



HAL
open science

Knowledge Engineering

Nathalie Aussenac-Gilles, Jean Charlet, Chantal Reynaud

► **To cite this version:**

Nathalie Aussenac-Gilles, Jean Charlet, Chantal Reynaud. Knowledge Engineering. Pierre Marquis; Odile Papini; Henri Prade. A Guided Tour of Artificial Intelligence Research, vol. I, 1: Knowledge Representation, Reasoning and Learning, Springer International Publishing, pp.733-768, 2020, 978-3-030-06163-0. 10.1007/978-3-030-06164-7_23 . hal-02857757

HAL Id: hal-02857757

<https://hal.science/hal-02857757v1>

Submitted on 25 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



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

© Springer Nature Switzerland AG 2020
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 bottleneck¹ 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

¹Knowledge 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.

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

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

81 2.2 Problem Solving Models

82 Problem solving models describe in an abstract way, using tasks and methods, the
83 reasoning process that the KBS must carry out. A task defines one or several goals
84 and sub-goals to be achieved by the system, and a method describes one of the ways
85 the task goals can be achieved. A task description also specifies the input and out-
86 put knowledge, constraints and resources required to perform the task. To describe
87 the way the system should behave to solve a problem, a hierarchy of tasks can be
88 defined, a general task being decomposed into several more specific tasks that specify
89 the sub-goals required to achieve the goal of the main task. Methods make explicit
90 how a goal can be reached thanks to an ordered sequence of operations. Methods
91 that decompose a task into sub-tasks are distinguished from methods that implement
92 a basic procedure to directly reach a particular goal. The distinction between tasks
93 and methods progressively emerged from research works after B. Chandrasekaran

94 proposed the notion of Generic Task (1983) and L. Steels proposed a componential
 95 modelling framework that included three types of components: tasks; methods and
 96 domain data models (1990). This distinction has been adopted to account for the
 97 reasoning process in many studies (Klinker et al. 1991; Puerta et al. 1992; Schreiber
 98 et al. 1994; Tu et al. 1995) because it provides a separate description of the targeted
 99 goal and the way to achieve it. Thus, several methods can be defined for one single
 100 task, making it easier to explicitly represent alternative ways to reach the same goal.
 101 This kind of model is similar to results established in task planning (Camilleri et al.
 102 2008; Hendler et al. 1990) where planning systems implement problem solving mod-
 103 els thanks to operational methods and tasks, as it is suggested in the CommonKADS
 104 methodology (Schreiber et al. 1999).

105 2.3 From Conceptual Models to Ontologies

106 Once solutions had been found to design explicit problem-solving models, build-
 107 ing the full conceptual model of an application consisted in reusing and adapting
 108 problem-solving components together with an abstract representation of domain data
 109 and concepts. Then an analysis of the domain knowledge was needed to establish
 110 a proper connection between each piece of the domain knowledge and the roles it
 111 played in problem solving (Reynaud et al. 1997). Domain knowledge models include
 112 two parts. The *domain ontology* forms the core part; it gathers concepts, i.e. class-
 113 sets of domain entities in a class/sub-class hierarchy, and relations between these
 114 classes, to which may be associated properties like constraints or rules. The second
 115 part extends this core with instances or entities belonging to the concepts classes,
 116 and relations between these entities. Thus an ontology defines a logical vocabulary
 117 to express domain facts and knowledge, in a formal way so that a system can use it
 118 for reasoning. Some concepts, called *primitive concepts*, are defined thanks to their
 119 situation in the concept hierarchy and thanks to properties that form necessary con-
 120 ditions for an entity to belong to this class. Other concepts, called *defined concepts*,
 121 are defined as classes equivalent to necessary and sufficient conditions that refer to
 122 properties and primitive concepts. The word *ontology* used to refer to a sub-field of
 123 philosophy. It has been first used in computer science, and particularly in AI, after
 124 the *Knowledge Sharing Effort* ARPA project (Neches et al. 1991) introduced it to
 125 refer to a structure describing the domain knowledge in a KBS. A little later, Gruber
 126 (1993) was the first to propose a definition of ontology in the field of KE. A more
 127 recent definition, proposed in Studer et al. (1998), is currently the acknowledged
 128 one:

129 *An ontology is a formal, explicit specification of a shared conceptualisation.*

130 *Conceptualisation refers to an abstract model of some phenomenon in the world by having*
 131 *identified the relevant concepts of that phenomenon. Explicit means that the type of concepts*
 132 *used, and the constraints on their use are explicitly defined. Formal refers to the fact that*
 133 *the ontology should be machine-readable.*

134 *Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not*
 135 *private of some individual, but accepted by a group.*

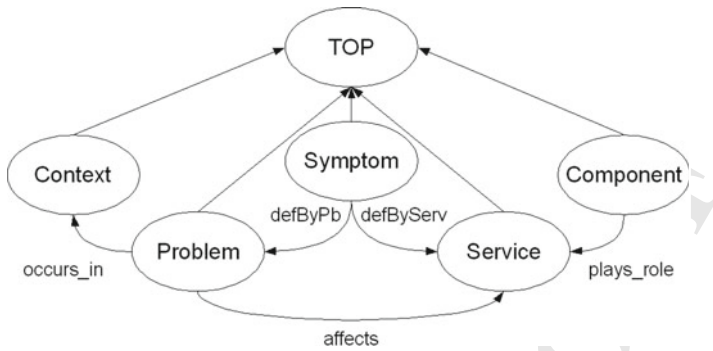


Fig. 1 High level concepts of an ontology used in the domain of electronic fault diagnosis

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

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

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

²Whereas 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.

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

- 169 • An *upper level ontology* or *top-ontology* is considered the highest level. It struc-
 170 tures knowledge with very general and abstract categories that are supposed to
 171 be universal and that are the fruit of philosophical studies on the nature of the
 172 main knowledge categories when formally representing human thinking in any
 173 domain. The major reference studies about top levels in ontologies are Sowa's
 174 *top-level categories*,³ SUMO,⁴ or DOLCE⁵ to name a few of them. As concluded
 175 by the SUO⁶ working group and the joint communiqué from the Upper Ontology
 176 Summit,⁷ trying to define a unique norm for high level categories is pointless as
 177 long as various philosophical schools or trends propose distinct ways to categorise
 178 the world entities. Top level ontologies are the anchor point of more specific lev-
 179 els (core ontologies and domain knowledge), and they are generic enough to be
 180 shared.
- 181 • A *core ontology* or *upper domain ontology* provides a domain description that
 182 defines the main concepts of a particular domain, together with properties and
 183 axioms applying on these concepts. For instance, a core ontology of medicine
 184 would contain concepts such as *diagnosis*, *sign*, *anatomic structure* and relations
 185 like *localisation* linking a pathology to the affected anatomic structure (cf. GFO-
 186 Bio⁸); in Law, the LKIF-Core⁹ ontology offers notions like *norm*, *legal action* and
 187 *statutory role*.
- 188 • A *domain ontology* describes the domain concepts practically handled by pro-
 189 fessionals and experts in everyday activities. It is the most specific kind of a
 190 knowledge model, and it becomes a knowledge base when instances of domain
 191 specific concepts are represented. Nevertheless, there may be no clear frontier
 192 between a *core-ontology* and an ontology of the same domain that includes the
 193 core one when both of them are designed within the same process. The distinction
 194 is more obvious when the domain ontology reuses and specialises an existing core
 195 ontology. Domain ontologies or the domain level of ontologies can be designed
 196 thanks to text-based approaches and reusing domain thesaurus or terminologies
 197 (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/>.

198 3 Issues and Major Results

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

215 3.1 Knowledge Sources

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

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

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

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

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

236 The historical evolution of knowledge-based information systems highlights vari-
 237 ous types of knowledge that were considered over the years: individual expert knowl-
 238 edge, in-use knowledge related to practice, activities and individual usage; knowledge
 239 about organisations, consensual and shared knowledge of an application field, com-
 240 mon sense knowledge, knowledge related to knowledge integration or distributed
 241 knowledge over the Web. It is to capture these various kinds of knowledge that new
 242 knowledge sources have been taken into account. Thus, documents have played an
 243 increasing role as more digital documents were available. Since the early works
 244 on knowledge acquisition for expert systems, KE relies on documents, in particu-
 245 lar textual documents, as they convey meaning and may contribute to reveal some
 246 knowledge. Documents are exploited for the language and information they contain,
 247 which is complementary or an alternative to interviews of domain experts or special-
 248 ists. Data can also become knowledge sources thanks to knowledge or information
 249 extraction processes from data or data mining. Last, components of existing knowl-
 250 edge models can be reused when they convey consensual and shared knowledge.
 251 These components can either be *problem solving models*, that can be reused across
 252 various domains, like the library of problem solving methods in CommonKADS (this
 253 library is one of the major results of the KADS and later CommonKADS¹¹ European
 254 projects Schreiber et al. 1999), or domain models, ontologies, semantic resources
 255 like lexical data-bases or thesauri. Ontologies represent domain concept definitions
 256 in a formal structure. A lexical data-bases like WordNet¹² registers, classifies and
 257 organises, according to semantic and lexical criteria, most of the vocabulary of the
 258 English language. Thesauri collect normalised domain vocabularies as structured
 259 sets of terms.

260 3.2 From Knowledge Sources to Models: Research Issues

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

¹¹<http://www.commonkads.uva.nl/>.

¹²<http://wordnet.princeton.edu/wordnet/>.

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

276 3.2.1 How to Design a Model?

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

293 3.2.2 How to Benefit from Complementary Knowledge Sources?

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

302 3.2.3 What Are Models Made of? What is the Optimal Formal Level?

303 Each model combines various types of knowledge. In a similar way, each KE method
304 questions and makes suggestions on the nature of the models to be designed, on the
305 way to structure them and to collect the appropriate knowledge that feel them as well
306 as on the representation formalism to select, which can be more or less formal as
307 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.

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

350 3.3.1 Human Expertise as Knowledge Source

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

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

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

378 Knowledge in software aims at better guiding users. By the way, it impacts their
379 working methods. So it raises the need to analyse their practices and the practices of
380 their collaborators, to study their activities and their use of support tools, to consider
381 their organisational context, which refers to ergonomics, sociological or management
382 approaches. Results of such analyses were first returned in a static way, as models
383 (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, Laflaquière 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 mentioned using nominal or verbal phrases with a strong coherence. According to the way they

425 are used, these phrases can be considered as terms denoting domain concepts or
426 relationships between domain concepts. Language may also provide clues with a
427 lower reliability, linking more diffuse knowledge elements. Then analysts have to
428 rebuild reference links in order to come up with knowledge-based elements, axioms
429 or rules. Results established by lexical semantics, terminology and corpus linguistics
430 research are set prior to the implementation of this kind of approach (Condamines
431 2002; Constant et al. 2008).

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

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

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

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

458 Vocabulary modelling motivated the design of dedicated software tools that
459 provide higher level results than standard NLP tools. For instance, results such as
460 terms and clusters of synonym terms can then be integrated in a model.
461 Examples of such tools are term extractors – Terminoweb (Barriere and Agbago
462 2006), Syntex-Upery (Bourigault 2002), TermExtractor (Drouin 2003) or TermRaider
463 in the GATE¹³ framework -; pattern-based relation extractors - Caméléon (Aussenac-
464 Gilles and Jacques 2008), RelExt (Schutz and Buitelaar 2005) or SPRAT (Maynard
465 et al. 2009) that implements three types of lexico-syntactic patterns (Hearst’s pat-
466 terns, patterns derived from Ontology design patterns and contextual patterns) in

¹³<http://gate.ac.uk/>.

467 GATE; pattern-based languages like Jape in GATE, Nooj,¹⁴ Unitex¹⁵; named-entity
 468 extractors (Poibeau and Kosseim 2000) that contribute to search for instances or rela-
 469 tions between instances (as with the KIM platform¹⁶). To sum up, designing models
 470 from texts has strongly benefited from NLP frameworks (GATE, Linguastream,¹⁷
 471 UIMA¹⁸) that support the development of adapted processing chains. Finally, spe-
 472 cific processing chains, as Text2Onto (Cimiano and Völker 2005), and the version
 473 integrated by NeOn,¹⁹ have allowed an assessment of the strengths and limitations of
 474 this approach by increasing automation and exploiting machine learning techniques.
 475 Current research works combine text analysis, reuse of ontological components and
 476 human interpretation. Cimiano et al. (2010) gives a reasonably full picture of these
 477 works.

478 3.3.3 Modelling Frameworks

479 Modelling frameworks provide access to knowledge sources, or to their traces, to
 480 knowledge extraction techniques and software tools, as well as to modelling tech-
 481 niques and languages. They suggest a methodology that defines a processing chain
 482 and guides the modelling task step by step. In the following Sub-section, we first
 483 present the most significant results about problem-solving modelling in the early
 484 1990s. Then we focus on methods and frameworks for ontology design which have
 485 been developed in the last ten years.

486 Methods for Problem-Solving Modelling

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

496 In the early 1980s and 1990s the notion of conceptual model evolved with an
 497 emphasis on problem-solving models, new related languages, inference and tasks
 498 notions articulated. From a methodological viewpoint, the research showed that
 499 modelling primitives provide a grid for collecting and interpreting knowledge; they
 500 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/>.

¹⁸http://domino.research.ibm.com/comm/research_projects.nsf/pages/uima.index.html.

¹⁹<http://www.neon-toolkit.org/>.

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

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

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

535 Methods and Frameworks for Designing Ontologies

536 The design process of ontologies took advantage of these methodologies. It started
537 when the reuse of domain models put forward the interest in high quality consensual
538 models designed according « good » principles facilitating reuse and adaptation.
539 The specific challenges encountered during the ontology design process are the fol-
540 lowings:

- 541 1. Define the ontology content and ensure its quality;
- 542 2. Exploit efficiently all available knowledge sources using, for instance, text anal-
543 ysis or ontology reuse processes;

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

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

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

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

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

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

579 The main component of the frameworks used for designing ontologies is usually
 580 an ontology editor. Therefore, Protégé²¹ is an editor extensively used to create
 581 or modify RDFS or OWL ontologies, and can be available as a web service
 582 (Web-Protégé) which is particularly appropriate for cooperative ontology design.
 583 Swoop²² has been designed for lightweight ontologies, whereas Hozo²³'s original-

²⁰For 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/>.

584 ity lies in the notion of role and the ability to distinguish concepts depending on
585 particular contexts from basic concepts to ensure an easier ontology reuse. Besides
586 this editing function, several other functionalities can be provided in ontology engi-
587 neering frameworks, such as Schema XML translating functions, graph display of
588 parts of the ontology, ontology modules management, ontology partition, transla-
589 tion of vocabularies, import functions of Web ontologies, access to ontology search
590 engines, text processing modules (like Tree-Tagger²⁴ or Stanford Parsing tools), help
591 for personalizing ontologies, generating documentation, managing ontology evolu-
592 tion, ontology evaluation, ontology alignment, reasoning and inference services,
593 navigation assistance services, visualisation services, ... As an illustration, most of
594 these functionalities are available as plug-ins in the Neon²⁵ framework.

595 Some frameworks are designed to deal with a specific kind of data. Therefore,
596 Text2Onto, successor of TextToOnto, and DaFOE4App are specially designed to use
597 text documents and thesaurus as input knowledge sources. Text2Onto (Cimiano and
598 Völker 2005) includes a text mining software and modules that generate structured
599 information from weakly structured documents. Text2Onto is associated with KAON
600 (Karlsruhe Ontology Management Infrastructure) framework (Oberle et al. 2004) in
601 order to design ontologies. DaFOE4App (Differential and Formal Ontology Editor
602 for Applications) (Szulman et al. 2009) focuses on the linguistic dimension while
603 its design uses some of the ARCHONTE methodology principles (Bachimont et al.
604 2002). DaFOE4App covers all stages from corpora analysis (using a NLP framework)
605 to the definition of a formal domain ontology. It guarantees persistence, traceability
606 and the dimensioning of models (several millions of concepts). The TERMINAE
607 framework (Aussenac-Gilles et al. 2008), designed before DaFOE4App, has evolved
608 with the specifications of DaFOE4App. TERMINAE²⁶ was used and evaluated in
609 many projects. To end this non-exhaustive list, PlibEditor is more specially tailored
610 to databases. With PlibEditor, users can perform all the tasks required to design
611 ontologies, import or export ontologies as well as data. PlibEditor is complementary
612 to OntoDB, an ontology-based database system and it enables a database approach
613 based on domain ontologies (Fankam et al. 2009).

614 3.4 Model Reuse

615 Just as software engineering aims to reuse software components, knowledge acqui-
616 sition promotes the reuse of knowledge components. This reusability can be achieved
617 in various ways.

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

²⁴<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>.

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

²⁶<http://lipn.univ-paris13.fr/terminae/>.

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

633 Based on the KADS project's outcome, some frameworks support the combi-
634 nation of generic components. They include rich libraries of components as well
635 as graphical editors dedicated to knowledge formalisation, task representation, and
636 the selection and configuration of the methods allowing to solve the tasks (Musen
637 et al. 1994). Solution to adapt generic models to a specific application are diverse,
638 ranging from manual instantiation procedures (Beys et al. 1996) to automated pro-
639 cesses including mechanisms that check the specification consistency (Fensel et al.
640 1996). The CommonKADS project settings led to the most successful results to
641 design problem-solving models. The CommonKADS expertise model can be built
642 by abstraction process or reusing components of problem-solving models. Its partic-
643 ular strength lies in the library of components with different granularities, and with a
644 reuse and adaptation process guided by a questions grid which ensures the relevancy
645 of designed model.

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

654 Ontology reuse has motivated the design of ontology search engines such as
655 Watson,²⁷ Swoogle,²⁸ or OntoSearch.²⁹ Using key words, these engines provide a
656 list of ontologies containing at least one concept, one relationship or another ele-
657 ment labelled or identified by one of the key words. Then selecting the most relevant
658 ontologies in this list requires that each ontology could be evaluated individually and
659 that ontologies could be compared to eachother according to various criteria. There-
660 fore, how to assess an ontology and to compare several ontologies is currently one
661 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>.

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

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

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

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

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

704 erogeneity, significant progress has been made on *model reconciliation*. Models can
 705 be reconciled at two different levels. At the schema level, reconciliation consists in
 706 identifying correspondences or mappings between semantically-related entities of
 707 two ontologies. In the past years, considerable efforts have been made to build ontol-
 708 ogy alignment tools (Euzenat and Shvaiko 2013), many of which are available on
 709 the internet such as OnAGUI³² or TAXOMAP (Hamdi et al. 2009). Each year since
 710 2004, OAEI international campaigns aim at comparing ontology matching systems.
 711 At the data level, reconciliation consists in determining if two data descriptions refer
 712 to the same entity of the real world (e.g. the same person or the same hotel). This
 713 problem is referred to as *reference reconciliation* (Sais et al. 2009) and it is close to
 714 coreference resolution in NLP.

715 3.5 Knowledge Representation in Models

716 Even though designing knowledge representation languages is not KE's main objec-
 717 tive, researchers, when specifying knowledge and models, contribute to develop,
 718 evaluate and evolve these languages within normalisation groups, such as W3C.
 719 Knowledge representation languages as well as modelling languages were first dedi-
 720 cated to problem-solving and reasoning. Then, they related to ontologies (cf. Sects. 2,
 721 2.1, 2.2); nowadays knowledge representation languages are back hand in hand with
 722 reasoning.

723 In the 1980s, ontology representation languages successfully took advantage of
 724 logic and conceptual graphs (Sowa 1984). Conceptual graphs could provide both a
 725 logic formalisation and a graphical symbolism when no powerful HMI was available
 726 to display semantic networks or trees, and to deploy or close them upon request.
 727 OWL was later developed as an evolution of DAML+OIL,³³ a language resulting
 728 from the merge of the DAML³⁴ and OIL project outcomes (Fensel et al. 2001).
 729 Drawn also on description logic (cf. Sect. I.5), and defined as a layer above XML,
 730 OWL became stable and included three languages OWL Lite, OWL-DL, OWL-full
 731 according to the W3C recommendations. Each of these three languages specificities
 732 results from the trade-off representativity *versus* calculability. In 2007, OWL was
 733 extended with new features. A new version, called OWL 2,³⁵ was formally defined
 734 in 2012 with three sub-languages³⁵ (called *profiles*) offering distinct advantages,
 735 computational properties or implementation possibilities, in particular application
 736 scenarios: OWL 2 EL enables polynomial time algorithms for all standard reason-
 737 ing tasks; OWL 2 QL enables conjunctive queries to be answered in LogSpace;

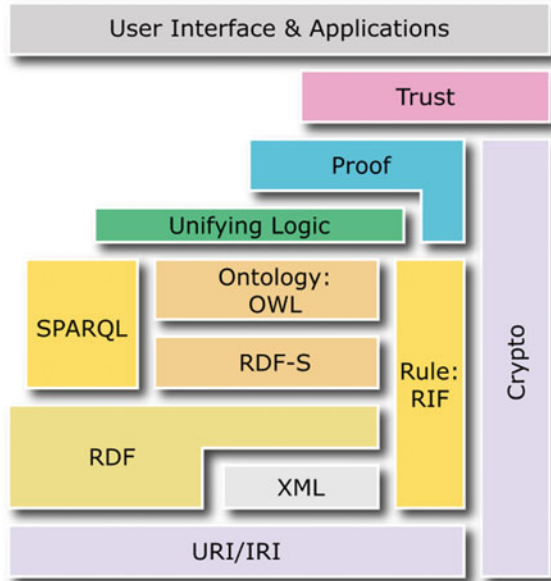
³²<https://github.com/lmazuel/onagui>.

³³<http://www.w3.org/TR/daml+oil-reference>.

³⁴<http://www.daml.org/>.

³⁵https://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



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

740 In the Semantic Web Stack proposed by Tim B. Lee (cf. Fig. 2), representing the
 741 stacking order of the Semantic Web languages, we can notice that RDF,³⁶ located
 742 in the bottom part, is the basic language of the Semantic Web. RDF is the common
 743 ground to all the languages of interest for KE (i.e. RDF, RDF-S, OWL, SPARQL
 744 and RIF). These languages allow applications to consistently use ontologies and
 745 associated rules. RDF is a simple language to express data models as a graph
 746 where nodes are web resources and edges properties. RDF Schema³⁷ is a seman-
 747 tic extension of RDF. It is written in RDF and provides mechanisms to structure
 748 data models, by describing groups of related resources and the relationships between
 749 these resources. OWL is another and more expressive extension allowing a better
 750 integration of ontologies and easier inferences. SPARQL³⁸ is an RDF semantic query
 751 language for databases, able to retrieve and manipulate data stored in RDF format.
 752 RIF³⁹ (Rule Interchange Format) is the rule layer in the Semantic Web Stack. RIF is
 753 not a rule language but rather a standard for exchanging rules among rule systems.
 754 Other rule languages may apply on ontologies, like SWRL,⁴⁰ or Description Logic
 755 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>.

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

758 Another W3C recommendation defined as an application of RDF is SKOS⁴² (for
 759 Simple Knowledge Organisation System). SKOS provides a model for expressing
 760 the basic structure and content of concept schemes such as thesauri, taxonomies,
 761 folksonomies, and other similar types of controlled vocabulary. In basic SKOS, con-
 762 ceptual resources (concepts) are related to each other in informal hierarchies but
 763 no logical inference is possible. Using SKOS, generalisation versus specialisation,
 764 (*broader-than* and *narrower-than* – BT/NT) relations that are very often used in the-
 765 saurus can be represented without logical inferences associated to the subsumption
 766 relationship in OWL.

767 SKOS was even more necessary in that logical inferences based on the subsump-
 768 tion relationship are only valid if ontologies comply with the associated constraints
 769 (whereas such relationship is not valid on thesaurus). Furthermore, the applications
 770 using thesaurus and ontologies are increasingly efficient and the resources them-
 771 selves – i.e. thesaurus and ontologies – are involved in the development processes
 772 using different knowledge representation languages at different steps in the devel-
 773 opment process and not always as intended by the language designers. For instance,
 774 a thesaurus and an ontology jointly used in an application can be modelled in OWL
 775 for that application. However, one could be originally developed in SKOS and the
 776 other one in OWL, and they could further be distributed in a format like CTS2.⁴³

777 4 Methodological Issues and Today's Applications

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

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

789 KE-related applications form a vast field of research, experimentation and transfer
 790 of AI technologies in which innovative methods must be developed. The articulation
 791 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>.

792 **4.1 Linking Language, Knowledge and Media**

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

798 **4.1.1 Designing Problem-Solving Models and Ontologies from Natural** 799 **Language in Textual Documents**

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

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

826 **4.1.2 Information Retrieval with Ontologies**

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

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

841 4.2 Coping with Data Explosion

842 For nearly 20 years, the amount of available data exploded. In a parallel movement,
843 the Semantic Web turned out to be a web of Data in addition to a web a document.
844 This means that the semantics should also be brought to data by labeling them with
845 ontology concepts. Thus applications address increasingly numerous and diverse
846 data that generate new needs in particular for their description and their integration.
847 The so-called *Big Data* is frequently characterised by the four (or more) versus
848 (4Vs): Volume, Velocity, Variety, Veracity. Velocity has to do with efficiency and
849 calculability of knowledge representation, which is out of the scope of this chapter.
850 In the following paragraphs, we explore the three others characteristics: Veracity,
851 Variety, and, for the Volume problematic, we focus more specifically on the question
852 of the size of designed models.

853 4.2.1 Volume

854 The description of these very numerous data requires the development of models in
855 which the amount of information to be taken into account can be large enough to open
856 new perspectives to statistical approaches and models. In order to maintain the use
857 and management of symbolic models, the challenge is to be able to design models of
858 very large size, for example by reducing the amount of information to be taken into
859 account simultaneously. In this way, work on ontology modularity aims at designing
860 very large ontologies needed for applications, and to consider these ontologies as sets
861 of (more or less independent) modules. Modularity, in the general sense of the word,
862 refers to the perception of a large knowledge repository (i.e. an ontology, a knowledge
863 or data base) as a set of smaller repositories. Although the concept of modularity
864 is widely used in computer science, it is a relatively new idea in KE. For example,

865 the Knowledge Web project⁴⁴ (2004–2007) provided guidelines to design modular
866 ontologies (Stuckenschmidt et al. 2009). This project showed the diversity of views
867 on modularity and pointed out the important research directions to be developed:
868 guidelines to design modules (how to determine a coherent and meaningful set of
869 concepts, relationships, axioms and instances), metadata to describe, to select and to
870 use or re-use modules, specification of how they can be linked to one another, their
871 composition and their reuse in different contexts. Managing a large mass of data in
872 a distributed context can also lead to designing on a set of existing ontologies that
873 need to be redesigned, aligned, transformed into modules or integrated with non-
874 ontological resources such as databases, folksonomies or thesauri. The networked
875 ontology construction method defined by the NeOn⁴⁵ project (2006–2010) includes
876 a support for cooperative design and takes into account the dynamic and evolutionary
877 features of ontologies (Gómez-Pérez and Suárez-Figueroa 2009), which are major
878 issues for the development of large ontology-based applications.

879 4.2.2 Variety or Managing Knowledge Integration Through Ontologies

880 Both in the fields of databases and information retrieval, ontologies are experimented
881 as a promising solution for data integration. When integrating data from multiple
882 and heterogeneous sources, ontologies can help to understand and interpret data
883 belonging to the same domain but represented in heterogeneous structures. Then
884 ontologies are also a good support to relate them more easily (Assele Kama et al.
885 2010). In some domains, such as geography, few ontologies are practically available
886 for data integration (Buccella et al. 2009) or they describe targeted domains, such
887 as Townology for planning and urbanism (Roussey et al. 2004) or FoDoMuSt in
888 the field of image processing (Brisson et al. 2007). The challenge then consists in
889 designing useful ontologies.

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

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

⁴⁵<http://www.neon-project.org/>.

904 This context, reinforced by the need to exploit diversified knowledge or several
905 partial models (or modules), requires to face the problem of heterogeneity between
906 models/ontologies/knowledge, and motivates the current interest in semantic inter-
907 operability. Research work on semantic interoperability bears on automatic mapping
908 tools that set links between elements of semantically heterogeneous concept schemes,
909 ontologies or other knowledge sources. They define processes for schema matching,
910 ontology alignment (cf. Sect. 3.4), or data reconciliation. For instance, recent medi-
911 cal studies have tried to integrate most of the knowledge needed to make a diagnosis
912 – e.g. clinical, imaging, genomics knowledge – thanks to a pivotal ontology based
913 on various available ontologies or models (Hochheiser et al. 2016; Sarnitivijai et al.
914 2016).

915 4.2.3 Veracity

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

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

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

930 4.3 Managing Distributed Data

931 The web and web standards have greatly changed the way data is distributed. In par-
932 ticular, new types of systems, web services, rely on a new communication protocol
933 between machines. Thanks to web services, the Web became a distributed com-
934 puting device where programs (services) can interact intelligently by being able to
935 automatically discover other services, to negotiate among themselves and to com-
936 pose themselves into more complex services. A considerable amount of knowledge
937 is mandatory to get intelligible services from machines. When added a knowledge
938 base, web services become *semantic web service*.

939 Semantic web services are the bricks to create a semantic Web of services whose
940 properties, capabilities, interfaces and effects are described in an unambiguous way
941 and can be exploited by machines. The semantics thus expressed must facilitate

942 the automatic management of services. Semantic web services are essential for the
943 effective use of web services in industrial applications. However, they still raise a
944 number of issues for the research community, including for the KE field because
945 they use ontologies to explain which service they provide to other services or to end
946 users. Semantic modelling contributes to evaluate the quality of a Web service and to
947 take it into account in the process of discovery or composition of services. Peer-to-
948 peer (P2P) systems have also grown significantly, and a substantial body of research
949 work has recently sought to improve the search function in unstructured systems
950 by replacing random routing with semantically guided routing. Several dimensions
951 of the problem are analysed: Which semantics should be remembered? Which rep-
952 resentation to adopt? How to design it? What is shared among peers? How to use
953 semantics? How to disseminate it? These issues remained unresolved and have been
954 brought into sharper focus by KE.

955 **4.4 Leveraging New Knowledge Sources**

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

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

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

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

982 The number of data published according to the principles of linked data is grow-
983 ing rapidly (we are talking about billions of RDF triplets available on the Internet).
984 The site <http://lov.okfn.org/dataset/lov/> gives a snapshot of existing vocabularies
985 (more than 600) and highlights the numerous mutual reuse of terms between these
986 vocabularies. Among this large number of data sources, DBpedia⁴⁷ structures the
987 content of Wikipedia⁴⁸ into RDF triples so as to make the information of the ency-
988 clopedia reusable. DPpedia is a very powerful source as it is interconnected with
989 other data sources, such as Geonames⁴⁹ and MusicBrainz⁵⁰) and it has been linked
990 to even larger data sets like YAGO⁵¹ (Rebele et al. 2016) or BabelNet⁵² (Navigli and
991 Ponzetto 2012). These large generic knowledge bases are also used by search engines
992 to display structured content in response to users' queries. Because of they propose
993 unambiguous and linked vocabularies, these masses of data represent promising
994 sources for KE.

995 4.5 Coping with Knowledge Evolution

996 The dynamic nature of the data on the Web gives rise to a multitude of problems
997 related to the description and analysis of the evolution of such data. The existing
998 models of knowledge representation are inadequately addressing the challenges of
999 data evolution and, above all, they do not benefit from any adaptive mechanism that
1000 would allow them to rigorously follow the evolutions of a domain. Research work
1001 on ontology evolution underlines how much the Semantic Web and KE communities
1002 need to find appropriate solutions to this complex issue. Early studies defined the
1003 stages of an evolution process (Noy and Klein 2004; Stojanovic 2004), they spec-
1004 ified a typology of changes (Plessers et al. 2007) and change descriptions. Other
1005 works proposed mechanisms, sometimes borrowed to belief revision (Flouris 2006)
1006 to keep the modified ontology consistent and logically sound (Haase and Stojanovic
1007 2005) and defined how to propagate changes in distributed ontologies and in the
1008 applications that use them (Stuckenschmidt and Klein 2003). With similar purposes
1009 to ontology engineering, ontology evolution can be fed thanks to the knowledge
1010 identified in textual documents using NLP tools (Buitelaar and Cimiano 2008) and
1011 relying on document structure, like in (Nederstigt et al. 2014). More recently, when
1012 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/>.

1013 with the evolution of these semantic annotations when the textual corpus or when the
1014 indexing vocabularies evolve (Tissaoui et al. 2011; Da Silveira et al. 2015; Cardoso
1015 et al. 2016).

1016 Zablieth et al. (2015) propose a recent overview of the major trend in this domain.
1017 Characterizing and representing domain data evolution raises issues both at the data
1018 level (Stefanidis et al. 2016) and at the model scheme level (Guelfi et al. 2010).
1019 Ontology evolution remains a hard issue, even at the era of machine learning, because
1020 a statistic processing of a massive amount of documents is relevant for building large
1021 knowledge bases like DBpedia, but produces poor results when trying to fix errors
1022 or to identify local changes in an existing model. Processing large amounts of data is
1023 much more appropriate to feed and update the data level in knowledge bases, which
1024 corresponds to instances of ontological classes.

1025 4.6 *Collective Versus Personal Knowledge*

1026 Most of the previous approaches place little emphasis on the social dimension of
1027 knowledge management. This dimension is strong enough in some professional
1028 communities to consider them as communities of interest or as communities of prac-
1029 tices. Communities of practices designate social groups in which learning processes
1030 emerge through the sharing of networked knowledge. KE models need to capture
1031 these learning processes or to integrate them into their knowledge management pro-
1032 cess. To this end, Lewkowicz and Zacklad (2001) propose a new form of knowledge
1033 management based on the structuring of collective interactions. This approach aims
1034 at better using of the shared knowledge, at facilitating its reuse, the knowledge of an
1035 organisation being considered as above all a matter of collective competence.

1036 The identification of communities of interest that emerged thanks to the develop-
1037 ment of Web 2.0 or the analysis of users' digital traces sharing similar thematic
1038 information implies the representation of individual knowledge about the fields
1039 of interest and activities of their members, together with the collective dimension
1040 of knowledge. This collective dimension is the focus of the Computer Supported
1041 Cooperative Work (CSCW) research community, that designs specific solutions
1042 to manage collective and in-use knowledge. For instance, M. Zacklad proposes a
1043 conceptual model mid-way between thesauri and formal ontologies, called *semiotic*
1044 *ontologies*, that should be more easily shared by a working community in an infor-
1045 mation retrieval framework (Zacklad 2007). Conversely, more and more software
1046 systems and Web interfaces are designed to be context sensitive or user customised.
1047 To do so, they adapt to the user profile, environment or interactions with the system,
1048 which requires the acquisition, the modelling and the processing of the interaction
1049 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

1089 results of the domain complement each other over time and can be taken from a new
1090 angle when the context evolves.

1091 For a long time, KE has been interested in producing knowledge models in a well-
1092 structured process under the control of knowledge engineers. The resulting models,
1093 generally complex, were used in specific applications. Today, applications in which
1094 knowledge is used as support for reasoning or activity have become much more
1095 diversified. Since 2000, they have been devoted to knowledge management in the
1096 broadest sense, including semantic information retrieval, navigation aids, decision
1097 support, and many semantic Web applications. This enlargement continues and new
1098 fields of application are still emerging, posing the problems of KE in new terms.

1099 Thus, in the age of ubiquitous computing, it is the living room, the train, the auto-
1100 mobile, the workshop, the classroom or meeting room, the smallest kitchen device
1101 that become “smart” tools. Within these tools, a dynamic process is required to con-
1102 tinuously acquire context knowledge on the flow from a wide variety of sources (sen-
1103 sors, databases, the Internet, users with various profiles). In addition, these intelligent
1104 tools must have a pro-active behaviour that enables them to initiate communication
1105 or action based on their understanding of the current situation and on their goals.
1106 So, for example, phones know where we are at a given time and become capable
1107 of automating some operations, such as when taking pictures, labeling them with
1108 geographic and temporal metadata.

1109 The last decade has seen a major transformation in the way individuals interact
1110 and exchange. Information is now co-produced, shared, filed and evaluated on the
1111 Web by thousands of people. These uses and the underlying technologies are known
1112 as Web 2.0. Web 3.0 is the latest evolution to date that combines the social web and
1113 the semantic technologies of the semantic Web. In the context of communities of
1114 interest or practices where spontaneous emergence and activity are allowed by these
1115 evolutions of the Web, KE and knowledge management are thus major stakes of the
1116 future decade.

1117 Finally, KE must feed and evaluate all these new developments, compare them
1118 with previous models (reasoning models, rules bases), estimate the need to use ontolo-
1119 gies and their alignment to type or organise data, to define new techniques and
1120 languages if necessary, to justify the use of metadata to enrich and reuse data, and so
1121 on. The speed of Web evolutions can be seen as a crazy accelerator of the research
1122 pace or as an alarm that invites us to step back and pose the problems at a higher
1123 abstraction level, necessarily interdisciplinary, in order to better qualify the essence
1124 of knowledge, their dissemination and their formalisation for digital processing.

1125 References

- 1126 Assele Kama A, Mels G, Choquet R, Charlet J, Jaulent M-C (2010) Une approche ontologique pour
1127 l’exploitation de données cliniques. In: Despres S (ed) Acte des 21èmes Journées Francophones
1128 d’Ingénierie des Connaissances, Nîmes, France. Ecole des Mines d’Alès, pp 183–194

- 1129 Aussenac N (1989) Conception d'une méthodologie et d'un outil d'acquisition de connaissances
 1130 expertes. Thèse de doctorat, Université Paul Sabatier, Toulouse, France
- 1131 Aussenac-Gilles N, Jacques M-P (2008) Designing and evaluating patterns for relation acquisition
 1132 from texts with caméléon. *Terminology* 14(1):45–73 (special issue on Pattern-based approaches
 1133 to semantic relations)
- 1134 Aussenac-Gilles N, Krivine J, Sallantin J (1992) Editorial du numéro spécial Acquisition des con-
 1135 naissances. *Revue d'Intelligence Artificielle* 6(2):7–18
- 1136 Aussenac-Gilles N, Bourigault D, Condamines A (1995) How can knowledge acquisition benefit
 1137 from terminology? In: Proceedings of the 9th knowledge acquisition workshop, Banff, University
 1138 of Calgary (CA)
- 1139 Aussenac-Gilles N, Laublet P, Reynaud C (eds) (1996) Acquisition et ingénierie des connaissances:
 1140 tendances actuelles. Cepadues Editions, Toulouse
- 1141 Aussenac-Gilles N, Biébow B, Szulman S (2000) Revisiting ontology design: a method based
 1142 on corpus analysis. In: 12th international conference on knowledge engineering and knowledge
 1143 management, Juans-Les-Pins, 03/10/2000–06/10/2000. Springer, Heidelberg, pp 172–188
- 1144 Aussenac-Gilles N, Desprès S, Szulman S (2008) The TERMINAE method and platform for ontol-
 1145 ogy engineering from texts. *Ontology learning and population: bridging the gap between text and
 1146 knowledge*, pp 199–223
- 1147 Bachimont B, Isaac A, Troncy R (2002) Semantic commitment for designing ontologies: a proposal.
 1148 In: EKAW, pp 114–121
- 1149 Barriere C, Agbago A (2006) Terminoweb: a software environment for term study in rich contexts.
 1150 In: International conference on terminology, standardization and technology transfer, Beijing, pp
 1151 103–113
- 1152 Beys B, Benjamins V, Van Heijst G (1996) Remediating the reusability - usability trade-off for
 1153 problem-solving methods. In: Gaines B, Musen M (eds) Proceedings of the 10th knowledge
 1154 acquisition workshop (KAW), Banff, Canada, pp 2–1/2-20
- 1155 Bourigault D (2002) Upery: un outil d'analyse distributionnelle étendue pour la construction
 1156 d'ontologies à partir de corpus. In: Actes de la 9ème conférence annuelle sur le Traitement
 1157 Automatique des Langues (TALN 2002), Nancy, France, pp 75–84
- 1158 Bourigault D, Slodzian M (1999) Pour une terminologie textuelle. *Terminol Nouv* 19:29–32
- 1159 Bourigault D, Aussenac-Gilles N, Charlet J (2004) Construction de ressources terminologiques ou
 1160 ontologiques à partir de textes: un cadre unificateur pour trois études de cas. *Revue d'Intelligence
 1161 Artificielle* 18(1/2004):87–110
- 1162 Brank J, Grobelnik M, Mladenic D (2005) A survey of ontology evaluation techniques. In: Data
 1163 mining and data warehouses conference (SIKDD), Lubiana, Slovénie
- 1164 Brewster C, Alani H, Dasmahapatra S, Wilks Y (2004) Data driven ontology evaluation. In: LREC
 1165 Brisson R, Boussaid O, Gançarski P, Puissant A, Durand N (2007) Navigation et appariement
 1166 d'objets géographiques dans une ontologie. In: EGC, pp 391–396
- 1167 Buccella A, Cechich A, Fillottrani P (2009) Ontology-driven geographic information integration: a
 1168 survey of current approaches. *Comput Geosci* 35(4):710–723 (Geoscience knowledge represen-
 1169 tation in cyberinfrastructure)
- 1170 Buitelaar P, Cimiano P (eds) (2008) Proceedings of the 2008 conference on ontology learning and
 1171 population: bridging the gap between text and knowledge, Amsterdam, The Netherlands. IOS
 1172 Press
- 1173 Camilleri G, Soubie J-L, Zaraté P (2008) A replanning support for critical decision making situa-
 1174 tions: a modelling approach. In: Intelligent decision making: an AI-based approach, pp 173–192
- 1175 Cardoso SD, Pruski C, Da Silveira M, Ying-Chi L, Anika G, Erhard R, Reynaud-Delaître C (2016)
 1176 Leveraging the impact of ontology evolution on semantic annotations. In: Knowledge engineering
 1177 and knowledge management - 20th international conference, EKAW, Bologna, Italy
- 1178 Chandrasekaran B (1983) Towards a taxonomy of problem solving types. *AI Mag* 4(1):9–17
- 1179 Charlet J (1991) ACTE: a strategic knowledge acquisition method, pp 85–93
- 1180 Charlet J, Zacklad M, Kassel G, Bourigault D (eds) (2000) Ingénierie des connaissances: Evolutions
 1181 récentes et nouveaux défis. Eyrolles, Paris

- 1182 Cimiano P, Völker J (2005) Text2onto. In: NLDB, pp 227–238
- 1183 Cimiano P, Buitelaar P, Völker J (2010) Ontology construction. In: Indurkha N, Damerau, FJ (eds)
- 1184 Handbook of natural language processing, 2nd edn. CRC Press, Taylor and Francis Group, Boca
- 1185 Raton. ISBN 978-1420085921
- 1186 Clark P, Thompson JA, Porter BW (2000) Knowledge patterns. In: KR, pp 591–600
- 1187 Condamines A (2002) Corpus analysis and conceptual relation patterns. *Terminol. Int J Theor Appl*
- 1188 *Issues Spec Commun* 8(1):141–162
- 1189 Constant M, Dister A, Ermikanian L, Piron S (2008) Description linguistique pour le traitement
- 1190 automatique du français. Cahier du CENTAL
- 1191 Cordier M-O, Reynaud C (1991) Knowledge acquisition techniques and second-generation expert
- 1192 systems. *Appl Artif Intell* 5(3):209–226
- 1193 Da Silveira M, Dos Reis J, Pruski C (2015) Management of dynamic biomedical terminologies:
- 1194 current status and future challenges. *Yearb Med Inform* 24:125–133
- 1195 Daga E, Blomqvist E, Gangemi A, Montiel E, Nikitina N, Presutti V, Villazon-Terrazas B (2010)
- 1196 NeOn project: NeOn D2.5.2. Pattern-based ontology design: methodology and software report.
- 1197 Rapport de contrat
- 1198 Darses F, Montmollin M (eds) (2006) L'ergonomie. La Découverte - Col. Repères, Paris
- 1199 Dieng-Kuntz R, Corby O, Gandon F, Gibouin A, Golebiowska JNM, Ribière M (eds) (2005) Knowl-
- 1200 edge management: Méthodes et outils pour la gestion des connaissances. Dunod
- 1201 Drouin P (2003) Term extraction using non-technical corpora as a point of leverage. *Terminology*
- 1202 9:99–117
- 1203 Euzenat J, Shvaiko P (2013) Ontology matching, 2nd edn. Springer, Heidelberg
- 1204 Fankam C, Bellatreche L, Hondjack D, Ameer YA, Pierra G (2009) Sisro, conception de bases de
- 1205 données à partir d'ontologies de domaine. *Technique et Science Informatiques* 28(10):1233–1261
- 1206 Fensel D, Schnanegge R, Wielinga B (1996) Specification and verification of knowledge-based
- 1207 systems. In: Proceedings of the 10th knowledge acquisition workshop (KAW), Banff (Can).
- 1208 University of Calgary (Can)
- 1209 Fensel D, van Harmelen F, Horrocks I, McGuinness DL, Patel-Schneider PF (2001) Oil: an ontology
- 1210 infrastructure for the semantic web. *IEEE Intell Syst* 16(2):38–45
- 1211 Fernández-López M, Gómez-Pérez A (2002) Overview and analysis of methodologies for building
- 1212 ontologies. *Knowl Eng Rev* 17(2):129–156
- 1213 Flouris G (2006) On belief change in ontology evolution. *AI Commun* 19(4):395–397
- 1214 Gangemi A (2005) Ontology design patterns for semantic web content. In: International semantic
- 1215 web conference, pp 262–276
- 1216 Gangemi A, Catanacci C, Battaglia M (2004) Inflammation ontology design pattern: an exercise in
- 1217 building a core biomedical ontology with descriptions and situations. In: Maria PD (ed) *Ontologies*
- 1218 *in medicine*. IOS Press, Amsterdam
- 1219 Garlatti S, Prié Y (2004) Adaptation et personnalisation dans le web sémantique. *Revue I3 - Numéro*
- 1220 *hors série Web Sémantique*
- 1221 Gómez-Pérez A, Suárez-Figueroa M-C (2009) Scenarios for building ontology networks within the
- 1222 neon methodology. In: K-CAP 2009, pp 183–184
- 1223 Gómez-Pérez A, Fernández-López M, Corcho O (2007) Ontological engineering: with examples
- 1224 from the areas of knowledge management, e-commerce and the semantic web. (Advanced infor-
- 1225 mation and knowledge processing). Springer, New York
- 1226 Gruber TR (1993) A translation approach to portable ontology specifications. *Knowl Acquis* 5:199–
- 1227 220
- 1228 Guarino N, Welty CA (2004) An overview of ontoclean. *Handbook on ontologies*, pp 151–172
- 1229 Guelfi N, Pruski C, Reynaud C (2010) Experimental assessment of the target adaptive ontology-
- 1230 based web search framework. In: NOTERE, pp 297–302
- 1231 Haase P, Stojanovic L (2005) Consistent evolution of owl ontologies. In: ESWC, pp 182–197
- 1232 Hamdi F, Safar B, Niraula N, Reynaud C (2009) TaxoMap in the OAEI 2009 alignment contest. In:
- 1233 The fourth international workshop on ontology matching, Chantilly, Washington DC, États-Unis

- 1234 Hendler JA, Tate A, Drummond M (1990) AI planning: systems and techniques. *AI Mag* 11(2):61–
 1235 77
- 1236 Hitzler P, Sure Y, Studer R (2005) Description logic programs: a practical choice for the modelling
 1237 of ontologies. In: Principles and practices of semantic web reasoning
- 1238 Hochheiser H, Castine M, Harris D, Savova G, Jacobson RS (2016) An information model for
 1239 computable cancer phenotypes. *BMC Med Inform Decis Mak* 16(1), 121
- 1240 Jacob-Delouis I, Krivine J (1995) Lisa: un langage réflexif pour opérationnaliser les modèles
 1241 d'expertise. *revue d'Intelligence Artificielle* 9(1):53–88
- 1242 Kamel M, Aussenac-Gilles N (2009) Utiliser la Structure du Document dans le Processus de Con-
 1243 struction d' Ontologies (regular paper). In: L'Homme M-C, Szulman S (eds) Conférence Interna-
 1244 tionale sur la Terminologie et l'Intelligence Artificielle (TIA), Toulouse (France), 18–20/11/2009,
 1245 page (on line). <http://www.irit.fr/> (IRIT)
- 1246 Kassel G (2002) Ontospec: une méthode de spécification semi-informelle d'ontologies. In: Actes
 1247 d'IC, pp 75–87
- 1248 Klinker G, Bholra G, Dallemagne G, Marquès D, Dermott M (1991) Usable and reusable program-
 1249 ming constructs. *Knowl Acquis* 3:117–136
- 1250 Laflaquière J, Prié Y, Mille A (2008) Ingénierie des traces numériques d'interaction comme inscrip-
 1251 tions de connaissances. In: Actes d'IC, pp 183–195
- 1252 Lewkowicz M, Zacklad M (2001) Une nouvelle forme de gestion des connaissances basée sur
 1253 la structuration des interactions collectives. In: Grundstein M, Zacklad M (eds) Ingénierie et
 1254 Capitalisation des connaissances. Hermes Sciences Europe LTD, pp 49–64
- 1255 Lozano-Tello A, Gomez-Perez A (2004) ONTOMETRIC: a method to choose the appropriate
 1256 ontology. *J Database Manag* 15(2):1–18
- 1257 Luong PH (2007) Gestion de l'évolution d'un web sémantique d'entreprise. Thèse de doctorat,
 1258 Ecole des Mines de Paris, Paris, France
- 1259 Maedche A (2002) Ontology learning for the semantic web. Kluwer Academic Publisher, Boston
- 1260 Maedche A, Staab S (2002) Measuring similarity between ontologies. In: EKAW, pp 251–263
- 1261 Manning C, Schütze H (1999) Foundations of statistical natural language processing. MIT Press,
 1262 Cambridge
- 1263 Marcus S, McDermott J (1989) SALT: a knowledge acquisition language for propose and revise
 1264 systems. *Artif Intell* 39(1):1–38
- 1265 Maynard D, Funk A, Peters W (2009) SPRAT: a tool for automatic semantic pattern-based ontology
 1266 population. In: International conference for digital libraries and the semantic web
- 1267 McAfee A (2006) Enterprise 2.0: the dawn of emergent collaboration. *MIT Sloan Manag Rev*
 1268 47(3):21–28
- 1269 Meyer I (2000) Extracting knowledge-rich contexts for terminography: a conceptual and method-
 1270 ological framework. In: Bourigault D, L'Homme M-C, Jacquemin C (eds) Recent advances in
 1271 computational terminology
- 1272 Musen MA, Eriksson H, Gennari JH, Tu SW, Puert AR (1994) PROTEGE-II: a suite of tools
 1273 for development of intelligent systems for reusable components. In: Proceedings of the annual
 1274 symposium on computer application in medical care
- 1275 Navigli R, Ponzetto SP (2012) BabelNet: the automatic construction, evaluation and application of
 1276 a wide-coverage multilingual semantic network. *Artif Intell* 193:217–250
- 1277 Neches R, Fikes R, Finin TW, Gruber TR, Patil RS, Senator TE, Swartout WR (1991) Enabling
 1278 technology for knowledge sharing. *AI Mag* 12(3):36–56
- 1279 Nederstigt LJ, Aanen SS, Vandic D, Frasinca F (2014) FLOPPIES: a framework for large-scale
 1280 ontology population of product information from tabular data in e-commerce stores. *Decis Support*
 1281 *Syst* 59:296–311
- 1282 Newell A (1982) The knowledge level. *Artif Intell* 18(1):87–127
- 1283 Noy NF, Hafner CD (1997) The state of the art in ontology design: a survey and comparative review.
 1284 *AI Mag* 18(3):53–74
- 1285 Noy NF, Klein MCA (2004) Ontology evolution: not the same as schema evolution. *Knowl Inf Syst*
 1286 6(4):428–440
- 1287 Oberle D, Volz R, Staab S, Motik B (2004) An extensible ontology software environment. *Handbook*
 1288 *on ontologies*, pp 299–320

- 1289 O'Reilly T (2007). What is web 2.0: design patterns and business models for the next generation of
1290 software. *Commun. Strat* (1):17
- 1291 Pan J, Lancieri L, Maynard D, Gandon F, Cuel R, Leger A (2007) Success stories and best practices.
1292 Knowledge web deliverable d.1.4.2.v2
- 1293 Pédaque RT (ed) (2003) Le document: forme, signe et medium les re-formulations du numérique.
1294 STIC-CNRS
- 1295 Pédaque RT (ed) (2005) Le texte en jeu, permanence et transformations du document. STIC-SHS-
1296 CNRS
- 1297 Plessers P, Troyer OD, Casteleyn S (2007) Understanding ontology evolution: a change detection
1298 approach. *J Web Semant* 5(1):39–49
- 1299 Poibeau T, Kosseim L (2000) Proper name extraction from non-journalistic texts. In: CLIN, pp
1300 144–157
- 1301 Porzel R, Malaka R (2004) A task-based approach for ontology evaluation. In: ECAI - workshop
1302 on ontology, learning and population
- 1303 Presutti V, Gangemi A, David S, De Cea GA, Surez-Figueroa MC (2008) NeOn project: NeOn
1304 D2.5.1. a library of ontology design patterns: reusable solutions for collaborative design of net-
1305 worked ontologies - NeOn project. Rapport de contrat
- 1306 Puerta A, Egar JW, Tu SW, Musen M (1992) Method knowledge-acquisition shell for the automatic
1307 generation of knowledge-acquisition tools. *Knowl Acquis* 4(2):171–196
- 1308 Rastier F (2009) *Sémantique interprétative*. PUF
- 1309 Rebele T, Suchanek FM, Hoffart J, Biega J, Kuzey E, Weikum G (2016) YAGO: a multilingual
1310 knowledge base from wikipedia, wordnet, and geonames. In: The semantic web - ISWC 2016
1311 - 15th international semantic web conference, Kobe, Japan, 17–21 October 2016, proceedings,
1312 part II, pp 177–185
- 1313 Rector A, Rogers J (2004) Patterns, properties and minimizing commitment: reconstruction of the
1314 GALEN upper ontology in OWL. In: EKAW
- 1315 Reymonet A, Thomas J, Aussenac-Gilles N (2007) Modélisation de ressources termino-
1316 ontologiques en owl. In: Actes d'IC, pp 169–181
- 1317 Reynaud C, Aussenac-Gilles N, Tchounikine P, Trichet F (1997) The notion of role in conceptual
1318 modeling. In: EKAW, pp 221–236
- 1319 Rosenbloom S, Miller RA, Johnson KB (2006) Interface terminologies: facilitating direct entry data
1320 into electronic health record systems. *J Am Med Inform* 13(3):277–288
- 1321 Rosse C, Mejino JL (2003) *J Biomed Inform* 36(6):478–500
- 1322 Roussey C, Laurini R, Beaulieu C, Tardy Y, Zimmermann M (2004) Le projet towntology: un retour
1323 d'expérience pour la construction d'une ontologie urbaine. *Revue Internationale de Géomatique*
1324 14(2):217–237
- 1325 Saïs F, Pernelle N, Rousset M-C (2009) Combining a logical and a numerical method for data
1326 reconciliation. *J Data Semant* 12:66–94
- 1327 Sarntivijai S, Vasant D, Jupp S, Saunders G, Bento AP, Gonzalez D, Betts J, Hasan S, Koscielny G,
1328 Dunham I, Parkinson H, Malone J (2016) Linking rare and common disease: mapping clinical
1329 disease-phenotypes to ontologies in therapeutic target validation. *J Biomed Semant* 7:8
- 1330 Schreiber G, Wielinga B (eds) (1992) KADS: a principled approach to knowledge-based system
1331 development. Academic, London
- 1332 Schreiber G, Wielinga BJ, Akkermans H, de Velde WV, Anjewierden A (1994) CML: the com-
1333 monKADS conceptual modelling language. In: EKAW, pp 1–25
- 1334 Schreiber G, Akkermans A, Anjewierden A, DeHoog R, Shadbolt N, Van de Velde W, Wielinga B
1335 (eds) (1999) Knowledge engineering and management: the CommonKADS methodology. MIT
1336 Press, Cambridge
- 1337 Schutz A, Buitelaar P (2005) RelEx: a tool for relation extraction from text in ontology extension.
1338 In: International semantic web conference, pp 593–606
- 1339 Shadbolt N, O'Hara K, Crow L (1999) The experimental evaluation of knowledge acquisition tech-
1340 niques and methods: history, problems and new directions. *Int J Hum-Comput Study* 51(4):729–
1341 755

- 1342 Sowa JF (1984) *Conceptual structures: information processing in mind and machine*. Addison-
 1343 Wesley, London
- 1344 Spackman KA (2005) Rates of change in a large clinical terminology: three years experience with
 1345 SNOMED clinical terms. In: AMIA annual symposium proceedings, pp 714–718
- 1346 Steels L (1990) Components of expertise. *AI Mag* 11(2):28–49
- 1347 Stefanidis K, Flouris G, Chrysakis I, Roussakis Y (2016) D2V - understanding the dynamics of
 1348 evolving data: a case study in the life sciences. *ERCIM News* 2016(105)
- 1349 Stefik M (1995) *Introduction to knowledge systems*. Morgan Kaufmann, San Francisco
- 1350 Stojanovic L (2004) *Methods and tools for ontology evolution*. PhD thesis
- 1351 Stuckenschmidt H, Klein MCA (2003) Integrity and change in modular ontologies. In: *IJCAI*, pp
 1352 900–908
- 1353 Stuckenschmidt H, Parent C, Spaccapietra S (eds) (2009) *Modular ontologies: concepts, theories
 1354 and techniques for knowledge modularization*, vol 5445. Lecture notes in computer science.
 1355 Springer, Berlin
- 1356 Studer R, Benjamins VR, Fensel D (1998) Knowledge engineering: principles and methods. *Data
 1357 Knowl Eng* 25(1–2):161–197
- 1358 Suárez-Figueroa M-C, Gómez-Pérez A, Motta E, Gangemi A (eds) (2012) *Ontology engineering
 1359 in a networked world*. Springer, Berlin
- 1360 Svatek V (2004) Design patterns for semantic web ontologies: motivation and discussion. In: *Con-
 1361 ference on business information systems*
- 1362 Szulman S, Charlet J, Aussenac-Gilles N, Nazarenko A, Sardet E, Teguiak V (2009) DAFOE: an
 1363 ontology building platform from texts or thesauri. In: Dietz J (ed) *Proceedings of the international
 1364 joint conference on knowledge discovery, knowledge engineering and ontology development
 1365 (KEOD 2009)*, Madeira (Portugal). Poster, pp 1–4
- 1366 Tissaoui A, Aussenac-Gilles N, Hernandez N, Laublet P (2011) EVONTO - joint evolution of
 1367 ontologies and semantic annotations. In: *KEOD 2011 - proceedings of the international confer-
 1368 ence on knowledge engineering and ontology development*, Paris, France, 26–29 October 2011,
 1369 pp 226–231
- 1370 Tu SW, Eriksson H, Gennari JH, Shahar Y, Musen MA (1995) Ontology-based configuration
 1371 of problem-solving methods and generation of knowledge-acquisition tools: application of
 1372 PROTÉGÉ-II to protocol-based decision support. *Artif Intell Med* 7:257–289
- 1373 Vandenbussche P-Y, Charlet J (2009) Méta-modèle général de description de ressources termi-
 1374 nologiques et ontologiques. In: *Actes d'IC*, pp 193–204
- 1375 Virbel J, Luc C (2001) Le modèle d'architecture textuelle: fondements et expérimentation XXII
 1376 I(1):103–123
- 1377 Zabli F, Antoniou G, d'Aquin M, Flouris G, Kondylakis H, Motta E, Plexousakis D, Sabou M
 1378 (2015) *Ontology evolution: a process-centric survey*. *Knowl Eng Rev* 30(1):45–75
- 1379 Zacklad M (2007) Classification, thesaurus, ontologies, folksonomies: comparaisons du point de
 1380 vue de la recherche ouverte d'information (roi). In: *Conférence CAIS/ACSI*