questions for which the ontology must provide the answers. This method treats the low-level, detailed, technical issues of ontology design at a more abstract, domain accesible level.

While competency questions provide a powerful way to develop and express the scope and requirements of a model, they do not in themselves provide an answer for how an ontology can anticipate the needs of future work contexts.

Because of the power of competency questions to focus ontology engineering, we position the contribution of domain modeling in the phases of project development leading up to competency questions. That is, the outcome of the domain modeling analyses that we recommend here will be a more informed way to create a set of competency questions. In short, the competency questions themselves are regarded as an engineered workproduct, rather than as a starting point. This view is not at all at odds with other views of competency questions; the contribution of this paper is the application of domain analysis insights into the design process for competency questions.

We borrow heavily from the methodological foundation and process of Organization Domain Modeling (ODM) [4, 5] for our characterization of domaion modeling methodology. ODM is an elaborate and fully-documented methodology (including process diagrams and workproduct templates) supporting all phases of domain engineering for software reuse (including engineering assets for reuse and asset deployment). Among the basic tenets of ODM, for the purposes of this investigation we will concentrate on the following:

- In order to support reuse, a domain model (ontology) must reflect the needs of a particular organization,
- The stakeholders in that organization will have some stakes in common, and others that vary,
- The primary function of a domain model (ontology) is to represent and manage the commonality and variability among stakeholder interests.

We apply these basic ideas to the design of an effective set of competency questions to drive ontology design.

2 Stakeholder Analysis

The basic problem in ontology design that Domain Analysis addresses is how to design a domain model (ontology) in such a way that it can anticipate the needs of users in novel work contexts. From a naïve point of view, this cannot be done at all; if a requirement can come from absolutely any work context at all, then there is no way to anticipate anything about it. In the context of simply using an ontology to mark up content, consider the example of an ontology designed for the description of mergers and acqusitions as given in [6]. If it were to be used, say, to annotate a report on the chemical properties of a newly discovered compound, then clearly the ontology would provide no guidance. An ontology must have a specified scope of applicability.

From a less naïve point of view, there are some clear ways around problems of this sort. [6] recommends a checklist of ontology scoping and requirements considerations, many of which include noting the target application(s) of the ontology. But in the absense of information about these target applications, how can one proceed?

Domain Analysis in general, and ODM in particular, have an answer to this question. And that answer begins with "Stakeholder Analysis". It might be impossible to know the applications of an ontology in advance, but it is not impossible to know the stakeholders.

Conventional wisdom holds that a reasonable approach to getting stakeholders to agree on a set of terminology is based on getting all of them together; reasonable people can work out what the important concepts are, then agree to some standard set of names. Then each stakeholder can map the standard names into the names they use in their own practice. In this way, each stakeholder group can continue to use familiar terminology, but a mapping can still be made between stakeholder groups.

As is the case for most conventional wisdom, this approach has its advantages as well as its pitfalls.

Consider the simple question that might be of interest to NASA; "what is a space shuttle?". Naively, the question could be answered extensionally by listing all of them; Atlantis, Columbia, Challenger, Discovery, Endeavor, and Enterprise. Now let's look at some possible stakeholders, and how they might view the situation.

- An historian interested in the names of the shuttles might consider all of these (as well as "Constitution", the original name of the Enterprise) to be shuttles
- An aerodynamics engineer, who is studying the lift properties of the fuselage in the atmosphere would agree, since the Enterprise was designed for tests of aerodynamics of the fuselage.
- However, an archivist of the design documents of the ships would not find it very useful to consider the Enterprise to be a space shuttle, since it was prototype that was never designed to actually fly. As such, it is missing all the relevant drawings and design rationales that are important to a shuttle project.
- Finally, the mission planner whose job it is to determine which shuttles are used on what missions, planning their repair and down time, would unfortunately no longer find it useful to consider Challenger or Columbia in the line-up of shuttles.

In answer to the question, "what is a space shuttle?", who is right?

Of course, all of them are right. Each stakeholder has a different view, not because each one is holding stubbornly to their own traditional viewpoint, but because each one has a genuine need in their work context for a particular conceptualization. In this sense, all these stakeholders can identify their requirements for the concept "Space Shuttle", and should be able to agree on these different definitions.

In real life, it isn't usually that simple. It is often, perhaps not surprisingly, difficult for a practitioner who has spent a career using terms in a particular way to take an objective viewpoint when it comes to discussing the boundaries between neighboring concepts. [1] Fortunately, Ontology representations are able to play a mitigating role in this difficulty. Since an ontology allows a modeler to represent not only several concepts, but the relationships between the, it provides grounding for an objective perspective on the domain.

In the following simple example, we will show various options of how to "re-factor" a concept structure. The point of this example is not to complete an optimal factoring in the sense used in object modelng, but simply to demonstrate that concept factoring decisions can be based directly on stakeholder issues, rather than system requirements, as is usually the case in object modeling. This is necessary in a domain analysis setting, and in many ontology engineering settings, because the object decomposition reflects issues about systems that have not yet been specified.

In this simple example, the various notions of "Space Shuttle" can be related to one another as follows:

- **Space Shuttle**
  - **In-Service Shuttle**
    - *Atlantis*
    - *Endeavor*
    - *Discovery*
  - **Out-of-Service Shuttle**
    - *Columbia*
    - *Challenger*
    - **Prototype Shuttle**
      - *Enterprise*

The related concepts "Prototype Shuttle", "Out-of-Service Shuttle", "In-Service Shuttle" and "Space Shuttle" are represented in a class/subclass hierarchy, showing the relationships between these concepts.

3 Commonality and Variability

An ontology description includes more semantics about class/subclass than is being exploited by simply using the ontology as a neutral worksheet to encourage

[1] Another important consideration in any stakeholder analysis is the possibility that some stakeholders might have motivations to block certain communications. While this is an important aspect, it is not the topic of this paper. A more complete analysis can be found in [7]

objectivity among stakeholders. Features and properties assigned to superclasses are inherited by subclasses, and by their instances. Unlike system analysis to a single specification, domain analysis for reuse involves modeling the commonality and variability among various stakeholders. That is, the model should be able to support a single point of access for items relating to the *common* interests of multiple stakeholders, while allowing them to represent the *variations* in their stakeholder contexts. In this simple example, the inheritance mechanism of the ontology allows them to do just that.

The commonalities among all the stakeholders interests in spacce shuttles (e.g., the shuttles have names, a fuselage, total weight, design documentation, etc.) can be represented at the top level. The historian and the aerodynamics engineer use this concept to mean "Space Shuttle". For the mission scheduler, information about shuttle availability, maintenance state, etc. can be represented at the concept "In-Service Shuttle".

One stakeholder has been left out - the design archivist, who is interested in managing all the drawings and design documents for the space shuttles (and whose job becomes particularly important when investigating accidents). For this stakeholder, the Enterprise, as a prototype, really doesn't count as a space shuttle. But there is no concept in the ontology that does correspond to this notion of a shuttle.

This suggests a re-organization of the ontology, into something like this:

- **Space Shuttle**—
  - **Prototype**
    - ***Enterprise***
  - **Full-service Shuttle**
    - ***Columbia***
    - ***Challenger***
    - **In-Service Shuttle**
      - ***Atlantis***
      - ***Endeavor***
      - ***Discovery***

3.1 Sources of Variability

So far, we have examined the commonalities and variabilities among stakeholders, as if each stakeholder was a consistent monolith of interests. Again, in the real world this clearly is not so. There are a number of factors that contribute to variability within a single stakeholder group, or even within a single individual. Some variability is inherent in the practice of any knowledge worker. In the example above, we have (somewhat artificially) separated out the aerodynamics engineer from the design archivist. More realistically, part of the experise of the design archivist would include knowing that for purposes of aerodynamic design, the Enterprise should be included in a design archive, while for purposes of understanding the electrical system, engine dynamics, etc., only the other shuttles' records are relevant. Fortunately, for the most part, this sort of variation can be handled in much the same way as variation among stakeholders, in effect viewing a single real-world stakeholder as a collection of ”virtual stakeholders”.

However, there are other sources of variation within stakeholder groups that can pose more troublesome problems to the modeler. These issues have to do with work culture and motivations of the stakeholders.

Disagreements. The simplest source of variation in expert stakeholder groups stems from professinal disagreements. Especially in expertise-intensive areas, individual experts, each of which may be recognized in the field, can have differences of opinion. Often these have to do with recognition of situations (e.g., does the patient have disease A or disease B?), but can also have to do with the organization of categories, assignment of properties, etc. Ontology modeling technology can play a role in resolving such conflicts (in a similar fashion as above, by providing an objective whiteboard for all views), but especially in cases where the experts have large emotional investments in their viewpoints, this problem may require more careful planning.

Party-line reports. In many practices, there is a difference between accepted knowledge and de-facto knowledge used in practice [8]. This adds another dimension of variability into the domain. While ontology modeling technology provides some technical assistance to this problem, careful management of the knowledge acquisition process is needed to gather the information necessary for modeling de-facto knowledge.

Managing ontology development in the face of such stakeholder issues and sources of variability requires a planning process that is beyond the scope of this paper. Details of a knowledge planning method for addressing these issues of variation, other stakeholder issues, as well as anecdotes of specific situations in which they were needed, can be found in [7].

4 Competency questions

Controlling the scope of a model is a common concern for both ontology modeling and domain modeling. In the case of ontology modeling, a common technique for recording and organizing the intended scope of a model is the use of Competency Questions; a set of questions the the model should be able to answer. [9, 10, 2] As a modeler is making a decision about whether to express some variability as a subclass or as a restriction on a property, or whether some entity should be an instance or a concept, or even whether a particular level of detailed variability should be expressed at all, each of these questions can be held up to a gold standard of the competency questions. This method provides a simple way to express the boundaries of a model, and to check whether a particular modeling or knowledge acqusition activity is in scope for the effort, or not.

In a normal systems analysis setting, the competency questions come directly from the requirements of the system that is to be built. In some sense, the competency questions are a reflection of the part of the system requirements that will be satisfied by the ontology. In a reuse setting, the situation is more complex, in that systems analysis must be replaced by domain analysis. The requirements are no longer those of a single, planned system, but rather requirements for a set of systems, some of which have not yet been built or designed. This means that the competency questions, rather than reflecting the system requirements, must reflect the commonality and variability requirements among the stakeholders.

Competency questions for systems engineering are often given as specific examples of kinds of questions; for example, in [2] we see competency questions like "Is Bordeaux red or white wine?" This question stands in as an example of several similar questions, e.g., "is Bourgogne a red or white wine?", "is Chianti a red or white wine?". When composing competency questions based around stakeholder commonality and variability, the appearance of literal domain concepts often has specific importance, in that it discriminates between one stakeholder group and another. In the space shuttle example above, the competency questions "Where can I find design documents about the aerodynamics of a shuttle fuselage?" and "Where can I find design documents about the power train of a shuttle engine?" are quite specifically directed at specifying the commonality and variability between two stakeholder groups, or variant interests of a particualar stakeholder group. For this reason, we recommend annotating each competency question with information about the intended variability implicit in the question. So, for instance, the question "Is *Bordeaux* a red or white wine?" we indicate that Bordeaux stands in for any wine category by setting it in italics; similarly, we would say "Where can I find *design documents* about the aerodynamics of the shuttle fuselage?". The elements in brackets are known to vary, while the elements not in brackets (aerodynamics, red and white wine) do not. This is not a foolproof method; it might be important to know the precise range of variation of these items. Such distinctions are beyond the capability of informal competency questions, and require a more formal approach, such as that suggested in [9].

With the exception of the treatment of variation in the competency questions, once they have been determined, their use can proceed in familiar ways, as outlined in [2] and [9]. Decisions about when to make a new class or a new property, its scope, etc. can be checked against the competency questions as a check of appropriateness.

5 Conclusion

Ontology engineering is a form of modeling resulting in a specification of a conceptualization [11]; what the conceptualization is about depends on the context of use of the ontology. Domain modeling for software reuse is a form of modeling resulting in a framework that can be used to store reusable assets of some sort [5]. These efforts overlap in the case where an ontology is used as a way to organize information from multiple sources, for multiple uses. This is the case when an ontology is used as the basis for a collaborative portal, in which content providers make knowledge assets available for the use of a wider knowledge community. It is also the case when an ontology is used as part of an architecture for mediating or integrating among information sources (data bases, document collections, etc.) with differing backgrounds, but considerable overlap in coverage of content. The intution behind using an ontology to perform integration of this sort is that new data sources can be integrated into the shared federation by specifying how that data source is described by the intermediating ontology(ies). In order for this process to be realistically dynamic, it must be possible to perform such a mapping for new data sources that might not have been encountered during the modeling process. This means that the ontology must be able to cover cases that have not seen before. This is exactly the problem addressed by the process of domain analysis for software reuse.

This paper has examined how an organizational approach to domain modeling (as outlined in the ODM methodology) can be applied to ontology engineering in such cases. The key insight of ODM, that a domain model must reflect the commonality and variability among a set of stakeholders, can be applied as well to ontology engineering, utilizing the familiar method of competency questions to organize and connect the stakeholder analysis to the ontology modeling process.

References

1. Arango, G., Prieto-Diaz, R. In: Domain Analysis Concepts and Research Directions. IEEE Press (1991)
2. Noy, N., McGuinness, D.: Ontology 101: A guide to creating your first ontology. Technical report, Stanford University (2000)
3. Guarino, N., Welty, C.: Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. In: Proceedings of ECAI-2000. (2000)
4. Simos, M.A.: Organization domain modeling (odm): Formalizing the core domain modeling life cycle. In: Proceedings of the 1995 Symposium on Software reusability, Seattle, Washington (1995)
5. Simos, M., Creps, D., Klingler, C., Levine, L., Allemang, D.: Organization domain modeling (odm) guidebook (1996)
6. Staab, S., Schnurr, H., Studer, R., Sure, Y.: Knowledge processes and ontologies (2001)
7. Simos, M., Allemang, D., Levine, L., Klingler, C., Creps, D., Hammons, C., Mantock, L.: Canvas knowledge acquisition guidebook (1996)
8. Welbank, M.: British telecom report on knowledge acquisition. Technical report, British Telecom (1983)
9. Gruninger, M., Fox, M.S.: Methodology for the design and evaluation of ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal (1995)
10. Gruninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway (1994)
11. Gruber, T.R.: A translation approach to portable ontologies. Knowledge Acquisition **5** (1993) 199–220