

Éléments de réflexion sur les composants d'ontologies et leur manipulation par RàPC

Béatrice Fuchs, Amedeo Napoli

► **To cite this version:**

Béatrice Fuchs, Amedeo Napoli. Éléments de réflexion sur les composants d'ontologies et leur manipulation par RàPC. 17ième séminaire sur le raisonnement à partir de Cas, Paris, 2009, Paris, France. pp.107-113. hal-00608168

HAL Id: hal-00608168

<https://hal.archives-ouvertes.fr/hal-00608168>

Submitted on 12 Jul 2011

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.

Éléments de réflexion sur les composants d’ontologies et leur manipulation par RàPC

Béatrice Fuchs¹ and Amedeo Napoli²

¹ LIRIS, Universités de Lyon 1, Nautibus, 8 bd Niels Bohr, F-69100 Villeurbanne
(bfuchs@liris.univ-lyon1.fr)

² LORIA, Bâtiment B, BP 239 F-54506 Vandœuvre-lès-Nancy
(Amedeo.Napoli@loria.fr)

Résumé

Dans cet article, nous proposons quelques premiers éléments de réflexion sur les patrons de conception et les composants d’ontologies, et sur les liens existant avec le raisonnement à partir de cas. Les patrons d’ontologies permettent de guider la conception d’une ontologie et jouent encore le rôle de guides de bonnes pratiques. La conception d’une ontologie est une tâche complexe et nous analysons différentes façons de détecter ce que peut être un patron d’ontologie, la façon de l’adapter et de la réutiliser. Cet article est une première ébauche de réflexion qui demande à être précisé et étendu, notamment avec des exemples pratiques opérationnels, voire de nouvelles propositions de patrons de conception d’ontologie.

1 Introduction

Le Web sémantique peut se concevoir comme un Web où les “agents logiciels” communiquent avec d’autres agents logiciels pour rendre des services et résoudre des problèmes pour les “agents humains” [16]. Ainsi, le Web sémantique se veut un espace de communication et d’interactions où les agents logiciels tirent parti des ressources disponibles pour traiter les problèmes. Parmi les ressources disponibles, il faut distinguer les documents de toutes sortes et les “ontologies”. Les ontologies établissent une terminologie commune entre les agents logiciels et humains leur permettant de partager un même point de vue et un même sens sur les concepts et relations manipulés. C’est pourquoi les ontologies prennent une très grande place dans de nombreux domaines liés plus ou moins directement au Web sémantique, comme la gestion des connaissances, les systèmes collaboratifs et notamment les Wikis sémantiques [9, 21], l’intégration de données, la recherche d’information et le commerce électronique.

Du point de vue de la représentation des connaissances, une ontologie matérialise un “modèle” d’un domaine particulier du monde réel. Pratiquement, un tel modèle se représente comme un ensemble de définitions de concepts munis d’attributs et de relations entre ces concepts. L’utilisation effective d’une ontologie nécessite un langage de représentation des ontologies — de la famille des langages de représentation des connaissances — et des modules associés de raisonnement [3]. S’il est important de disposer d’ontologies, il est néanmoins difficile de les construire [29]. Trois grandes stratégies de conception peuvent être considérées :

- la conception manuelle,
- la réutilisation d’ontologies existantes [15],
- la conception semi-automatique d’ontologies à partir de ressources (par l’intermédiaire de méthodes de fouille de données ou de fouille de textes [5, 4]).

Pour ces trois stratégies, il s’avère utile d’être guidé lors de la conception à l’image de ce qui se pratique en génie logiciel, où la production de code s’appuie l’utilisation de “patrons de conception” et la réutilisation de “composants” [17]. Les principes appliqués au logiciel peuvent être ici repris et adaptés pour donner naissance à des *patrons de conception d’ontologies* ou ODP pour *Ontology Design Patterns* [7, 18, 19]. Comme en ingénierie du logiciel un patron propose une solution générique et réutilisable pour un problème de conception, un ODP propose un modèle — implanté en OWL — pour résoudre un problème local de conception d’ontologie. Par exemple, parmi les ODP se distinguent les “patrons logiques” dont certains recouvrent des opérations

logiques élémentaires comme la conjonction et la disjonction ou d'autres des modes d'inférences comme le modus ponens pour faire émerger des connaissances implicites dans une hiérarchie de concepts, les "patrons d'architecture" pour factoriser et contrôler l'architecture de la hiérarchie des concepts, et des "patrons de contenu" comme la composition (et objets composites, ce qui sera notre exemple courant par la suite), les relations n-aires et les séquences (d'actions).

D'assez nombreux parallèles peuvent être faits entre la recherche et la mise en œuvre de patrons de conception et le raisonnement à partir de cas (abrégé en RÀPC dans la suite) comme évoqué dans [7, 6]. Auparavant, il nous faut faire une différence entre "composant d'ontologie" et "patron de conception d'ontologie" :

- Un composant dénote un bloc élémentaire qui va entrer dans la conception d'une ontologie et qui peut être très spécifique, comme par exemple la description d'une molécule spécifique en chimie organique [25].
- Un patron de conception dénote un bloc générique qui va entrer dans la conception de plusieurs ontologies ou encore qui va servir de base via une adaptation à la création de plusieurs composants, comme par exemple la structure de graphe peut servir de base à la description d'une molécule mais aussi à d'autres structures composites.

Les patrons d'ontologies sont plutôt en petit nombre et donc s'ils se retrouvent assez facilement ils doivent être le plus souvent adaptés. Curieusement un lien direct et assez naturel peut être établi avec le système CADRE [20] où quatre cas génériques d'architecture de bâtiments étaient fournis mais où tout l'effort portait sur le processus d'adaptation de ces cas (en fonction de connaissances du domaine disponibles). Le lien entre architecture et patrons de conception est d'ailleurs explicitement mentionné dans [19].

Le but des éléments de réflexion proposés ici est de faire un point sur les rapports entre conception d'ontologies et techniques de RÀPC, plus particulièrement entre les composants d'ontologies, les patrons de conception d'ontologies et le processus d'adaptation dans la conception d'ontologies. Des liens seront faits avec certaines manières de produire des composants et des patrons, qui sont essentiellement manuelles ou semi-automatiques. Ainsi, un certain nombre de travaux en découvertes de connaissances à partir de documents textuels ou de ressources hétérogènes peuvent être repris et étendus dans cette voie comme par exemple [5, 4].

Cet article est une première ébauche de réflexion qui demande à être précisé et étendu, notamment avec des exemples pratiques opérationnels, voire de nouvelles propositions de patrons de conception d'ontologie. Pour simplifier le texte et lorsqu'il n'y a pas de différence à faire entre patron et composant, nous parlerons simplement de "composant d'ontologie".

2 Quelques problèmes liés à la conception d'ontologies

Dans ce qui suit, nous considérons qu'une ontologie fournit un modèle explicite, non ambigu et opérationnel d'un certain domaine. Une ontologie est codée dans un formalisme de représentation des connaissances comme une logique de descriptions ou OWL [3]. Ce qui nous intéresse en premier lieu ici est ce qui peut être nommé le *schéma d'ontologie*, qui est constitué de l'ensemble des concepts et des relations qui existent entre eux, à la façon d'un réseau sémantique. Les attributs propres d'un concept tout comme le caractère primitif ou défini du concept ne sont pas nécessairement pris en compte dans ce schéma.

Il y a plusieurs façons de considérer les composants et patrons de conception d'ontologies. Tout d'abord, comme pour l'acquisition de connaissances à partir d'expertise, un patron peut se voir comme un bloc de conception générique, comme l'expression d'un savoir-faire ou encore un guide de bonne pratique pour la conception [18, 19]. De tels patrons interviennent dans la mise en forme et la concision de l'ontologie et jouent un rôle tangible sur la qualité de l'ontologie résultante. Dans l'industrie manufacturière par exemple, ces patrons prennent souvent la forme de "gabarits". En parallèle et plus en relation avec la découverte de connaissances, un composant — qui n'est pas nécessairement un patron de conception — peut se voir comme un bloc de conception élémentaire, comme un motif présentant une certaine régularité ou fréquence dans un ensemble de données, qu'il faut savoir caractériser, extraire et ensuite interpréter. Il est possible de combiner et d'associer ces deux visions des composants et patrons : un composant — opération ou structure — devient un patron de conception s'il montre une certaine fréquence et un certain intérêt, traduisant par là qu'il se retrouve souvent employé et jugé utile (par exemple à l'aide d'une fonction de score comme pour les motifs d'intérêt maximal

dans [25]).

Quelques développements récents en ingénierie des ontologies mettent en jeu des processus de découverte de connaissances qui peuvent avoir un intérêt en matière de construction d'ontologies, en particulier à partir de documents textuels. Ces processus mettent en jeu des méthodes de fouille de données et de textes comme l'analyse formelle de concepts [10]. Par exemple, dans [5, 4], l'analyse formelle de concepts sert à extraire et formaliser des concepts à partir de tables de données binaires `objets × attributs` tandis que l'analyse relationnelle de concepts permet d'extraire des concepts munis d'attributs relationnels (codant des relations entre objets) à partir de données relationnelles cette fois (dans des tables `objets × objets`). Les treillis résultant peuvent servir de schémas d'ontologies où un concept est muni d'attributs propres et de rôles qui représentent des relations avec d'autres concepts. Toutefois, ces approches ne peuvent être entièrement automatiques pour plusieurs raisons. Un des problèmes est relatif à la couverture du corpus de textes en entrée : il est difficile de prétendre avoir toutes les connaissances explicitement disponibles pour interpréter effectivement l'ensemble des textes et souvent les connaissances sont disséminées temporellement et géographiquement. Un autre problème est qu'il est difficile d'extraire un savoir-faire et une bonne pratique à partir de documents textuels et que l'expertise est là directement nécessaire (ou encore que les bonnes pratiques sont rarement explicites mais plutôt "enfouies").

Pour en revenir au processus de conception d'ontologies lui-même, comme indiqué dans [18, 19], un utilisateur moyen essayant de mettre en place une ontologie n'a finalement que peu d'assistance disponible : des ontologies qu'il a pu trouver mais qu'il peut quelquefois très difficilement réutiliser, des constructions d'un langage de représentation de connaissances pas toujours adaptées aux besoins courants, et encore dans le meilleur des cas une brassée de bonnes pratiques glanées ça et là : par exemple les bonnes pratiques données dans [2] et ontologydesignpatterns.org/, qui est le site de référence en la matière et qui répertorie les composants d'ontologie disponibles jusqu'à présent. En suivant un scénario assez habituel, il aura fallu rechercher des ontologies en rapport avec le problème, les étudier et en choisir des éléments réutilisables pour les adapter. Parmi ces ontologies, certaines ontologies dites de référence sont supposées être directement réutilisables ou spécialisables comme les ontologies de haut niveau que sont DOLCE ou SUMO [24]. Là encore la tâche de compréhension est difficile et les trop nombreuses propriétés — quelquefois d'ordre métaphysique — sont rarement adaptées aux besoins tout comme la couverture de telles ontologies est bien trop grande dans la plupart des cas pratiques.

À l'opposé, il existe de petites ontologies, bien faites, concises, simples à comprendre et à utiliser (voir `ProtegeOntologiesLibrary` par exemple) mais aussi des "ontologies simplifiées" (voir encore le site ontologydesignpatterns.org/, avec DOLCE+DnS Ultralite par exemple). En parallèle, un certain nombre de travaux se penchent sur le problème de l'intégration d'ontologies et de la réutilisation d'ontologies par morceaux [12, 13]. Pour terminer, il faut encore mentionner des "formalismes portables" et "adaptables" qui ont acquis une certaine popularité et qui permettent de construire des ontologies avec efficacité, et en particulier : (i) SKOS [23] (pour *Simple Knowledge Organization System* qui permet de représenter des vocabulaires, des thésaurus, des partitions et des "folksonomies", et (ii) FOAF [8] qui permet de représenter des réseaux de personnes en relation.

Un parallèle avec la pratique du raisonnement à partir de cas peut être faite ici : recherche de cas similaires, évaluation des cas retrouvés puis adaptation pour réutilisation effective. L'idée est bien de comprendre comment la méthodologie du RÀPC peut aider à choisir les bons éléments dans le bon catalogue qui autorisent une adaptation minimale et efficace (ce même problème se retrouve dans la gestion des services Web [30]).

Pour aller dans ce sens, et en supposant qu'il existe des classes de problèmes qui peuvent être résolus en employant des méthodes génériques, la réutilisation d'éléments pour concevoir des ontologies doit s'appuyer sur de "petites ontologies modulaires" qui vont jouer le rôle de patrons de conception d'ontologies. Ces composants génériques particuliers doivent offrir des fonctionnalités simples et d'intérêt général et doivent être faciles d'accès et de manipulation.

3 La gestion de composants et de patrons d'ontologies

3.1 La recherche de composants et de patrons d'ontologies

La recherche d'ontologies réutilisables, de composants ou de patrons de conception d'ontologies est une tâche qui peut se faire à l'aide de moteurs spécialisés comme Sindice [31], Swoogle [14] et Watson [11, 13], par l'intermédiaire de mots-clés ou encore de “sous-structures” — ici des composants de hiérarchies de concepts. Des mesures peuvent également être utilisées pour focaliser la recherche [1, 26] :

- La mesure d'appariement de classe combine un appariement de chaînes exact et inexact et mesure le degré de correspondance entre les chaînes données en entrée et celles qui font partie des définitions de classe, pour trouver une classe qui satisfait tout ou partie des mots-clés recherchés.
- La centralité dénote une valeur qui est censée mesurer la représentativité d'un concept dans une ontologie sur la base de son placement dans la hiérarchie des concepts. Plus un concept est central en fonction de ses subsumés, de ses subsumants et de sa position par rapport au sommet de la hiérarchie des concepts, plus ce concept est “important”, y compris en terme de réutilisation.
- La densité mesure le degré de détail en termes d'attributs et de rôles pour les concepts retenus. Ainsi, un concept dont la description est plus dense (donc plus riche) sera préféré à un concept de description plus minimale.
- La similarité sémantique mesure la proximité de deux concepts mis en correspondance en termes de longueur de chemins dans la hiérarchie des concepts et des relations. Ici, un appariement qui met en jeu des chemins plus courts va être préféré à un chemin plus long. Ce sont les chemins de similarité qui réapparaissent ici dont le formalisme et les opérations peuvent être repris de la même façon [22].

D'autres mesures existent aussi comme la couverture (comment un concept ou un ensemble de concepts prennent en compte les termes présents qui doivent être représentés), la structure (qui est en rapport avec la densité telle qu'elle est introduite ci-dessus), et la connectivité (qui reflète la connexion qu'un composant peut avoir avec d'autres composants dans d'autres ontologies).

Il faut faire une remarque ici, qui montre certaines limites de ces mesures : les “classes abstraites” en programmation par objets sont introduites pour factoriser des ensembles de propriétés et les distribuer entre classes. Par suite, ce sont des classes denses mais aussi centrales, toutefois elles n'ont pas de véritable existence — elles ne sont pas instanciables et instanciées — et en tout cas n'existent pas (au niveau de l'interprétation) sans les classes qu'elles alimentent.

En termes de RÀPC, la base de cas peut être considérée comme un catalogue de composants qui possèdent tous une représentation formelle et qui recouvrent de petites ontologies quelquefois réduites à un seul concept.

- L'opération de recherche de cas similaires nécessite une analyse des besoins exprimés en entrée, puis un appariement avec les cas de la base (en utilisant par exemple des mesures comme celles données ci-dessus). Les composants sélectionnés, qui répondent aux critères, tout ou partie, sont classés en fonction d'une mesure de similarité ou encore en fonction de leur adaptabilité (donc en termes de chemins de similarité comme évoqué ci-dessus).
- la recherche de composants peut aussi se voir dans ce cadre comme un alignement de composant ou plus généralement comme ce qui pourrait être qualifié “d'alignement par morceaux” comme dans [27, 28] où est définie la notion de patron de correspondances entre ontologies, qui permet d'exprimer des correspondances complexes entre parties ou morceaux d'ontologies.
- Lors de la recherche, la requête est donnée sous la forme d'une liste de mots-clés comme dans la plupart des cas, mais aussi comme une sous-structure, c'est à dire une liste de concepts vérifiant des relations de subsomption entre eux. Dans ce dernier point de vue, l'appariement est plus complexe mais va aussi retourner des cas similaires plus adaptés et adaptables.

Cela donne en particulier des chemins de similarité “bi-dimensionnels” où une dimension fait référence à la structure (des concepts) et une autre dimension fait référence aux relations (dans lesquelles sont impliqués les concepts). Ensuite, des concepts plus généraux ou plus spécifiques peuvent être préférés selon que ce qui prime est le degré de généralité des composants ou la structure. Les concepts les plus spécifiques qui satisfont la requête vont également fournir les composants les plus spécifiques et les plus adaptables.

- Après avoir sélectionné un des composants retournés, l'étape de réutilisation consiste à spécialiser le composant et à l'adapter au problème courant, pour construire une partie d'ontologie. Une révision peut avoir lieu qui permet d'ajuster la réutilisation du composant. Une réutilisation combinée à partir de plusieurs cas peut être envisageable.
- Une opération finale de mémorisation du processus de résolution peut être entreprise si l'intérêt du composant est démontré (les mesures détaillées ci-dessus peuvent être utilisées à ce moment là). En conséquence, le composant peut se voir associer le statut de patron de conception d'ontologie.

3.2 L'adaptation de composants et patrons

Dans cette partie, il est question de composants spéciaux qui sont appelés "patrons de contenu de conception d'ontologie", pour "Content Ontology Design Patterns", abrégés ici en COP [19]. Ces composants ont une nature spéciale et permettent de traiter des problèmes de modélisation. Ils sont considérés selon un double point de vue : le domaine pour une description statique de l'environnement et la tâche pour description des opérations applicables dans l'environnement, ce qui est exactement la façon dont est perçue une classe en programmation par objets. Un même domaine peut autoriser la réalisation de plusieurs tâches tandis qu'une même tâche peut aider à résoudre un problème dans plusieurs domaines. Les ontologies quant à elles jouent plutôt un rôle au niveau du modèle de domaine mais leur conception n'implique aucune utilisation particulière : plusieurs problèmes peuvent être résolus grâce à une même ontologie.

En tant que solution à un problème de conception d'ontologie, un COP fait explicitement mention d'un domaine et d'une tâche. Une classification en catalogue de tels COPs en fonction des tâches et opérations associées peut s'avérer très utile pour fournir des scénarios standards de résolution directement ou encore en combinant les COPs. Imaginons ici la structure de graphe comme un COP qui modélise un ensemble d'entités connectées entre elles. Dans le domaine particulier des molécules en chimie organique, parmi les structures moléculaires, certaines sont génériques mais aussi d'intérêt très général et elles vont donc donner naissance à l'équivalent des COPs, les groupes fonctionnels qui décrivent des familles de réactifs. D'autres structures moléculaires sont individuelles comme les molécules particulières.

L'intuition ici est que les ontologies peuvent être construites en fonction de certaines tâches ou de certains besoins qui sont relatifs à des "questions de compétences" ("competency questions") : à quoi veut-on aboutir et quels types de problèmes est-on censé résoudre à l'aide de l'ontologie. Une question de compétence est une requête (typique) qu'un expert aurait envie de voir satisfaite à l'aide de l'ontologie. Dans l'idéal, une ontologie devrait encoder le nécessaire et pouvoir servir à satisfaire toutes les questions de compétences qui ont servi à sa conception. Dans ce cadre, un COP apparaît comme un guide de bonne pratique qui doit être décrit sous une certaine forme. Un COP a les caractéristiques d'être opérationnel, petit en taille, autonome, hiérarchique ou organisable en hiérarchie, pertinents au niveau cognitif et linguistique, et témoin d'une bonne pratique [19].

Il reste ici à développer un exemple concret et illustratif.

4 Conclusion

Cet article est une première ébauche de réflexion sur les patrons de conception d'ontologies et le raisonnement à partir de cas. Il existe un bon nombre de liens entre les deux disciplines, la conception d'ontologies et le RÀPC, dont il faut pouvoir profiter pour guider globalement la conception d'une ontologie par la recherche et l'adaptation de patrons de conception d'ontologies. En outre, une autre discipline qui est celle de la découverte de connaissances à partir de données (ressources textuelles ici) doit aussi être naturellement associée à cette étude. Comme indiqué, ceci est une première ébauche d'article qui doit encore être travaillée et approfondie.

Références

- [1] H. Alani, C. Brewster, et N. Shadbolt. Ranking ontologies with aktiverank. In I.F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, et L. Aroyo, editors, *Proceedings of the 5th International Semantic Web Conference, ISWC 2006*, pages 1–15, 2006.

- [2] D. Allemang et J. Hendler. *Semantic Web for the Working Ontologist*. Morgan Kaufmann, 2008.
- [3] G. Antoniou et F. van Harmelen. *A Semantic Web Primer*. The MIT Press, Cambridge, Massachusetts, 2004.
- [4] R. Bendaoud, A. Napoli, et Y. Toussaint. Formal concept analysis : A unified framework for building and refining ontologies. In A. Gangemi et J. Euzenat, editors, *Knowledge Engineering : Practice and Patterns – Proceedings of the 16th International Conference EKAW*, Lecture Notes in Computer Science 5268, pages 156–171, 2008.
- [5] R. Bendaoud, Y. Toussaint, et A. Napoli. Pactole : A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In P. Eklund et O. Haemmerlé, editors, *Proceedings of the 16th International Conference on Conceptual Structures, ICCS 2008, Toulouse, France*, Lecture Notes in Computer Science 5113, pages 203–216, 2008.
- [6] E. Blomqvist. Case-based reasoning for ontology engineering. In A. Holst, P. Kreuger, et P. Funk, editors, *Tenth Scandinavian Conference on Artificial Intelligence, SCAI 2008*, Frontiers in Artificial Intelligence and Applications 173, pages 36–43. IOS Press, 2008.
- [7] E. Blomqvist. Pattern ranking for semi-automatic ontology construction. In R.L. Wainwright et H. Haddad, editors, *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC 2008)*, pages 2248–2255. ACM, 2008.
- [8] D. Brickley et L. Miller. FOAF vocabulary specification. Working draft, <http://www.foaf-project.org>, 2005.
- [9] M. Buffa, F.L. Gandon, G. Ereteo, P. Sander, et C. Faron. Sweetwiki : A semantic wiki. *Journal of Web Semantics*, 6(1) :84–97, 2008.
- [10] P. Cimiano, A. Hotho, et S. Staab. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. *Journal of Artificial Intelligence Research*, 24 :305–339, 2005.
- [11] M. d’Aquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, et E. Motta. Watson : Supporting next generation semantic web applications. In *Proceedings of WWW/Internet Conference*, 2007.
- [12] M. d’Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, et D. Guidi. Toward a new generation of semantic web applications. *IEEE Intelligent Systems*, 23(3) :20–28, 2008.
- [13] M. d’Aquin, M. Sabou, et E. Motta. Reusing knowledge from the semantic web with the watson plugin. In *International Semantic Web Conference (Session Posters & Demos)*, CEUR Workshop Proceedings 401, 2008.
- [14] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, et P. Kolari. Finding and ranking knowledge on the semantic web. In Y. Gil, E. Motta, V. R. Benjamins, et M. A. Musen, editors, *The Semantic Web - ISWC 2005 : 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland*, Lecture Notes in Computer Science 3729, 2005.
- [15] J. Euzenat et P. Shvaiko. *Ontology Matching*. Springer, Berlin, 2007.
- [16] D. Fensel, J. Hendler, H. Lieberman, et W. Wahlster, editors. *Spinning the Semantic Web*. The MIT Press, Cambridge, Massachusetts, 2003.
- [17] E. Gamma, R. Helm, R. Johnson, et J. Vlissides. *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading (MA), USA, 1994.
- [18] A. Gangemi. Ontology design patterns for semantic web content. In Y. Gil, E. Motta, V. R. Benjamins, et M. A. Musen, editors, *The Semantic Web - ISWC 2005 : 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland*, Lecture Notes in Computer Science 3729, 2005.
- [19] A. Gangemi et V. Presutti. Ontology Design Patterns. In S. Staab et R. Studer, editors, *Handbook on Ontologies (Second Edition)*. Springer, Berlin, 2008.
- [20] K. Hua, B. Faltings, et I. Smith. CADRE : case-based geometric design. *Artificial Intelligence in Engineering*, 10 :171–183, 1996.

- [21] M. Krötzsch, D. Vrandečić, et M. Völkel. Semantic mediawiki. In *The Semantic Web – ISWC 2006*, Lecture Notes in Computer Science 4273, pages 935–942. Springer, 2006.
- [22] J. Lieber et A. Napoli. Correct and Complete Retrieval for Case-Based Problem-Solving. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI'98)*, Brighton, UK, pages 68–72. John Wiley & Sons Ltd, Chichester, 1998.
- [23] A. Miles et D. Brickley. SKOS Core Vocabulary Specification. Technical report, World Wide Web Consortium (W3C), 2005.
- [24] D. Oberle, A. Ankolekar, P. Hitzler, P. Cimiano, M. Sintek, M. Kiesel, B. Mougouie, S. Baumann, S. Vembu, et M. Romanelli. DOLCE ergo SUMO : On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO). *Journal of Web Semantics*, 5 :156–174, 2007.
- [25] F. Pennerath, G. Polajlon, et A. Napoli. Mining intervals of graphs to extract characteristic reaction patterns. In J.-F. Boulicaut, M.R. Berthod, et T. Horváth, editors, *Discovery Science*, Lecture Notes in Computer Science 5255, pages 210–221. Springer, Berlin, 2008.
- [26] S. Peroni, E. Motta, et M. d'Aquin. Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In J. Domingue et C. Anutariya, editors, *Proceedings of the 3rd Asian Semantic Web Conference, ASWC 2008*, Lecture Notes in Computer Science 5367, pages 242–256. Springer, 2008.
- [27] F. Scharffe, J. Euzenat, et D. Fensel. Towards design patterns for ontology alignment. In R.L. Wainwright et H. Haddad, editors, *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC 2008)*, pages 2321–2325. ACM, 2008.
- [28] F. Scharffe et D. Fensel. Correspondence patterns for ontology alignment. In A. Gangemi et J. Euzenat, editors, *Knowledge Engineering : Practice and Patterns – Proceedings of the 16th International Conference EKAW*, Lecture Notes in Computer Science 5268, pages 83–92, 2008.
- [29] S. Staab et R. Studer, editors. *Handbook on Ontologies*. Springer, Berlin, 2004.
- [30] R. Studer, S. Grimm, et A. Abecker, editors. *Semantic Web Services – Concepts, Technologies and Applications*. Springer, Berlin, 2007.
- [31] G. Tummarello, R. Delbru, et E. Oren. Sindice.com : Weaving the open linked data. In K. Aberer, K.-S. Choi, N. Fridman Noy, D. Allemang, K.-I. Lee, L.J.B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, et P. Cudré-Mauroux, editors, *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference, ISWC/ASWC 2007*, Lecture Notes in Computer Science 4825, pages 552–565. Springer, 2007.