Knowledge Engineering

Nathalie Aussenac-Gilles, Jean Charlet, Chantal Reynaud

Knowledge Engineering

Nathalie Aussenac-Gilles, Jean Charlet and Chantal Reynaud

Abstract Knowledge engineering refers to all technical, scientific and social aspects involved in designing, maintaining and using knowledge-based systems. Research in this domain requires to develop studies on the nature of the knowledge and its representation, either the users' knowledge or the knowledge-based system's knowledge. It also requires the analysis of what type of knowledge sources is considered, what human-machine interaction is envisaged and more generally the specific end use. To that end, knowledge engineering needs to integrate innovation originating from artificial intelligence, knowledge representation, software engineering as well as modelling. This integration enables both users and software systems to manage and use the knowledge for inference reasoning. Other advances are fuelling new methods, software tools and interfaces to support knowledge modelling that are enabled by conceptual or formal knowledge representation languages. This chapter provides an overview of the main issues and major results that are considered as milestones in the domain, with a focus on recent advances marked by the raise of the semantic web, of ontologies and the social web.

1 Introduction

Knowledge engineering (KE) became a research domain in the early 1980s, its research object being designing, maintaining and using knowledge-based systems

N. Aussenac-Gilles (✉)
IRIT-CNRS, Université de Toulouse, Toulouse, France
e-mail: Nathalie.Aussenac-Gilles@irit.fr

J. Charlet
Sorbonne Université, INSERM, Université Paris 13, LIMICS, 75006 Paris, France
e-mail: Jean.Charlet@upmc.fr

J. Charlet
Assistance Publique-Hôpitaux de Paris, DRCI, Paris, France

C. Reynaud
LRI, Université Paris-Sud, CNRS, Université Paris-Saclay, Orsay, France
e-mail: Chantal.Reynaud@lri.fr

P. Marquis et al. (eds.), *A Guided Tour of Artificial Intelligence Research*,
https://doi.org/10.1007/978-3-030-06164-7_23

(KBS). Many of the early expert systems were developed using traditional software engineering methods combined with rapid prototyping. In this context, building conceptual models in the early stages of the process became a major and critical issue. The further population of these models with the appropriate knowledge presented also substantial challenges. The so-called knowledge acquisition bottleneck[1] became the subject of a large amount of research work, Ph.D. theses and international projects, either with a cognitive and methodological perspective (Aussenac 1989) or targeting the definition of new knowledge representations (Cordier and Reynaud 1991; Charlet 1991). In the late 1990s, the perspective broadened and gave birth to KE as a cross-disciplinary research domain. Mainly located in the field of Artificial Intelligence (AI), KE refers to all technical, scientific and social aspects involved in designing, maintaining and using KBS. KE defines the concepts, methods, techniques and tools to support knowledge acquisition, modelling and formalisation in organisations with the aim of structuring the knowledge and making it operational.

KE is expected to address knowledge modelling and sharing issues when designing any KBS that supports human activities and problem solving. Such knowledge intensive applications include knowledge management (KM) systems, Information Retrieval (IR) tools, both semantic or not, document or knowledge browsing, Information Extraction (IE), decision making or problem solving to name but a few. When the Semantic Web (to which the chapter "Semantic Web" of Volume 3 of this book is dedicated) emerged as a promising perspective to turn web data into knowledge and to define more powerful web services, research in KE started waving close relations with this domain. Indeed, the Semantic Web overlaps KE in various ways, both domains use the same languages, standards and tools like ontologies, knowledge representation languages and inference engines.

In the rest of this chapter, we propose a chronological and historical presentation of the major paradigms that marked milestones in KE during the last 25 years in Sect. 2. Then in Sect. 3, we detail the main research issues that KE is dealing with. Section 4 offers a synthetic view of the remaining methodological and representation challenges before we conclude in Sect. 5.

2 Knowledge Modelling

2.1 The Notion of Conceptual Model

Around the 1990s, KE methods proposed to design KBS starting with a knowledge modelling stage that aimed to collect and describe the system knowledge in

[1] Knowledge acquisition refers to the process of gathering expert knowledge (called "knowledge mining" at that time) and representing it in the form of rules and facts in the hope that the KBS behaves like the expert would in a similar situation. The difficulty to precisely collect or capture this knowledge, which is implicit and hard to elicit in many ways, reduces the amount and quality of knowledge actually represented, as the term "bottleneck" illustrates.

an operational form, regardless of the implementation. Knowledge representation in the model was both abstract and with an applicative purpose. It was expected to account for the multiple necessary knowledge features and types to meet the system requirements. Practically, this representation formed the so-called *conceptual model*. A conceptual model should fit the kind of knowledge to be described and would then be formalised using the appropriate formalisms required by the KBS (i.e. inference rules in many applications of the 1990s). Then, conceptual models became key components in knowledge engineering and they significantly evolved over the years to cover a large variety of models depending on the needs they should satisfy, thus being adapted to new approaches and to every recent research work in the field.

The way in which knowledge is described and represented impacts the implementation of the targeted KBS, and even more, the ability to understand or explain its behaviour. Knowledge acquisition and engineering have long referred to A. Newell's notion of *Knowledge Level* (1982). Newell was one of the first to establish a clear separation between the knowledge to be used in a system to produce a behaviour and its formal "in-use" representation in the system implementation. In other words, Newell stressed the necessity to describe the system knowledge at a level that would be independent from the symbols and structure of a programming language, level that he called the *Knowledge Level*. At this level, the system is considered as a rational agent that will use its knowledge to achieve some goals. Such system behaves in a rational way because, thanks to its knowledge, he intends to select the best sequence of actions leading to one of its goals as directly as possible. Newell's Knowledge Level not only prompted researchers to define conceptual models, but it also influenced the structuring of these models in several layers corresponding to various types of knowledge required to guarantee the system behaviour. In conceptual models, *domain knowledge*, that gathers entities or predicates and rules, is distinct from *problem solving knowledge* that consists in actions and goals modelled using *methods* and *tasks*.

2.2 Problem Solving Models

Problem solving models describe in an abstract way, using tasks and methods, the reasoning process that the KBS must carry out. A task defines one or several goals and sub-goals to be achieved by the system, and a method describes one of the ways the task goals can be achieved. A task description also specifies the input and output knowledge, constraints and resources required to perform the task. To describe the way the system should behave to solve a problem, a hierarchy of tasks can be defined, a general task being decomposed into several more specific tasks that specify the sub-goals required to achieve the goal of the main task. Methods make explicit how a goal can be reached thanks to an ordered sequence of operations. Methods that decompose a task into sub-tasks are distinguished from methods that implement a basic procedure to directly reach a particular goal. The distinction between tasks and methods progressively emerged from research works after B. Chandrasekaran proposed the notion of Generic Task (1983) and L. Steels proposed a componential modelling framework that included three types of components: tasks; methods and domain data models (1990). This distinction has been adopted to account for the reasoning process in many studies (Klinker et al. 1991; Puerta et al. 1992; Schreiber et al. 1994; Tu et al. 1995) because it provides a separate description of the targeted goal and the way to achieve it. Thus, several methods can be defined for one single task, making it easier to explicitly represent alternative ways to reach the same goal. This kind of model is similar to results established in task planning (Camilleri et al. 2008; Hendler et al. 1990) where planning systems implement problem solving models thanks to operational methods and tasks, as it is suggested in the CommonKADS methodology (Schreiber et al. 1999).

2.3 From Conceptual Models to Ontologies

Once solutions had been found to design explicit problem-solving models, building the full conceptual model of an application consisted in reusing and adapting problem-solving components together with an abstract representation of domain data and concepts. Then an analysis of the domain knowledge was needed to establish a proper connection between each piece of the domain knowledge and the roles it played in problem solving (Reynaud et al. 1997). Domain knowledge models include two parts. The *domain ontology* forms the core part; it gathers concepts, i.e. class-sets of domain entities in a class/sub-class hierarchy, and relations between these classes, to which may be associated properties like constraints or rules. The second part extends this core with instances or entities belonging to the concepts classes, and relations between these entities. Thus an ontology defines a logical vocabulary to express domain facts and knowledge, in a formal way so that a system can use it for reasoning. Some concepts, called *primitive concepts*, are defined thanks to their situation in the concept hierarchy and thanks to properties that form necessary conditions for an entity to belong to this class. Other concepts, called *defined concepts*, are defined as classes equivalent to necessary and sufficient conditions that refer to properties and primitive concepts. The word *ontology* used to refer to a sub-field of philosophy. It has been first used in computer science, and particularly in AI, after the *Knowledge Sharing Effort* ARPA project (Neches et al. 1991) introduced it to refer to a structure describing the domain knowledge in a KBS. A little later, Gruber (1993) was the first to propose a definition of ontology in the field of KE. A more recent definition, proposed in Studer et al. (1998), is currently the acknowledged one:

> *An ontology is a formal, explicit specification of a shared conceptualisation.*
>
> *Conceptualisation refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. 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 reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group.*

420043_1_En_23_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:8/7/2019 Pages: 36 Layout: T1-Standard

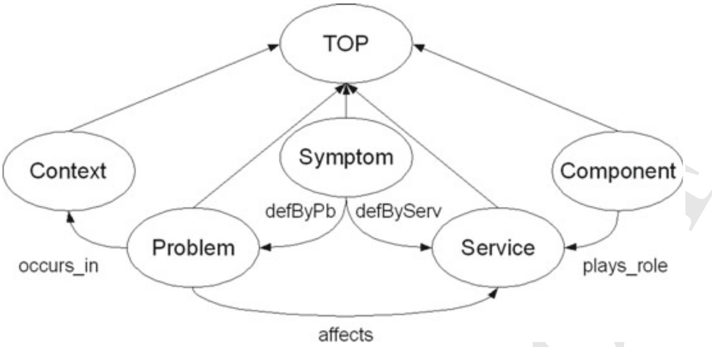

Fig. 1 High level concepts of an ontology used in the domain of electronic fault diagnosis

To sum up, ontologies meet complementary and symmetric requirements: (a) as specifications, they define a *formal semantics* so that software tools may process them; (b) as knowledge models, they reflect a – partial – point of view on a knowledge domain, that designers try to build as consensual as possible, and they provide semantic bridges that connect machine processable representations with their actual meanings for humans – supporting what Rastier calls *interpretative semantics* (2009).

The fact that an ontology be formal is both a strength because it enables to produce inferences (e.g. entity classification) and a limitation, using a formal language for its representation making it more difficulty to build. Figure 1 presents the main high level concepts of an ontology designed for an IR system in the domain of electronic fault diagnosis for cars. The `symptom` concept is defined by the identification of a car `component`, that provides a `service` to the vehicle user, that has been affected by a `problem` in a particular `context`. In the formal representation of this model, cardinality constraints on the `defByPb` and `defByServ` relations contribute to express that an instance of symptom cannot be identified unless a service and a problem have been identified too.

According to how the ontology will be used, it needs to be more or less rich in defined concepts and relations. For instance, if the ontology will be used in a standard information retrieval system, its role will be to structure domain concepts in a hierarchy and to provide labels (terms) for these concepts. This kind of ontology is called a *light-weight ontology*: it contains a concept hierarchy (or taxonomy) and very few defined concepts. When concept labels are represented with a specific formal class and properties, either called (formal) term or lexical entry, this kind of ontology is called *Lexical Ontology*.[2] If the ontology is to be used to produce inferences on domain knowledge, it will generally be larger and it will contain more relations, more axioms involved in the definition of defined concepts or any concept required for reasoning. This second kind of ontology is called a *heavy-weight ontology*.

[2]Whereas the KE English-speaking community uses "lexical ontology", many French research groups refer to Termino-Ontological Resource (TOR) (Reymonet et al. 2007) for very similar knowledge structures.

Due to their genericity and potentially high reusability, ontologies were expected to be easy to design. Several research lines have tried to characterise which parts of an ontology could be generic, and consequently reusable, on the one hand, and which techniques and methods could support the design of the non-generic parts. This distinction led to define the following typology of ontologies, which may also correspond to knowledge levels in a single ontology:

- An *upper level ontology* or *top*-ontology is considered the highest level. It structures knowledge with very general and abstract categories that are supposed to be universal and that are the fruit of philosophical studies on the nature of the main knowledge categories when formally representing human thinking in any domain. The major reference studies about top levels in ontologies are Sowa's *top-level categories*,[3] SUMO,[4] or DOLCE[5] to name a few of them. As concluded by the SUO[6] working group and the joint communiqué from the Upper Ontology Summit,[7] trying to define a unique norm for high level categories is pointless as long as various philosophical schools or trends propose distinct ways to categorise the world entities. Top level ontologies are the anchor point of more specific levels (core ontologies and domain knowledge), and they are generic enough to be shared.
- A *core ontology* or *upper domain ontology* provides a domain description that defines the main concepts of a particular domain, together with properties and axioms applying on these concepts. For instance, a core ontology of medicine would contain concepts such as *diagnosis, sign, anatomic structure* and relations like *localisation* linking a pathology to the affected anatomic structure (cf. GFO-Bio[8]); in Law, the LKIF-Core[9] ontology offers notions like *norm, legal action* and *statutory role*.
- A *domain ontology* describes the domain concepts practically handled by professionals and experts in everyday activities. It is the most specific kind of a knowledge model, and it becomes a knowledge base when instances of domain specific concepts are represented. Nevertheless, there may be no clear frontier between a *core*-ontology and an ontology of the same domain that includes the core one when both of them are designed within the same process. The distinction is more obvious when the domain ontology reuses and specialises an existing core ontology. Domain ontologies or the domain level of ontologies can be designed thanks to text-based approaches and reusing domain thesaurus or terminologies (cf. Sect. 4.1).

[3] http://www.jfsowa.com/ontology/toplevel.htm.

[4] http://www.ontologyportal.org/.

[5] http://www.loa-cnr.it/DOLCE.html.

[6] http://suo.ieee.org/.

[7] http://ontolog.cim3.net/cgi-bin/wiki.pl?UpperOntologySummit/UosJointCommunique.

[8] http://www.onto-med.de/ontologies/gfo-bio/index.jsp.

[9] http://www.estrellaproject.org/lkif-core/.

3 Issues and Major Results

If we consider the KE evolution over the last 30 years, changes have been driven by the diversification of what could be considered as a knowledge source for "intelligent" or AI information systems. This wealth in knowledge sources came together with changes in computers that impacted any software system: the amazing increase in storage capacities and always higher computing performance of computers. Knowledge source diversification offered the advantage to benefit from complementary sources together with available techniques to analyse them. In the following we first outline the various knowledge sources that KE has successively focused on over the years, as well as the research issues raised by the passage from these sources to models. By model, we mean here the different types of knowledge models presented in Sect. 2 used to represent either the knowledge in a KBS (*conceptual models*), some problem-solving process (*problem-solving models*) or domain specific knowledge (*domain models*). Then we show the research paradigms that deal with these issues, as well as the variety of modelling methods and techniques produced in KE to overcome them. We end with the presentation of major results about model reuse and with the connection of this research with the one on knowledge representation.

3.1 Knowledge Sources

Historically, *knowledge* for KBS first referred to human expertise, for which the knowledge base of *expert systems* should account according to a human-inspired paradigm. Knowledge was thus both technical and specialised. It gathered high-level skills and know-how that generally never had been verbalised before, and that were hard to explicit. The expected role of expert systems was to capitalise and make this expertise explicit so that it could be sustained and transferred to the KBS, or to humans via the KBS. Knowledge was then represented with inference rules.[10]

In a second period, expert systems evolved and became *Knowledge-Based systems* because their role was no longer to replace the expert but rather to provide an intelligent help to the end-user. Efficiency was privileged against the accuracy towards human reasoning. Then reference knowledge became shared knowledge, that KBS used for reasoning according to their own problem solving engines.

Today, many applications (i.e. spelling checkers, decision support systems, billing systems, but also chest players or search engines) include some *model-based modules*. Their goal is to perform some of the system tasks either in an autonomous way or in a cooperative way together with other modules or in cooperation with the user, adapting to the use context and to users' profiles. The knowledge required for these support tasks to solve problems or to perform activities includes technical,

[10] For a historical outline on knowledge-based system, one can read Aussenac (1989), Stefik (1995), Aussenac-Gilles et al. (1996), or Charlet et al. (2000).

consensual and shared knowledge, that is modelled as rules or action maps, and as structured and goal-oriented domain models.

The historical evolution of knowledge-based information systems highlights various types of knowledge that were considered over the years: individual expert knowledge, in-use knowledge related to practice, activities and individual usage; knowledge about organisations, consensual and shared knowledge of an application field, common sense knowledge, knowledge related to knowledge integration or distributed knowledge over the Web. It is to capture these various kinds of knowledge that new knowledge sources have been taken into account. Thus, documents have played an increasing role as more digital documents were available. Since the early works on knowledge acquisition for expert systems, KE relies on documents, in particular textual documents, as they convey meaning and may contribute to reveal some knowledge. Documents are exploited for the language and information they contain, which is complementary or an alternative to interviews of domain experts or specialists. Data can also become knowledge sources thanks to knowledge or information extraction processes from data or data mining. Last, components of existing knowledge models can be reused when they convey consensual and shared knowledge. These components can either be *problem solving models*, that can be reused across various domains, like the library of problem solving methods in CommonKADS (this library is one of the major results of the KADS and later CommonKADS[11] European projects Schreiber et al. 1999), or domain models, ontologies, semantic resources like lexical data-bases or thesauri. Ontologies represent domain concept definitions in a formal structure. A lexical data-bases like WordNet[12] registers, classifies and organises, according to semantic and lexical criteria, most of the vocabulary of the English language. Thesauri collect normalised domain vocabularies as structured sets of terms.

3.2 From Knowledge Sources to Models: Research Issues

One of the core and typical issues in KE is to provide or develop tools, techniques and methods that support the transition from the knowledge sources listed in Sect. 3.1 to the models presented in Sect. 2. These techniques not only rely on software systems but also on analysis frameworks or observation grids borrowed to other disciplines. Research in KE actually follows an engineering paradigm in the sense that it requires innovation to design new tools, languages and methods or to select and adapt existing ones. It requires as much innovation to organise them in an appropriate way within methodological guidelines and integrated or collaborative platforms. Expected innovations concern the nature and development of these tools as well as the definition of their use conditions, their synergy and interactions so that they could manage particular knowledge types at each stage of the development process of an application.

[11] http://www.commonkads.uva.nl/.

[12] http://wordnet.princeton.edu/wordnet/.

For the last twenty years, methodological research in KE raised cross-functional issues that have been reformulated and renewed when new knowledge sources were addressed, new types of models were designed or new use-cases and problems had to be solved using these models.

3.2.1 How to Design a Model?

Two complementary methodological streams first defined diverging stages and techniques (Aussenac-Gilles et al. 1992). *Bottom-up methods* privilege data analysis, first driven by the identified users' needs and later guided by the model structure and the components to be filled. Bottom-up approaches focus on tools that support data collection and mining, knowledge identification and extraction, and later on tools that produce abstract representations of knowledge features (classification, structuring and identification of methods and problem solving models). In contrast, the alternative process follows a *top-down approach* that privileges the *reuse* and adaptation of existing knowledge components. Then knowledge gathering starts with the selection of appropriate components, that further guides the extraction of new knowledge and the model instantiation process. A unified view considers that modelling follows a cyclic process where bottom-up and top-down stages alternate. The process moves from stages dedicated to knowledge collection or reuse towards knowledge representation stages using more and more formal languages. Most methods and tools presented in Sect. 3.3 combine both processes, whereas we focus on results about model reuse in Sect. 3.4.

3.2.2 How to Benefit from Complementary Knowledge Sources?

Diversifying knowledge sources and knowledge types is one of the solutions to get more precise and richer models, or to automatically design a part of them. As a consequence, KE methods start with the identification of appropriate knowledge sources. They suggest also a set of relevant tools and techniques that explore and efficiently process these sources. Most of all, they propose methodological guidelines to articulate the use of these tools in a coordinated way that ensures a complementary exploitation of their results to design an appropriate model. Results in Sect. 3.3 illustrate this process.

3.2.3 What Are Models Made of? What is the Optimal Formal Level?

Each model combines various types of knowledge. In a similar way, each KE method questions and makes suggestions on the nature of the models to be designed, on the way to structure them and to collect the appropriate knowledge that feel them as well as on the representation formalism to select, which can be more or less formal as discussed in Sect. 3.5.

3.2.4 How Does Model Engineering Take into Account the Target Use of a Model?

Several research studies have shown that conceptual models were all the more relevant than they were dedicated to a specific range of systems. KE does not restrict its scope to design models; it is highly concerned by their actual use because it is one of the ways to validate the engineering process, and because it is this specific use that determines the model content, its structure and, as a side effect, the way the model is designed. In short, the targeted use of a model has a strong impact on methodological options and on the selection of a knowledge representation in the model (Bourigault et al. 2004).

3.2.5 How to Promote Model Reuse?

The reuse of structured knowledge fragments is often the best option to reduce the cost of knowledge modelling. However, reuse is not possible unless the principles that guided the model design are available, unless models can be compared and combined, and unless the selection of some of their components and their combination are technically feasible and sound. These very same questions also arise in research work about ontology or KB alignment, reuse and composition to build new knowledge bases.

3.2.6 How to Ensure Model Evolution in Relation with the Use Context?

The knowledge models used in KBS are involved in a life cycle that includes their evolution. This parameter became increasingly significant as a consequence of the evolution of the knowledge sources, of domain knowledge and users' needs. Since the early 2000s, ontology evolution is one of the major challenges to be solved to promote their actual use. Various research studies define an evolution life-cycle, several means to identify and to manage changes while keeping the model consistent (Stojanovic 2004; Luong 2007).

3.3 Designing Models: Techniques, Methods and Tools

In order to make practical proposals in getting access to knowledge coming from people or documents deemed to provide indications, KE has its own solutions: techniques and tools that may be integrated into methodologies and frameworks. These solutions are largely inspired by close disciplines, depending on the considered source of knowledge, sequentially covering cognitive psychology, ergonomics, terminology and corpus linguistics since KE emerged as a discipline.

Designing models requires access to knowledge available through various sources. Access techniques depend on the nature of the sources, with potentially generation of new knowledge that had not been made explicit before. *Technique* makes reference here to operating modes requiring specific ways to choose or create knowledge production or use situations, then ways to discover/collect/extract or analyse data, and finally proposals to interpret, evaluate and structure the results of the analysis. We focus on the two knowledge sources that have been most widely used in this process: human expertise and textual documents.

3.3.1 Human Expertise as Knowledge Source

Regarding human expertise, research approaches have evolved from a *cognitivist* perspective, assuming a possible relation between mental and computer representations, to *constructivist* approaches, considering that models as artifacts that enable the system to behave as the human would, and then *situated cognition*, taking into account a contextual or collective dimension. In the first case, the task is to locate, make explicit and represent technical expertise. According to this view, which historically lead to design expert systems, one or several human experts possess the knowledge that has to be made explicit in order to design a system that produces the same reasoning. Cognitive psychology has provided guidelines on how to carry out interviews, on how to analyse them and gave the pros and cons of each form of interview in relation to the study of human cognitive phenomena (Darses and Montmollin 2006). These techniques have been adapted and then used to extract knowledge from experts, as in the works of Aussenac (1989), Shadbolt et al. (1999) or Dieng-Kuntz et al. (2005). We can distinguish the *direct* methods that consist in querying the expert to get him to speak in a more or less guided way and the *indirect* methods as *repertory grids* based on the interpretation of acquired elements as the expert performs tasks using his expertise.

This *cognitivist* perspective has been increasingly brought into question to better satisfy the situated aspect of the knowledge. As expertise is only accessible when applied in problem solving situations, KE has taken up task and activity analysis techniques from the area of ergonomics.

One main result was to lay the foundations of knowledge acquisition as a discipline focusing on knowledge *itself* prior to considering its formalisation and its use within a given system. Both adopting the *constructivist* view and taking into account existing methods in software engineering then led to new methodological proposals guiding the whole knowledge acquisition process. Several methods defined in important projects, mainly European projects, are presented in Sect. 3.3.3.

Knowledge in software aims at better guiding users. By the way, it impacts their working methods. So it raises the need to analyse their practices and the practices of their collaborators, to study their activities and their use of support tools, to consider their organisational context, which refers to ergonomics, sociological or management approaches. Results of such analyses were first returned in a static way, as models (task, interaction and organisation models for instance in CommonKADS) (Schreiber et al. 1999). These models were made operational using task languages and methods such as LISA, Task (Jacob-Delouis and Krivine 1995) or CML (Schreiber et al. 1994). The notion of trace of activities has then been widely explored to take into account activities in a more in-depth way. Traces are integrated to provide users with a precise and context sensitive help based on the knowledge of their behaviour. Therefore, Laflaquiére et al. (2008) define the notion of trace for software use or documentation system activities in order to be able to discover, represent, store traces and then exploit and reuse them.

3.3.2 Textual Documents as Knowledge Sources

Regarding textual documents, whether technical, linked to an activity or to an application domain, two problems arise when exploiting them as knowledge sources: their selection and their analysis. Document analysis is mainly based on the natural language in the text. Some approaches also exploit the text structure identified on the paper or screen layout and electronically manageable thanks to tags or annotations (Virbel and Luc 2001). The latter is generally referred as structured or semi-structured documents (XML documents). We first describe the strengths of textual document analysis, then the techniques and the tools used for that.

Strengths of Textual Document Analysis

Textual documents are rich knowledge sources. Text analysis has always been a part of KE but the way to address it changed drastically after 1990. We do not try anymore to recover automatically the understanding of a text by an individual (Aussenac-Gilles et al. 1995). The increasing importance of textual analysis is a consequence of the progress achieved by natural language processing (NLP), which has delivered robust specialised software programs to process written language. NLP maturity has been synchronous with ontology deployment. Designing ontologies and using them to semantically annotate documents became two applications of the analysis of written natural language. A strong assumption behind automatic text processing is that text provide stable, consensual and shared knowledge of an application domain (Bourigault and Slodzian 1999; Condamines 2002). However, this is not always the case, and two key points influence the quality of the extracted data: first, the creation of a relevant corpus early on in the process, then a regular contribution of domain experts or experts in modelling for interpreting the results. Text analysis is used to design ontologies and similar resources such as thesauri, indexes, glossaries or terminological knowledge bases.

Techniques and Tools for Textual Analysis

The aim of textual analysis in KE is to discover, in an automatic or cooperative way, linguistic elements and their interpretation and to help designing parts of conceptual models.

Linguistic approaches are based on wordings in the text to identify knowledge rich contexts (Barriere and Agbago 2006). Domain notions are expected to be mentionned using nominal or verbal phrases with a strong coherence. According to the way they are used, these phrases can be considered as terms denoting domain concepts or relationships between domain concepts. Language may also provide clues with a lower reliability, linking more diffuse knowledge elements. Then analysts have to rebuild reference links in order to come up with knowledge-based elements, axioms or rules. Results established by lexical semantics, terminology and corpus linguistics research are set prior to the implementation of this kind of approach (Condamines 2002; Constant et al. 2008).

Statistical approaches process a text as a whole and take advantage of redundancies, regularities, co-occurrences in order to discover idioms and terms, but also words or sets of words (clusters) with a similar behaviour or linguistic context. Several such techniques are described in the book *Foundations of Statistical Natural Language Processing* from Manning and Schütze (1999).

In both cases, preliminary text analysis, as cutting a text into sentences and into token words or grammatical parsing of words, is needed. A description of this research work is given in chapter "Artificial Intelligence and Natural Language" of Volume 3. The more sophisticated the pre-processing is (as complete syntactic analysis of sentences), the easier it is to automatically define precise interpretation rules. Unfortunately, software performing sophisticated analyses are often less robust, and they are available in fewer languages, English being often favoured. Furthermore, resources are sometimes needed (such as glossaries or semantic dictionaries) and few of them are available in some languages.

When the structure of the documents is available as a result of explicit markers, linguistic approaches can be combined with the exploitation of the structure in order to benefit of their complementary semantics (Kamel and Aussenac-Gilles 2009). The underlying idea is that structural cutting process of documents contributes to the semantic characterisation of their content.

Regarding the design of ontologies, text analysis serves two purposes (Maedche 2002; Cimiano et al. 2010): the identification of concepts with their properties and relationships, or *ontology learning* process; and the identification of concept instances and relations holding between them, the *ontology population* process. Similar tools can be used in both cases: text corpora have to be parsed in order to discover linguistic *knowledge-rich* elements (Meyer 2000), linguistic clues that can be interpreted as knowledge fragments.

Vocabulary modelling motivated the design of dedicated software tools that provide higher level results than standard NLP tools. For instance, results such as terms and clusters of synonym terms can then be integrated in a model. Examples of such tools are term extractors – Terminoweb (Barriere and Agbago 2006), Syntex-Upery (Bourigault 2002), TermExtractor (Drouin 2003) or TermRaider in the GATE[13] framework -; pattern-based relation extractors - Caméléon (Aussenac-Gilles and Jacques 2008), RelExt (Schutz and Buitelaar 2005) or SPRAT (Maynard et al. 2009) that implements three types of lexico-syntactic patterns (Hearst's patterns, patterns derived from Ontology design patterns and contextual patterns) in

[13] http://gate.ac.uk/.

GATE; pattern-based languages like Jape in GATE, Nooj,[14] Unitex[15]; named-entity extractors (Poibeau and Kosseim 2000) that contribute to search for instances or relations between instances (as with the KIM platform[16]). To sum up, designing models from texts has strongly benefited from NLP frameworks (GATE, Linguastream,[17] UIMA[18]) that support the development of adapted processing chains. Finally, specific processing chains, as Text2Onto (Cimiano and Völker 2005), and the version integrated by NeOn,[19] have allowed an assessment of the strengths and limitations of this approach by increasing automation and exploiting machine learning techniques. Current research works combine text analysis, reuse of ontological components and human interpretation. Cimiano et al. (2010) gives a reasonably full picture of these works.

3.3.3 Modelling Frameworks

Modelling frameworks provide access to knowledge sources, or to their traces, to knowledge extraction techniques and software tools, as well as to modelling techniques and languages. They suggest a methodology that defines a processing chain and guides the modelling task step by step. In the following Sub-section, we first present the most significant results about problem-solving modelling in the early 1990s. Then we focus on methods and frameworks for ontology design which have been developed in the last ten years.

Methods for Problem-Solving Modelling

Methodological guidelines have been established to better design large knowledge-based system projects. Their principles are similar to those in software engineering because of the importance assigned to modelling. In both cases, development cycles have to be managed and one or several models of the system to be designed must be built. The design of an application is considered as a model transformation process with conceptual models defined in Sect. 2.1. This requires a set of epistemological primitives that characterises at a high level (knowledge level) inference capabilities of the system to be designed. These primitives define generic knowledge representation structures that can be further instantiated.

In the early 1980s and 1990s the notion of conceptual model evolved with an emphasis on problem-solving models, new related languages, inference and tasks notions articulated. From a methodological viewpoint, the research showed that modelling primitives provide a grid for collecting and interpreting knowledge; they guide modelling. The utility of having elements coming from generic models and

[14] http://www.nooj4nlp.net/.
[15] http://www-igm.univ-mlv.fr/~unitex/.
[16] http://www.ontotext.com/kim/.
[17] http://linguastream.org/.
[18] http://domino.research.ibm.com/comm/research_projects.nsf/pages/uima.index.html.
[19] http://www.neon-toolkit.org/.

of being able to reuse them by instantiation on a particular application has then emerged, in particular from results on Generic Tasks from Chandrasekaran (1983). Later, the CommonKADS methodology showed the interest of adaptable and modular elements. All these principles are general as they apply irrespective of the task, the domain and the problem-solving method performed. Modelling techniques and reusable components are integrated in frameworks including as well expertise extraction techniques.

Following the work on Generic Task and role-limited methods (Marcus and McDermott 1989), and the proposals made by L. Steels in the componentional COMMET approach and in the KREST framework (1990), several works distinguished explicitly the notions of tasks and methods. This distinction has the advantage to describe separately the goal to be reached from the way to reach it and it allows for the explicit definition of several ways to reach a same goal by associating several problem-solving methods to a same task. These works have been taken into account by the European project KADS (Schreiber and Wielinga 1992), a pioneer in KE, which has resulted in the most accomplished methodology and framework CommonKADS (Schreiber et al. 1999).

CommonKADS allows for the construction of several models related to each other and required to specify a KBS with an organisational model reflecting in-use knowledge. The expertise model of the system is now recognised as very different from a cognitive model of a human expert. It is described according to three viewpoints: tasks, domain models, methods. Each problem-solving method can be parametrised and its adaptation is defined using a questionnaire guiding for the choice of one of the solution methods corresponding to each main task of the reasoning process of a specific application. Tasks describe what must be performed by the KBS. Domain models describe the knowledge required for reasoning. Methods describe how the knowledge is used to solve a task. A method can decompose a task into sub-tasks or solve one or several task(s). The methodology suggests an iterative construction of an application model according to the three different viewpoints. These perspectives are all necessary and complementary. The choice of a domain model depends on the selection of a problem-solving method as problem-solving methods define the role of the knowledge to be filled. Specifically, methods largely define the nature of the controlled sub-tasks. The aim of the methodology is thus to identify and model all the relations between methods, tasks and domain models.

Methods and Frameworks for Designing Ontologies

The design process of ontologies took advantage of these methodologies. It started when the reuse of domain models put forward the interest in high quality consensual models designed according ≪ good ≫ principles facilitating reuse and adaptation. The specific challenges encountered during the ontology design process are the followings:

1. Define the ontology content and ensure its quality;
2. Exploit efficiently all available knowledge sources using, for instance, text analysis or ontology reuse processes;

3. Facilitate the knowledge engineer design by providing specific tools; and
4. Define a methodological setting and the relevant approach to perform the various tasks.

Ontology engineering frameworks are uniform and coherent environments supporting the ontology design. They help achieve the different tasks by providing various tools and supporting a methodology that guarantees that all tasks are run one after the other.

Various methods can be used to design ontologies.[20] In this paper, we present three methodologies that are paying close attention to the quality of the ontology content: OntoClean, ARCHONTE and OntoSpec.

The OntoClean methodology has been designed by Guarino and Welty (2004). The first ideas were presented in a series of articles published in 2000, the OntoClean name appeared in 2002. Inspired by the notion of formal ontology and by principles of analytical philosophy, OntoClean made a significant contribution as the first formal methodology in ontology engineering. It proposes to analyse ontologies and to justify ontological choices using metaproperties of formal classes independent of all application domains. These metaproperties were originally four (i.e. identity, unity, rigidity and dependence).

The ARCHONTE (ARCHitecture for ONTological Elaborating) methodology, designed by Bachimont et al. (2002), is a bottom-up methodology to design ontologies from domain texts in three steps. First, relevant domain terms are selected and then semantically normalised as concepts by indicating the similarities and differences between each concept, its siblings and its father (principle of *differential semantic*). The second step consists in knowledge formalisation (*ontological commitment*). The aim is to design a differential ontology by adding properties or annotations, by defining domains and ranges of relationships. Finally, the third step consists in ontology operationalisation using knowledge representation languages. This process results in a *computational ontology*.

OntoSpec (Kassel 2002) is a semi-informal ontology specification methodology. It finds its origins in the definitions that are associated in natural language with conceptual entities which allow users to collaborate with knowledge engineers in order to design ontologies. In addition, this methodology proposes a framework including a typology of properties that can be used in the definition of concepts, relationships or rules, in order to paraphrase properties using natural language. The framework serves as a guide to model and facilitate the design of formal ontologies.

The main component of the frameworks used for designing ontologies is usually an ontology editor. Therefore, Protégé[21] is an editor extensively used to create or modify RDFS or OWL ontologies, and can be available as a web service (Web-Protégé) which is particularly appropriate for cooperative ontology design. Swoop[22] has been designed for lightweight ontologies, whereas Hozo[23]'s original-

[20]For a survey of the main existing methodologies, see Fernández-López and Gómez-Pérez (2002).
[21]http://protege.stanford.edu/.
[22]http://code.google.com/p/swoop/.
[23]http://www.hozo.jp/ckc07demo/.

ity lies in the notion of role and the ability to distinguish concepts depending on particular contexts from basic concepts to ensure an easier ontology reuse. Besides this editing function, several other functionalities can be provided in ontology engineering frameworks, such as Schema XML translating functions, graph display of parts of the ontology, ontology modules management, ontology partition, translation of vocabularies, import functions of Web ontologies, access to ontology search engines, text processing modules (like Tree-Tagger[24] or Stanford Parsing tools), help for personalizing ontologies, generating documentation, managing ontology evolution, ontology evaluation, ontology alignment, reasoning and inference services, navigation assistance services, visualisation services, … As an illustration, most of these functionalities are available as plug-ins in the Neon[25] framework.

Some frameworks are designed to deal with a specific kind of data. Therefore, Text2Onto, successor of TextToOnto, and DaFOE4App are specially designed to use text documents and thesaurus as input knowledge sources. Text2Onto (Cimiano and Völker 2005) includes a text mining software and modules that generate structured information from weakly structured documents. Text2Onto is associated with KAON (Karlsruhe Ontology Management Infrastructure) framework (Oberle et al. 2004) in order to design ontologies. DaFOE4App (Differential and Formal Ontology Editor for Applications) (Szulman et al. 2009) focuses on the linguistic dimension while its design uses some of the ARCHONTE methodology principles (Bachimont et al. 2002). DaFOE4App covers all stages from corpora analysis (using a NLP framework) to the definition of a formal domain ontology. It guarantees persistence, traceability and the dimensioning of models (several millions of concepts). The TERMINAE framework (Aussenac-Gilles et al. 2008), designed before DaFOE4App, has evolved with the specifications of DaFOE4App. TERMINAE[26] was used and evaluated in many projects. To end this non-exhaustive list, PlibEditor is more specially tailored to databases. With PlibEditor, users can perform all the tasks required to design ontologies, import or export ontologies as well as data. PlibEditor is complementary to OntoDB, an ontology-based database system and it enables a database approach based on domain ontologies (Fankam et al. 2009).

3.4 *Model Reuse*

Just as software engineering aims to reuse software components, knowledge acquisition promotes the reuse of knowledge components. This reusability can be achieved in various ways.

Initially proposed in the settings of the KADS project, reuse of problem-solving models consists in taking up task models expressed in a domain-independent terminology and adapting them to specific tasks. This approach is attractive. However,

[24] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

[25] http://www.neon-toolkit.org/wiki/Neon_Plugins.

[26] http://lipn.univ-paris13.fr/terminae/.

two specific problems are of paramount importance when adapting a problem-solving model to a specific domain. First, an application often performs several types of reasoning, with several models associated to each of them that have to be distinguished and combined. Second, the reuse and adaptation of predefined generic models to a specific application is difficult and highly time consuming. Indeed, both the task to be completed and the knowledge base of the system must be expressed in the terms of the same application domain, whereas reusable methods coming from libraries, are expressed using a generic vocabulary. Therefore, adapting problem-solving elements to an application is first and mainly a problem of term matching. Consequently, these challenges have led to more flexible approaches with reusable and adaptable elements of a finer granularity. Such approaches imply reusing parts of reasoning models instead of full generic problem-solving models.

Based on the KADS project's outcome, some frameworks support the combination of generic components. They include rich libraries of components as well as graphical editors dedicated to knowledge formalisation, task representation, and the selection and configuration of the methods allowing to solve the tasks (Musen et al. 1994). Solution to adapt generic models to a specific application are diverse, ranging from manual instantiation procedures (Beys et al. 1996) to automated processes including mechanisms that check the specification consistency (Fensel et al. 1996). The CommonKADS project settings led to the most successful results to design problem-solving models. The CommonKADS expertise model can be built by abstraction process or reusing components of problem-solving models. Its particular strength lies in the library of components with different granularities, and with a reuse and adaptation process guided by a questions grid which ensures the relevancy of designed model.

Ontology design is also shaped by the need to reuse existing models. The number of domain ontologies has grown significantly, their popularity being explained in part by the ability to reuse them from one information system to another. Specifically, ontology reuse aims at reducing the difficulties in ex-nihilo developments that constitute real obstacles to some applications. Issues raised by ontology reuse include: the selection of reusable and relevant ontologies, the specific support required to reuse large and complex ontologies that are hard to comprehend, and the integration of various reused ontologies in the under development ontology.

Ontology reuse has motivated the design of ontology search engines such as Watson,[27] Swoogle,[28] or OntoSearch.[29] Using key words, these engines provide a list of ontologies containing at least one concept, one relationship or another element labelled or identified by one of the key words. Then selecting the most relevant ontologies in this list requires that each ontology could be evaluated individually and that ontologies could be compared to eachother according to various criteria. Therefore, how to assess an ontology and to compare several ontologies is currently one of the main challenges in the field. Various questions should be addressed in order

[27] http://kmi-web05.open.ac.uk/WatsonWUI/.

[28] http://swoogle.umbc.edu/.

[29] http://asaha.com/ebook/wNjE3MzI-/OntoSearch--An-Ontology-Search-Engine.pdf.

to tackle this challenge: What criteria can be used? How to understand the modelling perspective adopted in an ontology? How to merge two ontologies? To what extend do two ontologies reflect the same conceptualisation of a given domain? Can we describe the differences in relation to level of detail, compatibility, key concepts and coverage? Are the differences artificial shifts (i.e. consequences of technical or terminological choices) or profound semantic differences that reflect diverging conceptualisations? A major area of research work focused on the development of algorithms and tools to identify and solve differences between ontologies (i.e. analysis of differences between terms, concepts, definitions). Moreover, some research studies bear on global ontologies comparison providing an overview on commonalities and differences. One interesting research direction is to best exploit ontology visualisation results. Visualisation software tools applied to large ontologies provide global views and some of them specifically enable the identification of the ontology main concepts.

The notion of knowledge pattern, directly based on the *design patterns* used in software engineering, aims at reducing the significant difficulties occurring when designing large ontologies or when adapting reusable ontologies. Knowledge pattern has been introduced in Ontology Engineering by Clark et al. (2000) and then in semantic web applications by Gangemi et al. (2004), Rector and Rogers (2004) and Svatek (2004). Knowledge patterns are recurrent and shared representations of knowledge, explicitly represented as generic models and validated through a cooperative process by the research community. Therefore, they are easily reusable after a further processing by symbolic relabelling required to obtain specific representations. Knowledge patterns provide "building blocks" that ensure faster ontology design.[30] Moreover, they lead to better results by solving, for instance, design problems and content-related issues independently of the conceptualisation (Gangemi 2005). Additionally, patterns can facilitate the application of good modelling practices (Pan et al. 2007). The "Semantic Web Best Practices and Deployment" W3C working group promotes the use of ODPs to design ontologies. A library of knowledge patterns is provided in the settings of the European NeOn project. It includes structural, correspondence, content, reasoning, presentation and lexico-syntactic patterns (Presutti et al. 2008). The eXtreme Design (XD) methodology provides guidelines for pattern-based ontology design (Daga et al. 2010).[31]

Reuse of knowledge models requires also to manage their integration within the system under development in order to allow for an easy communication between the reused model and the other models. Although ontologies aim at facilitating interoperability between applications they usually originate from different designers and refer to various modelling perspectives. Therefore, their use within a same application requires to solve specific issues associated with semantic heterogeneity. In practice, the same terms may be used to label different concepts in each reused ontology or ontology module; the same concepts may have different labels; and a particular concept can be characterised by different features in each model. Facing this het-

[30] Referred to as *Ontology Design Pattern* or ODP.

[31] http://ontologydesignpatterns.org/wiki/Main_Page.

erogeneity, significant progress has been made on *model reconciliation*. Models can be reconciled at two different levels. At the schema level, reconciliation consists in identifying correspondences or mappings between semantically-related entities of two ontologies. In the past years, considerable efforts have been made to build ontology alignment tools (Euzenat and Shvaiko 2013), many of which are available on the internet such as OnAGUI[32] or TAXOMAP (Hamdi et al. 2009). Each year since 2004, OAEI international campaigns aim at comparing ontology matching systems. At the data level, reconciliation consists in determining if two data descriptions refer to the same entity of the real world (e.g. the same person or the same hotel). This problem is referred to as *reference reconciliation* (Saïs et al. 2009) and it is close to coreference resolution in NLP.

3.5 Knowledge Representation in Models

Even though designing knowledge representation languages is not KE's main objective, researchers, when specifying knowledge and models, contribute to develop, evaluate and evolve these languages within normalisation groups, such as W3C. Knowledge representation languages as well as modelling languages were first dedicated to problem-solving and reasoning. Then, they related to ontologies (cf. Sects. 2, 2.1, 2.2); nowadays knowledge representation languages are back hand in hand with reasoning.

In the 1980s, ontology representation languages successfully took advantage of logic and conceptual graphs (Sowa 1984). Conceptual graphs could provide both a logic formalisation and a graphical symbolism when no powerful HMI was available to display semantic networks or trees, and to deploy or close them upon request. OWL was later developed as an evolution of DAML+OIL,[33] a language resulting from the merge of the DAML[34] and OIL project outcomes (Fensel et al. 2001). Drawn also on description logic (cf. Sect. I.5), and defined as a layer above XML, OWL became stable and included three languages OWL Lite, OWL-DL, OWL-full according to the W3C recommendations. Each of these three languages specificities results from the trade-off representativity *versus* calculability. In 2007, OWL was extended with new features. A new version, called OWL 2, was formally defined in 2012 with three sub-languages[35] (called *profiles*) offering distinct advantages, computational properties or implementation possibilities, in particular application scenarios: OWL 2 EL enables polynomial time algorithms for all standard reasoning tasks; OWL 2 QL enables conjunctive queries to be answered in LogSpace;

[32]https://github.com/lmazuel/onagui.

[33]http://www.w3.org/TR/daml+oil-reference.

[34]http://www.daml.org/.

[35]https://www.w3.org/TR/owl2-new-features/#F15:_OWL_2_EL.2C_OWL_2_QL.2C_OWL_2_RL.

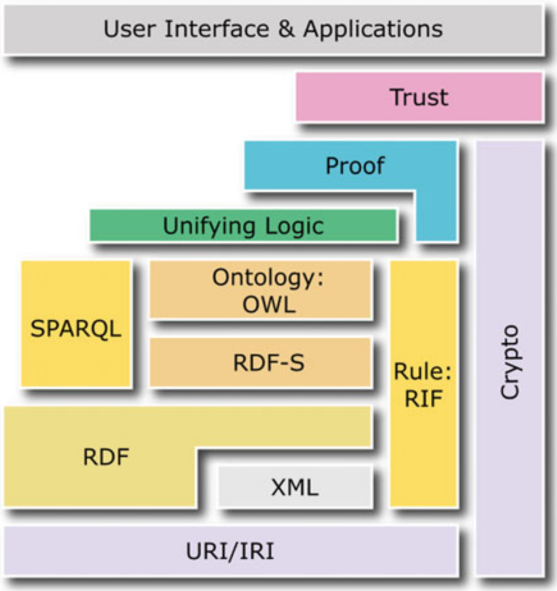

Fig. 2 The layer *cake* of the semantic Web as proposed in 2009

OWL 2 RL enables the implementation of polynomial time reasoning algorithms using rule-extended database technologies.

In the Semantic Web Stack proposed by Tim B. Lee (cf. Fig. 2), representing the stacking order of the Semantic Web languages, we can notice that RDF,[36] located in the bottom part, is the basic language of the Semantic Web. RDF is the common ground to all the languages of interest for KE (i.e. RDF, RDF-S, OWL, SPARQL and RIF). These languages allow applications to consistently use ontologies and associated rules. RDF is a simple language to express data models as a graph where nodes are web resources and edges properties. RDF Schema[37] is a semantic extension of RDF. It is written in RDF and provides mechanisms to structure data models, by describing groups of related resources and the relationships between these resources. OWL is another and more expressive extension allowing a better integration of ontologies and easier inferences. SPARQL[38] is an RDF semantic query language for databases, able to retrieve and manipulate data stored in RDF format. RIF[39] (Rule Interchange Format) is the rule layer in the Semantic Web Stack. RIF is not a rule language but rather a standard for exchanging rules among rule systems. Other rule languages may apply on ontologies, like SWRL,[40] or Description Logic Programs (DLP)[41] (Hitzler et al. 2005). None of them is proposed as a standard for

[36] https://www.w3.org/RDF/.
[37] https://www.w3.org/TR/rdf-schema/.
[38] https://www.w3.org/TR/rdf-sparql-query/.
[39] https://www.w3.org/TR/rif-overview/.
[40] http://www.w3.org/Submission/SWRL/.
[41] http://logic.aifb.uni-karlsruhe.de/wiki/DLP.

the semantic web, because the W3C assumes that a single language would not satisfy the needs of many popular paradigms for using rules in knowledge representation.

Another W3C recommendation defined as an application of RDF is SKOS[42] (for Simple Knowledge Organisation System). SKOS provides a model for expressing the basic structure and content of concept schemes such as thesauri, taxonomies, folksonomies, and other similar types of controlled vocabulary. In basic SKOS, conceptual resources (concepts) are related to each other in informal hierarchies but no logical inference is possible. Using SKOS, generalisation versus specialisation, (*broader-than* and *narrower-than* - – BT/NT) relations that are very often used in thesaurus can be represented without logical inferences associated to the subsumption relationship in OWL.

SKOS was even more necessary in that logical inferences based on the subsumption relationship are only valid if ontologies comply with the associated constraints (whereas such relationship is not valid on thesaurus). Furthermore, the applications using thesaurus and ontologies are increasingly efficient and the resources themselves – i.e. thesaurus and ontologies – are involved in the development processes using different knowledge representation languages at different steps in the development process and not always as intended by the language designers. For instance, a thesaurus and an ontology jointly used in an application can be modelled in OWL for that application. However, one could be originally developed in SKOS and the other one in OWL, and they could further be distributed in a format like CTS2.[43]

4 Methodological Issues and Today's Applications

The current KE challenges are both methodological and application oriented. A few founding principles tackle those issues and provide a general framework:

- The need for a multidisciplinary approach taking into account the recommendations of other disciplines such as cognitive psychology, ergonomics, management, linguistics, information retrieval, natural language processing or document management.
- The importance of a thorough modelling approach, bringing together different models whenever required during the system development process.
- The need to consider upstream the system ergonomic design, prior to any modelling stage; more specifically, the targeted uses of the system should be taken into account as well as its integration in the broader information processing architecture.

KE-related applications form a vast field of research, experimentation and transfer of AI technologies in which innovative methods must be developed. The articulation between methodology and applications guides the stakes described below.

[42]https://www.w3.org/TR/2009/REC-skos-reference-20090818/.

[43]http://www.3mtcs.com/resources/hl7cts.

4.1 *Linking Language, Knowledge and Media*

As said in Sect. 3.1, natural language is an ideal vector of knowledge, and written natural language is now a good support for knowledge extraction thanks to recent advances in NLP and machine learning techniques. To represent and manage knowledge from text, KE has to deal with various interdisciplinary methodological issues that appear in concordance with classes of applications related to various media.

4.1.1 Designing Problem-Solving Models and Ontologies from Natural Language in Textual Documents

In the 1990s, the first KE studies on knowledge acquisition for expert systems focused on text to identify heuristic knowledge and more or less explicitly explain human reasoning. At that time, text sources were either existing documents or documents elaborated for modelling purposes, such as transcriptions of interviews. Later, the focus on domain ontologies accentuated the sometimes provisional dissociation between the heuristic reasoning and the description of the concepts (and vocabulary) used by these heuristics. Subsequently, at the end of the 1990s, under the impetus of research studies like the one of the French TIA Group, textual corpora generated in relation with an activity were used to help design ontologies for support systems of this same activity. Thus textual corpora were considered as a complementary or alternative source of knowledge to experts and specialists in the field. Processing such corpora requires not only NLP tools but also platforms able to use the result of these tools to design ontologies, terminologies or any conceptual scheme. (cf. Sect. 3.3.2).

Moreover, in this perspective, the document as such is a valuable knowledge conveyer in its own right. The management of documents produced and used in the individual and collective activity, but also, as such, the management of documentary collections (images, sounds, videos) is of interest to KE. KE can then rely on document management technologies that support the sharing, dissemination, archiving, indexing, structuring or classification of documents or document flows. A major difficulty is to select the right documents in order to best meet the users' needs and to find the useful task supports (including knowledge). Because more and more KE projects integrate document management in a large variety of forms, researchers in the field cannot free themselves from an in-depth reflection on the notion of a document, particularly a digital document. To this end, several researchers contributed to the work of the multidisciplinary thematic network on the document (RTP-DOC) and its productions (Pédauque 2003, 2005).

4.1.2 Information Retrieval with Ontologies

Thanks to the Semantic Web, where ontologies provide metadata for indexing documents, ontologies are now at the heart of Information Retrieval (IR) applications. In

420043_1_En_23_Chapter ✓ TYPESET DISK LE ✓ CP Disp.:8/7/2019 Pages: 36 Layout: T1-Standard

this context, they make it easier to access to relevant resources, because they can be used to link and integrate distributed and heterogeneous sources at both the schema and data level. Ontologies are also a means to query multiple sources using a unified vocabulary, to enrich queries with close concepts or synonym terms, to filter out and classify the query results. Given that thesauri are already in use in this field, this line of work obviously leads to compare the gains and limitations of ontologies with those of thesauri or terminologies and to evaluate their respective contributions to IR. These analyses contribute to specify which kind of ontology is more likely to support IR: those having a strong linguistic component, with at least many terms labeling the concepts. As a consequence, a new need emerged: the implementation of application environments where ontologies and thesaurus co-exist to serve the purpose of IR (Vandenbussche and Charlet 2009).

4.2 Coping with Data Explosion

For nearly 20 years, the amount of available data exploded. In a parallel movement, the Semantic Web turned out to be a web of Data in addition to a web a document. This means that the semantics should also be brought to data by labeling them with ontology concepts. Thus applications address increasingly numerous and diverse data that generate new needs in particular for their description and their integration. The so-called *Big Data* is frequently characterised by the four (or more) versus (4Vs): Volume, Velocity, Variety, Veracity. Velocity has to do with efficiency and calculability of knowledge representation, which is out of the scope of this chapter. In the following paragraphs, we explore the three others characteristics: Veracity, Variety, and, for the Volume problematic, we focus more specifically on the question of the size of designed models.

4.2.1 Volume

The description of these very numerous data requires the development of models in which the amount of information to be taken into account can be large enough to open new perspectives to statistical approaches and models. In order to maintain the use and management of symbolic models, the challenge is to be able to design models of very large size, for example by reducing the amount of information to be taken into account simultaneously. In this way, work on ontology modularity aims at designing very large ontologies needed for applications, and to consider these ontologies as sets of (more or less independent) modules. Modularity, in the general sense of the word, refers to the perception of a large knowledge repository (i.e. an ontology, a knowledge or data base) as a set of smaller repositories. Although the concept of modularity is widely used in computer science, it is a relatively new idea in KE. For example, the Knowledge Web project[44] (2004–2007) provided guidelines to design modular ontologies (Stuckenschmidt et al. 2009). This project showed the diversity of views on modularity and pointed out the important research directions to be developed: guidelines to design modules (how to determine a coherent and meaningful set of concepts, relationships, axioms and instances), metadata to describe, to select and to use or re-use modules, specification of how they can be linked to one another, their composition and their reuse in different contexts. Managing a large mass of data in a distributed context can also lead to designing on a set of existing ontologies that need to be redesigned, aligned, transformed into modules or integrated with non-ontological resources such as databases, folksonomies or thesauri. The networked ontology construction method defined by the NeOn[45] project (2006–2010) includes a support for cooperative design and takes into account the dynamic and evolutionary features of ontologies (Gómez-Pérez and Suárez-Figueroa 2009), which are major issues for the development of large ontology-based applications.

4.2.2 Variety or Managing Knowledge Integration Through Ontologies

Both in the fields of databases and information retrieval, ontologies are experimented as a promising solution for data integration. When integrating data from multiple and heterogeneous sources, ontologies can help to understand and interpret data belonging to the same domain but represented in heterogeneous structures. Then ontologies are also a good support to relate them more easily (Assele Kama et al. 2010). In some domains, such as geography, few ontologies are practically available for data integration (Buccella et al. 2009) or they describe targeted domains, such as Towntology for planning and urbanism (Roussey et al. 2004) or FoDoMuSt in the field of image processing (Brisson et al. 2007). The challenge then consists in designing useful ontologies.

In other domains, like agriculture or medicine, ontologies exist but are very large and therefore difficult to exploit. In this case, the challenge is to enable the understanding of their content in order to help extract the relevant subset for an application. In the medical field, many classifications contain several tens of thousands of concepts and an ontology includes several hundred thousand concepts. Ontology reuse and management reaches an additional level of complexity: ontologies are developed to represent knowledge of a precise sub-domain, we speak of *Interface ontology*. Other large ontologies are developed to provide broad representations and to serve as references for future epidemiological studies, we speak of *Reference ontology* (Rosenbloom et al. 2006). In this context, the best known models are SNOMED-CT that covers the whole medical domain (Spackman 2005) and FMA for representing human anatomy in whole (Rosse and Mejino 2003). Between the two types of ontologies, we need alignment services and the possibility of extracting the relevant subsets for a target system. This is what a standard like CTS2 allows (cf. Sect. 3.5).

[44] http://cordis.europa.eu/ist/kct/knowledgeweb_synopsis.htm.

[45] http://www.neon-project.org/.

This context, reinforced by the need to exploit diversified knowledge or several partial models (or modules), requires to face the problem of heterogeneity between models/ontologies/knowledge, and motivates the current interest in semantic interoperability. Research work on semantic interoperability bears on automatic mapping tools that set links between elements of semantically heterogeneous concept schemes, ontologies or other knowledge sources. They define processes for schema matching, ontology alignment (cf. Sect. 3.4), or data reconciliation. For instance, recent medical studies have tried to integrate most of the knowledge needed to make a diagnosis – e.g. clinical, imaging, genomics knowledge – thanks to a pivotal ontology based on various available ontologies or models (Hochheiser et al. 2016; Sarntivijai et al. 2016).

4.2.3 Veracity

Veracity points out, with a step backwards, two things.

The quality of data is often a problem. For example, in medicine, the medical staff generally inputs data into information systems through poor interfaces, with little time, in difficult working conditions or with little involvement. As a consequence, the data quality is poor too. In a KE point of view, it is important to stress that quality ontologies, and quality Knowledge Organisation Systems in general, are necessary.

Secondly, it appears that medical data are coded (or tagged with concepts) with precise goals and strict coding rules. This process involves a reduction of the meaning, and raises difficulty when interpreting the data, which often requires to read again the original text or resource. Indeed, when reusing data in a new context or when trying to merge it with other data, we observe that the data is biased by the first context. It is then necessary to closely analyse the bias and to check that it can be taken into account or even compensated for in another way. Knowledge engineers must be aware of these limitations and anticipate them before data reuse.

4.3 Managing Distributed Data

The web and web standards have greatly changed the way data is distributed. In particular, new types of systems, web services, rely on a new communication protocol between machines. Thanks to web services, the Web became a distributed computing device where programs (services) can interact intelligently by being able to automatically discover other services, to negotiate among themselves and to compose themselves into more complex services. A considerable amount of knowledge is mandatory to get intelligible services from machines. When added a knowledge base, web services become *semantic web service*.

Semantic web services are the bricks to create a semantic Web of services whose properties, capabilities, interfaces and effects are described in an unambiguous way and can be exploited by machines. The semantics thus expressed must facilitate the automatic management of services. Semantic web services are essential for the effective use of web services in industrial applications. However, they still raise a number of issues for the research community, including for the KE field because they use ontologies to explain which service they provide to other services or to end users. Semantic modelling contributes to evaluate the quality of a Web service and to take it into account in the process of discovery or composition of services. Peer-to-peer (P2P) systems have also grown significantly, and a substantial body of research work has recently sought to improve the search function in unstructured systems by replacing random routing with semantically guided routing. Several dimensions of the problem are analysed: Which semantics should be remembered? Which representation to adopt? How to design it? What is shared among peers? How to use semantics? How to disseminate it? These issues remained unresolved and have been brought into sharper focus by KE.

4.4 Leveraging New Knowledge Sources

Two knowledge sources currently raise major challenges: data from the Web 2.0 and data from the Web data-bases (web of data).

The Web 2.0 or social Web (OReilly 2007) devotes a considerable attention to users compared to the Web in its initial version, by allowing them to become active. Both authors and actors, Internet users can use the web 2.0 tools to store, implement and manage their own content and share it. These tools include blogs, social networks, collaborative sites, linking platforms, and on-line sharing services. These tools and services are increasingly used in organisations. However, the software tools managing these contents have their own data format and they are increasingly distributed and heterogeneous. These features raise important problems of information integration, reliable identification of the authors or history tracking to name but a few. Similarly, *tagging* or *labeling*[46] is a common practice to characterize and group similar contents and to facilitate data search. This process presents several limitations due to the ambiguity and heterogeneity of the labels, called *tags*. Enterprise 2.0 systems (McAfee 2006) recently tend to develop as a field of experimentation and promotion for KE techniques. It enables a kind of renewal within the KE domain by making new proposals for facilitating navigation, querying or retrieval. As proposed by Tim Berners-Lee, linked Web data refer to an RDF-based publication and interconnection of structured data on the Web, based on the RDF model. Tim Berners-Lee talks about a *Web of data*. It thus promotes a W3C project that goes in this direction, i.e. the Linking Open Data (LOD). The Web of Data, following the web of documents, intends to face the flood of information by connecting the data. Linked data has the advantage of providing a single, standardised access mechanism rather than using different interface and result formats. Data sources can be more easily searched

[46] I.e. content indexing with user's metadata. The sets of labels then form *folksonomies*.

by search engines, accessed using generic data browsers, and linked to different data sources.

The number of data published according to the principles of linked data is growing rapidly (we are talking about billions of RDF triplets available on the Internet). The site http://lov.okfn.org/dataset/lov/ gives a snapshot of existing vocabularies (more than 600) and highlights the numerous mutual reuse of terms between these vocabularies. Among this large number of data sources, DBPedia[47] structures the content of Wikipedia[48] into RDF triples so as to make the information of the encyclopedia reusable. DPpedia is a very powerful source as it is interconnected with other data sources, such as Geonames[49] and MusicBrainz[50]) and it has been linked to even larger data sets like YAGO[51] (Rebele et al. 2016) or BabelNet[52] (Navigli and Ponzetto 2012). These large generic knowledge bases are also used by search engines to display structured content in response to users' queries. Because of they propose unambiguous and linked vocabularies, these masses of data represent promising sources for KE.

4.5 *Coping with Knowledge Evolution*

The dynamic nature of the data on the Web gives rise to a multitude of problems related to the description and analysis of the evolution of such data. The existing models of knowledge representation are inadequately addressing the challenges of data evolution and, above all, they do not benefit from any adaptive mechanism that would allow them to rigorously follow the evolutions of a domain. Research work on ontology evolution underlines how much the Semantic Web and KE communities need to find appropriate solutions to this complex issue. Early studies defined the stages of an evolution process (Noy and Klein 2004; Stojanovic 2004), they specified a typology of changes (Plessers et al. 2007) and change descriptions. Other works proposed mechanisms, sometimes borrowed to belief revision (Flouris 2006) to keep the modified ontology consistent and logically sound (Haase and Stojanovic 2005) and defined how to propagate changes in distributed ontologies and in the applications that use them (Stuckenschmidt and Klein 2003). With similar purposes to ontology engineering, ontology evolution can be fed thanks to the knowledge identified in textual documents using NLP tools (Buitelaar and Cimiano 2008) and relying on document structure, like in (Nederstigt et al. 2014). More recently, when the ontology is used to generate semantic annotations of text, research studies deal

[47] http://wiki.dbpedia.org/.
[48] https://fr.wikipedia.org.
[49] http://www.geonames.org/.
[50] https://musicbrainz.org/.
[51] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/.
[52] http://babelnet.org/.

420043_1_En_23_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:8/7/2019 Pages: 36 Layout: T1-Standard

with the evolution of these semantic annotations when the textual corpus or when the indexing vocabularies evolve (Tissaoui et al. 2011; Da Silveira et al. 2015; Cardoso et al. 2016).

Zablith et al. (2015) propose a recent overview of the major trend in this domain. Characterizing and representing domain data evolution raises issues both at the data level (Stefanidis et al. 2016) and at the model scheme level (Guelfi et al. 2010). Ontology evolution remains a hard issue, even at the era of machine learning, because a statistic processing of a massive amount of documents is relevant for building large knowledge bases like DBpedia, but produces poor results when trying to fix errors or to identify local changes in an existing model. Processing large amounts of data is much more appropriate to feed and update the data level in knowledge bases, which corresponds to instances of ontological classes.

4.6 Collective Versus Personal Knowledge

Most of the previous approaches place little emphasis on the social dimension of knowledge management. This dimension is strong enough in some professional communities to consider them as communities of interest or as communities of practices. Communities of practices designate social groups in which learning processes emerge through the sharing of networked knowledge. KE models need to capture these learning processes or to integrate them into their knowledge management process. To this end, Lewkowicz and Zacklad (2001) propose a new form of knowledge management based on the structuring of collective interactions. This approach aims at better using of the shared knowledge, at facilitating its reuse, the knowledge of an organisation being considered as above all a matter of collective competence.

The identification of communities of interest that emerged thanks to the development of Web 2.0 or the analysis of users' digital traces sharing similar thematic information implies the representation of individual knowledge about the fields of interest and activities of their members, together with the collective dimension of knowledge. This collective dimension is the focus of the Computer Supported Cooperative Work (CSCW) research community, that designs specific solutions to manage collective and in-use knowledge. For instance, M. Zacklad proposes a conceptual model mid-way between thesauri and formal ontologies, called *semiotic ontologies*, that should be more easily shared by a working community in an information retrieval framework (Zacklad 2007). Conversely, more and more software systems and Web interfaces are designed to be context sensitive or user customised. To do so, they adapt to the user profile, environment or interactions with the system, which requires the acquisition, the modelling and the processing of the interaction contexts (Garlatti and Prié 2004).

4.7 Model Quality Assessment

Finally, a fundamental question for KE concerns the quality assessment of the models used and the results produced. The use of poor quality knowledge may lead to errors, duplications and inconsistencies that must be avoided. Beyond its interest in research, the theme of quality has become critical with the deployment of systems in companies.

The quality of the models/ontologies can be guaranteed methodologically, when the ontology was designed following a rigorous method based on the theoretical and philosophical foundations of what an ontology is (such as the methods presented in Sect. 5). Other methodological works aim to move from manual and approximative approaches, the cost and duration of which are difficult to estimate, to more systematic, equipped and better controlled processes. Of course, they focus on reuse such as Methontology (Gómez-Pérez et al. 2007) and NEON in Suárez-Figueroa et al. (2012), on practical guidelines (Noy and Hafner 1997) or on systematic text analysis using NLP tools and modelling platforms such as Terminae (Aussenac-Gilles et al. 2000) or GATE and methods listed in Maedche (2002). In the case of Brank et al. (2005), a state of the art classifies the ontology evaluation techniques into four categories: (1) syntactic evaluations check whether the model complies the syntactic rules of a reference language (RDF, OWL, …) such as Maedche and Staab (2002), (2) in-use evaluations test the ontology when used by a targeted system, e.g. Porzel and Malaka (2004) (3) comparison with a reference source in the domain (either a gold model or a representative set of textual documents), such as Brewster et al. (2004) or, finally (4) human evaluation tests how well the ontology meets a set of predefined criteria, standards or needs, for example Lozano-Tello and Gomez-Perez (2004). Moreover, in Brank et al. (2005), validation approaches are organised into six levels: lexical level, level of taxonomic relations, level of other semantic relationships, application level (looking how the ontology impacts on the system that uses it), context level (how the ontology is reused by or reuses another ontology), syntactic level or, finally, the level of design principles. Practically, it may be easier to evaluate an ontology level by level because of its complexity.

5 Conclusion

KE has undergone successive changes of direction. This research field constantly evolves from the inside (experimenting new analyses, new perspectives, original ways of posing problems, new theoretical concepts) and from outside (targeting new types of applications, dealing with new types of data, in particular with the upheavals of the Web, integrating the contributions of other disciplines that come to bring new methods and concepts). Over the years, these developments gradually broadened the scope of KE. Each new proposed theoretical framework includes parts of the previous work. Even if some changes of perspective correspond to actual breaks, the results of the domain complement each other over time and can be taken from a new angle when the context evolves.

For a long time, KE has been interested in producing knowledge models in a well-structured process under the control of knowledge engineers. The resulting models, generally complex, were used in specific applications. Today, applications in which knowledge is used as support for reasoning or activity have become much more diversified. Since 2000, they have been devoted to knowledge management in the broadest sense, including semantic information retrieval, navigation aids, decision support, and many semantic Web applications. This enlargement continues and new fields of application are still emerging, posing the problems of KE in new terms.

Thus, in the age of ubiquitous computing, it is the living room, the train, the automobile, the workshop, the classroom or meeting room, the smallest kitchen device that become “smart” tools. Within these tools, a dynamic process is required to continuously acquire context knowledge on the flow from a wide variety of sources (sensors, databases, the Internet, users with various profiles). In addition, these intelligent tools must have a pro-active behaviour that enables them to initiate communication or action based on their understanding of the current situation and on their goals. So, for example, phones know where we are at a given time and become capable of automating some operations, such as when taking pictures, labeling them with geographic and temporal metadata.

The last decade has seen a major transformation in the way individuals interact and exchange. Information is now co-produced, shared, filed and evaluated on the Web by thousands of people. These uses and the underlying technologies are known as Web 2.0. Web 3.0 is the latest evolution to date that combines the social web and the semantic technologies of the semantic Web. In the context of communities of interest or practices where spontaneous emergence and activity are allowed by these evolutions of the Web, KE and knowledge management are thus major stakes of the future decade.

Finally, KE must feed and evaluate all these new developments, compare them with previous models (reasoning models, rules bases), estimate the need to use ontologies and their alignment to type or organise data, to define new techniques and languages if necessary, to justify the use of metadata to enrich and reuse data, and so on. The speed of Web evolutions can be seen as a crazy accelerator of the research pace or as an alarm that invites us to step back and pose the problems at a higher abstraction level, necessarily interdisciplinary, in order to better qualify the essence of knowledge, their dissemination and their formalisation for digital processing.

References

Assele Kama A, Mels G, Choquet R, Charlet J, Jaulent M-C (2010) Une approche ontologique pour l’exploitation de données cliniques. In: Despres S (ed) Acte des 21èmes Journées Francophones d’Ingénierie des Connaissances, Nîmes, France. Ecole des Mines d’Alès, pp 183–194

Aussenac N (1989) Conception d'une méthodologie et d'un outil d'acquisition de connaissances expertes. Thése de doctorat, Université Paul Sabatier, Toulouse, France

Aussenac-Gilles N, Jacques M-P (2008) Designing and evaluating patterns for relation acquisition from texts with caméléon. Terminology 14(1):45–73 (special issue on Pattern-based approaches to semantic relations)

Aussenac-Gilles N, Krivine J, Sallantin J (1992) Editorial du numéro spécial Acquisition des connaissances. Revue d'Intelligence Artificielle 6(2):7–18

Aussenac-Gilles N, Bourigault D, Condamines A (1995) How can knowledge acquisition benefit from terminology? In: Proceedings of the 9th knowledge acquisition workshop, Banff, University of Calgary (CA)

Aussenac-Gilles N, Laublet P, Reynaud C (eds) (1996) Acquisition et ingénierie des connaissances: tendances actuelles. Cepadues Editions, Toulouse

Aussenac-Gilles N, Biébow B, Szulman S (2000) Revisiting ontology design: a method based on corpus analysis. In: 12th international conference on knowledge engineering and knowledge management, Juans-Les-Pins, 03/10/2000–06/10/2000. Springer, Heidelberg, pp 172–188

Aussenac-Gilles N, Desprès S, Szulman S (2008) The TERMINAE method and platform for ontology engineering from texts. Ontology learning and population: bridging the gap between text and knowledge, pp 199–223

Bachimont B, Isaac A, Troncy R (2002) Semantic commitment for designing ontologies: a proposal. In: EKAW, pp 114–121

Barriere C, Agbago A (2006) Terminoweb: a software environment for term study in rich contexts. In: International conference on terminology, standardization and technology transfer, Beijing, pp 103–113

Beys B, Benjamins V, Van Heijst G (1996) Remedying the reusability - usability trade-off for problem-solving methods. In: Gaines B, Musen M (eds) Proceedings of the 10th knowledge acquisition workshop (KAW), Banff, Canada, pp 2–1/2-20

Bourigault D (2002) Upery: un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. In: Actes de la 9ème conférence annuelle sur le Traitement Automatique des Langues (TALN 2002), Nancy, France, pp 75–84

Bourigault D, Slodzian M (1999) Pour une terminologie textuelle. Terminol Nouv 19:29–32

Bourigault D, Aussenac-Gilles N, Charlet J (2004) Construction de ressources terminologiques ou ontologiques à partir de textes: un cadre unificateur pour trois études de cas. Revue d'Intelligence Artificielle 18(1/2004):87–110

Brank J, Grobelnik M, Mladenic D (2005) A survey of ontology evaluation techniques. In: Data mining and data warehouses conference (SIKDD), Lubiana, Slovénie

Brewster C, Alani H, Dasmahapatra S, Wilks Y (2004) Data driven ontology evaluation. In: LREC

Brisson R, Boussaid O, Gançarski P, Puissant A, Durand N (2007) Navigation et appariement d'objets géographiques dans une ontologie. In: EGC, pp 391–396

Buccella A, Cechich A, Fillottrani P (2009) Ontology-driven geographic information integration: a survey of current approaches. Comput Geosci 35(4):710–723 (Geoscience knowledge representation in cyberinfrastructure)

Buitelaar P, Cimiano P (eds) (2008) Proceedings of the 2008 conference on ontology learning and population: bridging the gap between text and knowledge, Amsterdam, The Netherlands. IOS Press

Camilleri G, Soubie J-L, Zaraté P (2008) A replanning support for critical decision making situations: a modelling approach. In: Intelligent decision making: an AI-based approach, pp 173–192

Cardoso SD, Pruski C, Da Silveira M, Ying-Chi L, Anika G, Erhard R, Reynaud-Delaître C (2016) Leveraging the impact of ontology evolution on semantic annotations. In: Knowledge engineering and knowledge management - 20th international conference, EKAW, Bologna, Italy

Chandrasekaran B (1983) Towards a taxonomy of problem solving types. AI Mag 4(1):9–17

Charlet J (1991) ACTE: a strategic knowledge acquisition method, pp 85–93

Charlet J, Zacklad M, Kassel G, Bourigault D (eds) (2000) Ingénierie des connaissances: Evolutions récentes et nouveaux défis. Eyrolles, Paris

Cimiano P, Völker J (2005) Text2onto. In: NLDB, pp 227–238
Cimiano P, Buitelaar P, Völker J (2010) Ontology construction. In: Indurkhya N, Damerau, FJ (eds) Handbook of natural language processing, 2nd edn. CRC Press, Taylor and Francis Group, Boca Raton. ISBN 978-1420085921
Clark P, Thompson JA, Porter BW (2000) Knowledge patterns. In: KR, pp 591–600
Condamines A (2002) Corpus analysis and conceptual relation patterns. Terminol. Int J Theor Appl Issues Spec Commun 8(1):141–162
Constant M, Dister A, Ermikanian L, Piron S (2008) Description linguistique pour le traitement automatique du français. Cahier du CENTAL
Cordier M-O, Reynaud C (1991) Knowledge acquisition techniques and second-generation expert systems. Appl Artif Intell 5(3):209–226
Da Silveira M, Dos Reis J, Pruski C (2015) Management of dynamic biomedical terminologies: current status and future challenges. Yearb Med Inform 24:125–133
Daga E, Blomqvist E, Gangemi A, Montiel E, Nikitina N, Presutti V, Villazon-Terrazas B (2010) NeOn project: NeOn D2.5.2. Pattern-based ontology design: methodology and software report. Rapport de contrat
Darses F, Montmollin M (eds) (2006) L’ergonomie. La Découverte - Col. Repères, Paris
Dieng-Kuntz R, Corby O, Gandon F, Gibouin A, Golebiowska JNM, Ribière M (eds) (2005) Knowledge management: Méthodes et outils pour la gestion des connaissances. Dunod
Drouin P (2003) Term extraction using non-technical corpora as a point of leverage. Terminology 9:99–117
Euzenat J, Shvaiko P (2013) Ontology matching, 2nd edn. Springer, Heidelberg
Fankam C, Bellatreche L, Hondjack D, Ameur YA, Pierra G (2009) Sisro, conception de bases de données à partir d’ontologies de domaine. Technique et Science Informatiques 28(10):1233–1261
Fensel D, Schnanegge R, Wielinga B (1996) Specification and verification of knowledge-based systems. In: Proceedings of the 10th knowledge acquisition workshop (KAW), Banff (Can). University of Calgary (Can)
Fensel D, van Harmelen F, Horrocks I, McGuinness DL, Patel-Schneider PF (2001) Oil: an ontology infrastructure for the semantic web. IEEE Intell Syst 16(2):38–45
Fernández-López M, Gómez-Pérez A (2002) Overview and analysis of methodologies for building ontologies. Knowl Eng Rev 17(2):129–156
Flouris G (2006) On belief change in ontology evolution. AI Commun 19(4):395–397
Gangemi A (2005) Ontology design patterns for semantic web content. In: International semantic web conference, pp 262–276
Gangemi A, Catanacci C, Battaglia M (2004) Inflammation ontology design pattern: an exercise in building a core biomedical ontology with descriptions and situations. In: Maria PD (ed) Ontologies in medecine. IOS Press, Amsterdam
Garlatti S, Prié Y (2004) Adaptation et personnalisation dans le web sémantique. Revue I3 - Numéro hors série Web Sémantique
Gómez-Pérez A, Suárez-Figueroa M-C (2009) Scenarios for building ontology networks within the neon methodology. In: K-CAP 2009, pp 183–184
Gómez-Pérez A, Fernández-López M, Corcho O (2007) Ontological engineering: with examples from the areas of knowledge management, e-commerce and the semantic web. (Advanced information and knowledge processing). Springer, New York
Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5:199–220
Guarino N, Welty CA (2004) An overview of ontoclean. Handbook on ontologies, pp 151–172
Guelfi N, Pruski C, Reynaud C (2010) Experimental assessment of the target adaptive ontology-based web search framework. In: NOTERE, pp 297–302
Haase P, Stojanovic L (2005) Consistent evolution of owl ontologies. In: ESWC, pp 182–197
Hamdi F, Safar B, Niraula N, Reynaud C (2009) TaxoMap in the OAEI 2009 alignment contest. In: The fourth international workshop on ontology matching, Chantilly, Washington DC, États-Unis

Hendler JA, Tate A, Drummond M (1990) AI planning: systems and techniques. AI Mag 11(2):61–77

Hitzler P, Sure Y, Studer R (2005) Description logic programs: a practical choice for the modelling of ontologies. In: Principles and practices of semantic web reasoning

Hochheiser H, Castine M, Harris D, Savova G, Jacobson RS (2016) An information model for computable cancer phenotypes. BMC Med Inform Decis Mak 16(1), 121

Jacob-Delouis I, Krivine J (1995) Lisa: un langage réflexif pour opérationnaliser les modèles d'expertise. revue d'Intelligence. Artificielle 9(1):53–88

Kamel M, Aussenac-Gilles N (2009) Utiliser la Structure du Document dans le Processus de Construction d' Ontologies (regular paper). In: L'Homme M-C, Szulman S (eds) Conférence Internationale sur la Terminologie et l'Intelligence Artificielle (TIA), Toulouse (France), 18–20/11/2009, page (on line). http://www.irit.fr/ (IRIT)

Kassel G (2002) Ontospec: une méthode de spécification semi-informelle d'ontologies. In: Actes d'IC, pp 75–87

Klinker G, Bhola G, Dallemagne G, Marquès D, Dermott M (1991) Usable and reusable programming constructs. Knowl Acquis 3:117–136

Laflaquière J, Prié Y, Mille A (2008) Ingénierie des traces numériques d'interaction comme inscriptions de connaissances. In: Actes d'IC, pp 183–195

Lewkowicz M, Zacklad M (2001) Une nouvelle forme de gestion des connaissances basée sur la structuration des interactions collectives. In: Grundstein M, Zacklad M (eds) Ingénierie et Capitalisation des connaissances. Hermes Sciences Europe LTD, pp 49–64

Lozano-Tello A, Gomez-Perez A (2004) ONTOMETRIC: a method to choose the appropriate ontology. J Database Manag 15(2):1–18

Luong PH (2007) Gestion de l'évolution d'un web sémantique d'entreprise. Thèse de doctorat, Ecole des Mines de Paris, Paris, France

Maedche A (2002) Ontology learning for the semantic web. Kluwer Academic Publisher, Boston

Maedche A, Staab S (2002) Measuring similarity between ontologies. In: EKAW, pp 251–263

Manning C, Schütze H (1999) Foundations of statistical natural language processing. MIT Press, Cambridge

Marcus S, McDermott J (1989) SALT: a knowledge acquisition language for propose and revise systems. Artif Intell 39(1):1–38

Maynard D, Funk A, Peters W (2009) SPRAT: a tool for automatic semantic pattern-based ontology population. In: International conference for digital libraries and the semantic web

McAfee A (2006) Enterprise 2.0: the dawn of emergent collaboration. MIT Sloan Manag Rev 47(3):21–28

Meyer I (2000) Extracting knowledge-rich contexts for terminography: a conceptual and methodological framework. In: Bourigault D, L'Homme M-C, Jacquemin C (eds) Recent advances in computational terminology

Musen MA, Eriksson H, Gennari JH, Tu SW, Puert AR (1994) PROTEGE-II: a suite of tools for development of intelligent systems for reusable components. In: Proceedings of the annual symposium on computer application in medical care

Navigli R, Ponzetto SP (2012) BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif Intell 193:217–250

Neches R, Fikes R, Finin TW, Gruber TR, Patil RS, Senator TE, Swartout WR (1991) Enabling technology for knowledge sharing. AI Mag 12(3):36–56

Nederstigt LJ, Aanen SS, Vandic D, Frasincar F (2014) FLOPPIES: a framework for large-scale ontology population of product information from tabular data in e-commerce stores. Decis Support Syst 59:296–311

Newell A (1982) The knowledge level. Artif Intell 18(1):87–127

Noy NF, Hafner CD (1997) The state of the art in ontology design: a survey and comparative review. AI Mag 18(3):53–74

Noy NF, Klein MCA (2004) Ontology evolution: not the same as schema evolution. Knowl Inf Syst 6(4):428–440

Oberle D, Volz R, Staab S, Motik B (2004) An extensible ontology software environment. Handbook on ontologies, pp 299–320

OReilly T (2007). What is web 2.0: design patterns and business models for the next generation of software. Commun. Strat (1):17

Pan J, Lancieri L, Maynard D, Gandon F, Cuel R, Leger A (2007) Success stories and best practices. Knowledge web deliverable d.1.4.2.v2

Pédauque RT (ed) (2003) Le document: forme, signe et medium les re-formulations du numérique. STIC-CNRS

Pédauque RT (ed) (2005) Le texte en jeu, permanence et transformations du document. STIC-SHS-CNRS

Plessers P, Troyer OD, Casteleyn S (2007) Understanding ontology evolution: a change detection approach. J Web Semant 5(1):39–49

Poibeau T, Kosseim L (2000) Proper name extraction from non-journalistic texts. In: CLIN, pp 144–157

Porzel R, Malaka R (2004) A task-based approach for ontology evaluation. In: ECAI - workshop on ontology, learning and population

Presutti V, Gangemi A, David S, De Cea GA, Surez-Figueroa MC (2008) NeOn project: NeOn D2.5.1. a library of ontology design patterns: reusable solutions for collaborative design of networked ontologies - NeOn project. Rapport de contrat

Puerta A, Egar JW, Tu SW, Musen M (1992) Method knowledge-acquisition shell for the automatic generation of knowledge-acquisition tools. Knowl Acquis 4(2):171–196

Rastier F (2009) Sémantique interprétative. PUF

Rebele T, Suchanek FM, Hoffart J, Biega J, Kuzey E, Weikum G (2016) YAGO: a multilingual knowledge base from wikipedia, wordnet, and geonames. In: The semantic web - ISWC 2016 - 15th international semantic web conference, Kobe, Japan, 17–21 October 2016, proceedings, part II, pp 177–185

Rector A, Rogers J (2004) Patterns, properties and minimizing commitment: reconstruction of the GALEN upper ontology in OWL. In: EKAW

Reymonet A, Thomas J, Aussenac-Gilles N (2007) Modélisation de ressources termino-ontologiques en owl. In: Actes d'IC, pp 169–181

Reynaud C, Aussenac-Gilles N, Tchounikine P, Trichet F (1997) The notion of role in conceptual modeling. In: EKAW, pp 221–236

Rosenbloom S, Miller RA, Johnson KB (2006) Interface terminologies: facilitating direct entry data into electronic health record systems. J Am Med Inform 13(3):277–288

Rosse C, Mejino JLV (2003) J Biomed Inform 36(6):478–500

Roussey C, Laurini R, Beaulieu C, Tardy Y, Zimmermann M (2004) Le projet towntology: un retour d'expérience pour la construction d'une ontologie urbaine. Revue Internationale de Géomatique 14(2):217–237

Saïs F, Pernelle N, Rousset M-C (2009) Combining a logical and a numerical method for data reconciliation. J Data Semant 12:66–94

Sarntivijai S, Vasant D, Jupp S, Saunders G, Bento AP, Gonzalez D, Betts J, Hasan S, Koscielny G, Dunham I, Parkinson H, Malone J (2016) Linking rare and common disease: mapping clinical disease-phenotypes to ontologies in therapeutic target validation. J Biomed Semant 7:8

Schreiber G, Wielinga B (eds) (1992) KADS: a principled approach to knowledge-based system development. Academic, London

Schreiber G, Wielinga BJ, Akkermans H, de Velde WV, Anjewierden A (1994) CML: the commonKADS conceptual modelling language. In: EKAW, pp 1–25

Schreiber G, Akkermans A, Anjewierden A, DeHoog R, Shadbolt N, Van de Velde W, Wielinga B (eds) (1999) Knowledge engineering and management: the CommonKADS methodology. MIT Press, Cambridge

Schutz A, Buitelaar P (2005) RelExt: a tool for relation extraction from text in ontology extension. In: International semantic web conference, pp 593–606

Shadbolt N, O'Hara K, Crow L (1999) The experimental evaluation of knowledge acquisition techniques and methods: history, problems and new directions. Int J Hum-Comput Study 51(4):729–755

Sowa JF (1984) Conceptual structures: information processing in mind and machine. Addison-Wesley, London

Spackman KA (2005) Rates of change in a large clinical terminology: three years experience with SNOMED clinical terms. In: AMIA annual symposium proceedings, pp 714–718

Steels L (1990) Components of expertise. AI Mag 11(2):28–49

Stefanidis K, Flouris G, Chrysakis I, Roussakis Y (2016) D2V - understanding the dynamics of evolving data: a case study in the life sciences. ERCIM News 2016(105)

Stefik M (1995) Introduction to knowledge systems. Morgan Kaufmann, San Francisco

Stojanovic L (2004) Methods and tools for ontology evolution. PhD thesis

Stuckenschmidt H, Klein MCA (2003) Integrity and change in modular ontologies. In: IJCAI, pp 900–908

Stuckenschmidt H, Parent C, Spaccapietra S (eds) (2009) Modular ontologies: concepts, theories and techniques for knowledge modularization, vol 5445. Lecture notes in computer science. Springer, Berlin

Studer R, Benjamins VR, Fensel D (1998) Knowledge engineering: principles and methods. Data Knowl Eng 25(1–2):161–197

Suárez-Figueroa M-C, Gómez-Pérez A, Motta E, Gangemi A (eds) (2012) Ontology engineering in a networked world. Springer, Berlin

Svatek V (2004) Design patterns for semantic web ontologies: motivation and discussion. In: Conference on business information systems

Szulman S, Charlet J, Aussenac-Gilles N, Nazarenko A, Sardet E, Teguiak V (2009) DAFOE: an ontology building platform from texts or thesauri. In: Dietz J (ed) Proceedings of the international joint conference on knowledge discovery, knowledge engineering and ontology development (KEOD 2009), Madeira (Portugal). Poster, pp 1–4

Tissaoui A, Aussenac-Gilles N, Hernandez N, Laublet P (2011) EVONTO - joint evolution of ontologies and semantic annotations. In: KEOD 2011 - proceedings of the international conference on knowledge engineering and ontology development, Paris, France, 26–29 October 2011, pp 226–231

Tu SW, Eriksson H, Gennari JH, Shahar Y, Musen MA (1995) Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools: application of PROTÉGÉ-II to protocol-based decision support. Artif Intell Med 7:257–289

Vandenbussche P-Y, Charlet J (2009) Méta-modèle général de description de ressources terminologiques et ontologiques. In: Actes d'IC, pp 193–204

Virbel J, Luc C (2001) Le modèle d'architecture textuelle: fondements et expérimenation XXII I(1):103–123

Zablith F, Antoniou G, d'Aquin M, Flouris G, Kondylakis H, Motta E, Plexousakis D, Sabou M (2015) Ontology evolution: a process-centric survey. Knowl Eng Rev 30(1):45–75

Zacklad M (2007) Classification, thesaurus, ontologies, folksonomies: comparaisons du point de vue de la recherche ouverte d'information (roi). In: Conférence CAIS/ACSI