Building Ontologies: Towards a Unified Methodology

Mike Uschold
AIAI-TR-197
September 1996

This paper appears in the proceedings of Expert Systems '96, the 16th Annual Conference
of the British Computer Society Specialist Group on Expert Systems,
held in Cambridge, UK on 16-18 December 1996

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

Abstract

The use and importance of ontologies is becoming more widespread, however building ontologies is largely a black art. The aim of this paper is to identify and characterise what we currently know and to move towards the longer term goal of developing a comprehensive unified methodology.

We first identify dimensions for characterising ontologies, to be used as a basis for noting which techniques and guidelines for building ontologies apply in different circumstances. We then give an overview of the current state of the art, noting that most work addresses just a small part of the life cycle. The very few more complete methods are limited to case studies involving single ontologies and they are hard to compare. In the main part of this paper, we examine two such methods and give a framework for comparing and unifying them. We emphasise that different approaches are required for difference circumstances, and give some guidelines for when to use which techniques. We conclude by considering how to further advance our understanding of building ontologies.

Keywords Ontological engineering, ontology, conceptual modelling, methodology

1 Introduction

There is a growing body of work in the area of how to build ontologies, but to date, it is limited and falls well short of a comprehensive methodology. For example Gruber [6] reported a set of overall guidelines for building ontologies; Gómez-Pérez [4] has proposed detailed guidelines and provided automated support for evaluating ontologies; Skuce [17] discusses how to reach agreement on shared ontologies; and Uschold [20] has reported a case study describing experiences of converting an informal specification of an ontology into the formal language, Ontolingua [5].

More complete methodologies have been described, where some attempt is made to identify and describe a set of stages in the ontology development process [8, 16, 21, 23], however these are limited to case studies of the development of a single ontology, or limited to a particular project.

A long range goal is to put this all together into a coherent framework which might be in the form of a handbook, which would provide useful guidance for anyone wishing to build an ontology. Such a handbook should clearly characterise the dimensions of variation for ontologies and give guidelines for how to build any given ontology, matching the particular circumstances with appropriate methods.

The main barrier to the production of such a coherent unified framework embracing all of these techniques and methods for building ontologies is that *there is no clear indication of how general the individual techniques and methods reported to date are*. Consequently, in a given set of circumstances, there are no guidelines for deciding what techniques and methods are likely to apply.

We make no attempt to take into account *all* techniques and methods reported to date, but instead concentrate on two independently developed methodologies used in the domain of enterprise modelling. Gruninger and Fox [8], address formal aspects of ontology development in describing how the TOVE ontology was created. Uschold and King [21] emphasise informal aspects in their description of how the Enterprise Ontology was developed. Although more complete discussions of each methodology were subsequently given [23], until now, no attempt has been made to compare them.

To address these barriers we proceed by

1. generalising and merging the independently developed TOVE and Enterprise methodologies;
2. identifying how other techniques and guidelines presented in the literature (*e.g.* [6, 4]) may be merged into the various steps in the methodology.

We intend for this to be a significant step in the direction of having a unified methodology for building ontologies. We anticipate that future work will concentrate on incorporating a wider range of methods and techniques, paying special attention to characterising in what situations they do and do not apply. The methods developed in the related fields of data modelling and software engineering should also be incorporated.

Background

Why ontologies? — In his technical keynote presentation for the ES94 conference [18], Sleeman noted the general area of knowledge base reuse, in conjunction with the sub-area of ontologies as one of four 'hot issues' in knowledge base system building. While there is a considerable amount of work going on, as yet, little has been reported to this audience.

Tools and techniques for development and use of ontologies are developing rapidly; *e.g.* there have been a series of workshops in recent years at the major AI conferences. As yet there is no comprehensive *review* of this emerging field, which summarises the key issues and contributions from a research perspective. However, a thorough *introduction* to the field may be found in [23].

Related Efforts Much of our work concerning ontologies was undertaken as part of the Enterprise Project [12]. This is the UK government's major initiative to promote the use of knowledge-based systems in enterprise modelling. In the project we have devoted much of our efforts to the derivation of the Enterprise Ontology, which is intended to provide a common view of the information and activities involved in enterprise modelling (see appendix A for further details).

In addition to Enterprise, other projects that have advanced our understanding of the development and use of ontologies include: 1) playing a leading role in the EuroKnowledge Project (ESPRIT P9806) which aims to facilitate the reuse and exchange of knowledge by establishing recommendations for "knowledge level" modelling [24]; 2) development of planning and process modeling ontologies. This includes Knowledge Representation Specification Language (KRSL) [15] and Process Interchange Format (PIF) [14].

2 What is an Ontology?

Before we proceed with describing how to build ontologies, we first clarify what we mean by this term. We pay special attention to the different kinds of ontologies that exist and purposes that they serve because this will impact heavily on how to build one.

2.1 Definition

There is no universally agreed meaning for the term, 'ontology', (see [10] for a competent analysis of this situation). Below we give a definition that we believe best conforms with common usage of the term.

Conceptualisation First we introduce the important idea of a *conceptualisation*. Broadly, a conceptualisation is a world view; it corresponds to a way of thinking about some domain. It can be seen as "a set of informal rules that constrain the structure of a piece of reality

[9]". It is typically conceived and/or expressed as a set of concepts (*e.g.* entities, attributes, processes), their definitions and their inter-relationships.

A conceptualisation may be implicit, *e.g.* existing only in someone's head, or embodied in a piece of software. For example, an accounting package presumes some world view encompassing such concepts as invoice, and a department in an organisation. A conceptualisation that is explicit, is usually called an *ontology*.

Ontology We adopt the following as our working definition of an ontology:

> an *explicit* account or representation of some part of a conceptualisation [adapted from [10]].

An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms and some specification of their meaning. An ontology is virtually always the manifestation of a shared understanding of a domain that is agreed between a number of parties. Such agreement facilitates accurate and effective communication of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing.

2.2 Kinds of Ontologies

As noted above, the way to build ontologies depends very heavily on the particular circumstances under which an ontology is desired. Below are three key dimensions along which ontologies vary; implicitly these give rise to many 'kinds' of ontologies. We will use these dimensions when describing an initial proposal for a unified methodology for building ontologies. The main dimensions are:

Formality: the degree of formality by which a vocabulary is created and meaning is specified;

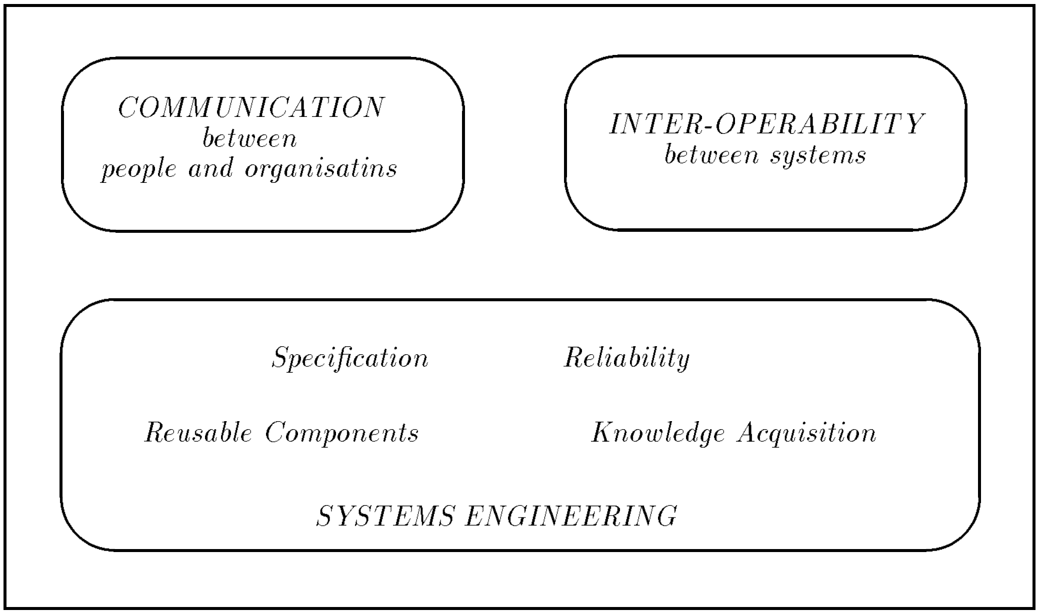
Purpose: the intended use of the ontology;

Subject Matter: the nature of the subject matter that the ontology is characterising.

2.2.1 Formality

Four somewhat arbitrary points along what might be thought of as a formality continuum are:

- highly informal: expressed loosely in natural language
  e.g. many glossaries fit into this category;
- structured informal: expressed in a restricted and structured form of natural language, greatly increasing clarity by reducing ambiguity,
  e.g. the text version of the 'Enterprise Ontology' [22];



We identify three main categories of uses for ontologies. Within each, other distinctions may be important, such as the nature of the software, who the intended users are, and how general the domain is.

Figure 1: Uses for Ontologies

- semi-formal: expressed in an artificial formally defined language,
  e.g. the Ontolingua version of the Enterprise Ontology[1];
- rigorously formal: meticulously defined terms with formal semantics, theorems and proofs of such properties as soundness and completeness.
  e.g. TOVE [7].

The best method for building an ontology will depend on the degree of formality required, which in turn depends a great deal on the intended *purpose* of the ontology.

2.2.2 Purpose

The literature is currently rich with descriptions of ontologies and their intended purposes. At a high level, most seem to be intended for some manner of reuse. Some of these purposes are implicit in the various interpretations of the word 'ontology' that are commonly found in the literature, as noted in [10].

[1] Available from "http://www.aiai.ed.ac.uk/ entprise/enterprise/ontology.html"

Other factors include the nature of the software with which the ontology will be used, whether it is intended to be shared within a small group and reused within that context for a variety of applications, or whether it is intended to be reused by a larger community. Some view their ontologies mainly as a means to structure a knowledge base; others conceive an ontology to be used as part of a knowledge base, *e.g.* by loading it in as a set of sentences which will be added to as appropriate; still others view their ontology as an application-specific inter-lingua (e.g. ATOS [13]).

Based on these observations, we identify three main categories of uses for ontologies (see figure 1); for further details and examples see [23]).

Communication between people. Here, an unambiguous but informal ontology may be sufficient.

Inter-Operability among systems achieved by translating between different modelling methods, paradigms, languages and software tools; here, the ontology is used as an interchange format.

Systems Engineering Benefits: In particular,

- *Re-Usability:* the ontology is the basis for a formal encoding of the important entities, attributes, processes and their inter-relationships in the domain of interest. This formal representation may be (or become so by automatic translation) a reusable and/or shared component in a software system.
- *Knowledge Acquisition:* speed and reliability may be increased by using an existing ontology as the starting point and basis for guiding knowledge acquisition when building knowledge-based systems [1].
- *Reliability:* A formal representation also makes possible the automation of consistency checking resulting in more reliable software.
- *Specification:* the ontology can assist the process of identifying requirements and defining a specification for an IT system (knowledge based, or otherwise).

Genericity Related to purpose, is the notion of *genericity*, which is the extent to which an ontology can or is intended to be reused in a range of different situations. Very generic ontologies (*e.g.* [2, 16, 19]) are sometimes referred to as upper-level models and are used for organising substantial portions of human knowledge – *e.g.* for natural understanding. Less generic ontologies for particular applications are sometimes referred to as 'application ontologies' [25]. The problem solving framework described in [1] uses genericity an explicit principle for organising and using ontologies.

2.2.3 Subject Matter

The subject matter that an ontology characterises can be anything at all. Three widely accepted categories are:

1. subjects such as medicine, geology, or finance, considered separately from the problems or tasks that may arise relevant to the subject;
2. the subject matter of problem solving,
3. the subject matter of knowledge representation languages.

An ontology in the first category is frequently called a *domain* ontology; an ontology for the second is usually called a *task*, *method*, or *problem solving* ontology. The terms *representation ontology* or *meta-ontology* are used to refer to ontologies in the third category.

This is by no means intended to be a complete characterisation of how subject matter may differ; in particular, the first category is very fuzzy. Many sub-dimensions are possible such as uncertainty, or imprecision in the domain. In what follows, we are mainly concerned with subject matter in the first category.

2.2.4 Example

The terms and definitions comprising the informal version of the Enterprise Ontology [22] may be characterised as follows:

- Level of formality: structured informal
- Purpose: to facilitate communication between members of the project by giving a consistent terminology; to document and specify a subsequent formal encoding to facilitate inter-operation of enterprise modelling tools.
- Subject matter: business enterprises (fairly generic)

.

3 The Limitations of Current Methodologies

Having presented a way to characterise a wide range of ontologies, we now turn our attention to how to build them. We pay particular attention to how the method varies depending on the particular kind of ontology one is building.

Above we note that there are a range of methods and techniques for building ontologies reported in the literature. It is too ambitious at this point to attempt to merge all of this into a single all-encompassing methodology. Instead, we explore two methodologies that have been described previously in some detail and show how they may be merged. In the future, other methods may be considered and where possible, incorporated into a more general framework.

We will consider the methodologies for building the Enterprise Ontology and the TOVE Ontology; these were originally described in [21] and [8]. Enhanced and refined versions of these methodologies were presented in a joint work by both authors [23] as part of a comprehensive introduction to the emerging ontology field. This is a good starting point for furthering our understanding of building ontologies because there is sufficient similarity in the circumstances for each (*e.g.* both are in the enterprise modelling domain), yet there are also sufficient differences so that merging them is both useful and non-trivial.

It is useful because, as they were developed and reported independently, it is not immediately obvious what their similarities and differences are, where they overlap and where there are conflicts. Furthermore, given the similarity of the two projects, it is not clear which methodological differences are incidental, and which ones naturally arise from important differences in the context and goals of the respective projects. Thus, someone who wished to build their own ontology, in a similar domain would not know which (if either) methodology to use, or how to choose individual techniques from each method and apply them to their problem.

Comparisons & Critique The unified methodology we describe in the next section is aimed at the imaginary 'someone', just mentioned. Before we proceed, we first give a brief comparison of the two approaches and note some of the specific problems that such a person would face.

The two approaches both explicitly emphasise separating out the informal from the formal stages, though in very different ways. Both seem to produce the same output from what is a scoping activity (i.e a set of informal terms). However, one uses brainstorming and a requirements document during the scoping phase, the other discusses neither of these activities and has no explicit scoping phase. One uses competency questions (for scoping and evaluation), the other does not. One has an explicit evaluation phase, the other does not. One takes an essentially formal view, the other essentially informal. Both produce a formally encoded ontology at the end, but only one produces a separate informal ontology during an intermediate stage. Each have various guidelines that may or may not apply in the others' circumstances.

Overall, it is not clear why any of these differences exist, nor whether or how much it matters.

4 A Unified Approach

In this section, we turn our attention to the process by which a vocabulary is identified and definitions produced. The goal is to enable future ontology builders to better understand how to choose the most appropriate techniques for their particular set of circumstances. In particular, we:

- identify the steps and techniques that are of general applicability;
- identify the circumstances in which the non-general steps and techniques apply;

- attempt to put it all together in a coherent framework.

The framework we present is illustrated in figure 2. It is derived from and directly compatible with both the TOVE and Enterprise methodologies. Our aim is that it effectively enhances both and broadens the range of circumstances for which there is explicit methodological guidance for building ontologies. We address each main phase in turn.

4.1 Purpose

Ideally, an ontology builder should first have a clear idea of why the ontology is wanted, what it will be used for and possible mechanisms for use. In practice, these things are not always easily come by. Ways to proceed include:

- identify and characterise the range of intended users (*e.g.* managers, technical people, programmers);
- consult the range of purposes noted in § 2.2.2 and compare to one's own circumstances;
- identify [fairly general] motivating scenarios and competency questions and use these to help clarify specific uses and mechanisms (see below);
- produce a user requirements document for the target software system(s) (with respect to the role of the ontology)[2]

If you really cannot clearly identify your purpose, you should consider whether it is worth proceeding to build the ontology, for you may encounter problems later. For example, there will be no clear guidelines for determining the nature and content of the ontology; nor will it be clear by what criteria it can be evaluated.

4.2 Level of Formality

A prospective ontology builder also needs to decide how formal the ontology needs to be. This is determined in large part by the purpose and users of the ontology. For example, if the users are non-technical people and the primary purpose is to provide a shared vocabulary to facilitate communication between humans, then an informal glossary may suffice. In general, the degree of formality required increases with the degree of automation in the tasks that the ontology will support. For example, if the intended use is to support inter-operability or reuse and sharing of knowledge bases, then a more formal representation will be required.

In some cases both an informal and a formal ontology may be required to satisfy both technical and non-technical users. Where there are only technical users, it may still be useful to generate a complete informal ontology which can serve to both document and specify a subsequent formal encoding. There may be good reasons for any or all of these approaches depending on the specific circumstances.

[2]N.B. these requirements are not to be confused with the requirements (i.e. purpose) of the ontology.

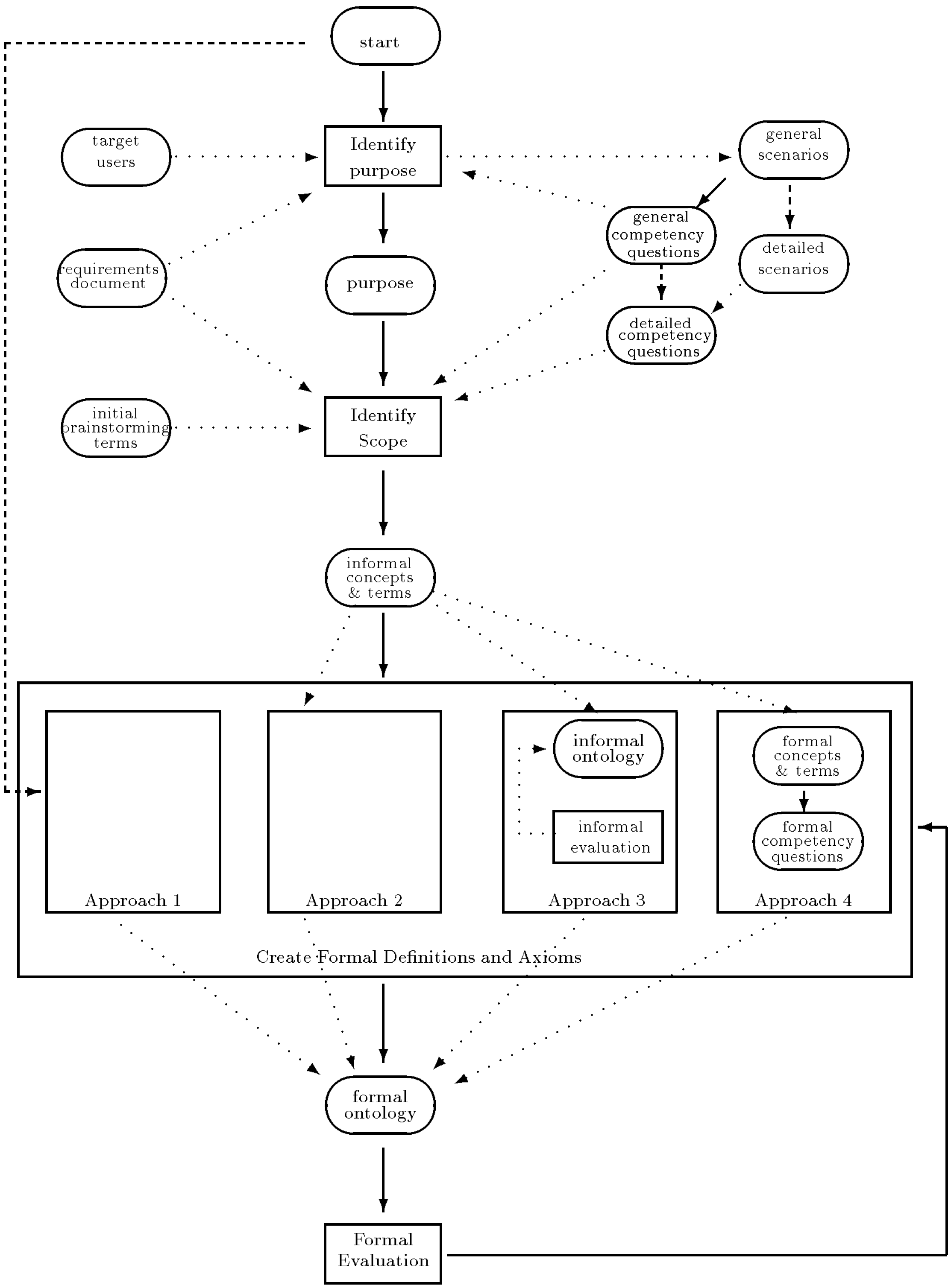


Figure 2: Various Approaches to Building Ontologies

4.3 Scope: subject matter

Once the purpose and level of formality of the ontology is fairly well determined, the next step is to identify the scope. The output for this phase is a set of concepts and terms covering the full range of information that the ontology must characterise to satisfy the requirements already identified. Below we consider two main ways to proceed with this process.

4.3.1 Motivating Scenarios and Informal Competency Questions

One excellent way to get a clear picture of the scope of the ontology is to create detailed scenarios that arise in the applications. These correspond to story problems and the scenarios should include possible solutions to the problem. This may have been done at a high level previously to help clarify the purpose of the ontology. At this stage, more detail and more coverage is appropriate. Ideally, exemplary scenarios for all envisaged kinds of situations and uses should be identified.

It may be that the concepts and terms defining the scope of the ontology can be gleaned directly from the scenarios themselves, by casual inspection. However, this is somewhat ad hoc. A more thorough approach is to use the scenarios as the basis for defining a complete set of competency questions. Competency questions are based on the scenarios and express different reasoning problems that must be supported. A set of questions is complete in the sense that if the ontology can support providing answers to all the questions, then the ontology can serve its intended purpose. In other words, the set of competency questions collectively specifies the expressive and reasoning requirements of the ontology.

Thus, we know what must be *in* the ontology. However, scoping also entails knowing what must *not* be in the ontology. The competency questions can also be used for this. Very simply, if there is no competency question that requires the use of a term or concept, then that term or concept is not included. Later we will see this principle used to determine whether an axiom needs to be included in the formal encoding of an ontology.

Competency questions should be devised in a hierarchical manner, starting with general ones, which in turn give rise to more specific ones analogous to a goal reduction tree (see [23] for details of the competency question approach).

4.3.2 Brainstorming and Trimming

Such motivating scenarios and competency questions are not always ready to hand. Or, they may not sufficiently cover the expected requirements. This may arise if no clear purpose emerged in the initial step. In this case, brainstorming could also effectively be used instead of or in conjunction with motivating scenarios and competency questions to do a more thorough and accurate job of scoping.

Proceed as follows. First brainstorm to produce a list of all potentially relevant concepts. If collectively, the persons involved possess insufficient expertise, then another corpus of knowledge may need to be consulted to ensure adequate coverage. The nature of brainstorming is that initially, nothing is excluded. Therefore, some way to trim the set of concepts down to size is required. There are two reasons for removing terms: lack of relevance and duplication (*i.e.* [near] synonyms).

To determine relevance, compare with the output of the previous stages which should have identified one or more of: purposes, a requirements document, motivating scenarios and/or competency questions.

For each and every term you should make a conscious decision as to whether to keep or get rid of it. To make the task more manageable, it may help to *group* the terms in to semantically similar categories. This will also facilitate identifying duplicate terms which can be eliminated, as well as be a basis for organising the ontology in later stages of development.

Scoping: concluding remarks

The primary and necessary output of the scoping phase is a set of concepts and terms that must be included in the ontology. The concepts may or may not have been structured in some way, *e.g.* into groups, or perhaps implicitly by hierarchy 'inheriting' structure that may have existed in the set of competency questions from which the terms were derived.

As indicated above, the degree to which there is confidence in these terms being the right ones varies according to how they were identified. In all cases, however, it should be expected that there will be some later modification and fine-tuning. This will be the result of the careful thought required to produce the definitions and structure the ontology as it is being designed and built, as well as during the evaluation phase.

4.4 Building the Ontology

With a set of terms and concepts that the ontology must include, the next step is to build the ontology itself. The main activity is to produce the definitions, but some decisions must also be made as to how and whether to arrange the definitions in any particular way, thus structuring the ontology.

4.4.1 A Range of Approaches

How to proceed at this point depends a great deal on what level of formality is required for the ontology to serve its intended purposes. It also depends on what has been done in the prior stages. Here we describe four possible approaches, there are other combinations and variations (see figure 2).

Approach 1 At one extreme, someone might more or less ignore all of the above stages and start the whole ontology building process by sitting down at a terminal and loading up, say, the KSL ontology editor [3] and defining terms and axioms. Surprisingly, some of the design decisions for this tool assume that the user will be doing this. Nevertheless, the advantages of a more principled approach are well known. This may be adequate for prototyping and/or for small or simple ontologies, but otherwise is perhaps best viewed as a hacker's approach.

Approach 2 Alternatively, one might have gone through the above steps and properly scoped the ontology. They might start form there and begin a formal encoding. This is likely to give better results than the previous approach, and may be suitable for small or simple ontologies. For more complex and/or larger ontologies, this approach may be too ad-hoc.

Approach 3 A third approach is to take great care and produce a complete intermediate document, an informal ontology consisting of terms and definitions in a structured form of natural language (e.g. the Enterprise method). Ideally, scenarios and competency questions would drive this process, but not necessarily. This informal ontology can serve three possible purposes: 1) it may be the final result, if no formal encoding is required; 2) it can serve as a specification for the subsequent formal code as well as 3) be documentation for it. If this approach is taken, there should be an evaluation/revise cycle before the informal ontology is deemed to be complete.

Approach 4 Instead of creating an intermediate informal ontology, one might proceed by identifying formal terms from the set of informal terms, using these to convert the informal competency questions into formal ones, and then specify the axioms and definitions that comprise the ontology (e.g. the TOVE method). This is really a more principled and formal variation on approach 2, in that no intermediate ontology is produced.

4.4.2 Guidelines

Whichever approach is taken, it is during the production of the actual definitions and determining how and whether there will be any explicit structure to the ontology that many previously reported guidelines and techniques apply. We briefly summarise these below, further details may be found in [23].

General Criteria

Various general criteria should be born in mind when creating definitions and organising an ontology. These can also be used after-the-fact as evaluation criteria. Below are some important ones adapted from those reported in [6].

Clarity: Definitions should be maximally clear and unambiguous, whether expressed in natural language or formally encoded. Use examples where possible to illustrate what is intended, negative examples can also be useful to clearly show what is *not* intended, where mistakes may be anticipated[3]. Also, state all underlying assumptions, especially where they are not explicitly formalised as axioms.

Consistency and Coherence: An ontology should be internally consistent; circularity should be avoided, especially if a formal encoding is desired. It should also be externally consistent, so terms best conform to common usage. Avoid introducing new terms, consult dictionaries, thesauri, and technical glossaries.

Extensibility and Reusability: An ontology should be designed in such a way as to maximise subsequent reuse and extensibility. This can be achieved by getting the right balance between being specific enough to perform the required tasks, but not so specific that it will be of little use to others. During formal coding, symbol-level biases should be avoided – *e.g.* those made purely for the convenience of notation or implementation. It also helps to be very careful to avoid introducing several terms that mean roughly the same thing; instead identify the key underlying term and reuse it to define other terms. This achieves parsimony, which in turn facilitates reusability.

Other Guidelines

Some specific guidelines to assist in identifying terms and producing definitions:

Go middle-out – In choosing which terms to define first, proceed in a *middle-out* fashion rather than top-down or bottom up. The choice of whether to go top-down, middle-out or bottom-up has a number of effects. A bottom-up approach results in a very high level of detail. This, in turn 1) increases overall effort, 2) makes it difficult to spot commonality between related concepts and 3) increases risk of inconsistencies which leads in turn to 4) re-work and yet more effort. The middle-out approach was used on the Enterprise Project and has been used successfully for many years as part of BSDM, developed by IBM[11]; the problems noted above are largely avoided.

Handling ambiguity – To reach agreement when terms are used ambiguously, concentrate on the underlying ideas first, ignoring the terms. Define each related idea, inventing meaningless labels for each, then decide on the most important idea(s) and *lastly*, choose appropriate terms.

4.5 Evaluation / Revision Cycle

There are various sorts of criteria that may be used to evaluate an ontology. Some are general, likely to apply to any ontology, others are more specific to a particular example.

[3] I'm indebted to Pat Hayes for this excellent suggestion.

General Some general criteria include those noted in the previous section, such as clarity, consistency and reusability. However, these are of limited use because there is no obvious way to measure them. To address that problem, Gómez-Pérez [4] has operationalised these criteria by identifying a wide range of detailed measurable criteria such as:

- The domains and ranges of all defined functions must themselves be defined;
- Redundancies should be avoided, some implicit ones can be recognised;

Automated support is available that uses these criteria for evaluating ontologies written in Ontolingua. However, the principles could be applied to other languages.

Specific Some techniques that use project-specific criteria include: manually checking the ontology against the identified purpose of the ontology, the user requirements document and/or informal competency questions, depending on what has been produced in the early stages of ontology development. These are appropriate for evaluating informal ontologies.

Approach 4 above, for building ontologies lends itself to a *formal* and functional approach for ontology evaluation. Specifically, the ontology must be able to answer all the formal competency questions, as well as contain *only* terms, definitions or axioms that are *required* to answer at least one competency question. This approach can be used in conjunction with other approaches and evaluation criteria.

5 Summary and Conclusion

In this paper, we have taken a step toward the development of a unified methodology for building ontologies. We first noted that there are a wide range of methodological techniques and methods reported to assist in building ontologies, but most are either project-specific, or only address a relatively small portion of the overall process. We have not attempted to develop a general framework that incorporates all previously reported work as we regard that as too ambitious at this point.

Instead, we began this process by considering two independently developed methodologies. We explained that a person who wished to build a new ontology would not very easily be able to determine which methodology to use, nor whether or which parts of each might be used. We proceeded to address this problem by merging these two methodologies, in an attempt to broaden the range of circumstances for which explicit methodological support is provided.

Future Work In the future more of the existing methods and techniques for ontology development must be taken into account, and if possible, merged into a coherent framework. Also, the level of granularity for the guidelines we suggest here for choosing different methods is very coarse – this should be refined.

Efforts should also be made to incorporate existing work in the closely related areas of semantic data modelling and conceptual modelling during the requirements acquisition phase in software engineering.

Acknowledgements

I am indebted to Martin King for inciting my interest in the methodological aspects of building ontologies, and for co-developing the methodology used to build the Enterprise Ontology [21]. Martin, along with Michael Gruninger, Ann Macintosh and an anonymous referee provided helpful feedback on earlier drafts.

Although I take full responsibility for any 'wrong' opinions and views expressed in this paper, many of the ideas came during stimulating discussions (both email and otherwise) with Michael Gruninger and Asuncion Gómez-Pérez. Nicola Guarino and Gertjan van Heijst provided feedback on the dimensions of variation for characterising ontologies.

References

[1] A. Abu-Hanna and W. Jansweijer. Modelling domain knowledge using explicit conceptualisation. *IEEE Expert*, 9(5):53–64, 1994.

[2] J.A. Bateman, R.T. Kasper, J.D. Moore, and R.A. WHitney. General organisation of knowledge for natural language processing:the penman upper model. Technical report, USC / Information Sciences Institute Institute, Marina del Rey , CA., 19.

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

[4] A. Gómez-Pérez, N. Juristo, and J. Pazos. Evaluation and assessment of knowledge sharing technology. In N.J. Mars, editor, *Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing 1995*, pages 289–296. IOS Press, Amsterdam, 1995.

[5] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[6] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5/6):907–928, 1995.

[7] M. Gruninger and M.S. Fox. The logic of enterprise modelling. In J. Brown and D. O'Sullivan, editors, *Reengineering the Enterprise*, pages 83–98. Chapman and Hall, 1995.

[8] M. Gruninger and M.S. Fox. Methodology for the design and evaluation of ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing*. International Joint Conference on Artificial Intelligence, 1995.

[9] N. Guarino. Understanding, building and using ontologies – a commentary to 'using explicit ontologies in kbs development' by by van heijst, schreiber, and wielinga. *International Journal of Human and Computer Studies*, 1996. *to appear (draft version).*

[10] N. Guarino and P Giaretta. Ontologies and knowledge bases- towards a terminalogical clarification. In N.J. Mars, editor, *Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing 1995*, pages 25–32. IOS Press, Amsterdam, 1995.

[11] IBM. Introduction to business system development method. Technical Report GE19-5387-01, International Business Machines Corporation, 1990.

[12] Fraser. J., A. Tate, and M. Uschold. The enterprise toolset - an open enterprise architecture. In *The Impact of Ontologies on Reuse, Interoperability and Distributed Processing*, pages 42–50. Unicom Seminars, London, 1995. Further information about the Enterprise Project and Ontology is available on the World Wide Web from: *http://www.aiai.ed.ac.uk/~entprise/enterprise/.*

[13] M. Jones, J. Wheadon, D. Whitgift, M. Niezatte, R. Timmermans, I. Rodriguez, and R. Romero. An agent based approach to spacecraft mission operations. In N.J. Mars, editor, *Towards Very Large Knowledges Bases - Knowledge Building and Knowledge Sharing 1995*, pages 259–269. IOS Press, Amsterdam, 1995.

[14] J. Lee (editor), M. Gruninger, Y. Jin, T. Malone, G. Yost, and other members of PIF Working Group. The pif process interchange format and framework – version 1.1. Technical Report 194, MIT Center for Coordination Science, 1996.

[15] N. Lehrer. Knowledge representation specification language. Technical report, DARPA/Rome Laboratory Planning and Scheduling Initiative, 1993. Reference Manual.

[16] D. Lenat and R.V. Guha. *Building Large Knowledge-based Systems: Representation and Inference in the CYC Project.* Addison Wesley, 1990.

[17] D. Skuce. Conventions for reaching agreement on shared ontologies. In *Proceedings of the 9th Knowledge Acquisition for Knowledge Based Systems Workshop*, 1995.

[18] D. Sleeman. Some recent advances in kb system building. In *The Proceedings of Expert Systems 94, of the*, pages 3–16, December 1994. the 14th Annual Technical Conference, Cambridge.

[19] J. Sowa. Top-level ontological categories. *International Journal of Human-Computer Studies*, 43(5/6):669–686, 1995.

[20] M. Uschold. Converting an informal ontology into ontolingua: Some experiences. In *Workshop on Ontological Engineering.* ECAI, 1996. Also available as AIAI-TR-192 from AIAI, The University of Edinburgh.

[21] M. Uschold and M. King. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing.* International Joint Conference on Artificial Intelligence, 1995. Also available as AIAI-TR-183 from AIAI, The University of Edinburgh.

[22] M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. Technical Report AIAI-TR-195, Artificial Intelligence Applications Institute, The University of Edinburgh, 1996. This ontology was developed as part of the Enterprise Project, see *http://www.aiai.ed.ac.uk/~entprise/enterprise/* for further information.

[23] M. Uschold and Gruninger M. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2), 1996. Also available as AIAI-TR-191 from AIAI, The University of Edinburgh.

[24] M. (editor) Uschold. Knowledge level modelling: Concepts and terminology. Technical Report AIAI-TR-196, Artificial Intelligence Applications Institute, The University of Edinburgh, 1996. To appear in a future issue of the Knowledge Engineering Review.

[25] G. van Heijst, B. Wielinga, and W. Jansweijer. Using explicit ontologies in kbs development. *International Journal of Human and Computer Studies*, 1996. (Forthcoming).

A The Enterprise Project

The overall objective of the Enterprise Project[4] [12] is to improve and where necessary replace existing modelling methods with a framework for integrating methods and tools which are appropriate to enterprise modelling and the management of change. This framework is based on an ontology for enterprise modelling.

A goal of the Enterprise Project is to provide a computer-based toolset which will help a) capture aspects of a business and b) analyse these aspects to identify and compare options for meeting the business requirements. The toolset will provide task management support to users by helping them perform enterprise modelling activities and guiding them through the toolset facilities. These facilities will enable 1) capture and description of an enterprise, 2) specifications of business problems/requirements (consistent with the ontology), 3) identification and evaluation of solution options and alternative design and implementation paths at strategic, tactical and operational levels, and 4) representations for the definition of relevant metrics and advanced simulation support.

The Enterprise Ontology — The Enterprise Ontology is conceptually divided into several major sections. These are listed below, along with a few of the most important concepts for each.

Meta-Ontology: Entity, Relationship, Role, Actor, State of Affairs;

Activities and Processes: Activity, Resource, Plan, Capability;

Organisation: Organisational Unit, Legal Entity, Manage, Ownership;

Strategy: Purpose, Strategy, Help Achieve, Assumption;

[4] The Enterprise Project is led by AIAI at The University of Edinburgh and the partners are IBM UK, Lloyd's Register, Logica and Unilever. The project is supported by the Department of Trade and Industry.

Marketing: Sale, Product, Vendor, Customer, Market.

Figure ?? illustrates how the ontology is intended to facilitate inter-operation among tools.

B TOVE

The goal of the TOVE (TOronto Virtual Enterprise) [7] project is to create an enterprise ontology that has the following characteristics: 1) provides a shared terminology for the enterprise that every application can jointly understand and use, 2) defines the meaning (semantics) of each term in a precise and as unambiguous manner as possible using First Order Logic, 3) implements the semantics in a set of Prolog axioms that enable TOVE to automatically deduce the answer to many 'common sense' questions about the enterprise, and 4) defines a symbology for depicting a term or the concept constructed thereof in a graphical context[5].

The TOVE ontologies constitute an integrated enterprise model, providing support for more powerful reasoning in problems that require the interaction of the following ontologies:

- Activities, states, and time
- Organisation
- Resources
- Products
- Services
- Manufacturing
- Cost
- Quality

This framework provides a characterisation of classes of enterprises by sets of assumptions over their processes, goals, and organisation constraints.

[5]This summary of TOVE is taken from [23] and was prepared by Michael Gruninger