Towards a Methodology for Building Ontologies

Mike Uschold and Martin King

AIAI-TR-183

July 1995

Presented at: “Workshop on Basic Ontological Issues in Knowledge Sharing”; held in conjunction with IJCAI-95

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

Abstract

We outline some requirements for a comprehensive methodology for building ontologies, and review some important work that has been done in the area which could contribute to this goal. We describe our own experiences in constructing a significant ontology, emphasising the ontology capture phase.

We first consider the very general issue of categorisation in modelling, and relate it to the process of ontology capture. We then describe the procedure that we used to identify the terms and produce definitions. We describe a successful way to handle ambiguous terms, which can be an enormous obstacle to reaching a shared understanding. Other important findings include: it may not be necessary to identify competency questions before building the ontology; the meta-ontology can be chosen after detailed text definitions are produced; defining terms which are 'cognitively basic' first can lead to less re-work.

1 Introduction

Currently, a considerable body of experience exists in building ontologies. The literature describing various case studies and other general issues relating to ontologies is mainly concerned with such things as: what is an ontology? what are they used for? their specific content; languages for representing them; tools for building them, etc.

Regarding *how* one goes about building an ontology, there are many hints, guidelines and anecdotal experiences reported in the literature. However, we are aware of no proposals for a general methodology for building ontologies. Some significant steps in this direction are described in:

- IDEF5: [8] a language and method for describing ontologies;
- a paper by Gruber discussing principles for designing ontologies [2];
- A paper by Skuce addressing the important issue of reaching agreement as a necessary step to sharing and integrating ontologies between multiple parties [10];
- A set of papers by Gomez-Perez on evaluation of knowledge sharing technology (*e.g.* [1]).

The goal of this paper is to take a further step. We first propose a skeletal methodology for building ontologies. We define a small number of stages that we believe will be required for any future comprehensive methodology. For each, we briefly indicate what currently has been done in the area, and consider what would be desirable in a comprehensive methodology.

Following this, we report our experiences in the successful development of a significant ontology as part of a large collaborative project. We describe the process we went through in developing our ontology. The particular emphasis is in the early phase of identifying and defining the key concepts and relationships in the domain and the production of a set of definitions of terms in natural language. We do not address the task of coding the ontology in a formal language, nor how it might be evaluated.

The skeletal methodology and the specific process described in the ontology capture phase extends the set of proposed methodological guidelines for building ontologies that has been reported to date. We do not attempt to a full analysis of all the literature in the area, nor to develop a comprehensive methodology. However we do draw comparisons with experiences and guidelines reported elsewhere.

A Note on Terminology

For the purposes of this paper, we adopt the following terminology:

- *concept:* broadly used to include any thing, notion or idea; this differs from its more restricted use in description languages;
- *conceptualisation:* an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality;
- *ontological theory:* a set of formulas intended to be always true according to a certain *conceptualisation*;
- *ontology:* refers to either or both of the above; disambiguated by context (this and previous two terms are as given in [5]).

2 A Skeletal Methodology

We envisage a comprehensive methodology for developing ontologies to include the following stages:

- Indentify Purpose;
- Building the Ontology;
  - Ontology capture,
  - Ontology coding,
  - Integrating Existing Ontologies;
- Evaluation;
- Documentation.

In addition, it should include a set of techniques, methods, principles and guidelines for each stage, as well as indicating what relationships exist between the stages (e.g. recommended order, interleaving, inputs/output).

Below, we define each stage and indicate what if any work has been reported that could be used to develop a comprehensive methodology.

2.1 Purpose

It is important to be clear about why the ontology is being built and what its intended uses are. It will also be useful to identify and characterise the range of intended users of the ontology.

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 re-use. Some of these purposes are implicit in the various interpretations of the word 'ontology' as noted in [5] (*e.g.* a vocabulary for vs a meta-level specification of a logical theory). Other dimensions of variation 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 re-used 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 [7]).

We are aware of no comprehensive surveys which attempt to identify and classify the range of all such purposes and end users, though [5] may constitute a step in that direction, at a fairly general level. Competency questions [4] can be seen as relating to the purpose of an ontology in very specific terms. Such a survey would be a useful starting point for ontology developers, assisting them to clarify their own purpose(s). It could thus play the role of a reference document in a comprehensive methodology.

2.2 Building the Ontology

2.2.1 Capture

By ontology capture, we mean:

1. identification of the key concepts and relationships in the domain of interest, *i.e.* scoping;
2. production of precise unambiguous text definitions for such concepts and relationships;
3. identification of terms to refer to such concepts and relationships;
4. agreeing on all of the above.

Perhaps, the most directly relevant work reported is in [10], where Skuce argues for an intermediate representation of a conceptualisation which is more formal than loosely structured natural language, but less formal than a formal language. He proposes a specific format for such an intermediate representation, which is to include assumptions, justifications as well as precisely worded definitions.

In § 3.4 we add further to the methodological assistance in the ontology capture phase. It is largely consistent with and complimentary to Skuce's proposal.

The whole field of knowledge acquisition is also relevant to this stage, but ontology capture is especially concerned with the knowledge level, *i.e.* independent from concerns of a particular coding language. This is analogous to the KADS [13] recommendation to produce a domain model before coding the knowledge base.

2.2.2 Coding

By coding, we mean explicit representation of the conceptualisation captured in the above stage in some formal language. This will involve committing to some meta-ontology, choosing a representation language, and creating the code.

With regards to choosing a language, possibly the most extensive work done in this area is the Plinius Project [12, 15]. They have experimented with a large variety of languages for representing their ontology in the materials science domain; these include Prolog, Conceptual Graphs, L-Lilog, Ontolingua, and several languages from the KL-ONE family (Back, Back++, Loom, Classic). These experiences could serve as a starting point for developing guidelines in choosing representation languages for ontologies.

Coding and capture are sometimes merged into a single step. Indeed, some of the design decisions of the KSL Ontology Editor[1] presume that ontology builders may be developing the conceptualisation on the fly[2]. This may be appropriate in some cases, however our experience suggests that many benefits derive from separating the two.

Gruber's criteria [2] for designing ontologies are relevant to the capture and coding stages. and should be integrated into any methodology.

Insofar as an ontology is a kind of a knowledge base, there is a wealth of useful methodological guidance that is potentially applicable. A comprehensive methodology would make very clear what applies for building ontologies as opposed to knowledge bases in general. It will also clarify under what circumstances, if any, capture and coding stages may be merged.

[1]URL: http://www-ksl-svc.stanford.edu:5915/

[2]Email communication with James Rice

2.2.3 Integrating Existing Ontologies

During either or both of the capture and coding processes, there is the question of how and whether to use ontologies that already exist. In general this is a difficult problem, and although much progress has been made in the context of Ontolingua, the general problem is very hard. Provision of guidance and tools in this area may be one of the biggest challenges in developing a comprehensive methodology. Skuce [10] has made some progress in this area. His main point is that in order to agree on ontologies that can be shared among multiple user communities, much work must be done to achieve agreement. One way forward is to make explicit all assumptions underlying the ontology.

2.3 Evaluation

Gomez-Perez provides a good definition of evaluation in the context of knowledge sharing technology:

> "to make a technical judgement of the ontologies, their associated software environment, and documentation with respect to a frame of reference ... The frame of reference may be requirements specifications, competency questions, and/or the real world. [1]"

Some detailed work has been done on evaluation of ontologies which could contribute to a comprehensive methodology for building ontologies. The approach taken in this work, is to look first at what has been done in the field of KBS, and to adapt it for ontologies.

2.4 Documentation

It may be desirable to have established guidelines for documenting ontologies, possibly differing according to type and purpose of the ontology.

As pointed out by Skuce [10], one of the main barriers to effective knowledge sharing, is the inadequate documentation of existing knowledge bases and ontologies. To address these problems, Skuce argues that all important assumptions should be documented, both about the main concepts defined in the ontology, as well as the primitives used to express the definitions in the ontology (which we refer to as a meta-ontology).

The facilities provided by Ontolingua, and supported by the KSL Ontology Editor facilitate both formal and informal documentation of such assumptions. Though such facilities may be conceptually straightforward, they can have significant benefit.

3 A Case Study

This completes the presentation of the skeletal methodology for building ontologies. In this section, we report on our experience in building a significant ontology as a collaborative effort among several parties.

After giving some general background information on the project, we discuss some initial considerations, such as:

- what was the nature of the thing to be built (*i.e.* the ontology)?
- what was it for?

- how would we represent it?
- how would we go about producing the definitions?

The main emphasis will be on ontology capture phase. We first consider the very general issue of categorisation in modelling, and relate it to the process of ontology capture. Finally, we describe the procedure that we in fact used to identify the terms and produce definitions. The idea is not to propose or argue for these as normative guidelines; rather we are sharing our experiences, in hopes that others may benefit.

3.1 Background

We are involved in a large collaborative project, a major part of which is to produce a substantial ontology. The ontology is to be used in a large piece of software that will integrate with a variety of independent software tools. The idea is to use the ontology as an inter-lingua into which and from which representations from other tools can be translated/converted as required. This approach to ontology use is thus quite similar to that used in the ATOS project [7].

We have produced a document containing definitions of a substantial variety and number of terms expressed in natural language text. Although relatively informal, great lengths were taken to ensure precision, avoid ambiguity, and make connections between similar things explicit. This facilitated subsequent coding of the terms in Ontolingua [3].

3.2 Initial Considerations

Initially, the following main issues were addressed:

- Decide what exactly it is that we intend to build (*i.e.* agree on a working definition of the term 'ontology');
- Identification of who would use the ontology and how would it be used;
- Choosing the most appropriate language for representing the ontology;
- Choosing the most appropriate method for ontology capture.

What is an Ontology? After much discussion, we agreed to use the definition proposed by Gruber: 'an explicit specification of a conceptualisation'. However, it was noted that this definition left much open for interpretation.

Using Guarino's subsequently introduced terminology [5], our text document describes a *conceptualisation* and the code is the corresponding *ontological theory*.

Purpose and Users — Initially this was discussed at a fairly abstract level. In the subsequent months, as the architecture of the software in which the ontology was to be used was being designed, this became much clearer.

The main use of the ontology is as an inter-lingua, as noted above. Classes of users identified included: ontology developers, ontology maintainers, users of ontology as basis of KBs for end-user applications.

We agree in principle with the idea of precise competency questions [4] to guide the ontology development. However, these sorts of question were too specific to guide the early development of our ontology. No detailed scenarios for end applications had been prepared.

The first task was to agree and define a set of important concepts to include and to decide on appropriate terms. Competency questions may serve a useful purpose in the later stage of evaluating the ontology code and when scenarios are being developed.

Choosing a representation language — Candidate ontology languages identified included: Ontolingua [3], KADS domain modelling language [13], Conceptual Graphs [11], IDEF5 [8], and BSDM [6] mapping language for business entities and processes.

The criteria used for choosing the most appropriate ontology language included:

- perspicuity,
- 'conceptual distance': extent to which the semantic primitives are 'close' to how language users think about the concepts to be represented. In other words, is it direct and natural or will much work be necessary to 'translate' the mental representation into the language.
- expressive power: will it represent the concepts we need?
- is it aligned with any current or forthcoming standards (*e.g.* ISO)?
- translatability / transportability;
- is it supported by any methods or guidelines for using the language?
- does it have a formal semantics which may assist with consistency checking?
- how easily obtainable is it?
- what kind of user base does it have?
- flexibility: does it force you to represent things in certain ways?

The criteria were weighted and a rough evaluation was done on the above candidates; the choices were narrowed to: Ontolingua, Conceptual Graphs and KADS domain modelling language. We eventually decided to use Ontolingua. Important reasons included: it is an emerging de-facto standard, having been specifically designed to represent ontologies; there is a formal semantics; there is a variety of software support tools.

Choosing an ontology capture method — We went through a similar exercise for choosing a method for ontology capture. There are very few such methods per se, but various sources of information exist. Some of these include:

- BSDM: [6] a comprehensive method developed by IBM for modelling enterprises as a preliminary step to developing IT systems;
- KADS; [13]
- IDEF5 [8];
- Object-Oriented Analysis and Design techniques (e.g ORCA);

- Tom Gruber's principles for ontology design [2].

Criteria for choosing a method included:

- coverage/completeness: how many steps of the ontology capture process does it address? will such guidelines apply for the range of concepts required?
- level of detail/granularity: does it address things at the right and/or multiple levels of granularity?
- learning curve: is it easy to learn? (*e.g.* is there existing documentation?)

The choice was to use BSDM as the main method, but to use some of the other methods and guidelines to test/evaluate the resulting ontology. In the event, we did not use any particular method, *per se*; rather we were guided by all of the above, as embodied in our collective experience, most particularly in knowledge acquisition/representation and BSDM.

Having covered the general background of this effort, and some important initial considerations, we now explore the general issue of categorisation and modelling, and its relationship to ontology capture.

3.3 Categorisation and Modelling

A central activity in the development of ontologies, and modelling more generally, is to identify those aspects of the real world that are of interest, to define them, and create terms to refer to them. If we view the real world as a set of objects in a universe of discourse then the above activity can be seen as choosing and defining *categories* of objects found in the universe. One important issue is the generality, or conversely, granularity of selected categories.

As such, it is appropriate to consider how to apply general ideas on categorisation to the choice of categories in modelling. In this discussion, we draw on work in cognitive psychology and linguistics from [9] and considerable experience of modelling using the BSDM method [6] developed by IBM.

3.3.1 A Theory of Categorisation

Lakoff [9] presents categorisation as fundamental to human cognition, a view that few will argue with. Less classically, he holds that categorisation is very much more complex than the view of hierarchies of classes with clear common properties and membership criteria being the norm. The theory he presents is based on a view developed from the later philosophy of Ludwig Wittgenstein through to research of Eleanor Rosch and associates.

Lakoff summarises the work in eleven themes, two of which are most relevant to the present discussion and are described as follows:

Basic-level categorisation: The idea that categories are not merely organised in a hierarchy from the most general to the most specific, but are organized so that the categories that are cognitively basic are 'in the middle' of a general-to-specific hierarchy. Generalisation proceeds 'upward' from the basic level and specialisation proceeds 'downward'.

Basic-level primacy:: The idea that basic-level categories are functionally and epistemologically primary with respect to the following factors: gestalt perception, image formation, motor movement, *knowledge organisation* [our italics], ease of cognitive processing (learning, recognition, memory, etc.), and ease of linguistic expression. [p 13]

A third theme is also relevant to the usability of chosen categories:

Functional embodiment: The idea that certain concepts are not merely *understood intellectually*; rather, they are *used* automatically, unconsciously, and without noticeable effort as part of normal functioning. Concepts used in this way have a different, and more important, psychological status than those that are only thought about consciously. [p 12]

Lakoff gives some examples to illustrate the basic level [p 46]:

Superordinate	Animal	Furniture
Basic Level	Dog	Chair
Subordinate	Retriever	Rocker

He summarises the properties of basic-level categories as follows [p 47]:

Perception: Overall perceived shape; single mental image; fast identification;

Function: General motor program (referring to physical interaction with a category member);

Communication: Shortest, most commonly used and contextually neutral words, first learned by children and first to enter the lexicon;

Knowledge Organisation: Most attributes of category members are stored at this level.

Lakoff supports the last point by citing experimental evidence published by Tversky and Hemenway [14].

3.3.2 Categorisation and Ontology Capture

We considered it necessary to produce a document defining the terms of the Ontology in natural language. In other words to capture the conceptualisation. This has two roles:

- as a requirements specification for coding the ontology;
- as documentation suitable for non-technical people.

This required defining many terms and a strategy for the sequence for doing so. Top-down would have required starting with a few general terms and risking imprecision and rework. Bottom-up would have required definition of a large number of detail terms first with the risk that many of them would not be important in the final Ontology. The above theory suggested a middle-out approach which seemed less likely to suffer from these problems. We chose to define the 'basic' concepts before the super- and sub-ordinate ones. This seemed a promising approach because:

- The basic ones seem to be the most important ones, so getting them right first is likely to involve less backtracking when defining other concepts;
- The non-basic concepts would normally be defined in terms of the basic ones;
- This was likely to increase the clarity of the document, especially for the non-technical portion of the intended audience;
- It is backed by the considerable experience of one of this paper's authors in using BSDM.

We decided to use this approach as an experiment, albeit an uncontrolled one. We would note the pros and cons as we progressed.

3.4 A Procedure for Ontology Capture

In this section we describe in some detail, the process we went through in informally defining the ontology. We do not present this as a set of normative guidelines, supposing that it is better than any other approach. We felt it worked well; we offer it for others to examine and consider what may be useful for them. We consider the following four phases in turn: scoping, producing definitions, review, and development of a meta-ontology.

3.4.1 Scoping

BRAINSTORMING — Have a brain-storming session to produce all potentially relevant terms and phrases; at this stage the terms alone represented the concepts, thus concealing serious ambiguities and differences of opinion.

Brainstorming worked well for us; however if there was insufficient collective domain expertise of those involved, another corpus of knowledge may need to be consulted to ensure adequate coverage.

GROUPING Structure the terms loosely into work areas corresponding to naturally arising sub-groups. In our case, groups arose such that terms were more related to other terms *within* the same group than they were to terms in other groups.

For each term:

- provisionally categorise it for inclusion, exclusion or borderline; this was determined mainly by reference to a previously agreed requirements document;
- keep notes to record such decisions for future reference;
- group similar terms and potential synonyms together for further consideration.

Finally, identify 'semantic' cross-references between the areas (*i.e.* concepts that are likely to refer to or be referred to by concepts in other areas.

3.4.2 Deciding What To Do Next

Determining Meta-Ontology — Initially, do not formally commit to any particular meta-ontology. Doing so can constrain thinking and potentially lead to inadequate or incomplete definitions. Also, if it ends up being wrong, many definitions may have to be re-done.

Instead, let the careful consideration of the concepts and their inter-relationships determine the requirements for the meta-ontology. Keep in mind various possibilities, and use words and phrases in a consistent manner where appropriate (e.g. role, entity, relationship, type, instance).

If use of a term appears loaded due to commonly assumed technical meanings, use neutral ones instead (*e.g.* 'thing' rather than 'entity').

Work Areas — Address each work area in turn. Start with work areas that have the most semantic overlap with other work areas; these are the most important to get right in the first place. If there is little overlap between work areas, work on them in any order.

Terms — Define the cognitively basic terms in each work area before moving on to more abstract and more specific terms withing a work area. In our experience, this makes it easier to relate terms in different areas more precisely.

3.4.3 Produce Definitions

Complete all definitions in all work areas.

Degree of Effort — There was considerable variation in the degree of effort required to agree on definitions and terms for underlying concepts. For example:

- For some terms consensus on the definition of a single concept was fairly easy.
- In other cases several terms seemed to correspond with one concept definition. In particular, there were several cases where commonly used terms had significantly different informal usage, but no useful different definition could be agreed. This was recorded in notes against the definition.
- Some highly ambiguous terms area identified as corresponding with several closely related, but different concepts. In this situation, the term itself *gets in the way* of a shared understanding.

Handling Ambiguous Terms — In the above special case where a term has many possibly meanings, we proceeded as follows:

1. Suspend use of the term; it is too ambiguous.
2. Clarify the *ideas* by carefully defining each concept using as few technical terms as possible, or only those whose meaning is agreed – *consult the dictionary.*
3. It can be helpful to give these definitions meaningless labels such as x1, x2, x3 etc. so they can be conveniently referred to.

4. Determine which, if any of the concepts are important enough to be in the ontology [usually one];
5. Choose a term for the concept (avoiding the original ambiguous term, if at all possible).

A good example of this phenomenon is the introduction of technical terms like 'Thing' to avoid potentially confusing terms like 'Entity' and 'Object'.

GUIDELINES — In all cases the following guidelines were followed:

- Produce a natural language text definition, being as precise a possible;
- Ensure consistency with terms already in use; in particular:
  - make ample use of dictionaries and other technical glossaries,
  - *avoid introducing new terms* where possible;
- Indicate the relationship with other commonly used terms that are similar to the one being defined (*e.g.* synonyms or variants referring to the same underlying notion, but perhaps from different perspectives)
- The definition of each term is intended to be necessary and sufficient as far as this is possible in natural language. Provide clarification or additional information essential to understanding the definition as separate notes following the definition.
- Give examples where appropriate.

WORDING — Although the text version of the ontology served as the specification for producing code, there was a requirement that it be accessible to non-technical readers. To achieve an appropriate balance between technical precision and clarity, we :

1. kept the text definitions relatively informal;
2. equivalent, but more technically precise definitions cast using the primitives in the meta-ontology are used in documentation directly accompanying the code.

3.4.4 Review

Critically review definitions, revising as appropriate; where important decisions were made overturning previous decisions, keep track of the changes as a set of historical notes.

3.4.5 Meta-Ontology

Devise a meta-ontology, using the natural language definitions as an an implicit requirements specification. This serves as the basis for the coding stage, which entails further formalisation.

4 Conclusion

In this paper, we have outlined the skeleton for a methodology for building ontologies. We have suggested some guidelines derived from our experience in the area of ontology capture. The key points to note are:

- It may not be possible or useful to set out detailed competency questions before the main content and structure of the ontology is developed.

  A comprehensive methodology should incidate under what circumstances competency questions should be used to clarify purpose, drive the ontology capture and coding stages as well as provide input to the evaluation phase.

- We found that it can be beneficial to *not* first choose a meta ontology, but rather to allow the carefully considered natural language definitions serve as implicit requirements for the meta-ontology.

  Related to this, we recommend separating capture from coding so as not to be unduly influenced by accidental features of formalism.

- Define cognitively basic terms first; our experience was that this worked well. Our intuition is that this helped avoid re-work, but as it was not a controlled experiment, we can make no strong claims in this regard.

- *Terms get in the way.* Sometimes, one's pre-conceived ideas about what a term means can severely undermine communication and prevent any agreement on a proposed definition. Our approach for dealing with this is to ignore the terms and concentrate on the underlying ideas. Identify and carefully define each of the closely related ideas. Once the ideas are clear, you can agree on which must be included in the ontology and agree on what term(s) to use (preferably avoiding the original term).

References

[1] 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.

[2] 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.

[3] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 2:199–220, 1993.

[4] M. Gruninger and M. Fox. The role of competency in enterprise engineering. In *IFIP WG5.7 Workshop on Benchmarking- Theory and Practice*. IFIP, Trondheim, Norway, 1994.

[5] 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.

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

[7] 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.

[8] KBSI. Knowledge based systems incorporated,. Technical Report IDEF5 Method Report, AL/HRGA, Wright-Patterson Air Force Base, Ohio, September 1994.

[9] G. Lakoff. *Women, Fire and Dangerous Things*. University of Chicago Press, 1987.

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

[11] J. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison Wesley, Reading, MA, 1984.

[12] P.H. Speel, P.E. van Raalte, P.E. van der Vet, and N.J. Mars. Scalability of the performance of knowledge representation systems. In *Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing 1995*, pages 173–183. IOS Press, Amsterdam, 1995.

[13] D.S.W. Tansley and C.C. Hayball. *Knowledge-Based Systems Analysis and Design: a KAD Developers Handbook*. Prentice Hall, 1993.

[14] B. Tversky and Hemenway. K. Objects, parts, and categories. *Journal of Experimental Psychology*, 113:169–193, 1984.

[15] P.E. van der Vet and N.J. Speel, P.H.and Mars. Ontologies for very large knowledge bases in materials science: a case study. In N.J. Mars, editor, *Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing 1995*, pages 73–83. IOS Press, Amsterdam, 1995.