

# Chapter 5

## Ontology Design Approaches

### 5.1 Introduction

In Chapter 3 we discussed the concept of ontology and its characteristics, and the numerous advantages of ontologies. We may feel inspired to design an ontology for our information system but we still do not know: where to start, how to prepare ourselves, how many people are needed for the ontology design, how we ascertain that we are on the right track, whether similar ontologies already exist, how to determine that we are designing a valid ontology, what tool we can use to assist us with our design, and other similar issues.

There is not a single, consensual ontology-design methodology. Many different ontology design methodologies have been proposed. We will consider some of them in order to illustrate different approaches to ontology design. The methodologies presented are different from each other and together cover various aspects of the ontology design process.

### 5.2 Ontology Design Criteria

A set of criteria has proven to be useful as a guiding tool during the ontology design process. These principles have also been used to evaluate ontology design. Gruber suggests five design criteria for ontologies (Gruber 1995):

- 1) Clarity
- 2) Coherence
- 3) Extendibility
- 4) Minimal Encoding Bias
- 5) Minimal Ontological Commitment

**Clarity.** We need to be clear about our reasons for choosing a particular set of ontology terms and the intended meanings of those terms. The number of possible interpretations of a term needs to be restricted. Complete definitions of ontology terms, which are defined by necessary and sufficient conditions, are preferred over partial definitions of ontology terms, which are defined by necessary or sufficient conditions. The definitions need to be objective, documented in natural language but stated in formal axioms within the ontology.

Clear definitions of ontology terms will contribute to the effectiveness of communication between agents. Take the example presented in Figure 3.3 where we gave four different definitions of a gene. The definition that “a gene is situated on a chromosome” is incomplete because there are some other parts of chromosomes

that are not genes but have a regulatory function; for instance, promoters are situated on a chromosome and are responsible for the activation or repression of a particular gene. The condition that “a gene is situated on a chromosome” is necessary but not sufficient. If agent A gives information about promoters to agent B who thinks that everything situated on DNA is a gene, the agent B will conclude that the promoter is a new gene. It is difficult and sometimes impossible for different agents to reach an agreement if they have different understandings of the terms used in conversation.

**Coherence.** Propositions that can be inferred from the ontology definitions and axioms may not contradict other ontology definitions and axioms. Inferences need to be consistent with the existing definitions and axioms, and should be clear and logical. In the definition “gene is situated on a chromosome”, it has not been clearly stated if the gene is part of the chromosome sequence or whether this chromosome is used to carry genes on its surface. An agent could ‘infer’ from this definition that “a gene is superimposed on a chromosome structure”. We know that a gene is a part of a chromosome sequence and for domain experts this inferred statement may appear to be strange. But the knowledge that an agent holds is limited to the knowledge given to it and we need to be aware of this fact when defining ontology terms.

**Extendibility.** As new knowledge emerges each day, this information may need to be added to an already existing ontology. For this reason, the ontology needs to be easily extendable. The new ontology terms should be easily defined based on the existing vocabulary and/or preferably without the need to alter the existing ontology definitions.

Extending of an ontology may result in a new ontology that will contain more precise and correct definitions. Let us say that an ontology contains the definition “a gene is part of chromosome sequence”. When the new knowledge emerges that implies the presence of promoters within the chromosome sequence and when this knowledge needs to be added to the existing ontology, it will point to the mistakes in the definition of the term “gene”. The existing definition needs to be corrected in order to allow the new definition of the term “promoter” to be incorporated in the ontology. The resulting ontology will not only be extended but it will also be more precise.

**Minimal Encoding Bias.** Conceptualizations need to be specified at a knowledge-level and not at a symbol-level. The goal of the ontology is to represent true facts, even when it seems to be more convenient to compromise on ontology correctness. In our example of gene definition, it may appear to be more convenient to use the expression “positioned on” instead of “part of”, but this may be at the cost of much time and energy when we come to the implementation phase and our agent needs to communicate with other agents. We should not compromise the correctness of an ontology for the convenience of its notation or implementation.

**Minimal Ontological Commitment.** More commitments in the ontology make the ontology structure more rigid and limit the number of ontology users. Ontological commitment needs to be minimal but sufficient to support the intended knowledge sharing activities. For this purpose, it may be needed to specify an

ontology which is necessary for effective communication consistent with the domain conceptualization and easily extendable by individual agents.

All the agents can communicate effectively and efficiently between each other on the basis of the ‘necessary’ ontology. In our example of gene definitions, two agents need to find a shared understanding of the term “gene” and the ‘necessary’ ontology may contain the definition “gene codes for a protein”. Agents talk to each other on a higher conceptual level and for this they need the ‘necessary’ ontology. Each of these agents has certain tasks assigned to it. For this reason, the agents need to extend the ‘necessary’ ontology to a ‘specialized’ ontology. Suppose that the tasks of agent A are more gene-oriented and those of agent B are more protein-oriented. Agent A may need to extend this definition with the statement that “a gene is part of chromosome” and agent B with “protein is composed of amino acids”. The agents need to be able to extend the common ontology where needed for their individual purposes in order to perform their individual tasks efficiently and effectively.

Cooperative work between knowledge and software engineers, ontology designers and domain experts is required for the design and development of ontology-based information systems.

### **5.3 Ontology Design Methodologies**

In this section, we will discuss six different Ontology Design Methodologies, namely:

- 1) Knowledge Engineering Methodology
- 2) DOGMA Methodology
- 3) TOVE Methodology
- 4) Methontology
- 5) SENSUS Methodology
- 6) DILIGENT Methodology

#### ***5.3.1 Knowledge Engineering Methodology***

Knowledge engineering methodology (Uschold and Gruninger 1996) for designing and maintaining ontologies is identified as a six-stage process, as shown in Figure 5.1, namely:

- 1) definition of the domain, purpose and scope of ontology
- 2) acquisition and conceptualization of the domain knowledge
- 3) reusing of existing ontologies
- 4) formal specification of the ontology
- 5) population of the ontology with individual instances
- 6) evaluation and documentation

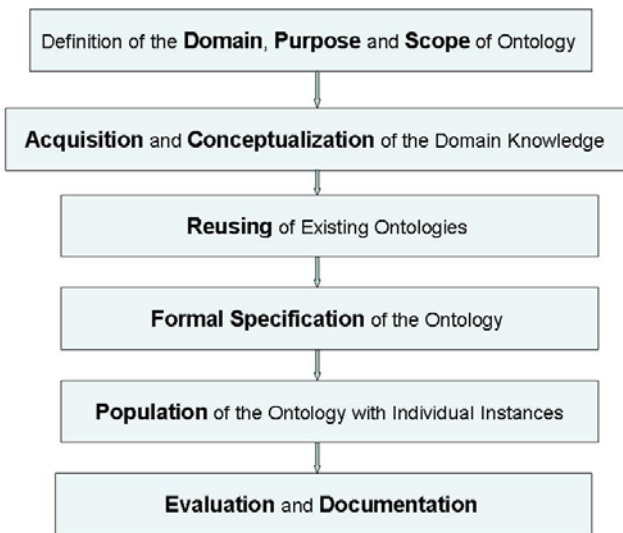
1. **Definition of the Domain, Purpose and Scope of Ontology.** Domain problem, purpose and scope of the ontology is defined in this stage. This stage involves description of the ontology based on the use of a motivating scenario.

A set of questions is used in order to facilitate the tasks associated with this phase (Noy and McGuinness 2001), such as:

- 1) What domain will the ontology cover?
- 2) For what will the ontology be used?
- 3) For what types of questions should the ontology provide answers?
- 4) Who will use and maintain the ontology?

The output of this phase is a natural language ontology specification document which supports the ontology design process. This document contains information such as the purpose of the ontology within a community, ontology scope, use case scenarios, degree of formality used to codify the ontology and other similar matters. An ontology specification document must include relevant terms without duplication, and the meanings of these terms and their relationships need to make sense in the domain, and coincide with their meaning in the real world.

For example, a genetics institute wants to design an intelligent information retrieval system that will look for functions and characteristics of a particular DNA sequence. A DNA ontology needs to be designed to support the intelligent searches performed by the information retrieval system. Does the system need the full DNA sequence in order to perform its searches, or does it only look for the numerically represented positions of the sequence? Those and similar kinds of ontology details and ontology scope will depend on how the ontology is intended to be used within the system and on the system requirements.



**Fig. 5.1** Knowledge engineering methodology

2. Acquisition and Conceptualization of the Domain Knowledge. This stage involves the process of acquiring knowledge from a given domain. Here, it is important to establish cooperation between the ontology engineer and the domain experts. Different methods for collecting and analyzing the data may be applied here such as domain expert interviews, questionnaires and text analysis of relevant documents.

The output of this step is an informal ontology. The informal ontology is a conceptualized domain model that represents domain concepts and the relationships between them. The informal ontology contains a glossary of terms (concepts), definition of the relationships between the terms and constraints on their usage.

Different conceptual modelling techniques can be used to represent the ontology. A suggested conceptual modelling approach consists of the following two steps:

- (1) make a list of domain terms and their meaning

The domain terms need to be identified and their properties described. A set of tables can be created to support the collection of information about domain terms. Domain concepts can be identified from the collected information sources and related data interpreted and analyzed for their significance in the ontology design. The identification of the relevant domain terms is crucial for the ontology design process.

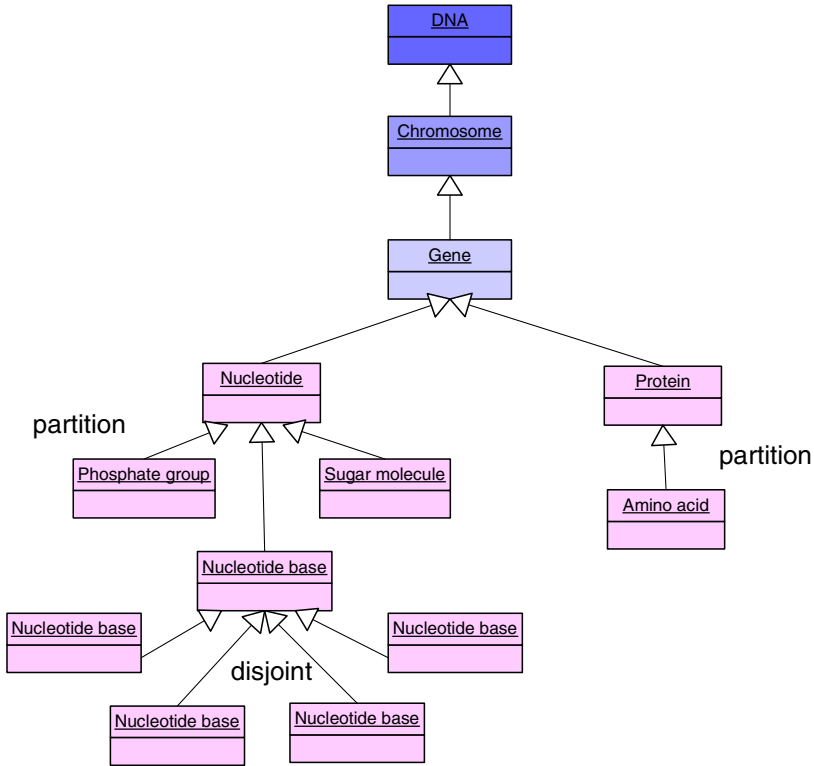
In our example, the DNA ontology may consist of the following terms: DNA, chromosome, sequence, gene, nucleotide, nucleotide base, sugar, phosphate group, protein, amino acid.

- (2) classify the concepts

Concept classification includes hierarchical conceptual structures, conceptual graphs, semantic nets and ontologies. The quality of the applied classification schema affects the effectiveness of the ontology design. Top-down and bottom-up approaches can be applied for the categorization and classification of domain concepts (Nakata 2001).

A top-down concept classification includes specification of all possible subconcepts of a concept. This approach is usually applied with the help of domain experts. Depending on the richness of the domain knowledge required in the analysis, specific limits on the granularity for this classification need to be introduced. For example, for our DNA ontology, do we need to say that the DNA consists of a sequence of nucleotides, or are we going to go a step further and add that each nucleotide consists of a sugar molecule, phosphate group and one of the four nucleotide bases (Adenine, Cytosine, Guanine and Thymine)? The choice will depend on the richness required by the ontology model.

The bottom-up concept classification includes identifying concepts in the documents. This includes analyzing terms, phrases, and segments of text and assigning them to a concept or group of concepts. The bottom-up concept classification is applied in situations such as analysis of the created and collected documents, interview results and questionnaire responses. Collected concepts are then grouped in a hierarchy, from which the ontology may begin to emerge.



**Fig. 5.2** Hierarchy of the DNA ontology

The abovementioned concepts can be represented by the hierarchy shown in Figure 5.2. DNA, chromosomes and genes are all sequences of nucleotides. Nucleotides consist of a phosphate group, a sugar molecule and a nucleotide base (Adenine, Cytosine, Guanine or Thymine). A gene is a part of a chromosome and a chromosome is part of a DNA. A gene codes for a protein which is a sequence of amino acids.

3. Reuse of Existing Ontologies. This stage focuses on the process of building a new ontology by reusing and adapting ontological terms from other published and consensual ontologies. Usually, a set of sharable and reusable ontologies is used during this process. The goal is to obtain some uniformity across different ontologies.

Two main processes associated with the reuse of existing ontologies are ontology merging and ontology alignment. The difference between these two processes is that the merging of different ontologies results in a single coherent ontology, while the aligning of different ontologies establishes links between the concepts of the aligned ontologies, allowing them to reuse information from one another. These processes both take as input two (or more) source ontologies and return, in

the case of ontology merging, a single merged ontology or, in the case of ontology alignment, a cluster of ontologies where the two (or more) original ontologies persist with links established between them. Usually, in ontology merging, the original ontologies cover similar or overlapping domains while in ontology alignment the original ontologies cover domains that are complementary to each other.

In our example from Figure 5.2, we may have two different ontologies: the first one is on the left hand side of the figure which consists of the terms: “DNA”, “chromosome”, “gene”, “nucleotide”, “phosphate group”, “sugar molecule”, “nucleotide base”, “Adenine”, “Cytosine”, “Guanine” and “Thymine” while the other ontology is on the right hand side of the figure and consist of the terms “protein” and “amino acid”. In this case, the joining point is between the terms “gene” and “protein: gene codes for a protein”. We can merge those two ontologies into a single ontology, or we can align them but still represent them as two separate ontologies with an indication of links between them.

4. Formal Specification of the Ontology. Ontologies need to be formally specified using a formal representation language. The output of the two previous steps is an informal ontology written in natural language that can be understood by humans, but not by computers. The information represented by this ontology needs to be translated into a formal language. This enables machines to effectively use the ontologies; the machines can ‘read’, ‘understand’ and ‘act’ upon this information. This is one of the main reasons that the concept of ontology was introduced into the domain of computer and information science. The choice of the formal representation language depends on the way the system needs to be implemented and/or integrated with the existing system. Currently, most of the ontologies are written in OWL.

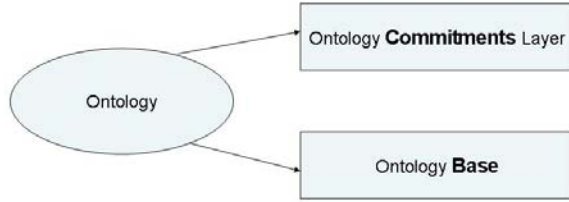
5. Population of the Ontology with Individual Instances. Ontology consistency is experimentally analyzed in this step. An ontology design tool, such as Protégé, can be used in this step. Defining an individual instance of a class requires:

- 1) Identification of a class to which this instance belongs
- 2) Creation of an individual instance of this class
- 3) Determination of slot attributes with the proper values for this instance

6. Evaluation and Documentation. The ontology is evaluated in respect to ontology consistency and completeness (Gómez-Pérez et al. 1996), as mentioned in Section 3.2. The terms in the ontology need to be clearly defined, the terminology needs to be coherent, and the relationships between the terms logically consistent. Some inconsistencies and redundancies may emerge during the implementation process. For this reason, ontology (re)modification follows the implementation process in most cases.

Each of the abovementioned methodology steps must be documented and explicitly written.

**Fig. 5.3** DOGMA methodology



### 5.3.2 DOGMA

Whether or not axioms should form part of an ontology is one of the main discussion topics in regard to ontology definition (Pisanelli et al. 2002). For some, an ontology is a domain description that organizes the concepts into a specific structure e.g. hierarchy; for others, an ontology is a complete theory consisting of a formal vocabulary as well as well-defined axioms that allow further deductions or inferences to be made in regard to the knowledge the ontology represents.

One way to address the above issue is to adopt an approach that neatly separates the specification of ontology concepts from their axioms. The DOGMA (Developing Ontology-Guided Mediations of Agents) framework has been developed at the VUB STARLab (Vrije Universiteit Brussel Semantic Technology and Applications Research Laboratory). The DOGMA approach is based on the principle of a double articulation (De Bo et al. 2003). An ontology designed using the DOGMA double articulation approach consists of two layers: ontology base and commitment layer, shown in Figure 5.3.

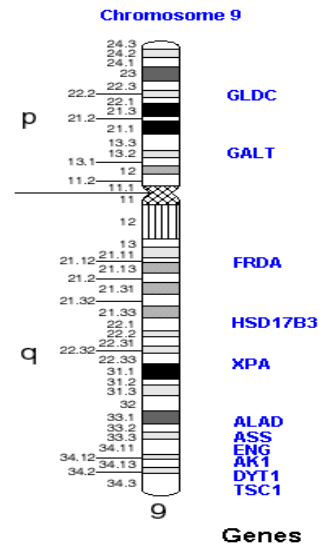
1. **Ontology Base.** The ontology base holds the conceptualization of a domain. Namely, ontology concepts and relationships between these concepts are formally defined in the ontology base. The ontology base is defined in such a way that it enables correct contextual identification of these concepts and the associated relationships.

According to the DOGMA methodology, an ontology base is a set of context-specific binary fact types, called lexons (notation:  $\langle \mu, t1, r, cr, t2 \rangle$ ). Here  $\mu \in \Delta$  is an abstract context identifier chosen from a set  $\Delta$ ;  $t1, r, cr, t2$  are term1, role, co-role and term2 respectively. The lexical terms ( $t1, r, cr, t2$ ) are constructed from a given alphabet. For each  $\mu \in \Delta$  and each term  $t$  occurring in a lexon, the pair  $(\mu, t)$  specifies exactly a unique concept.

In our example, we may have the following lexon of the form  $\langle \mu, t1, r, cr, t2 \rangle$ :

$\langle \text{DNA}, \text{gene}, \text{is part of}, \text{contain}, \text{chromosome} \rangle$ . Here, “DNA” corresponds to context identifier  $\mu$ , “gene” corresponds to term  $t1$ , “is part of” corresponds to role  $r$ , “contain” corresponds to co-role  $cr$  and “chromosome” corresponds to term  $t2$ . This statement means that in the DNA context, “gene is a part of a chromosome” and “chromosome contain(s) gene(s)”. An example shown in Figure 5.4, specifies that genes GLDC, GALT, FRDA, HSD17B3 etc. are found on chromosome 9 (out of 22 human chromosomes, and X and Y sex chromosomes).



**Fig. 5.4** Genes on chromosome 9

2. Commitment Layer. Each commitment within the commitment layer contains:

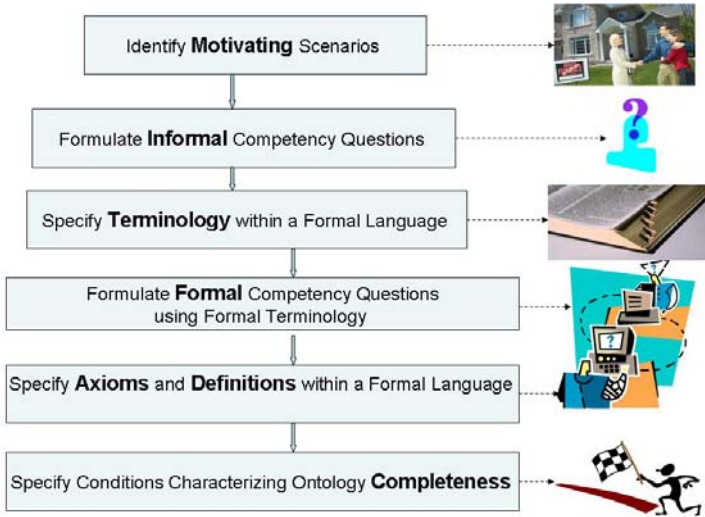
- 1) a set of constraints, derivation and domain rules applied to a specified subset of the ontology base (Jarrar et al.2003), and
- 2) a set of mappings between ontological elements and application elements (Deray and Verheyden 2003).

In our example, a commitment is “each chromosome contains at least one gene”. The corresponding statement from the ontology base, “chromosome contain(s) gene(s)”, has been more precisely described by this commitment.

An agent or application that operates in the ontology domain is able to further specialize the ontology base through modifications to its commitment layer. An agent may, or may not, require the presence of a specific commitment within its ontology. This depends on the agent’s function within the system. The ontology base plus the agent’s commitment to a part of it, is the real world to the application agents.

### 5.3.3 TOVE Methodology

The TOVE (Toronto Virtual Enterprise) (Gruninger and Fox 1995) project required the design of an ontology for the business processes and activities domain. Experiences gained during the development of this ontology were used as the basis for the development of a novel ontology design methodology, called the TOVE methodology. The TOVE methodology consists of the following six steps, shown in Figure 5.5.



**Fig. 5.5** TOVE methodology

- 1) Identify motivating scenarios
- 2) Formulate informal competency questions
- 3) Specify terminology within a formal language
- 4) Formulate formal competency questions using formal terminology
- 5) Specify axioms and definitions within a formal language
- 6) Specify conditions characterizing ontology completeness

1. Identify motivating scenarios. Various scenarios that may arise during the application process motivate the development of ontologies. The motivating scenarios are story problems or examples that cannot be solved using the existing knowledge base, ontologies or similar techniques. They require a new ontology or a current ontology to be extended in order to enable the system to solve the problems from motivating scenarios. This is the reason that the motivating scenario will usually also provide a set of possible solutions to the scenario problems that can be solved only once the proposed ontology has been developed. The motivating scenario, together with a set of possible solutions to the scenario problems, provide an informal semantics for the concepts and relationships between those concepts which later need to be included in the ontology. When designing a new ontology or extending an existing one, one or more motivating scenarios need to be described as well as the set of possible solutions for the problems presented in the scenarios.

In our example of DNA ontology, we may have the following scenarios where we need to:

- 1) determine the exact position of a specific gene within a chromosome
- 2) determine the number of genes that are found on a specific chromosome;

3) determine the length of a specific gene in nucleotide base pairs (length of a gene is usually expressed in nucleotide base pairs because DNA is a double helix; nucleotide bases of the one helix bind to the nucleotide bases from the other helix).

2. Formulate informal competency questions. The informal competency questions are written in natural language and are used to evaluate the expressiveness of the ontology. Once the ontology has been expressed in a formal language, it needs to be able to answer these questions. The ontology to be developed must be able to use its terminology, definitions and axioms to represent these questions and characterize the answers to these questions.

Examples of competency questions relating to the abovementioned motivating scenarios are:

- 3) Given the name of the gene (e.g. GLDC, GALT, FRDA or HSD17B3 from Figure 5.4) and constraints in regard to gene positions on a specific chromosome (in this example, constraints in regard to gene positions on chromosome 9), can the exact position of this gene within the specific chromosome be determined?
- 4) Given the name of the chromosome (e.g. chromosome 9) and constraint in regard to the number of genes on a specific chromosome (e.g. greater than or equal to 11), can the number of genes found on this specific chromosome be determined?
- 5) Given the name of the gene (e.g. GLDC, GALT, FRDA or HSD17B3 from Figure 5.4) and constraints in regard to gene length (e.g. greater than or equal to 114 nucleotide base pairs), can the exact length of this chromosome in nucleotide base pairs be determined?

3. Specify terminology within a formal language. This step consists of two parts: extraction of informal terminology and formal specification of the terminology.

Extraction of informal terminology. The set of terms used within informal competency questions is extracted. They serve as a basis for specification of the terminology in a formal language. In our example, we can extract terms such as GLDC, GALT, FRDA, D17B3, gene name, gene position, gene length, nucleotide base, nucleotide base pair, chromosome, and similar.

Formal specification of the terminology. A formalism, such as first-order logic, can be used to specify the ontology terminology. This will enable the use of axioms to express ontology definitions and constraints. If we aim to use first order logic to specify the ontology terminology, firstly we need to identify objects in the universe of discourse. Examples of objects are: GLDC, GALT, FRDA, HSD17B3, chromosome 9 etc. Secondly, we need to identify unary predicates which are used to represent concepts (such as gene, chromosome, nucleotide base), binary predicates for attributes (such as gene-name, gene-length, gene-position, chromosome-name) and predicates that represent binary relations (has-length, has-position).

4. Formulate formal competency questions using formal terminology. In this step, the competency questions identified in step 2 are defined formally according to the specifications in step 3.

5. Specify axioms and definitions within a formal language. An ontology consists of a set of terms and a set of axioms that define the semantics (meaning) of ontology terms. Ontology axioms define the ontology terms and specify the constraints on their interpretation. If the existing axioms are insufficient to represent and define the solutions to the competency questions in a formal language, additional axioms need to be added to the ontology. Therefore, development of the ontology with respect to the competency questions is an iterative process.

6. Specify conditions characterizing ontology completeness. As a final step in the ontology design, we need to define the conditions under which the solutions to the formal competency questions are complete. These conditions serve then as a basis for development of ontology completeness theorems.

### 5.3.4 METHONTOLOGY

The METHONTOLOGY (Fernandez et al. 1997) framework enables the construction of ontologies at the knowledge level and includes:

- 1) An ontology development process
- 2) An ontology life cycle based on evolving prototypes

1. The Ontology Development Process. The ontology development process refers to the activities that need to be carried out when building ontologies. Three different categories of these activities have been identified: ontology management activities, ontology development-oriented activities and ontology support activities. The diagram is shown in Figure 5.6.

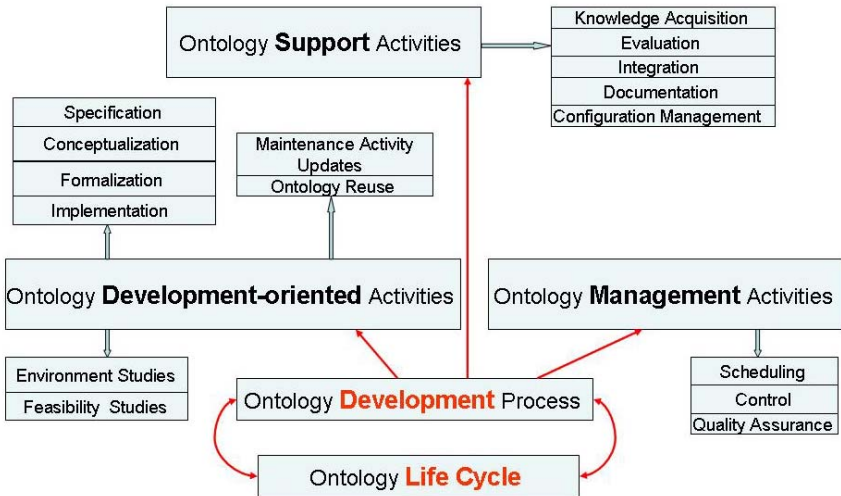


Fig. 5.6 METHONTOLOGY methodology

a. Ontology Management Activities include scheduling, control and quality assurance. Scheduling helps in the identification of tasks that need to be performed, their arrangement, and the time and resources needed for their completion. Control monitors the scheduled tasks and ensures that they are performed as planned. Quality assurance is responsible for the satisfactory quality of every product output (e.g. ontology, software or documentation).

b. Ontology Development-Oriented Activities are grouped into pre-development, development and post-development activities. Environment and feasibility studies are carried out during the pre-development phase. Environment study involves analysis of the platforms and applications where the ontology will be used and integrated, while the feasibility study analyzes possibility and suitability of ontology design. Specification, conceptualization, formalization and implementation activities that are carried out during the development. The purpose of the ontology, its intended uses and end-users are analyzed during the specification phase. Domain knowledge is represented as a meaningful model during the conceptualization phase. This conceptual model is transformed into a formal or semi-computable model during formalization. During implementation, computable models are built in a computational language. In the post-development phase, the ontology is updated and corrected on the basis of the maintenance activity. Also, other ontologies and applications can use the ontology.

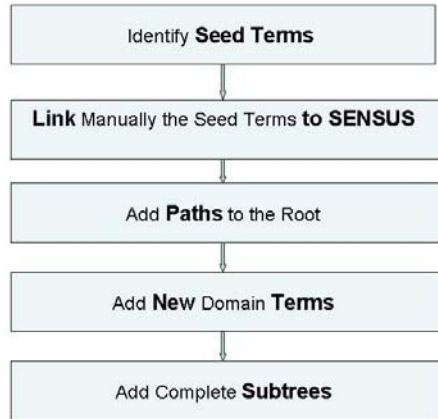
c. Ontology Support Activities include knowledge acquisition, evaluation, integration, documentation and configuration management. During the knowledge acquisition phase, domain experts or some kind of (semi)automatic process is used in order to acquire knowledge of a given domain. Ontologies, their associated software environments and documentation need to be technically evaluated. Integration of ontologies is required when designing a new ontology by reusing existing ontologies. Each completed stage, as well as any generated products, needs to be documented. All the versions of the documentation, software and ontology code are recorded by configuration management.

2. The Ontology Life Cycle. The ontology building life cycle is based on evolving prototypes. The ontology life cycle identifies the set of phases through which the ontology moves during its lifetime, describes activities that need to be performed in each phase, and relationships between the different phases. Ontology changes correspond to the improvements required in the evolving prototypes.

### 5.3.5 *SENSUS Methodology*

This approach is based on the assumption that the knowledge between two ontologies can be easily shared if they have a common underlying structure i.e. if they are based on a common ontology.

SENSUS (Swartout et al. 1997) is an ontology based on natural language and is composed of upper, middle and lower regions. It provides a broad coverage conceptual structure and contains more than 70.000 nodes. PENMAN Upper Model, ONTOS, WordNet and semantic categories from electronic dictionaries (English, Spanish and Japanese) are used for development of the SENSUS ontology.

**Fig. 5.7** SENSUS methodology

According to the SENSUS methodology, the steps shown in Figure 5.7 need to be followed when building an ontology for a particular domain:

- 1) Identify seed terms
- 2) Manually link the seed terms to SENSUS
- 3) Add paths to the root
- 4) Add new domain terms
- 5) Add complete subtrees

Seed terms act as representatives of relevant domain-specific concepts. They need to be identified and manually linked to the SENSUS ontology. All of the concepts from the seed terms to the linking node within the SENSUS ontology are included in the final ontology. Irrelevant SENSUS terms are pruned from the domain-specific ontology. Relevant domain terms absent from the SENSUS ontology need to be added to the final ontology. Some nodes may have a large number of paths through them. If many of the nodes in a subtree have been found to be relevant for the final ontology, then the other nodes in the subtree are likely to be relevant as well. For this reason, the entire subtree under the node may need to be added to the final ontology structure. Because this decision requires understanding of the domain, this step is best done manually.

In our example of the DNA ontology from Figure 5.2, “gene” is a seed term. This term needs to be linked to a node of a SENSUS ontology. The choice of the linking node depends on the intended purpose and use of the ontology. This linking node from the SENSUS ontology is called the “root”. We will add paths to the root by introducing terms “chromosome” and “DNA”. It may be necessary to add new domain terms such as “protein” and “amino acid”. Terms “Adenine”, “Guanine”, “Cytosine” and “Thymine” appear to have some relevance but the ontology designer is not sure whether the terms “nucleotide”, “nucleotide base”, “phosphate group”, “sugar molecule” are relevant as well. If these terms are found to be relevant, then the whole subtree under the term “nucleotide” needs to be added to

the final ontology. A domain expert will approve this because gene is a sequence of nucleotides that are composed of a nucleotide base (Adenine, Guanine, Cytosine or Thymine), a sugar molecule and a phosphate group.

### 5.3.6 DILIGENT Methodology

DILIGENT (Pinto et al. 2004) is a methodology for Distributed, Loosely-controlled and evolInG Engineering of oNTologies. This methodology may play an important role in the development of shared ontologies in distributed settings, such as the Semantic Web. This methodology was developed to support domain experts in a distributed setting to engineer and evolve ontologies using a fine-grained methodological approach based on Rhetorical Structure Theory (RST). RST is used to analyze the arguments exchanged when consensus is sought in evolving distributed ontology engineering processes. RST assumes a function for every part of a text and shows the evident role of this part of the text.

General process, roles and functions in the DILIGENT process are shown in Figure 5.8 and include five main activities:

- 1) build
- 2) local adaptation
- 3) analysis
- 4) revision
- 5) local update

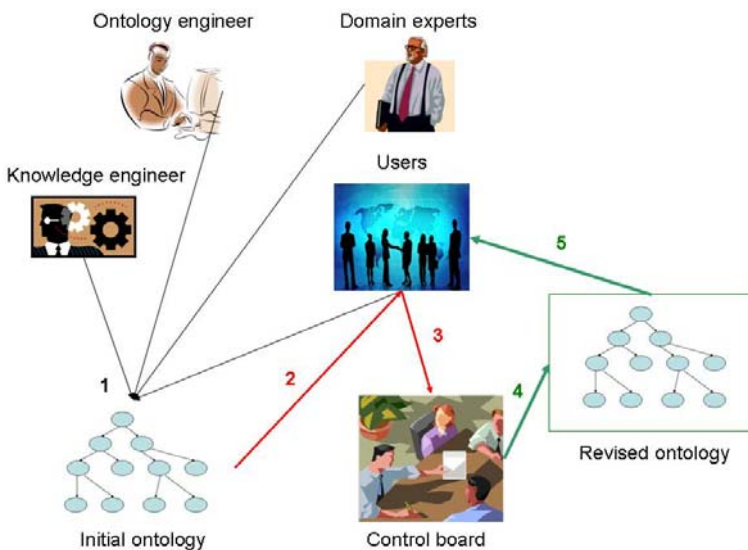


Fig. 5.8 DILIGENT methodology

Different stakeholders usually have different purposes, needs and locations. Such distributed ontology development requires online ontology engineering support. Domain experts, users, knowledge engineers and ontology engineers build an initial ontology (step 1 in Figure 5.8). This initial shared ontology is made available and users can start using it. The users need to adapt the initial ontology needs locally for their own purposes (step 2 in Figure 5.8). They can change the ontology in their local environment, but the original ontology that is shared by all users may not be changed. The control board collects and analyzes change requests to the shared ontology (step 3 in Figure 5.8). The board must decide which changes need to be introduced in the next version of the shared ontology. A balanced decision must be made that takes into account the different users' needs and meets their evolving requirements. The control board revises the shared ontology accordingly. Once a new version of the shared ontology has been released (step 4 in Figure 5.8), users can locally update their own ontologies (step 5 in Figure 5.8).

## 5.4 Conclusion

In this chapter, we introduced different ontology design methodologies: Knowledge Engineering, DOGMA, TOVE, MENTHONTOLOGY and SENSUS methodology. We took a simple DNA ontology as an example and illustrated the design of this ontology using each of the different methodologies. We have chosen these methodologies in particular in order to illustrate the diversity in ontology design methodologies. None of the abovementioned methodologies is similar to another. On the other hand, all of these different methodologies solve the same problem. It is very interesting that such diversity exists in the different approaches to the same problem.

In the following chapter, we will discuss the advantages of ontology- and agent-based systems, and the benefits of integrating these two complementary technologies.

## References

1. De Bo, J., Spyns, P., Meersman, R.: Creating a "DOGMAtic" multilingual ontology infrastructure to support a semantic portal. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2003. LNCS, vol. 2889, pp. 253–266. Springer, Heidelberg (2003)
2. Deray, T., Verheyden, P.: Towards a semantic integration of medical relational databases by using ontologies: a case study. In: Meersman, R., Tari, Z. (eds.) OTM-WS 2003. LNCS, vol. 2889, pp. 137–150. Springer, Heidelberg (2003)
3. Fernandez, M., Gomez-Perez, A., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: AAI 1997 Spring Symposium on Ontological Engineering, USA (1997)
4. Gómez-Pérez, A.: Towards a Framework to Verify Knowledge Sharing Technology. *Expert Systems with Applications* 11, 519–529 (1996)
5. Gruber, T.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies* 43, 907–928 (1995)



6. Gruninger, M., Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. In: IJCAI 1995 Workshop on Basic Ontological Issues in Knowledge Sharing, Canada (1995)
7. Jarrar, M., Demey, J., Meersman, R.: On Reusing Conceptual Data Modeling for Ontology Engineering. In: Spaccapietra, S., March, S., Aberer, K. (eds.) *Journal on Data Semantics I. LNCS*, vol. 2800, pp. 185–207. Springer, Heidelberg (2003)
8. Nakata, K.: A Grounded and Participatory Approach to Collaborative Information Exploration and Management. In: *Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS-34)* (2001)
9. Noy, N.F., McGuinness, D.L.: *Ontology Development 101: A Guide to Creating Your First Ontology* (2001), [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf) (retrieved: May 20, 2004)
10. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of ontologies. In: *The 16th European Conference on Artificial Intelligence (ECAI 2004)*, Spain, pp. 393–397 (2004)
11. Pisanelli, D.M., Gangemi, A., Steve, G.: Ontologies and Information Systems: The Marriage of the Century? In: *International workshop on LYEE methodology* (2002), <http://www.loa-cnr.it/Papers/lyee.pdf> (retrieved: January 18, 2005)
12. Swartout, W.R., Patil, R., Knight, K., Russ, T.: *Towards Distributed Use of Large-Scale Ontologies*. In: *AAAI 1997 Spring Symposium on Ontological Engineering, USA* (1997)
13. Uschold, M., Gruninger, M.: Ontologies: principles, methods, and applications. *Knowledge Engineering Review* 2, 93–155 (1996)