Knowledge Representation for Software Architecture Domain by Manual and Automatic Methodologies

Anabel Fraga, Sonia Sánchez, Juan Llorens

*Universidad Carlos III de Madrid
Madrid, Spain*

{afraga,ssanchez,llorens}@inf.uc3m.es

Hernán Astudillo

*Universidad Técnica Federico Santa María
Valparaíso, Chile*

{hernan}@inf.utfsm.cl

Abstract

At the moment, there is a need for new knowledge representation using Thesaurus or Ontologies because of the need to reuse knowledge. In this paper, a Software Architecture knowledge representation is created, for that purpose a manual and automatic methodology for creating it is used. A new manual methodology is provided in the paper. CAKE (Computer Aided Knowledge Environment) is the automatic process used as automatic methodology. The result is the first thesaurus in English for the Software Architecture Domain using the new manual methodology presented in the paper and the first ontology in Spanish for the Software Architecture Domain using the automatic methodology.

Keywords

Knowledge Representation, Software Architecture, Knowledge Management, Methodologies for Knowledge Representation, Thesaurus, Ontologies, Reuse

1. Introduction

Methodologies used for developing thesaurus following ISO 2788 exist [LA86] [AI97] [CU98] [VA91] [NIS]¹. Nevertheless, even with the unquestionable utility of thesaurus and controlled vocabulary as a base for the Semantic Web, there are difficulties for their generation. The main problem is the hard intellectual process needed; it does not generate motivation on experts to develop the process. Additionally, the low rate of available domain experts to collaborate in the large process for creating/evaluating a knowledge representation and the fast growing rate of information on domains repels on the methodologies. But it is intended to suggest a new manual methodology and how to use CAKE as an automatic methodology [CAK] [LFA] [DI05] [AN04] [BA99] [BSW].

The knowledge representation construction is supported by two references: ISO 2788:1986 for Monolingual thesaurus and ISO 5964:1995 for multilingual thesaurus. Recently the ANSI/NISO Z.39.19:2005 was published with the name "Guidelines for the Construction, Format and Management of Monolingual Controlled Vocabularies"; it was created for content representation, adapting different Knowledge Organization Systems and availing SKOS W3C's initiatives. The ISO and NISO recommendations established terminology taxonomy for domain representation and wide causal relations.

¹ Leonard Will, <http://www.willpowerinfo.co.uk/thesprin.htm>

All available recommendation is oriented to manual building but not automatic, so they were adapted to automatic building of knowledge representation systems. The translation of ISO2788 for Spanish is UNE 50-106-90 [UNE], and for ISO 5964 is UNE 50-125-97, both directives were published by AENOR in 1990 and 1997 respectively. They are guidelines and conventions for the contents, display, construction, testing, maintenance, and management of monolingual controlled vocabularies. It focuses on controlled vocabularies that are used for the representation of content objects in knowledge organization systems including lists, synonym rings, taxonomies, and thesauri.

CAKE as an environment allows creating software using a knowledge management development paradigm [LFA]. Many authors treated the advantages to make an automatic process in order to create thesaurus [LA86] [HE92]. Partial proposals for terms and relations selections are predominant. Terms selection [CH88] [EV91] is based on frequencies calculus for extended corpus. Relations selection has been focused on techniques based on co-occurrence analysis, statistic [JU91], linguistic [HE92] and mixed techniques [GR94].

The last few years have seen initiatives to identify concepts in a semantic way. Thus, controlled vocabularies and metadata have been used because of the compatibility these techniques' results have with ontology languages [OWL] [RDF]. Some new proposals that build on these ideas are SKOS 2005 [SKO] and SWAD [SW01], which deal with knowledge organization systems using thesaurus standards [TM1] [UNE]. The key problems remain knowledge structure generation [CON], translation into different languages, exchange and merging, and its expression in the Semantic Web [ODM].

This study uses a newly proposed manual method for creating knowledge representations. A Software Architecture thesaurus was created in English. That creation was supported by tmCAKE [REU] as a thesaurus management tool of the CAKE family tools available at The Reuse Company web page. On the other hand, a Software Architecture thesaurus was created in Spanish, in that case CAKE methodology was used for the process. Those are the first Software Architecture Thesaurus for the Domain. It helps to increase the bases for the semantic web and it is the first step in order to create, as a future work, a Software Architecture Ontology. It is not intended to make a difference between methodologies but it will be a future work.

The problem to solve is to build ontology-based tools to manipulate SA artefacts and knowledge. Currently, no controlled vocabularies for SA are widely known, much less accepted [CFD]. Some works are available on the field like Mehta's paper [MH00] for Software Architecture Connectors Taxonomies, or Keshav's investigation on a Taxonomy of Architecture Integration Strategies [KE98], or maybe a Taxonomy of Orthogonal Properties of Software Architectures [BR99], but a Taxonomy for the Software Architecture Domain as a whole at the moment is not available. One of the IASA (International Association of Software Architects) Workgroups of IT architects concern is a taxonomy for the domain in order to provide clarity, to chart out the domain for the entire worldwide IASA network [IASA]. For that, we have developed an ontology of the SA domain, which is described later in this article. To maximize the impact of this work, we have developed it using two different methods, one automatic (NLP-based) and one manual (traditional way to construct knowledge representations).

The reminder of this article is structured as follows: Section 2 explains some key concepts and problems of SA, Section 3 surveys some related previous work, Sections 4 and 5 describe the manual and automatic methods respectively, Section 6 presents both ontologies and compares them, and the final sections cover Future Work and Conclusions.

2. Software Architecture Domain introduction

The Software Architecture Domain has been active since the early 90's [GA94] [SH00], when researchers began to focus on design and the abstraction level required for architecture.

The IEEE defines Software Architecture as: “Architecture is the fundamental organization of a system, embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution”. Software architects privilege Non-Functional Requirements (NFRs) than on functional requirements because the former are much harder to satisfy in large and distributed systems. NFRs cannot be satisfied with local design decisions, but require global solutions because they correspond to systemic properties. Software Architecture are used for training of team, estimate and manage impacts of changes, integration with other systems, quality evaluation, reuse, and communication with stakeholders.

The literature and documentation available for this domain is quite extensive in English, but quite poor in Spanish; hence, the indexing process for the English language could be difficult and could not be accurate to the reality. But representing the domain in English could be easier because of the lot of information and books available. The automatic methodology could not be used for English because it is still working on this matter.

The advantages of building new thesaurus or ontologies for the domain could be interesting in order to improve communication between domain experts, makes a better retrieval of knowledge, allows classifying documents and could be useful if new architects need to get new structures knowledge. For Software Architecture domain does not exist that kind of structured knowledge representation in English or Spanish, so there is an obvious need in the community in order to reuse it. We aim to solve in some way this problem.

3. Manual methodology for creating thesaurus

The new manual methodology proposed follows the next steps:

1. Domain identification [AI97] and micro disciplines division [VA91]. Some internals consideration has been taking, like auxiliary themes, precision grade, document classes.
2. Terms and relationships selection. A search of domain information is needed at this point. Documents containing specific vocabulary have been chosen: glossaries, thesaurus, papers, and so on. In a next phase could be chose as well specific documents and natural language terms (NL).
3. Creating a base document that contains a resume of the most important knowledge in the domain, could be a key for better thesaurus to be obtained because people building thesaurus should be prepared at least on the basis of the domain to be represented. That is a new step in the traditional manual methodologies and it represents a suggestion for manual generation of knowledge representation. In order to complete that step, the processes of searching, reading and learning are essentials. A weakness is obvious, not all people are available for playing the role [AS98] [BA01] [SH00] [PE92] [LU99] [GA94].
4. Use of a thesaurus management tool, like tmCAKE², to include terms, manage relations, and manage scope notes. The result will be a thesaurus created from scratch using tmCAKE. The knowledge creation process is in into the creator.
5. Generate reports using tmCAKE: hierarchical, hyperlinked, and alphabetic in order to illustrate the product by standard views.

² Thesaurus Manager made by Reuse Company. Available version at: <http://www.reusecompany.com/> [03/07/05]

4. Automatic methodology for creating thesaurus

The CAKE³, an automatic methodology for creating thesaurus⁴ is based on electronic documents; those documents are not structured in many cases. For example, documents written in natural language are not structured at all. It is possible to select a set of terms following ISO 2788 and could be possible to identify relationships between terms like generalization-specialization, association, synonymic or equivalence. CAKE follows ISO 2788 as a standard for create knowledge representation.

CAKE uses a natural language process system (NLPS) for content analysis. On the other hand, it uses heuristic techniques for filtering the terms to be included.

CAKE methodology fundamentals

The CAKE fundamentals for creating thesaurus are based on personnel involved in the creation process and the family tools provided by the company. Personnel required for creating thesaurus should be: domain engineers, domain experts, and a domain responsible. For that, a set of tools and techniques had been developed in order to make the process. Those tools are based on textual treatment, terms identification, relations identification, automatic indexing, presentation and maintaining of the knowledge represented which evolved one day after another.

4.1. Personnel roles

The roles established on CAKE methodology are three: domain engineer, domain expert and a domain responsible. Each one must have specific knowledge and must observe some rules.

The domain engineer is an expert building domains and he/she must have a high level of abstraction. That role must be an expert on tools and CAKE methodology, and should be in contact with domain experts. The main task is to apply filters over documents and results extracted in the automatic process using the tools.

The domain expert must have a whole knowledge of studied domain. As any methodology dream, should be a goal to have more than one expert in order to obtain a better off verification.

The domain responsible is a domain expert as well, but he/she is the responsible for the final product. A person on that role should study and set the frontiers for the domain and the viability of the project. He/She is responsible of document searching guidelines with the domain engineer, and together they get quality for the domain. Also he/she is in charge of place domain expert's decisions and maintains the thesaurus generated.

As shown in figure 1, the roles used in the methodology are refined in a graphical definition and relations between them are shown in addition.

³ The Reuse Company is in the process of developing a tool for Ontology management, as well as tmCAKE is for Thesaurus management. A future work will be to complete the thesaurus using the new tool following CAKE paradigm in order to obtain a complete knowledge representation.

⁴ At the moment CAKE methodology was developed for Spanish language, it will support multilingual because it is on development for English. It should be a future work of this paper to develop the Software Architecture Domain Thesaurus using CAKE for the English language as soon as it is fully operational.

Figure 1: Roles in CAKE

4.2. CAKE tools

The CAKE tools for automatic knowledge⁵ creation are classified as follows: Vocabulary identification, relations identification and knowledge management. Vocabulary identification and relations identification follows natural language processing techniques (NLPT). NLPT uses heuristic methods. Thesaurus management uses NLP only if it is maintained by automatic indexing.

4.2.1. Vocabulary identification

Vocabulary identification uses a NLP system, a vocabulary filtering system and a keyword evaluating system.

NLP system

⁵ On this case should be understood the term automatic as an almost automatic process. Terms and relations extraction could be created using automatic techniques, but it is recommended to be supervised by specialist personnel in order to get a thesaurus with quality. On this case the process could be considered as semi-automatic.

The NLP system is in charge to process documents and it identifies simple and complex units by morphologic categorization for different words. Once identified the possible morphologic categories and after the disambiguation is done then there are applied techniques for syntactic analysis in order to identify unit's lexical-semantic candidates to be possible descriptors.

Vocabulary filtering system

The filtering process consists of selecting terms from documents depending of frequency of appearance (FT-FDI). This process must be supervised by domain engineers. The filtering system pretends to select relevant vocabulary for the automatic process. That process could be done validating the vocabulary and keeping it in a data base. A domain expert will be able to select valid terms in order to obtain relevant relation for the thesaurus/ontology.

Descriptor term validation system

The evaluation system for descriptors has two phases: keyword list validation system phase and evaluated by experts terms validation system phase.

By applying keyword list validation system via web, could be possible to obtain a collaborative environment where domain experts in different locations could interact. It is a simple application that allows accepting, rejecting, discarding or suggesting new candidates as valid terms.

Terms validation is done using a tool that takes suggestions made by domain experts. The domain responsible is in charge of that process. Decisions of keywords or descriptors candidates are made using statistics and heuristic methods. The result of this phase is a list of descriptors to conform the thesaurus/ontology.

4.2.2. Relations identification

Relations identification could be fulfilled in more than one phase. The methodology distributes it in three phases: relations extraction system, relations filtering system, and relation validation system. All relations found would be kept as RSHP relations into the database [DI05].

Relation extraction system

The relation extraction system uses NLP system. It extracts relations between known terms. Relations are identified by dependencies between different composed word structures, dependencies between composed word structures having verbal units, and identifying trigger words. The result of that process is an extended set of all possible relations between terms extracted from the documental corpus. The domain engineer should be in charge of that step process.

Figure 2: Relations extraction system

Relations filtering system

Relation filtering system detects and analyse contradictory relations like relations between descriptor terms and not descriptor terms (synonymies), generic-specific relation in both directions, generic-specific curled relation and generic-specific transitive relation. The involved role in this process is the domain engineer. The result is a list of relations between descriptor terms.

Relation evaluating system

Relation evaluating system is done using a tool able to manage evaluations and suggestions made by domain experts. The involved role in this process is the domain responsible. Taken decisions on the process about relations going out or staying in are done by statistics generated after domain experts' evaluation. The result of that phase is a list of candidate's relations.

4.2.3. Thesaurus management

Finally, two additional tools are needed to generate the knowledge representation: the generating tool and the management tool. Using that last process will be obtained the goal and it will be allowed to be maintained.

Thesaurus Generation

Once descriptor terms are extracted and relations between them are placed, the CAKE tool would be able to prepare terms and relations. It is a simple process because all information extracted from previous phases is available. The role involved on this phase is the domain engineer, but any involved personnel in the process are prepared to assume that role. It is recommended for the process that domain engineers help domain experts to evaluate the result.

Figure 3: Thesaurus generating system

Thesaurus Management

The result approved by experts could be presented and managed using tmCAKE. As well it could be possible to generate indexes and analyse results by statistical reports using relevant information.

On the other hand, tmCAKE could be used in an independent way to create thesaurus by a manual methodology from scratch, it is one of the strength of that tool. Figure 4 shows the tool working.

Thesaurus maintenance

The knowledge representation should be maintained using suggestions made by users and domain experts. Knowledge is not static and that step is important for maintaining knowledge updated. In case of using automatic indexing tools, the system proposes a list of candidates periodically as well as new relations but the corpus must be updated.

5. Comparing methodologies used for the domain and results

The manual method helps to represent knowledge with quality. It can be executed by a single person, and a domain expert may later assist in validating the results. The automatic method gives guidance to create new representations of knowledge in a fast way, and it is supported by tools to represent, treat and create knowledge. So, both methods can be used depending on: corpus available, expert of the domain, time available to elaborate the Thesaurus or Ontology, language, and other factors.

For the Software Architecture domain, the manual method has been more useful for constructing the English Ontology because English documents for this domain are everywhere, which is not so with Spanish language documents. The automatic method requires a solid corpus, well documented, following rules and acceptable writing syntactic, in order to be processed in an effective way.

The representation in Spanish (done using the automatic methodology) defines the "Software Architecture" domain with smaller precision than the one in English. Includes an amount of more general terms, because of the limit of texts founded in Spanish, this fact makes useful to include more general information at the moment of building the corpus. The English representation is more precise, i.e. it uses more terminology of the domain because the used sources come from the main domain experts. The precision refers to the quality of the domain terms as far as controlled vocabulary. The ontology must accurately reflect the terms and references between them to enable a good index process. For example, if comparing the terms "patterns" (in English) and "patrones" (in Spanish), the knowledge representation must reflect that the ontology in English represents some architecture patterns hierarchically faceted, that "architectural pattern" is a domain term, and it is a NT ("Narrowed Terms") of the generic term "Pattern" (see term "Patterns" in the English representation). On the other hand, the thesaurus in Spanish does not make this hierarchy so clear and it leaves a fuzzy definition being only alphabetical representation (see term "Patrones" in the Spanish representation). This is just an example of some better representations that could be obtained by a human specialist due to the use of existing documents for the domain, and it's not clear that the automatic methodology could generate the same knowledge representation even if the corpus has the documents in English.

Manual methodology results

Following the manual methodology exposed on this paper the result is a Software Architecture knowledge representation (Thesaurus). A clear disadvantage using that methodology could be the fact that it could take more time to be done because of the learning involved and the creating process is expensive for human beings. On the other hand, thesaurus could be generated with more quality in order to obtain a better based semantic web. But if the amount of information is unmanageable people becomes unable to create it in a short time. A maintenance is needed even the thesaurus is not finished.

The knowledge representation has over 500 terms. It is faceted-hierarchic and it has relevant information for some terms like scope notes, synonyms, and reference sources. The relation rate is about 1,47; it was calculated dividing total relations in the thesaurus (not synonymy) by total concepts. Estimated time in developing process by one person was over two months.

The results are available at <http://www.reusecompany.com/SAE/index.asp>

Figure 4: Software Architecture Thesaurus in English language by tmCAKE inspection.

A fraction of the hierarchy representation generated by the tmCAKE tool for the resulting thesaurus is:

..Be	...Communicator
...Abstract Level	...Creativity
..High	...Ethics
..Low	...Goal Oriented
..Medium	...Intuitive
..Architect	...Leadership
..Skills	...Parsimonious
..Architect Framework	

...Pragmatism	..Generic Models
...Pro-active	...Bottom-up
...Researcher	..Reference Models
...Teacher	...Top-down
...Tolerant in ambiguity	..Architecture (Software)
...Visionary	..Architectural Blueprints
..Do	..Architectural Decisions
...Commitment	..Architectural Design Process
...Consulting	..Architecture Qualities
...Leadership	..Architecture Representation
...Negotiation	..DCE
...Strategy	..Domain Architecture
..Know	..Enterprise Architecture
...Organizational Politics	...Business Transformation Methodology
...Technology	...
..Architectural Models	

CAKE automatic methodology results

Following CAKE, as an automatic methodology for creating knowledge representation, a Software Architecture Thesaurus in Spanish is created. It has over 1200 terms more or less and it is hierarchic-alphabetic. The relation rate is 0,80 and it was generated in a week.

The results are available at <http://www.reusecompany.com/SAS/index.asp>

A fraction of the hierarchy representation generated by the tmCAKE tool for the resulting thesaurus is:

...	
..4+1	..ABSTRACCIÓN CLAVE DEL DOMINIO OBJETOS/CLASES
..4+1 VIEWS	..ABSTRACCIÓN CLAVE DEL DOMINIO OBJETOS/CLASES
..4+1 VIEWS	..ABSTRACT
..9600 BPS	..ABSTRACT FACTORY
..ABSTRACCIÓN	

..ABSTRACT PRODUCT	..DISEÑO
..ABSTRACT FACTORY	...DISEÑO CORRECTO
..ABSTRACT PRODUCT	...DISEÑO DÚCTIL
..ABSTRACTAS	...DISEÑO PORTÁTIL
..ABSTRACTAS REUSABLE	...DISEÑO REUSABLE
..ABSTRACTAS REUSABLE	..ACTIVIDAD COMERCIAL
..ACID	..ACTIVIDAD CONCURRENTES
..ACOPLAMIENTO	..ACTIVIDAD PROPUESTAS
..ACTIVIDAD	..ACUERDO
..ACTIVIDAD COMERCIAL	..ADL
..ACTIVIDAD CONCURRENTES	..ADLS
..ACTIVIDAD PROPUESTAS	
..AUTOR	

6. Future works and Conclusions

Some future works had been pointed in previous sections, one work to be done will be to improve and maintain the knowledge represented having help from software architecture domain experts.

A comparison between manual and automatic methods could be interesting in a formal way, not only the scratch showed in the paper. Variables will be studied, defined and tested for evaluating quality and time in thesaurus creation.

Another improve for the work done could be to evolved in to a complete ontology the knowledge represented but first must be needed to improve and maintain it to be sure the result is updated to be evolved.

The possible integration of available taxonomies into the ontologies could be interesting, just to complete it in detailed fields [BR99] [ME00] [MH00] [KE98].

Knowledge representation is important to develop and promote development of new knowledge representation for domains. Quality should be evaluated because the results of the process for creating them fast and clearly could generate better support for the net to come. New ways to generate knowledge are important and considering the high growing rate of information in the network manual methodologies become ineffective. For that reason, new automatic methodologies should be used and promoted.

A manual method, as suggested, will generate products with quality for knowledge to represent and it could be done by a single person. The time the process takes is the main dilemma for that methodology to be used, but the effort every one makes in order to generate new knowledge will be the key of success or failure.

The automatic method offers guidance to represent new knowledge, treat it and generate it. Quality of results depends on evaluations that should be made by all the people involved in the process.

The time invested to classify/define controlled vocabulary diminishes the time to retrieve documents and knowledge that have been stored. If domains are documented in Spanish, it is recommendable to use the CAKE method, as long as the text is precise and the writing is clear, but we recommend using manual methods for domains little documented in Spanish but documented in other languages. In this case, the creation effort will be more than generation with CAKE.

In future cases of information retrieval for the Software Architecture domain, these ontologies can be used to improve search precision.

The result is the first global Software Architecture ontology for the domain using both methodologies and in both Spanish and English.

7. Bibliography

7.1. Architectural Bibliography

[AS98] Astudillo, H.; Hammer, S.: "Understanding the architect's job." Software Architecture Workshop of OOPSLA'98 (1998).

[BA01] Bass, L.; Clements, P.; Kazman, R.: *Software Architecture in Practice (2nd Ed.)* Addison Wesley (2001).

[BR99] Bratthall, Lars; Runeson, Per. *A Taxonomy of Orthogonal Properties of Software Architectures*. Workshop in NOSA. (1999).

[GA94] Garlan, D.; Shaw, M.: "An introduction to software architecture." http://www.cs.cmu.edu/afs/cs/project/able/www/paper_abstracts/intro_softarch.html Carnegie Mellon University Technical Report CMU-CS-94-166.(January 1994).

[IASA] International Association of Software Architects. http://www.iasahome.org/iasaweb/appmanager/home/workgroup?_nfpb=true&_pageLabel=working_group_holder [10.03.2006]

[KE98] Keshav, R.; Gamble, R. *Towards a taxonomy of architecture integration strategies*. Proceedings of the third international workshop on Software architecture (89-92). (1998).

[LU99] Lung, CH.; Kalaichelvan, K.: *An Approach to Quantitative Software Architecture Sensitivity Analysis*. Nortel Networks. (1999).

[ME00] Mehta, Nikunj; Medvidović, Nenad; Rakić, Marija. *Why Consider Implementation-Level Decisions in Software Architectures?*. University of Southern California Technical Report [usccse2000-514](#). (2000).

[MH00]Mehta, Nikunj R.; Medvidovic, Nenad; Phadke, Sandeep. *Towards a taxonomy of software connectors*. Proceedings of the 22nd international conference on Software engineering. (178-187). Ireland. (2000).

[PE92] Perry; Wolf: "Foundations for the study of software architecture." ACM SIGSOFT Software Engineering Notes, 17(4) (October 1992).

[SH00] Shaw, M. *The Coming-of-Age of Software Architecture Research*. International Conference on Software Engineering (656-664a). <http://citeseer.ist.psu.edu/shaw01comingage.html>. (2001).

7.2. Thesaurus and Ontology Bibliography

[AI97] Aitchison, Jean; Gilchrist, Alan; Bawden, David. *Thesaurus construction and use: a practical manual (3rd Ed.)*. Aslib, London (1997).

[AN04] Antoniou; Van Harmelen: *A Semantic Web Primer*. MIT Press, 2004.

[BAE99] Baeza-Yates, R.; Ribeiro-Neto, B.: *Modern Information Retrieval*. Addison-Wesley, Wokingham, UK, 1999.

[BSW] Dumbill, Edd: "Building The Semantic Web." XML.com, March 7 2001 <http://www.xml.com/pub/a/2001/03/07/buildingsw.html> [03.07.2005]

[CAK] CAKE. http://europa.eu.int/information_society/istevent/2004/cf/document.cfm?doc_id=533 [03.04.2005]

[CFD] Computing Free Dictionary. The Free Dictionary. <http://computing-dictionary.thefreedictionary.com/software+architecture> [20.02.2006]

[CH88] Choueka, Y.: "Looking for needles in a haystack or locating interesting collocational expressions in large textual databases." In: Proceedings of the RIA O Conference on User-Oriented Content-Based Text and Image Handling, Cambridge, MA (March 21-24, 1988).

[CON] Conocimiento. <http://www.ontoknowledge.org/oil> [03.10.2004]

[CU98] Currás, Emilia. *Tesauros. Manual de Construcción y Uso*. Madrid. KAGER II (1998).

[DI05] Díaz, I.; Llorens, J.; Génova, G.; Fuentes, J.M.: "Generating domain representations using a relationship model." Information Systems, 30(1): 1-19, March 2005 (March 2005).

[EV91] Evans, D.; Henderson, S.; Lefferts, R.; Monarch, I.: "A summary of the CLARIT project." Technical Report CMU-LCL-91-2, Laboratory for Computational Linguistics, Carnegie-Mellon University, Pittsburgh, PA USA (November 1991).

[GR94] Grefenstette, G.: *Explorations in Automatic Thesaurus Discovery*. Kluwer (1994).

[GO04] Gómez-Pérez, A.; Fernández-López, M.; Corcho, O.: *Ontological Engineer*. Springer (2004).

[HE92] Hearst, M.: "Automatic acquisition of hyponyms from large text corpora." In Proceedings of the Fourteenth International Conference on Computational Linguistics. Nantes, France: COLING'92 (1992).

[JU91] Justeson, J., S.; Katz, S.M.: "Co-occurrences of antonymous adjectives and their contexts." Computational Linguistics, 17: 1-19 (1991).

[LA86] Lancaster, F. W.: *Vocabulary control for information retrieval. 2nd ed.* Arlington, Virginia. Information Resources Press (1986).

[LFA] Llorens, J.; Fuentes J.: "Computer Aided Knowledge Environment CAKE". The Reuse Company, Winter (2004). <http://www.reusecompany.com> [23.06.2005]

[NIS] NISO: *Z39.19-200x Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. ISBN: 1-880125-653 <http://www.niso.org/standards/balloting.html> [14.09.2004]

[ODM] OMG: *Ontology Definition Metamodel*. <http://www.omg.org/cgi-bin/doc?ad/2003-3-40> [16.09.2005]

[OWL] W3C: *OWL* <http://www.w3.org/2004/OWL/> [02.06.2005]

[REU] The Reuse Company. <http://www.reusecompany.com> [12.07.2005]

[RDF] W3C: *Getting Into RDF & Semantic Web Using N3*. <http://www.w3.org/2000/10/swap/Primer> [10.06.2005]

[SKO] SKOS. <http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20050510> [10.06.2005]

[SW01] *SWAD-Europe Thesaurus Activity*. <http://www.w3.org/2001/sw/Europe/reports/thes/> (SWAD-E Thes, 2001) [11.04.2005]

[TM1] Site de Ontologías. <http://www.willpower.demon.co.uk/thessoft.htm> [10.07.2005]

[UNE] *Norma UNE 50-106-90. Directrices para el establecimiento y desarrollo de tesauros monolingües*. Madrid. AENOR. (1990).

[VA91] Van Slype; Georges: *Los lenguajes de indización. Concepción, construcción y utilización en los sistemas documentales*. Madrid. Biblioteca del Libro. Fundación Germán Sánchez Ruipérez (1991).