ResearchGate

See discussions, stats, and author profiles for this publication at:
https://www.researchgate.net/publication/220123854

A Survey on Ontology Creation Methodologies

Article *in* International journal on Semantic Web and information systems · April 2005

Impact Factor: 0.9 · DOI: 10.4018/jswis.2005040103 · Source: DBLP

CITATIONS
102

READS
459

2 authors:

Matteo Cristani
University of Verona

65 PUBLICATIONS **316** CITATIONS

SEE PROFILE

Roberta Cuel
Università degli Studi di Trento

48 PUBLICATIONS **409** CITATIONS

SEE PROFILE

All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Roberta Cuel
Retrieved on: 11 June 2016

A Survey on Ontology Creation Methodologies

Matteo Cristani, Università di Verona, Italy
Roberta Cuel, Università di Verona, Italy

ABSTRACT

In the current literature of knowledge management and artificial intelligence, several different approaches have been tried to the problem of developing domain ontologies from scratch. All these approaches deal fundamentally with three problems: (1) providing a collection of general terms describing classes and relations to be employed in the description of the domain itself; (2) organizing the terms into a taxonomy of the classes by the ISA relation; and (3) expressing in an explicit way the constraints that make the ISA pairs meaningful. Though a number of such approaches can be found, no systematic analysis of them exists which can be used to understand the inspiring motivation, the applicability context, and the structure of the approaches. In this chapter, we provide a framework for analyzing the existing methodologies that compare them to a set of general criteria. In particular, we obtain a classification based upon the direction of ontology construction; bottom-up are those methodologies that start with some descriptions of the domain and obtain a classification, while top-down ones start with an abstract view of the domain itself, which is given a priori. *The resulting classification is useful not only for theoretical purposes but also in the practice of deployment of ontologies in Information Systems, since it provides a framework for choosing the right methodology to be applied in the specific context, depending also on the needs of the application itself.*

Keywords: Please provide

INTRODUCTION

In the recent past, complex markets have been characterized by a huge specialization of work, a high level of outsourcing processes, and a more open Porter's chain that develop and increase the needs of intra- and interorganizational networks. Both intraorganizational networks among strategic units, divisions, groups, and other even smaller substructures and interorganizational networks, such as industrial districts and knowledge networks (Hamel & Prahalad, 1990) are

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

composed of a constellation of specialized units (Ashby; 1956; Numagami, Ohta & Nonaka, 1989), which might not be controlled totally by a unique subject and might grow and differentiate their activities, their system of artifacts, and their view of the world in an autonomous way. Although every unit uses a different view of the world (i.e., different conceptualizations), they should coexist as in a biofunctional system (Maturana & Varela, 1980) and communicate, coordinate, and share knowledge in a networked environment. Furthermore, this continuous and unpredictable encountering of different views might enable the creation of an unexpected and innovative combination of processes and products (Chandler, 1962).

Most of these processes are based nowadays on the Web, and in this scenario, tools and technologies that sustain knowledge telecommunication, coordination, and sharing are increasing their importance. Therefore, both the scientific community on computer science and industries are interested in what is called the Semantic Web. The Semantic Web community has grown in terms of dimension and specialization, and different viewpoints of creating and managing semantic tools have been taken on the nature of the Web. In studies, an excellent role is being assumed by discipline about development of systems and methodology that allow the combination of several different views (expressed through taxonomies, classifications, contexts, and ontologies) of the world. In particular, ontology can be considered the boundary topic between the Semantic Web research and Information Systems that mostly deserves special attention to the methodological aspects. This is the argument that urged researchers all over the world to create outstanding projects that involve methodological research, and that urged industries to create ontology-based applications that express the units' points of view and, at the same time, allow knowledge exchange among different units and their perspectives (Bonifacio, Bouquet & Cuel, 2002). In this work, we are interested especially in ontology, because that it is one of the most important systems and methods that allows an explicit representation of knowledge and expresses itself a view of the world. Even if implicitly, each organization, unit, or employee interconnected in a networked system uses a personal conceptual schema that could be shared or at least understood among them. Therefore, developing a good ontology seems to be one of the critical issues n this area. For this reason, this chapter focuses on the methodologies in literature that provide frameworks for developing *Formal Ontologies*. Frameworks are provided by ontology creation tools, which explicitly or implicitly endow a series of steps that the developer should follow. For instance, some ontology development tools make available documentation that describes the better way in which an ontology developer might create an ontology; other ontology development tools implicitly force the developer to follow some specific steps during the process of ontology creation.

BACKGROUND THEORIES

It is quite well established in recent investigation on Information Systems, that

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

formal ontologies are a crucial problem to deal with, and, in fact, they received a lot of attention in several different communities (e.g., knowledge management, knowledge engineering, natural language processing, intelligent information integration, etc.) (Fensel, 2000). Ontologies have been developed in Artificial Intelligence to facilitate knowledge sharing and reuse, and now are applied and intended to be used in expert systems of almost all types of industries. The concept of ontology that is adopted in this chapter is taken from the general considerations on the use of philosophical issues in Artificial Intelligence:

"the systematic, formal, axiomatic development of the logic of all forms and modes of being." (Cocchiarella, 1991, p. 000)

Another commonly accepted definition is that an ontology is an explicit specification of a shared conceptualization that holds in a particular context. The actual topic of ontology is one of those themes that epistemology dealt with in philosophical studies of Parmenides, Heraclitus, Plato, Aristotle, Kant, Leibnitz, Wittgenstein, and others. Ontologies define the kind of things that exist in the world and, possibly, in an application domain. In other words, an ontology provides an explicit conceptualization that describes semantics of data, providing a shared and common understanding of a domain. From an AI perspective we can say that:

"… ontology is a formal explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of phenomena in the world by having identified the relevant concepts of those phenomena. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine readable. Shared reflect that ontology should capture consensual knowledge accepted by the communities." (Gruber, 1993, p. 000)

"… an ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definition and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretation of terms." (Jasper & Ushold, 1999, p. 000)

The main idea is to develop an understandable, complete, and sharable system of categories, labels, and relations that represent the real world in an objective way. For instance, one of the interesting results achieved by Aristotle is the definition of general categories used to describe the main features of events, situations, and objects in the world: quality, quantity, activity, passivity, having, situated, spatial, temporal. Kant figured out only four macro categories used to describe the world: quantity, quality, relation, and modality. Nowadays, ontologies are not only an explicit and theoretical specification of a shared conceptualization or a common and abstract understanding of a domain, but they also are implied in projects as conceptual models that allow:

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

- communication among people and heterogeneous and widely spread application systems;
- content-bases access on corporate knowledge memories, knowledge bases, archives;
- agent understanding through interaction, communication, and negotiation of meanings; and
- understanding and agreement upon a piece of information structure.

In other words, ontologies provide qualitatively new levels of services in several application domains such as the Semantic Web (Ding & Foo, 2002) or federated databases. Moreover, they enable reuse of domain knowledge, make domain assumption explicit, and separate domain knowledge from the operational knowledge.

Unfortunately, in the real world or in practical applications (i.e., information systems, knowledge management systems, portals, and other ICT applications), the general and universal categories (studied by philosophers for along time) are not being used widely. In particular, two main problems come up in modern applications:

- it is difficult to implement a general ontology within specific domains;
- it is too expensive to create very complex, complete, and general ontologies.

Therefore, less complete, correct, and consistent ontologies have been used, and problems have arisen due to the fact that in the same project or domain, people might use different ontologies composed by various combinations of categories. This means that different ontologies might use different categories or systems of categories to describe the same kinds of entities; or even worse, they may use the same names or systems of category for different kinds of entities. Indeed, it might be that two entities with different definitions are intended to be the same, but the task of proving that they are, indeed, the same may be difficult, if not impossible (Sowa, 2000). The basic assumption of this behavior is that what we know cannot be viewed simply as a picture of the world, as it always presupposes some degree of interpretation. Different categories represent different perspectives, aims, and degrees of world interpretation. Indeed, depending on different interpretation schemas, people may use the same categories with different meanings or different words to mean the same thing. For example, two groups of people may observes the same phenomenon but still see different problems, different opportunities, and different challenges. This essential feature of knowledge was studied from different perspectives, and the interpretation schemas were given various names (e.g., paradigms [Kuhn, 1979], frames [Goffman, 1974], thought worlds [Dougherty, 1992], context [Ghidini & Giunchiglia, 2001], mental spaces [Fauconnier, 1985], cognitive path [Weick, 1979]). This view in which the explicit part of what we know gets its meaning from a typically implicit or taken-for-granted interpretation schema leads to some important consequences regarding the adoption and use of categories and ontologies.

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Therefore, an ontology is not a neutral organization of categories, but rather it is the emergence of some interpretation schema, according to which it makes sense to organize and define things in that way. In short, an ontology is often the result of a sense-making process and always represents the point of view of those who took part in that process (see Benerecetti, Bouquet, & Ghidini [2000] for a in-depth discussion of the dimensions along which any representation—including an ontology—can vary, depending on contextual factors). The person who takes part in ontology creation and management can be an expert in ontology creation or a person who, through daily work, creates and manages his or her point of view represented by an ontology.

In computer science, several languages and tools exist for helping final users and system developers in creating good and effective ontologies. In particular, various tools help people create, either manually or semi-automatically, categories, partonomies, taxonomies, and other organization levels of ontologies. Some of the most important ontology editors and ontology manager are:

- **Protégé-2000:** http://protege.stanford.edu;
- **SWOOP:** http://www.mindswap.org/2004/SWOOP;
- **KAON:** http://kaon.semanticweb.org;
- **WSMX:** Web Services Execution Environment is a dynamic discovery that allows selection, mediation, invocation, and interoperation of Semantic Web services. More information can be found at http://sourceforge.net/projects/wsmx;
- **OWL-S Editor:** A plug-in developed on Protégé. Its goal is to allow easy, intuitive OWL-S service development for users who are not experts in OWL-S. It's an open source software at http://owlseditor.projects.semwebcentral.org/;
- **OntoManager:** A workbench environment that facilitates the management of ontologies with respect to a user's needs. http://ontoware.org/details/ontomanager.

Behind these tools and techniques, different domain-independent languages have been developed and spread using RDF(S), DAML+OIL, OWL and its specifications, KIF, and so forth. DAML+OIL has been developed as a language for specifying descriptions, in particular to serialize in XML description logics. The expressive power of DAML+OIL is, therefore, at syntactic level, equivalent to T-box of, for instance, SHIQ, the last evolution in KL-ONE family. Finally OWL (Web ontology language) is able to integrate the expressive power of DAML+OIL along with the specification of ISA relations in an explicit way.

A FRAMEWORK FOR METHODOLOGY ANALYSIS

In this work, we do not argue about correctness, completeness, and consistency of ontologies, and we do not point out special characteristics that specific

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

development languages and systems should sustain. Moreover, in this chapter, we will not focus upon the ontologies languages, even if these would be an interesting research topic. Although the evolution of languages is, indeed, a single stream and focuses upon expressive power and computational expressive, it is out of the scope of this chapter.

The first phase of the investigation documented in this chapter had the objective of reviewing the existing methodologies for the creation of a domain ontology from scratch. We discovered that in the existing methodologies, there is a core structure that is independent of both languages and tools; therefore, we analyze ways in which ontology developers can create their own ontology and practices that should be adopted by developers to create the most effective ontology. In particular, we want to measure what are the common elements that each methodology take into account, in which way a methodology sustains the user in the ontology creation processes, and how non-expert people can adopt a methodology and create their own representations. Therefore, in the following section, some important methodologies will be described and analyzed. We tried to define:

- names of the phases;
- elements and knowledge that should be in input in each phase;
- description of the processes and operations that are developed in each phase; and
- expected output.

Unfortunately, we do not have all the framework's elements for all of the following methodologies, because some of them are more research oriented and are not focused on the description of how users and developers can utilize them in the process of ontologies creation. Although these methodologies are not well described in existing papers, we decided to analyze them due to their importance in the Semantic Web community.

In Section 4, we survey the most relevant methodologies of recent investigations. We cannot be exhaustive for at least three reasons: (1) in the practice of Semantic Web applications and in several real cases of intelligent information systems, people developed their own methodologies, which makes almost impossible a complete review; (2) several methodologies have been developed as an illustration of the principles that inspired tools and languages but are essentially equivalent to other methodologies; and (3) several cases of implicit methodologies exist; in particular, in the specific context of language or tools.

SOME RELEVANT METHODOLOGIES

In computer science, knowledge management, knowledge representation, and other fields, a lot of languages and tools are developed with the aim of helping people and system developers create good and effective ontologies. In particular, a lot of tools help people create, either manually or semi-automatically, categories, partonomies, taxonomies, and so

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

on. Indeed, behind these tools and techniques, different approaches, methods, and techniques are used to develop a big quantity of heterogeneous ontologies. In this section, we will describe some of them, and we will try to compare the more significant principles that are sustaining these approaches and that are all described in existing scientific papers.

DOLCE: Descriptive Ontology for Linguistic and Cognitive Engineering

The main authors' idea is to develop not a monolithic module, but rather a library of ontologies (WonderWeb Foundation Ontologies Library) that allows agents to understand one another despite forcing them to interoperate by adoption of a single ontology (Masolo et al., 2002). The authors intend the library to be:

- **minimal:** the library is as general as possible, including only the most reusable and widely applicable upper-level categories;
- **rigorous:** ontologies are characterized by means of rich axiomatizations;
- **extensively researched:** modules in libraries are added only after careful evaluation by experts and after consultation with canonical works.

One of the first modules of their foundational ontologies library is a Descriptive Ontology for Linguistic Cognitive Engineering (DOLCE). DOLCE is an ontology of particulars and refers to cognitive artifacts that depend on human perception, cultural imprints, and social conventions. Their ontology derives from armchair research in particular (in other terms, it is the result of an intellectual speculation), referring to enduring and perduring entities from philosophical literature.

Finally, basic functions and relations (according to the methodology introduced by Gangemi, Pisanelli & Steve [1998]) should be:

- general enough to apply to multiple domains;
- sufficiently intuitive and well studied in the philosophical literature;
- held as soon as their relata are given, without mediating additional entities.

The methodology that sustain this type of ontologies is based on a few starting points for building new ontologies:

1. Determine what things there are in the domain to be modeled.
2. Develop easy and rigorous comparisons among different ontological approaches.
3. Analyze, harmonize, and integrate existing ontologies and meta-data standards.
4. Describe a foundational ontology on paper, using a full first-order logic with modality.
5. Isolate the part of the axiomatization that can be expressed in OWL and implement it.
6. Add the remaining part in the form of KIF comments attached to OWL concepts.

This methodology doesn't describe phases that a developer should follow in

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

ontology creation, but rather focuses more on expressivity problems, on partial alignment to lower levels of WordNet, and on much more philosophical aspects of the ontology creation. Although it's not possible to complete our framework, we decided to add this methodology because we consider DOLCE as one of the most representative methodologies in the field.

Ontology Development 101

This methodology has been developed by the authors involved in ontology editing environment such as Protégé-2000, Ontolingua, and Chimaera (Noy & McGuinnes, 2001). They propose a very simple guide based on iterative design that helps developers, even the non-expert developer, to create an ontology using these tools. According to our framework, the sequence of steps to develop an ontology is described well in the Table 1.

OTK Methodology

The methodology developed within the On-To-Knowledge project is called OTK Methodology, and it is focused on application-driven development of ontology during the introduction of ontology-based knowledge management systems (Fensel et al., 2000; Lau & Sure, 2002; Sure et al., 2002; Sure, 2003). The main aim of this methodology is to sustain a new process based on human issues, in which experts in industrial contexts are capable of creating and managing their own ontology, and ontology engineers are allowed to create effective ontologies. In other words, according to this methodology, strong efforts should be focused on physical presence of engineering and industrial experts, brainstorming processes (in particular, during the early stages of ontology engineering, especially for domain experts not familiar with modelling), and advanced tools for supporting ontology creation. According to our framework analysis, the sequence of the steps to develop an ontology is described well in the Table 2.

Methontology

One of the most famous ontology design methodologies (supported by ontology engineering environment WebODE) is "Methontology." It tries to define the activities that people need to carry out when building an ontology (Fernández, Gòmez-Pérez, & Juristo, 1997). In other words, a flow of ontology development processes for three different processes: management, technical, and supporting.

The ontology development process is composed of the following steps:

- project management activities include:
  - **planning:** it identifies which tasks are to be performed, how they will be arranged, how much time and what resources are needed for their completion;
  - **control:** it guarantees that planned tasks are completed in the manner in which they were intended to be performed;
  - **quality assurance:** it assures that the quality of each and every product output is satisfactory.
- **development:** oriented activities include:

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Table 1: Ontology development 101 methodology

Name of the Phase	Input	Phase Description	Output
Determine domain and scope of the ontology	Nothing. It is the first step	Definition of - what is the domain that the ontology will cover, - what ontology will be used, - what types of question the ontology should provide answers to (competency questions are very important in this domain; they allow the designer to understand when ontology contains enough information and when it achieves the right level of detail or representation), - who will use and maintain the ontology.	The resulting document may change during the whole process, but at any time, this documentation helps to limit the scope of the model.
Consider reusing existing ontologies	Documents with the domain and the scope of the ontology	Looking for other ontologies that are defining the domain. There are libraries of reusable ontologies on the Web and in literature (e.g., Ontolingua ontology library, DAML ontology library, UNSPSC, RosettaNet, and DMOZ)	One or more domain ontologies, or part of them with their description
Enumerate important terms in the ontology	Documents with the domain, the scope of the ontology, and libraries on the domain	Write a list of all terms used within the ontology, and describe the terms, their meanings, and their properties	Terms and important aspects to model in the ontology
Define the classes and the class hierarchy	Important terms in the ontology, domain, and scope description	There are several possible approaches in developing a class hierarchy: - top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts; - bottom-up development process goes in the opposite direction; - a combination development process is a combination of the top-down and bottom-up approaches	Classes and class hierarchy
Define the properties of classes-slot	The taxonomy, and the domain and scope description	Add all the necessary properties and information that allow the ontology to answer the competency questions	Classes, class hierarchy, and properties
Define the facets of the slots	Slots and classes	There are different facets describing the value type, allowed values, the number of the values, and other features of the values the slot can take: slot cardinality, slot-value type, domain, and range	Ontology
Create instances	The ontology	Create individual instances of classes in the hierarchy, which means choosing a class, creating an individual instance of that class, and filling in the slot values.	Ontology and the modeled domain

- **specification:** it states why the ontology is being built, what its intended uses are, and who are the end-users;
- **conceptualization:** structures the domain knowledge as meaningful models at the knowledge level;
- **formalization:** transforms the conceptual model into a formal or semi-computable model;
- **implementation:** builds computable models in a computational language.

- support activities: include a series of activities performed at the same time as development-oriented activities:
  - knowledge acquisition;
  - **evaluation:** it makes a technical judgment of the ontologies, their associated software environment,

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Table 2. OTK methodology

Name of the Phase	Input	Phase Description	Output
Feasibility study: identify problem/opportunity areas and potential solutions, and put them into a wider organizational perspective. In general, a feasibility study serves as a decision support for economical, technical, and project feasibility in order to select the most promising focus area and target solution	Nothing. It is the first step	The process of studying the feasibility of the organization, the task, and the agent model proceeds in the following steps: - carry out a scoping and problem analysis study, consisting of two parts: o identifying problem/opportunity areas and potential solutions and putting them into a wider organizational perspective; o deciding about economic, technical, and project feasibility in order to select the most promising focus area and target solution; - carry out an impacts and improvements study for the selected target solution, again consisting of two parts: o gathering insights into the interrelationships among the business task, actors involved, and use of knowledge for successful performance, and what improvements may be achieved here; o deciding about organizational measures and task changes in order to ensure organizational acceptance and integration of a knowledge system solution.	The resulting document may change during the whole process. Indication of the most promising area and target solutions and of measures and task changes in order to ensure organizational acceptance and integration of knowledge systems solutions. An ontology requirements specification document (ORDS) should be formed. The ORSD describes what an ontology should support, sketching the planned area of the ontology application and listing, for example, valuable knowledge sources for the gathering of the semi-formal description of the ontology.
Kick-off phase	It starts with an ORSD	The ORSD should guide an ontology engineer to decide about inclusion and exclusion of concepts and relations and the hierarchical structure of the ontology. In this early stage, one should look for already developed and potentially reusable ontologies.	The result will be a document containing: - goal, domain, and scope of the ontology; - design guidelines; - knowledge sources; - (potential) users and usage scenarios; - competency questions; supported applications.
Refinement phase	Specification given by the kick-off phase	The goal of the refinement phase is to produce a mature and application-oriented "target ontology" according to the specification given by the kick-off phase. This phase is divided into different subphases: - a knowledge elicitation process with domain experts based on the initial input from the kickoff phase. - a formalization phase to transfer the ontology into the target ontology expressed in formal representation language like DAML+OIL.	- The knowledge about the domain is captured from domain experts in the previously mentioned competency questions or by using brainstorming techniques; - To formalize the initial semi-formal description of the ontology, we first form a taxonomy out of the semi-formal description of the ontology and add relations other than the "is-a" relation, which forms the taxonomical structure.
Evaluation phase	The refinement results	The refinement phase is closely linked to the evaluation phase. If the analysis of the ontology in the evaluation phase shows gaps or misconceptions, the ontology engineer takes these results as an input for the refinement phase. It might be necessary to perform several (possibly tiny) iterative steps to reach a sufficient level of granularity and quality	It serves as a proof for the usefulness of developed ontologies and their associated software environment.
Application and evolution phases		Define strict rules for the update, insert and delete processes within ontologies	User guide for future application and evolution phases

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Table 3. Methontology through our framework

Name of the Phase	Input	Description	Output
Planning	Nothing: first step	Plan the main tasks to be done, the way in which they will be arranged, the time and resources that are necessary to perform these tasks	A project plan
Specification	A series of question such as: "Why is this ontology being built and what are its intended uses and end-users?"	Identify ontology's goals	Ontology requirements specification document, specifying purposes and scopes. Its goal is to produce either an informal, semi-formal, or formal ontology specification document written in natural language, using a set of intermediate representations or using competency questions, respectively. The document has to provide at least the following information: the purpose of the ontology (including its intended uses, scenarios of use, end-users, etc.); the level of formality used to codify terms and meanings (highly informal, semi-informal, semi-formal, rigorously formal ontologies); the scope; its characteristics and granularity. Properties of this document are: concision, partial completeness, coverage of terms, the stopover problem and level of granularity of ache and every term, and consistency of all terms and their meanings.
Conceptualization	A good specification document	Conceptualize in a model that describes the problem and its solution. To identify and gather all the useful and potentially usable domain knowledge and its meanings	A complete glossary of terms (including concepts, instances, verbs, and properties). Then, a set of intermediate representations such as concepts classification trees, verb diagram, table of formulas, and table of rules. The aim is to allow the final user to ascertain whether or not an ontology is useful and to compare the scope and completeness of several ontologies, their reusability, and shareability.
Formalization	Conceptual model	Transform conceptual model into a formal or semi-compatible model, using frame-oriented or description logic representation systems	Formal conceptualization
Integration	Existing ontologies and the formal model	Processes of inclusion, polymorphic refinement, circular dependencies, and restriction. For example, select meta ontologies that better fit the conceptualisation	
Implementation	Formal model	Select target language	Create a computable ontology
Maintenance		Including, modifying definition in the ontology	Guidelines for maintaining ontologies
Acquisition		Searching and listing knowledge sources through non-structured interviews with experts, informal text analysis, formal text analysis, structured interviews with experts to have detailed information on concepts, terms, meanings, and so on	A list of the sources of knowledge and a rough description of how the process will be carried out and what techniques will be used.
Evaluation	Computable ontology	Technical judgment with respect to a frame of reference	A formal and correct ontology
Documentation			Specification document must have the property of concision

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

and documentation with respect to a frame of reference;
- integration;
- documentation.

In Table 3, we provide a listing of the single phases of the previously mentioned methodology. It is also worth depicting one fundamental fact about methontology: its specific explicit avoidance to commit to the user's interest in the development of tools that automate all the phases.

A Comprehensive Framework for Multilingual Domain Ontologies

The authors (Lauser et al., 2002) use the Methontology methodology defined by Fernández, Gòmez-Pérez, & Juristo (1997) and stress the specific actions for supporting the creation process for ontology-driven conceptual analysis.

The domain ontology is built using two different knowledge acquisition approaches (see Table 4):

- **acquisition approach 1:** creation of the core ontology. A small core ontology with the most important domain concepts and their relationships is created from scratch. This stage is comprised basically of the first three steps of Methontology development activities: requirement specification, conceptualization of domain knowledge, and formalization of the conceptual model in a formal language;
- **acquisition approach 2:** deriving a domain ontology from thesaurus;
- **ontology merging:** merging the manually created core ontology and the derived ontology using thesaurus terms;
- **ontology refinements and extension:** the frequent domain terms are used as possible candidate concepts or relationships for extending the ontology. These terms have to be assessed by subject specialists and checked for relevance to the ontology.

TOVE Methodology

Toronto Virtual Enterprise (TOVE) is a methodology for ontological engineering that allows the developer to build ontology following these steps:

- **motivating scenarios:** the starting point is the definition of a set of problems encountered in a particular enterprise;
- **informal competency questions:** based on the motivating scenario, it is the definition of ontology requirements described as informal questions that an ontology must be able to answer;
- **terminology specification:** the objects, attributes, and relations of the ontology are formally specified (usually first-order logic);
- **formal competency question:** the requirements of the ontology are formalized in terms of the formally defined terminology
- **axiom specification:** axioms that specify the definition of terms and constraints on their interpretations are given in first-order logic;

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Table 4. Multilingua domain methodology

Name of the Phase	Input	Description	Output
Acquisition approach 1	Nothing: first step	The creation of the core ontology. A small core ontology with the most important domain concepts and their relationships is created from scratch.	The goal of this step is to define a list of frequent terms and a list of domain specific documents to analyze.
Acquisition approach 2	Nothing: first step	Deriving a domain ontology from thesaurus. A thesaurus consists of descriptive keywords linked by a basic set of relationships. The keywords are descriptive in terms of the domain in which they are used. The relationships either may describe a hierarchical relation or an interhierarchical relation.	The goal of this step is to refine an RDFS ontology model to develop a pruned ontology and a list of frequent terms.
Ontology merging	Manually created core ontology and thesaurus terms	Merging the manually created core ontology and the derived ontology using thesaurus terms.	A first version of the ontology
Ontology refinements and extension	The first version of the ontology	The frequent domain terms are used as possible candidate concepts or relationships for extending the ontology. These terms have to be assessed by subject specialists and checked for relevance to the ontology	The final ontology version

- **completeness theorems:** an evaluation stage that assesses the competency of the ontology by defining the conditions under which the solutions to the competency question are complete.

The most distinctive aspect of TOVE is the focus on maintenance using formal techniques to address a limited number of maintenance issues (Gruninger, Atefi, & Fox, 2000; Kim, Fox, & Gruninger, 1999). According to our framework, this methodology can be summarized as shown in Table 5.

DILIGENT Methodology

The DILIGENT methodology (Tempich et al., 2004) aims at creating a set of effective ontologies that the user can share and, at the same time, expand for local use at their will and individual needs. The aim of this methodology is to overcome the misunderstanding of ontologies that are created by a very small group of people (the ontology engineers and the domain experts who represent the users) but are utilized by a huge number of users. The authors postulate that:

> Ontology engineering must take place in a Distributed evolvInG and Loosely-controlled setting.
>
> (Tempich et al., 2004, p.)

With DILIGENT, the authors provide a process template for distributed engineering of knowledge structures and intend to extend it toward a fully worked out and multiply-tested methodology. Ana-

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Table 5. TOVE methodology

Name of the Phase	Input	Description	Output
Motivating scenario	Nothing: first step	It's the starting point, the definition of a set of problems encountered in a particular enterprise	Identify intuitively possible applications and solutions.
Informal competency questions	The motivating scenario	It is the definition of ontology requirements described as informal questions that an ontology must be able to answer	Questions: terminology Answers: axioms and formal definitions
Terminology specification	Informal questions	The objects, attributes, and relations of the ontology are formally specified (usually first-order logic)	Objects: constants and variables Attributes and relations: functions and predicates
Formal competency question	Objects, attributes, and relations	The requirements of the ontology are formalized in terms of the formally defined terminology	Formally defined terminology
Axiom specification	Formally defined terminology	Axioms that specify the definition of terms and constraints on their interpretations are given in first-order logic;	First-order sentences, using predicates of ontology
Completeness theorems	First-order sentences	An evaluation stage that assesses the competency of the ontology by defining the conditions under which the solutions to the competency question are complete.	Conditions under which the solutions to the competency question are complete

lyzing DILIGENT, we can say that it is based on dominating roles and five high-level phases that are described in Table 6. The key roles are several experts with different and complementary competencies, involved in collaboratively building the same ontology.

Business Object Ontology

The authors Izumy and Yamaguchi (2002) have used this methodology to develop an ontology for business coordination. They constructed the business activity repository by employing WordNet as a general lexical repository. They have constructed the business object ontology in the following way:

- concentrating the case-study models of e-business and extracting the taxonomy;
- counting the number of appearances of each noun concept;
- comparing the noun hierarchy of WordNEt and the taxonomy obtained and adding the number counted for the similar concepts;
- choosing the main concept with high scores as upper concepts and building upper ontologies by giving all the nouns the formal *is-a* relation;
- merging all the noun hierarchies extracted from the whole process.

Although this methodology is quite general, it is important to notice that the continuous interdependencies among busi-

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Table 6. DILIGENT methodology

Name of the Phase	Input	Description	Output
Build	Domain experts, users, knowledge engineers, and ontology engineers	The definition of an initial ontology. The team involved in building the initial ontology should be relatively small in order to find more easily a small and consensual first version of the shared ontology	Built an initial ontology. It is not required completeness of the initial shared ontology with respect to the domain.
Local adaptation	Core ontology	Users work in the core ontology and adapt it to their local needs, business, and system of artefacts.	Logging local adaptation, the control board collects change requests to the shared ontology
Analysis	Local ontologies and the local requests	The board analyzes the local ontologies and tries to identify similarities in users' ontologies. One of the crucial activities of the board is to decide which changes should be introduced in the shared ontology	A refined version of the shared ontology, users' changes and board evaluations.
Revision	Shared ontology and local ontologies. The board should have well-balanced and representative participation of the different kinds of participants involved in the process.	The board should regularly revise the shared and local ontologies. The board should evaluate the ontology, mostly from a technical and domain point of view. The board should assure some compatibility with previous versions.	Final revision of the shared ontology, which entails its evolution
Local update	New version of the shared ontology	Users can update their own local ontologies to better suit the new version of the shared ontology for their local needs.	New locally defined ontologies represent new concepts and criteria with new meaning. The locally defined ontologies constitute new versions of the locally shared ontology.

ness people, industries, and case studies are the key factors in an effective ontology.

A Natural Language Interface Generator (GISE)

In Gatius and Rodríguez (1996), the authors developed three process steps to build a domain ontology:

- **first step:** building and maintenance of:
  - **the general linguistic knowledge:** It includes linguistic ontology that covers the syntactic and semantic information needed to generate the specific grammars and a general lexicon that includes functional, domain, and application independent lexical entries;
  - **the general conceptual knowledge:** it includes the domain and application-independent conceptual information as well as the meta-knowledge that will be needed in the following steps;
- **second step:** definition of the application in terms of the conceptual ontology. Both the domain description and the task structure description must be

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

built and linked to the appropriate components of the conceptual ontology;

- **third step:** definition of the control structure. It includes:
  - the meta-rules for mapping objects in the domain ontology with those in the task ontology;
  - the meta-rules for mapping the conceptual ontology onto the linguistic ontology and those for allowing the generation of the specific interface knowledge sources, mainly the grammar and the lexicon.

This methodology seems to be not specific enough to analyze it according to our framework. In any case, the most important point of this methodology is that it builds and maintains the general linguistic knowledge and conceptual ontology. This work is devoted to providing a mechanism based on the performance of a set of control rules relating general linguistic knowledge to the applications specifications in order to obtain automatically the natural language best suited for each application. The representation of the relevant knowledge in the process of generating an interface in separate data structures (conceptual ontology, linguistic ontology, general lexicon, and control rules) gives great flexibility in adapting linguistic resources to different applications and users.

SOME CONSIDERATION COMPARING THESE METHODOLOGIES

Although there are considerable differences between the methodologies previously described, a number of points clearly emerge:

- many of the methodologies take a specific task as a starting point: choosing domains and categories that allow the correct representation. From one point of view, it focuses on the acquisition, provides the potential for evaluation, and provides a useful description of the capabilities of the ontology, expressed as the ability to answer well-defined competency questions. On the other hand, it seems to provide limitations to the reuse of the ontology and to the possible interactions among ontologies;
- there are two different types of methodology models: the stage-based models (represented, for example, by TOVE) and evolving prototype models (represented by Methontology). Both approaches have advantages and disadvantages: the first one seems more appropriate when purposes and requirements of the ontology are clear; the second one is more useful when the environment is dynamic and difficult to understand;
- most of the time, there are both informal descriptions of the ontology and formal embodiment in an ontology language. These often are developed in separate stages, and this separation increases the gap between real world models and executable systems.

The common point in these methodologies is the starting point for creating an ontology that could arise from different situations (Ushold, 2000; Ushold & Gruninger, 1996):

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

- from scratch;
- from existing ontologies (whether global or local);
- from corpus of information sources only;
- a combination of the latter two approaches.

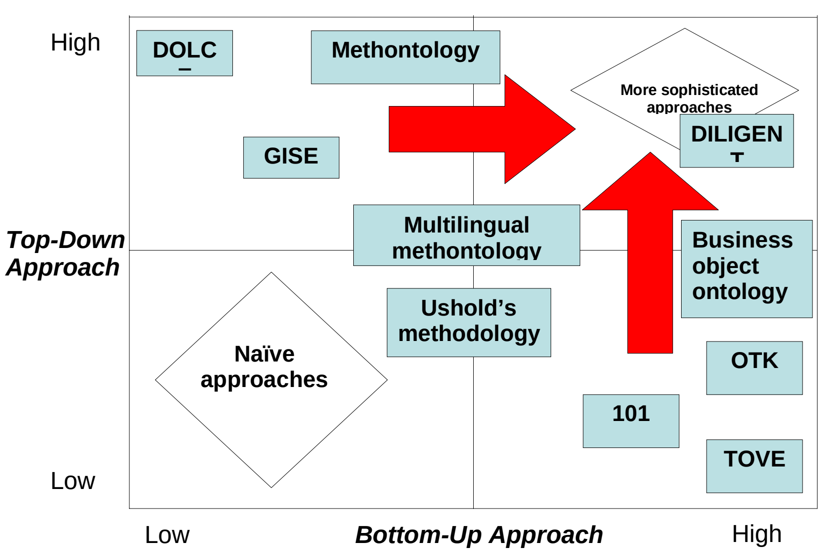
Normally, methods to generate ontology could be summarized (Ding & Foo, 2002):

- bottom-up: from specification to generalization;
- op-down: from generalization to specification such as KACTUS ontology;
- middle-out: from the most important concepts to generalization and specialization such as Enterprise ontology and Methondology.

Picture 1 shows how different ontologies deal with both the top-down approach and the bottom-up approach.

As depicted in Picture 1, on one side, the Naïve approaches present a low level of quality in both approaches and are not taken into account in this analysis. Although different technology-driven methodologies are used, new methodologies and tools are needed for creating and managing local schemata. These methodologies and tools should allow both the creation of a schemata from scratch (analyzing documents, repeated occurrences within databases, etc.) and the management of sense-making processes on concepts. This framework is called in the picture "more sophisticated approaches." These should provide guidelines to assist the knowledge owner in

Picture 1. Methodologies according the top down and bottom up approach



Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

making choices at different levels, from the high-level structure of the ontology to the fine detail of whether or not to include distinctions. These frameworks should aim at satisfying all the knowledge the owner needs and merging two different requirements, a more adequate representation of a local domain and a very effective development of a top-level ontology. In this study, the authors would consider that manually constructed ontologies are time-consuming, labor intensive, and error-prone (Ding & Foo, 2002) but are necessary to define a domain in which the quality and general comprehension of the ontology are good.

There are also a number of general ontology design principles that are proposed:

- Guarino, Borgo, Masolo, Gangemi, and Oltramari (2002) proposed a methodology to design a formal ontology, in particular, defining domain, identifying a basic taxonomic structure, and pointing to a role;
- Uschold and Gruninger (1996) proposed a skeletal methodology for building ontologies via a purely manual process: identify purpose and scope, ontology capture (identification of key concepts and relationships and provision of definitions), ontology coding (committing to the basic terms for ontology), language, integrating existing ontologies, evaluation, documentation, guidelines;
- Other researchers separate the construction and definition of complex expression from its representation;

Other authors proposed a number of desirable criteria for the final generated ontology to be open and dynamic, scalable and interoperable, easily maintained, and context independent.

Finally, there is no one correct way to model a domain; there are always viable alternatives. Most of the time, the best solution depends on the application that the developer has in mind and the tools that the developer uses to develop the ontology. In particular, we can notice some emerging problems (Blázquez et al., 1998; Gómez-Pérez, 2001):

- the correspondence between existing methodologies for building ontologies and environments for building ontologies causes this consequences:
  - conceptual models are implicit in the implementation codes, and a reengineering process usually is required to make the conceptual models explicit;
  - ontological commitments and design criteria are implicit in the ontology code;
  - ontology developer preferences in a given language condition the implementation of the acquired knowledge. So when people code ontologies directly in a target language, they are omitting the minimal encoding bias criterion defined by Gruber (1993);
- most of the tools only give support for designing and implementing the ontologies, but they do not support all the activities of the ontology life-cycle;

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

- ontology developers may find it difficult to understand implemented ontologies or even to build a new ontology.

ACKNOWLEDGMENTS

The first author gratefully thanks the European Union under Grant GRD1-2001-40739 (the UPTUN Project) for supporting the investigation that we have documented in this chapter.

The UPTUN-project, "Cost-effective, sustainable and innovative upgrading methods for fire safety in existing tunnels" is being carried out in the framework of the "Competitive and Sustainable Growth Programme" (project GRD1-2001-40739, Contract G1RD-CT-2002-0766), with a financial contribution of the European Community.

We also both gratefully thank Creative Consulting s.r.l. for the grant OASI (Ontologies of Artefacts and Services to Industry).

REFERENCES

Ashby, W.R. (1956). *An introduction to cybernetics*. New York: John Wiley & Sons.

Benerecetti, M., Bouquet, P., & Ghidini, C. (2000). Contextual reasoning distilled. *Journal of Theoretical and Experimental Artificial Intelligence, 12*(3), 279-305.

Blázquez, M., Fernández, M., García-Pinar, J.M., & Gómez-Pérez, A. (1998). Building ontologies at the knowledge level using the ontology design environment. *Proceedings of the KAW'98*, Banff, Canada.

Bonifacio, M., Bouquet, P., & Cuel, R. (2002). The role of classification(s) in distributed knowledge management. *Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies (KES 2002)*. Podere d'Ombriano, Crema, Italy.

Chandler, A.D. (1962). *Strategy and structure: Chapters in the history of the industrial enterprise*. Cambridge, MA: MIT Press.

Cocchiarella, N.B. (1991). Formal ontology. In H. Burkhardt, & B. Smith (Eds.), *Handbook of metaphysics and ontology*. Munich, Germany: Philosophia Verlag.

Ding, Y., & Foo, S. (2002). Ontology research and development, Part 1—A review of ontology generation. *Journal of Information Science, 3*(28), 123-136.

Dougherty, D. (1992). Interpretative barriers to successful product innovation in large firms. *Organization Science, 3*(2).

Fauconnier, G. (1985). *Mental spaces: Aspects of meaning construction in natural language*. MIT Press.

Fensel, D. (2000). *Ontologies: A silver bullet for knowledge management and electronic commerce*. Springer.

Fensel, D., van Harmelen, M., Klein, & Akkermans, H. (2000). On-to-knowledge: Ontology-based tools for knowledge management. *Proceedings of eBusiness and eWork*, Madrid, Spain.

Fernández, M., Gòmez-Pérez, A., & Juristo, N. (1997). METHONTOLOGY: From ontological art to-

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

wards ontological engineering. *Proceedings of the AAAI Spring Symposium on Ontological Engineering*, Stanford, California.

Gangemi, A., Pisanelli, D.M. & Steve, G. (1998). Ontology integration: Experiences with medical terminologies. In N. Guarino (Ed.), *Formal ontology in information systems* (pp. 163-178). IOS Press.

Gatius, M., & Rodríguez, H. (1996). A domain-restricted task-guided natural language interface generator. *Proceedings of the Second Edition of the Workshop Flexible Query Answering Systems (FQAS'96),* Denmark.

Ghidini, C., & Giunchiglia, F. (2001). Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence, 127*(2), 221-259.

Goffman, I. (1974). *Frame analysis*. New York: Harper & Row.

Gómez-Pérez, A. (2001). A proposal of infrastructural needs on the framework of the semantic Web for ontology construction and use. *Proceedings of the FP6 Programme Consultation Meeting 9*.

Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199-220.

Gruninger, M., Atefi, K., & Fox, M.S. (2000). Ontologies to support process integration in enterprise engineering. *Computational and Mathematical Organization Theory, 6*(4), 381-394.

Hamel, C., & Prahalad, G. (1990, May-June). The core competences of the corporation. *Harvard Business Review*.

Isumi, N., & Yamaguchi, T. (2002). Semantic coordination of Web services based on multi-layered repository. *PRICAI*, 597.

Jasper, R., & Ushold, M. (1999). A framework for understanding and classifying ontology applications. *Proceedings of the 12th Workshop on Knowledge Acquisition, Modelling, and Management*, Ban®, Canada.

Kim, H.M., Fox, M.S., & Gruninger, M. (1999). An ontology for quality management—Enabling quality problem identification and tracing. *BT Technology Journal, 17*(4), 131-140.

Kuhn, T. (1979). *The structure of scientific revolutions*. Chicago, IL: University of Chicago Press.

Lau, T., & Sure, Y. (2002). Introducing ontology-based skills management at a language insurance company. *Modellierung*, 123-134.

Lauser, B., et al. (2002). A comprehensive framework for building multilingual domain ontologies: Creating a prototype biosecurity ontology. *Proceedings of the Internatiional Conference on Dublin Core and Metadata for e-Communities*.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., & Oltramari, A. (2002). Wonderweb deliverable d17. *Intermediate Report 2.0, ISTC-CNR*.

Maturana, H.R., & Varala, F.J. (1980). *Autopoiesis and cognition: The realization of the living*. Dordrecht: D. Reidel.

Noy, N.F., & McGuinnes, D.L. (2001). *Ontology development 101: A guide to creating your first ontology*.

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

Numagami, T., Ohta, T., & Nonaka, I. (1989). *Self-renewal of corporate organizations: Equilibrium, self-sustaining, and self-renewing models* [working paper]. Berkeley, CA: University of California at Berkeley.

Sowa, J.F. (2000). *Knowledge representation. Logical, philosophical and computational foundations*. Brooks/Cole.

Sure, Y. (2003). Methodology, *tools and case studies for ontology based knowledge management* [doctoral thesis]. University of Karlsruhe, Department of Economics and Business Engineering.

Sure, Y., Staab, S., & Angele, J. (2002). OntoEdit: Guiding ontology development by methodology and inferencing. *Proceedings of the Confederated International Conferences CoopIS, DOA and ODBASE 2002*, Irvine, California.

Tempich, C., Pinto, S., Staab, S., & Sure, Y. (2004). A case study in supporting distributed, loosely-controlled and evolving engingeering of ontologies (DILIGENT). *Proceedings of the 4th International Conference on Knowledge Management (I-Know'04)*, Graz, Austria.

Ushold, M. (2000). Creating, integrating and maintaining local and global ontologies. *Proceedings of the First Workshop on Ontology Learning (OL-2000) in conjunction with the 14th European Conference on Artificial Intelligence (ECAI 2000).*

Ushold, M., & Gruninger, M. (1996). Ontologies: Principles, methods, and applications. *Knowledge Engineering Review, 11*(2), 93-155.

Weick, E.K. (1979). *The social psychology of organizing*. McGraw-Hill.

Please provide bios

Copyright © 2005, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.