5

Transforming Natural Language into Controlled Language for Requirements Elicitation: A Knowledge Representation Approach

Carlos Mario Zapata J and Bell Manrique Losada
Universidad Nacional de Colombia
Universidad de Medellín
Colombia

1. Introduction

Requirements elicitation is one of the first processes of the requirements engineering for software development. At the beginning of this process, a team of analysts and stakeholders must capture the requirements belonging to the domain—expressed in natural language—in which the future software application must work. Then, the analyst must transform the captured information into artifacts—mostly diagrams—(Leite, 1987), which specify this domain.

The requirements engineering research community proposes some models for understanding the requirements elicitation process. We discover such models in the state-of-the-art review. The models are used for representing knowledge on this conceptual domain by using a formalization that specifies the most common concepts. Among these models we can cite ontologies, meta-ontologies, and meta-models.

With a model, we can represent, describe, or specify something. In the requirements engineering field of knowledge, we can use a domain-model to describe the concepts of a domain and its interrelations, as well as a business-model to represent the business rules of a company. Ontologies, meta-models, and meta-ontologies are examples of models. A descriptive, structural model, representing reality by a set of concepts, their interrelations, and constraints ruled by open-world assumption (Assmann, 2006) is called *ontology*. A *meta-model* is the definition of a set of valid models, facilitating their transformation and exchange. In meta-models, statements are made about what can be expressed in the valid models of a certain modeling language (Seidewitz, 2003). Basically, a meta-model is the description of a set of other models and tries to describe the language elements that can be used to create models, along with aspects related to the modeling technique, the modeling language, and the modeling procedure. Finally, a *meta-ontology* is a model that formalizes the relationships among several categories of formal and semi-formal concepts and also the connection of these theories to the informal concepts of a domain.

The representation of the conceptual domain knowledge related to the requirements elicitation has been addressed by several models. Most of them fail to describe some issues: the elements represented in the natural language discourse vs. the models and diagrams

(specifications) expressed in controlled languages, the process followed by analysts for transforming the natural language discourse into controlled language specifications, and the relationship between structural and dynamic elements within the process.

In the search for a solution to these drawbacks, in this Chapter we present a meta-model to represent knowledge about these issues by using a pre-conceptual schema. The analysts can use this meta-model for improving the implementation of requirements elicitation processes in several domains.

The remainder of the Chapter is organized as follows: in Section 2 we describe the conceptual framework about requirements elicitation, knowledge representation, natural language, and controlled language. In Section 3 we review knowledge representation related to the requirements elicitation process and we discuss existing approaches to follow this process. In Section 4 we present the proposed meta-model and, then, we describe this kind of knowledge representation. Finally, in Section 5, we present the conclusions.

2. Conceptual framework

2.1 Requirements elicitation

Requirements elicitation is one of the first phases of requirements engineering for software development. The primary goal of the requirements engineering process is focused on discovering all the requirements which the future system needs to satisfy for being considered successful. In order to achieve this goal, two major activities—requirements elicitation and requirements analysis—are iteratively and incrementally carried out, involving an informal Natural Language and a formal modeling language (Li *et al*., 2003).

The requirements elicitation phase has several activities, including: understanding the domain, capturing and classifying requirements, establishing priorities, resolving conflicts, and negotiating system requirements (Robertson & Robertson, 2006).

Requirements elicitation is primarily concerned with the communication between the analyst and the stakeholder (customer, end-user, domain expert, and so on,) as a way to gather the essential and relevant domain information. Such information is considered the basis for the requirements elicitation process.

2.2 Natural language and controlled language in requirements elicitation

The requirements elicitation process needs, as a basis for the entire process, the usage of two kinds of languages: Natural Language and Controlled Language, which are explained as follows.

2.2.1 Unrestricted grammars

Grammars are intended to describe the structure of a language. They provide a precise and understandable syntactic specification. A grammar comprises mainly a set of production rules describing syntactically valid phrases made by using the language (Jiang & Li, 1998).

Chomsky defined a hierarchy (1995) with four classes of grammar (0-3). Type 0 is the unrestricted grammar and the other types are derived by increasing restrictions on the form

of the productions that may be used. Languages generated by using type 0 grammars are recognized by Turing machines. Also, they are called computably enumerable languages (CE-languages; Ruohonen, 2009).

2.2.2 Natural Language (NL)

The following issues are commonly addressed when working with NL: i) lexical ambiguity related to alternative parts of speech and word classes (article, preposition, adjective, noun, pronoun, adverb, verb, etc.) and ii) syntactic ambiguity, due to the complexity of sentence structure to avoid redundancy.

According to Berry (2003), the vast majority of requirements are written in natural language. The identification of the concepts used by the domain expert and the relationships among them are very important for the analyst. These concepts and relationships are considered the basis for the common language understanding used by the requirements engineer and the domain expert. According to Li *et al.* (2003), NL is highly informal in nature, because speakers and writers frequently create new forms and combinations of a language. This inventiveness is unfolded especially by the continuous evolution of nouns.

2.2.3 Controlled Language (CL)

Controlled Language is used—in requirements engineering and other similar knowledge areas—to improve the clarity of expression in the source text and the quality of the analysis phase (Mitamura & Nyberg, 1995).

The definition of a controlled language based on some subset of the natural language is important to establish a controlled vocabulary and a controlled grammar. The controlled languages restrict the translation so that only a pre-defined vocabulary is used. With this vocabulary, an analyst can write and standardize rule-based readable texts. If the grammatical constraints of the source language are formally specified, then a machine translation system may take advantage of less-complex and less-ambiguous texts (Mitamura & Nyberg, 1995).

2.3 Knowledge representation

The requirements elicitation and other similar processes are intended to gather information, which must be organized and modeled, in order to create a meaningful whole: the conceptual model. Either the clarification or the solution of a problem can apply the definition of a concept as a structure, derived from the information acquired. Consequently, the first task in creating a knowledge representation in a specific domain is the conceptualization, defined as the usage of concepts and relationships to deal with—and probably solve—a problem (Andrade *et al.,* 2003). Accordingly, a conceptual model is an abstraction of the universe of discourse, as well as a kind of knowledge representation, and a possible model of a possible solution for the problem.

Knowledge representation requires all the information elements to be identified from the analysis, and classified or grouped into levels, depending on what function they must fulfill in the specific domain. According to Andrade *et al.* (2003), such levels are: *static information*, which comprises the structural or declarative domain information about the problem (*i.e.*,

concepts, properties, relationships, and constraints), and *dynamic information*, needed to conform the behavior that takes place in the domain (functionality, actions, etc.)

Requirements elicitation is a negotiation process in which knowledge is intensively captured, combined, and disseminated (Falbo *et al.*, 2004). In this process, analysts and stakeholders should exchange information about the context and the activities to be supported by the software application under development. This process involves knowledge representation actions—from all possible information available—including the behavioral analysis of previous systems.

Zapata *et al.* (2006) propose the pre-conceptual schemas (PS) as knowledge representation models. The PS is an intermediate schema for facilitating the mapping process between natural language specifications and Unified Modeling Language (UML) diagrams. The main PS symbols are concepts (rectangles), relationships (ovals), connections (thin arrows), and implications (thick arrows). Also, conditionals in PS are expressed by rhombs.

3. State-of-the-art review

In the requirements elicitation process, analysts and stakeholders should exchange information concerning the context and activities from the domain. This information is useful to understand how we can obtain conceptual schemas from it. Obtaining a controlled language from the natural language discourse can be considered a useful idea to facilitate and improve this elicitation process. Along this line of thought, several approaches have been proposed.

Commonly, some approaches to knowledge representation have been identified in the requirements elicitation process. Because of this diversity, the state of the art encompasses a number of models from different viewpoints. These models are categorized as meta-models, meta-ontologies, and ontologies. The following are models of some currently deployed approaches, reflecting this trend.

The KAOS meta-model (Dardenne *et al*., 1991) is made up of meta-concepts, meta-relations linking meta-concepts, meta-attributes characterizing meta-concepts and meta-relations, and meta-constraints upon several kinds of components. The meta-model uses the "meta" prefix referring to the meta-level, where domain-independent abstractions are defined. This level is different to other levels involved: the domain level, where concepts specific to the application domain are defined, and the instance level, where particular instances of domain-specific concepts are introduced.

The requirements meta-model (see Figure 1) is based on instances of meta-concepts, linked to instances of the meta-relations, and featured by instances of meta-attributes. This model emphasizes both conceptual and structural aspects. Natural language is not considered in this meta-model as the source of the descriptions for the scenarios and controlled languages are not intended to represent the formal elements of the meta-model. Also, the usage of some sort of semantic network for describing the meta-model gives the orientation toward structural descriptions, excluding dynamic elements. Finally, from this meta-model we cannot deduce the process we must follow for transforming the scenarios into specifications.

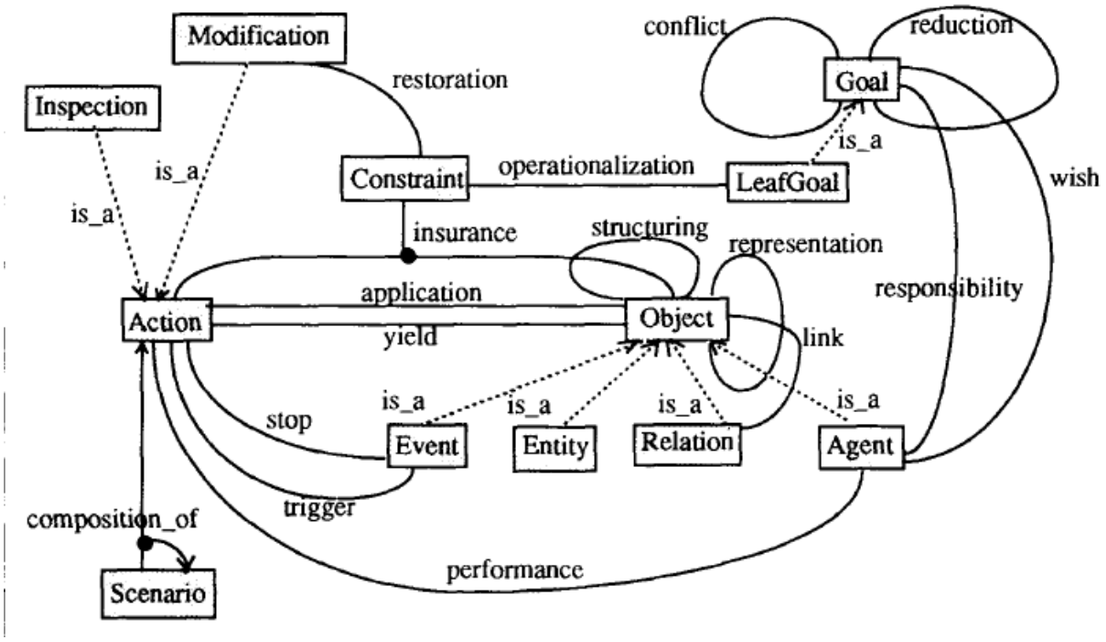


Fig. 1. The KAOS meta-model (Dardenne *et al.*, 1991)

Hu *et al*. (2010) propose a 'requirements meta-model' from the perspectives of *Role, Goal, Process*, and *Service* (RGPS). The meta-model is focused on modeling requirements from early requirements elicitation processes and, then, the process can be used to dynamically capture the stakeholder needs. The Relationships of the RGPS are defined in four layers (see Figure 2). The *Role Layer* contains actors (human or software), who are involved in the requirements engineering process, and information about what roles they play, respectively. The *Goal layer* depicts strategic goals (*i.e.,* Functional or Non-Functional goals) of service-oriented requirements. The *Process layer* is a multiple model, where processes can be composed by other processes. Finally, the *Service layer* represents the services composed when they are selected from the service repository. The RPGS model refers to the roles involved in requirements elicitation processes. Besides, it represents the functional elements implicated in the elicitation process.

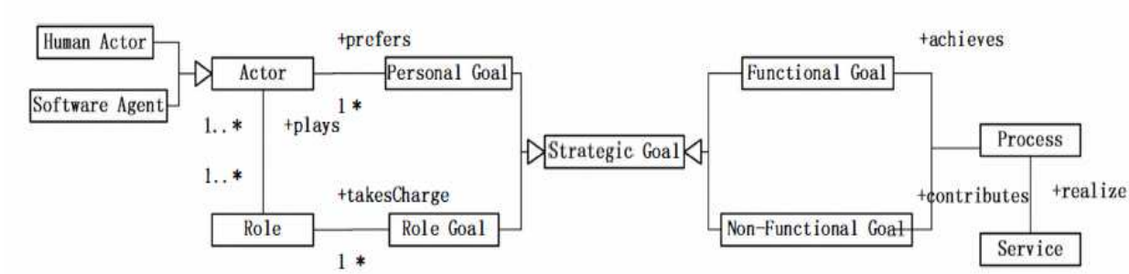


Fig. 2. The RPGS meta-model (Hu *et al.*, 2010)

Hu *et al.* (2010) use a class diagram for representing the meta-model, but avoiding the behavioral component of this kind of diagram (mainly directed by the class operations.) The

diagram is structural in nature, without elements leading to transformation processes. Also, the contents of natural language discourses or controlled languages specifications are not established in the meta-model. Consequently, the transformation process from natural language to controlled language cannot be identified.

Wei *et al.* (2008) present an approach nearest to the knowledge representation of requirements expressed in natural language. Also, they show how to define natural language patterns like the abstraction of the requirement statement. In this model, *patterns* (a part of the sentence) could be composed for describing several requirement statements, *i.e.*, a complete sentence of various requirements. The proposal can be seen in Figure 3, where the *behavior* represents a basic functional requirement, which is a kind of simple sentence. *Constraint* and *status* are other kinds of simple sentences. A *constraint* represents non-functional requirements (like quantitative and qualitative constrains), and the *status* represents the context requirements and services belonging to the stakeholders. An *event* is a behavioral trigger, which activates changes in either the entity state or the system state. A *condition* is a test of the current state of an entity or a test of the current system state.

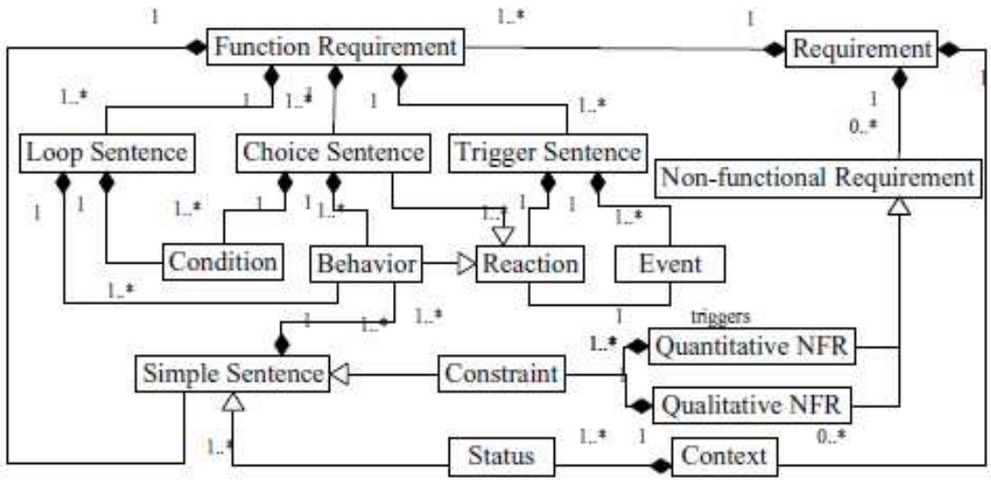


Fig. 3. Requirements statement model (Wei *et al.*, 2008)

As in the case of Hu *et al.* (2010), Wei *et al.* (2008) select a class diagram for representing their model, avoiding the behavioral component of the process. This model has a clear distinction between natural language discourses and controlled language requirements, but the transformation process is not clearly identified. However, some of the classes they use will be suitable for describing the entire requirements elicitation process, from natural to controlled language.

Rubin (2009) proposes a meta-model for representing the knowledge about the requirements engineering domain. This model comprises essential concepts belonging to the software development process. In this meta-model, Rubin proposes ontological constructs, that is, ontological components which can be combined to build up a model in a specific domain. The meta-model incorporates the domain-related constructs, their relationships, their link attributes, and their states (see Figure 4). This meta-model is used to bridge the data view (mainly composed by attributes) and the other elements focused on services.

Rubin (2009) expresses the meta-model by using a sort of entity-relationship diagram, mainly focused on structural elements. For this reason, the dynamic components are absent of the representation. Also, the distinction between natural language discourses and controlled languages specifications is absent.

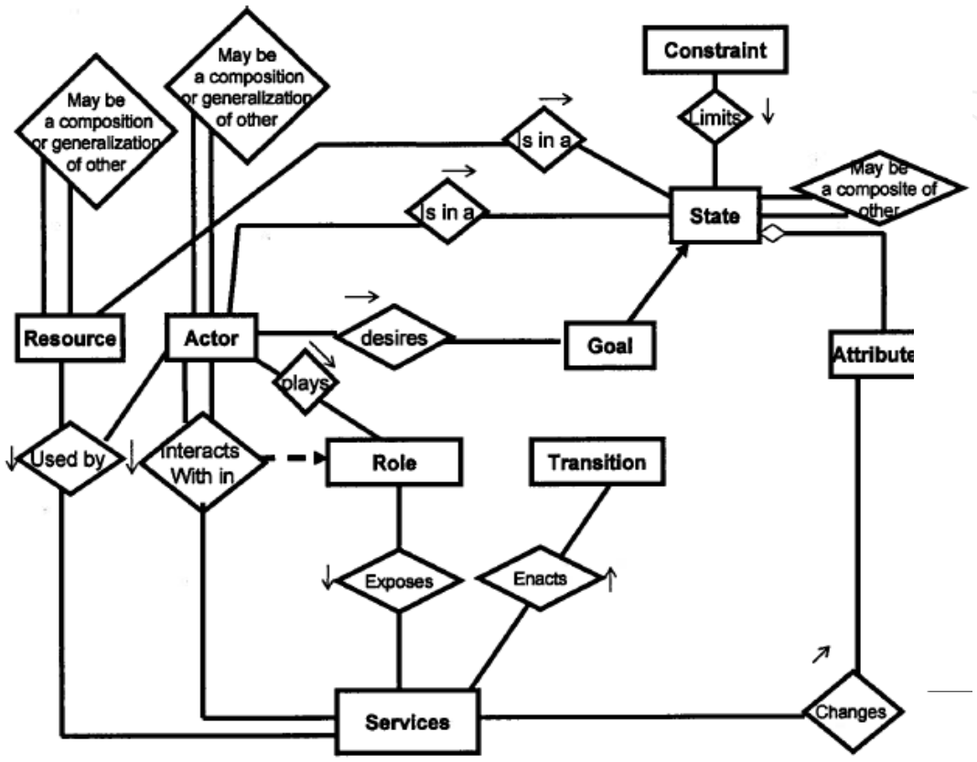


Fig. 4. Meta-model of requirements engineering domain knowledge (Rubin, 2009)

On the other hand, some other approaches have been introduced (*e.g.*, ontologies and meta-ontologies.) Zapata *et al.* (2010) propose a meta-ontology for the requirements elicitation process. This meta-ontology is incremental and independent of the problem domain. Zapata *et al.* (2010) use the meta-ontology to obtain stakeholder concepts from a dialogue expressed in natural language. The concepts identified in the meta-ontology are the following: *object*, *actor*, *constrains*, *actor-function*, *activity,* and *organization*. The meta-ontology is constructed in Protégé, an ontology editor. Due to the similarities between the hierarchical ontologies and the class diagram, we can say that the concepts of this meta-ontology are represented into a structural class diagram. In Figure 5 we show an image of the proposed meta-ontology. One of the main advantages in the usage of Protégé for representing the meta-ontology is the possibility of creating instances of it. So, we can define questions to be answered by the ontology and the entire process can be clarified by using the instances and the programmed questions.

An ontology to define semantics elements inside requirements descriptions is presented by Kaiya and Saeki (2006). The thesaurus of this proposal has concepts and relationships among the concepts. Also, it has several subclasses of concept classes and relationships (see Figure 6). In Figure 6, an "object" is a sub-class of a concept-class and an "apply" relationship can connect two concepts.

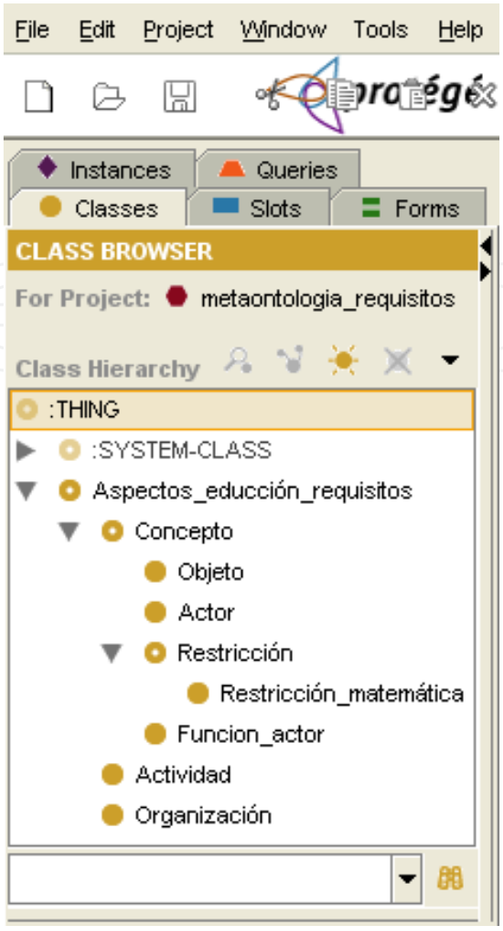


Fig. 5. Meta-ontology for requirements elicitation (Zapata *et al.*, 2010)

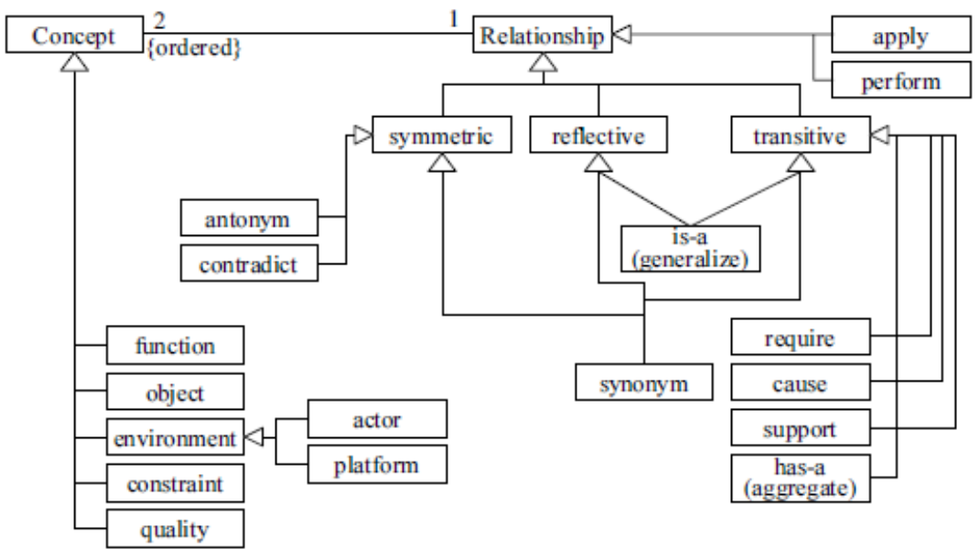


Fig. 6. Meta-model of Ontology for requirements (Kaiya & Saeki, 2006)

Both Zapata *et al.* (2010) and Kaiya and Saeki (2006) describe the meta-ontology in terms of class diagrams without the dynamic elements. Hence, the process for transforming the discourse into specifications is not clearly stated. Also, neither the natural language elements nor the controlled language elements are explicitly identified, leading to subjective interpretation of the meta-ontology concepts. Despite the mentioned drawbacks, the meta-ontology and the ontology can be used for defining some of the important concepts surrounding the requirements elicitation process and the transformation process from natural language discourses into controlled language specifications.

A meta-model for the requirements elicitation process is proposed by Dzung and Ohnishi (2009). This model is based on functional requirements (FR). Each FR is modeled as a node of the ontology, including one verb and several nouns. Inheritance and aggregation relationships can be represented as functional structures belonging to systems of certain domains. Functional requirements can include other relationships, for example: agent of the function (who), location of the function (where), time (when), reason (why), and non-functional requirements (NFR). In Figure 7, one of the diagrams used by Dzung and Ohnishi (2009) for representing the ontology is depicted. As in the case of the meta-ontology of Zapata *et al.* (2010), the main advantage of this approach is the possibility of instantiating the ontology. However, Dzung and Ohnishi (2009) require some other diagrams for representing the features of the ontology.

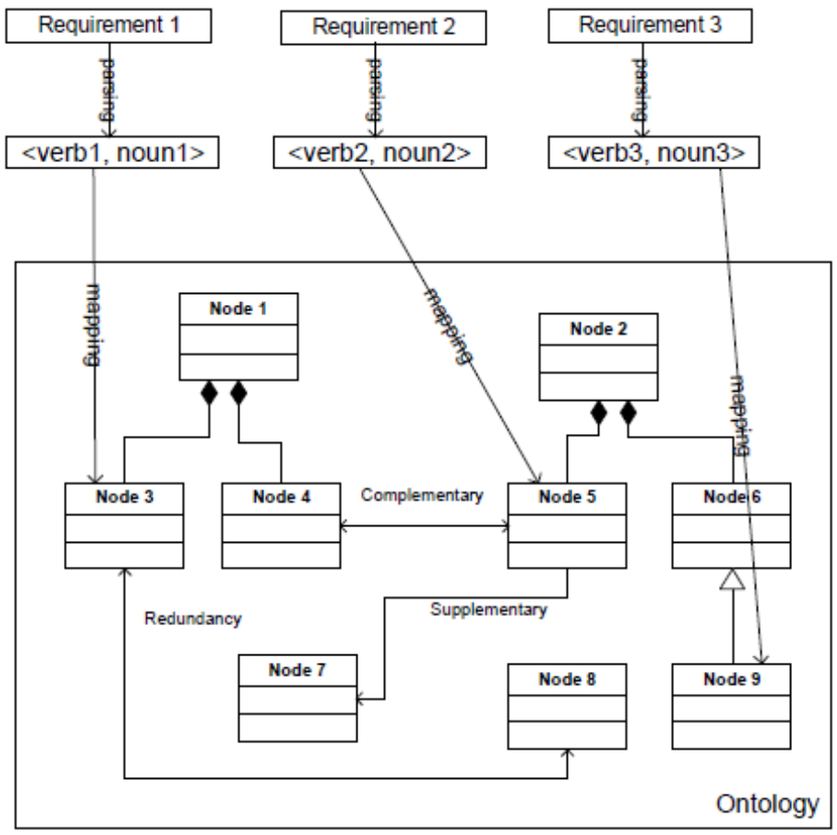


Fig. 7. Ontological meta-model for the requirements elicitation process (Dzung & Ohnishi, 2009).

Finally, another relevant approach is LEL (Language Extended Lexicon; Demitrio, 2005). This proposal focuses on identifying lexical elements from natural language. The lexicon comprises symbols representing: (i) active objects or subjects (perform actions;) (2) passive objects or subjects (actions are performed on them;) (iii) verbs; and (iv) meaningful states of the system. Demitrio (2005) proposes a framework for the automatic generation of LEL, based on information from specific documentation. Based on this lexicon, the analyst could generate the necessary scenarios to represent the knowledge in a specific context (see Figure 8). In this Figure, we can state the scenarios as the center of the model, generating many connections to other elements of the model like context, title, objectives, actors, resources, episodes, and exceptions. The diagram for representing the model is structural in nature, with constraints defined on some of the concepts used.

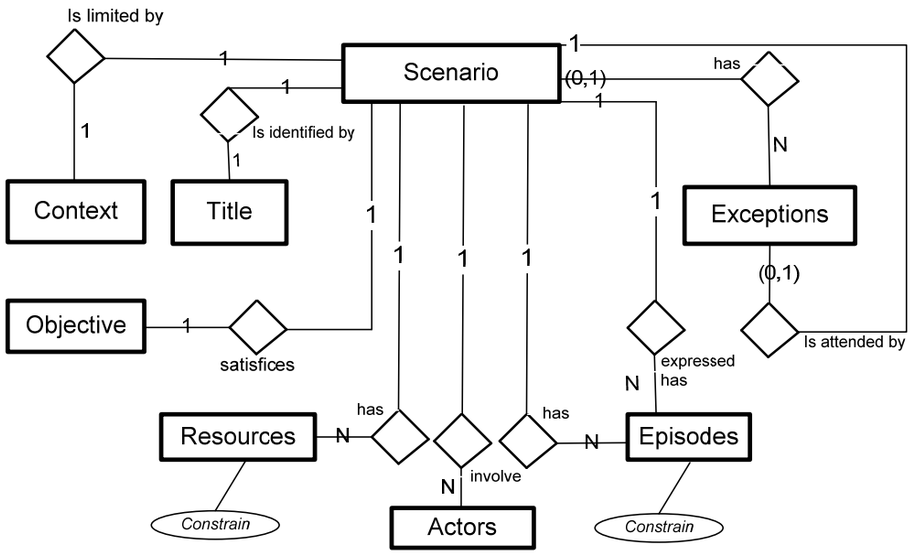


Fig. 8.Model to generate LEL from requirements (Demitrio, 2005)

Both the ontology suggested by Dzung and Ohnishi (2009) and the lexicon presented by Demitrio (2005) are related to the usage of natural language as the starting point for the requirements elicitation process. Dzung and Ohnishi (2009) selected an ontological approach for representing the process, defining the mapping process itself. However, the entire process comprises several diagrams and the whole structure and dynamics cannot be represented into one single diagram. Demitrio (2005) uses an entity-relationship diagram (mostly structural), and the elements of either natural or controlled languages are not explicitly identified. Consequently, the transformation process is also absent from this representation.

4. Knowledge representation proposal

In the state-of-the-art review, we mentioned a number of cases to represent knowledge related to the requirements elicitation process. Just a few of them are focused on the transformation process from natural language into controlled language. Also, they lack some explicit definition of the elements belonging either to natural or controlled language. Finally, the relationship between structural and dynamic elements is also absent from several knowledge representations, because they are structural in essence.

As we will describe in this Section, we propose a knowledge representation for this *transformation* from natural language discourses into controlled language specifications by using a dynamic and structural viewpoint in the requirements elicitation process. This knowledge representation approach is based on the pre-conceptual schemas (PS). A PS is an intermediate stage between natural language and conceptual schemas. The following are the main elements we can identify in the PS syntax:

- PS concepts (see Figure 9) are restricted to only two elements of the stakeholder discourse: nominal phrases noun-type (*e.g.*, grammatical element, domain-model, and technical-document), and single nouns (*e.g.*, analyst, user, requirement, and so forth.)

Fig. 9. Examples of PS Concepts (author elaboration).

- PS relationships (see Figure 10) are restricted to verbs from the stakeholder discourse. There are three kinds of PS relationships: structural relationships (single solid line), which refer to structural verbs or permanent relationships between concepts ("to be", "to have", and "to uniquely have" ,) dynamic relationships (with dotted line) which refer to dynamic verbs or temporal relationships between concepts (verbs such as "to review", "to generate", "to build", and so forth,) and achievement relationships (with double solid line) which refer to goal verbs (for example "to maximize", "to guarantee", and so on).

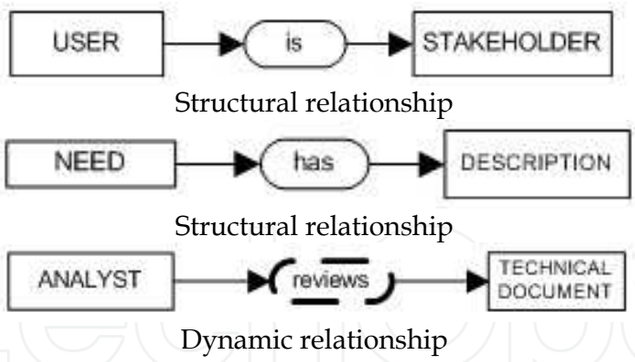


Fig. 10. Examples of PS relationships (author elaboration).

- PS conditionals represent preconditions—expressed in terms of concepts—that trigger some dynamic relationship. PS conditionals are formed by sets of concepts with comparative operators, such as "grade mark is greater than three."
- PS references (see Figure 11) are links between the same elements posted in different places at different distances from the PS.
- PS implications (see Figure 12) represent cause-and-effect relationships between dynamic relationships. Two uses of PS implications are: linking a dynamic relationship to another one and linking a PS conditional to a dynamic relationship.

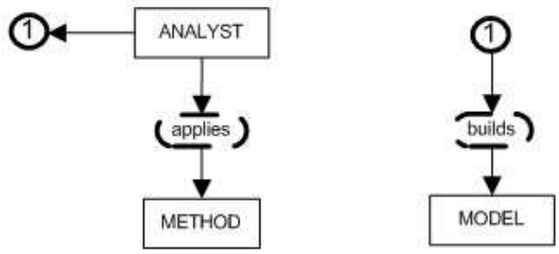


Fig. 11. Examples of PS references (author elaboration).

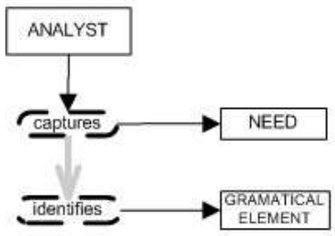


Fig. 12. Example of PS implication (author elaboration).

- Frames (represented by thick-lined rectangles with rounded edges as shown in Figure 13) are useful for gathering portions of the PS.

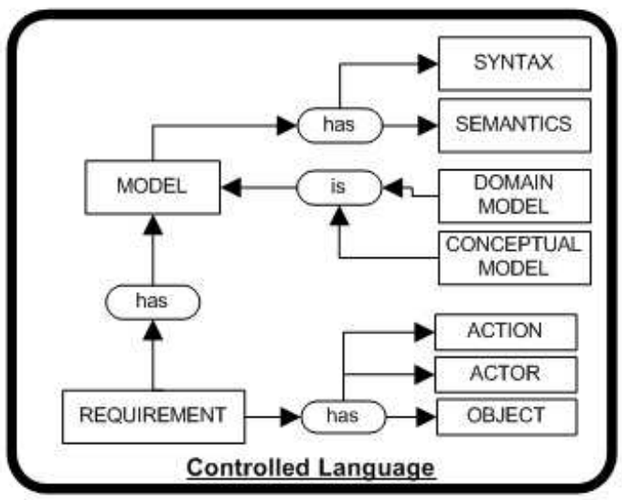


Fig. 13. Example of PS frame (author elaboration).

As mentioned before, we use a PS to represent several elements involved in the requirements elicitation process made by analysts. Furthermore, we represent the transformation process from natural language to controlled language. The proposed PS includes the relationship between structural and dynamic elements of the process. Analysts can use this PS to understand the transformation of stakeholder needs and expectations (from a discourse expressed in natural language) into requirements specifications and models (expressed in a controlled language.) We will explain the entire pre-conceptual schema as follows (see Figure 14).

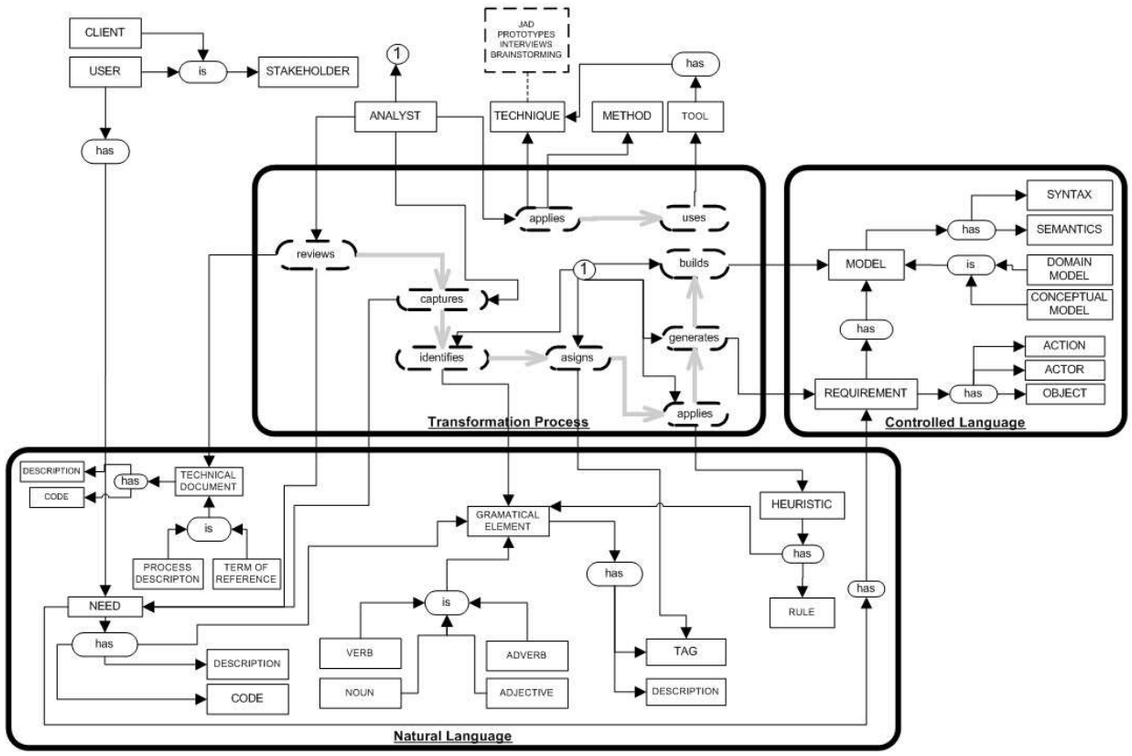


Fig. 14. Pre-conceptual Schema (author elaboration).

Analysts and stakeholders are involved in Requirements Elicitation. A STAKEHOLDER can be the CLIENT or the USER. A USER has a NEED. The ANALYST captures that NEED, based on a review of the TECHNICAL DOCUMENTATION, PROCESS DESCRIPTION, and TERM OF REFERENCE. The NEED has a *code*, a *description,* and a *grammatical element*. After capturing the NEEDS, the ANALYST identifies the GRAMATICAL ELEMENTS from NEED description. A GRAMATICAL ELEMENT can be: *Verb, Noun, Adverb*, or *Adjective*. It also has a TAG and a DESCRIPTION. Is the ANALYST who assigns a TAG to each GRAMATICAL ELEMENT identified, and applies a particular HEURISTIC as the tag. Immediately after, she/he generates a REQUIREMENT. The REQUIREMENT has an *action*, an *actor*, and an *object* associated. Finally, the ANALYST builds a MODEL, which has a particular *syntax* and *semantics*.

This way the NEED identified first by the ANALYST has a set of REQUIREMENTS, which will be drawn in a MODEL. The model can be a CONCEPTUAL MODEL or a DOMAIN MODEL. Through the whole process, the ANALYST applies a METHOD. This METHOD has a TECHNIQUE or set of techniques, supported by the use of one TOOL or more.

We include three frames in the PS: natural language, controlled language, and transformation process. By doing this, we are explicitly defining the elements of the requirements elicitation process regarding either natural or controlled language. With the third frame, we are signaling the activities belonging to the transformation process from natural language discourse into controlled language specifications defined by models and requirements. The natural language frame comprises the concepts belonging to the stakeholder discourse, headed by the description of the needs. Also, the grammatical

elements are included, and the tags related to the type of element we are using. The controlled language frame has the concepts related to the translated information originated from the description of the needs, mainly the requirements and the models. In the current state of the art, this information must be generated by the analyst, but in the proposed environment, some tools could help in elaborating the models and the requirements. Particularly, the environment surrounding the PS is currently searching for automated generation of diagrams and source code. Finally, the transformation process frame includes all the dynamic relationships we can identify in the process, like capturing needs, identifying grammatical elements, assigning tags, applying heuristics, generating requirements, and building models. Some PS elements are not grouped by frames, because they are related either to the agents who generates the activities of the process or to some properties we need to take for granted in the process, like the methods and tools.

With the proposed PS, we address the aforementioned drawbacks of the previous work in the following way:

- With frames we describe the elements represented in the natural language discourse and the results of the transformation process in terms of controlled language requirements and models.
- A third frame is used for describing the entire transformation process from natural to controlled language. The different dynamic relationships involved in this framework are the steps we need to follow for completing the transformation process.
- The distinction between structural and dynamic elements is accomplished by the nature of the pre-conceptual schema. The concepts are structural and the dynamic relationships are behavioral. They are linked by the intervention of the agents in the process, mainly the analyst and the stakeholder.

In order to illustrate how the process of transforming NL into controlled language occurs, we show—by means of an example—both the values assigned to each element and the fulfilled conditions to perform an action. This illustration shows the attribute values that bind with each concept and is based on executable pre-conceptual schemas (Zapata *et al.*, 2011). An executable PS adds elements to the basic notation; essentially differentiating concepts that contains attributes (class concepts) and atomic concepts (attribute concepts).

In the natural language frame, a User, which is a stakeholder, has a need. The scenario is characterized by steps, as follows. In the first step an *analyst* reviews a type of *technical document*. In this case the *technical document* is a *Term of Reference*, which has a *code* and a *description*. In our proposed example, the value of description is *Commonly, the manager is in charge of authorizing the payments represented by the payroll* and the code is labeled as *001*. The first step is represented in Figure 15.

This process is currently made by hand. The *need* has a *description*, a *code*, and *grammatical elements*. The need *description* is *Only the manager authorizes the payroll*, and the code is set as *001*, as shown in Figure 16.

In the step 3, the *analyst* identifies the *grammatical elements* from the *need*, which can be a *verb*, *adverb*, *noun*, or *adjective*. Each *grammatical element* has a *tag* and a *description*. The *analyst* assigns a *tag* to each *grammatical element*, after identifies it (implication). For the need coded

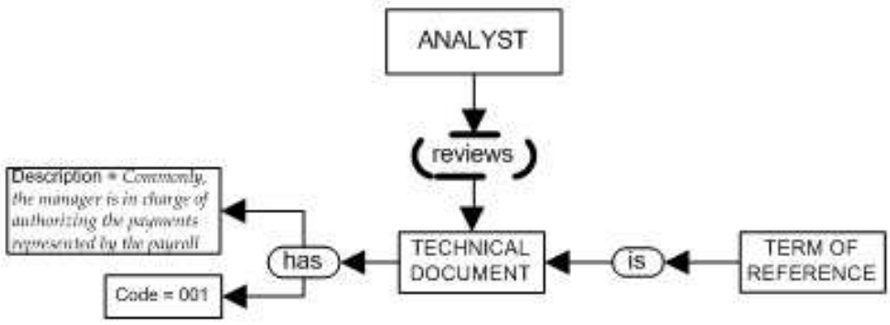


Fig. 15. PS including values of the step 1 (author elaboration).

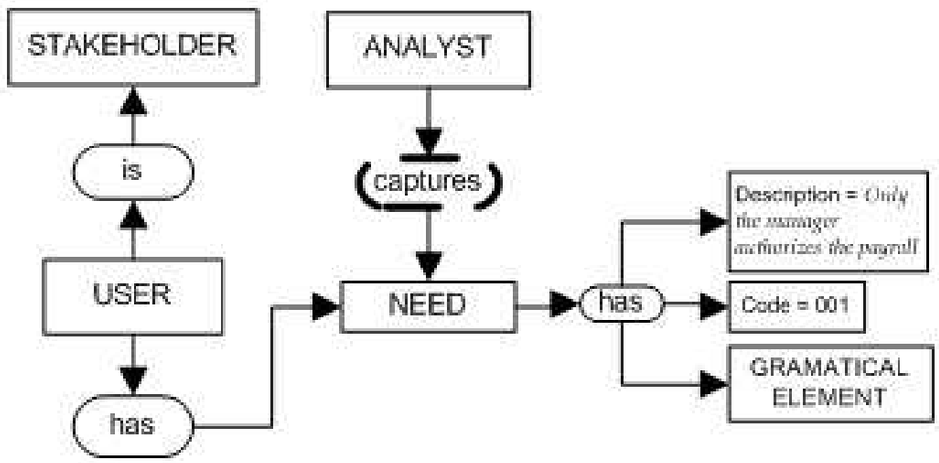


Fig. 16. PS including values of the step 2 (author elaboration).

001, he creates a list of the following values (Table 1). The PS (Figure 17) is supported by a table, such that for each concept-class containing another concept-class, analyst writes the attribute-concepts in a list, which describes the name as a head.

Type Grammatical Element	Tag	Description
Adverb	ADV	*Only*
Noun	N	*The manager*
Verb	VB	*Authorizes*
Noun	N	*The payroll*

Table 1. List attribute concept, Step 3 (author elaboration)

Finally, in the step 4, the analyst applies a heuristic, which in turn has rules and grammatical elements, to generate a requirement. The value of the rule to be applied is *transforming the noun before the verb into actor, the verb into action, and the noun after the verb in object*. Requirement has an action, an object, and an actor, as we show in Figure 18. The exemplification of the class-concept model and its attribute-concepts is beyond the scope of this chapter.

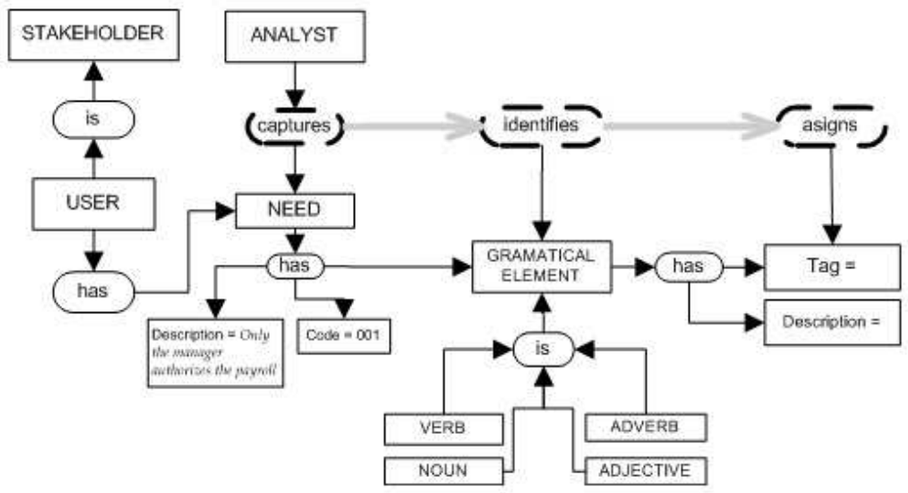


Fig. 17. PS including values of the step 3 (author elaboration).

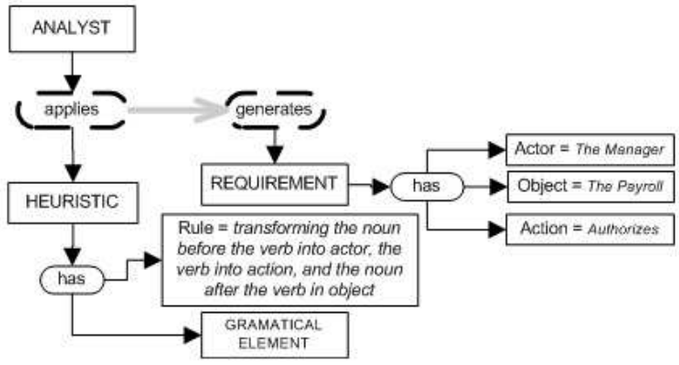


Fig. 18. PS including values of the step 4 (author elaboration).

For the sake of simplicity, we assume the entry of information in multiple attribute-concepts. The executable PS might be much more complex, e.g., using predefined descriptions for each of the values.

5. Conclusions

The requirements engineering research community has proposed several models to understand the requirements elicitation process. These models try to represent knowledge about this conceptual domain, by using ontologies, meta-ontologies, and meta-models. Most of such models fail to describe some issues: the elements belonging either to natural or controlled language, the transformation of natural language into controlled language in the requirements elicitation process made by the analysts, and the relationship between structural and dynamic elements in the process.

In this Chapter, we proposed a model for knowledge representation of the transformation process from a natural language discourse into controlled language specifications (within the context of the requirements elicitation process) by using pre-conceptual schemas. The model helps analysts to systematically understand the requirements elicitation process by

taking into account both the dynamic and structural aspects of requirements. In the model, we defined the behavior and functionality of the aforementioned transformation process.

In the proposed knowledge representation model, the following two kinds of structures are shown: functional structure and logical structure of the transformation process. The requirements elicitation process can be performed, based on the structural features in the model.

6. Acknowledgment

This work is founded by the Vicerrectoría de Investigación from both the Universidad de Medellín and the Universidad Nacional de Colombia, under the project: "Método de transformación de lenguaje natural a lenguaje controlado para la obtención de requisitos, a partir de documentación técnica."

7. References

Andrade, J., Ares, J., García, R., Pazos, J., Rodríguez, S., & Silva, A. (2003). A methodological framework for generic conceptualization: problem-sensitivity in software engineering. *Information and Software Technology*, Vol. 46, No. 10, (August 2004), pp. (635–649), 0950-5849

Berry, D. M. (2003) Natural language and requirements engineering - Nu?, In: *CSD & SE Program University of Waterloo*, 20-05-2011, Available from: http://www.ifi.unizh.ch/groups/req/IWRE/papers&presentations/Berry.pdf

Assmann, U., Zschaler, S., & Wagner, G. (2006). *Ontologies, Metamodels, and the Model-Driven Paradigm* (first edition), Springer, 978-3-540-34517-6, New York

Chomsky, N. (1965). Aspects of the Theory of Syntax. MIT Press, Cambridge, MA

Dardenne, A., Fickas, S., & van Lamsweerde, A. (1991). Goal-directed Concept Acquisition in Requirements Elicitation. *Proceedings of the 6th international workshop on Software specification and design,* 0-8186-2320-9, Italia, October 1991

Demitrio, D. (2005) *Framework para Elicitación Automática de Conocimientos*, Tesis Magíster en Ingeniería de software, Facultad de Informática Universidad Nacional de La Plata, Retrieved from http://postgrado.info.unlp.edu.ar/Carrera/Magister/Ingenieria%20de%20Software/Tesis/Daniel%20Demitrio.pdf

Dzung, D. & Ohnishi, A. (2009). Ontology-based Reasoning in Requirements Elicitation. *Seventh IEEE International Conference on Software Engineering and Formal Methods*, 978-0-7695-3870-9, Hanoi, November 2009

Falbo, R.A., Arantes, D.O., & Natali, A.C.C. (2004) Integrating knowledge management and groupware in a software development environment. *Proceedings of the International Conference on Practical Aspects of Knowledge Management*, 978-3-540-24088-4, Vienna, Austria, December 2004

Hu, B., He, K., Liang, P., & Li, R. (2010) REMO: A RGPS-based Requirements Modeling Process for Service Oriented Architecture. *Proceedings of Sixth International Conference on Networked Computing and Advanced Information Management (NCM),* 978-1-4244-7671-8, Seoul, September 2010

Jiang T. & Li, M. (1998). Formal Grammars and Languages. In Handbook on Algorithms and Theory of Computation. Editor: Mikhail J. Atallah, November 1998

Kaiya, H. & Saeki, M. (2006). Using Domain Ontology as Domain Knowledge for Requirements Elicitation. *Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06)*, 0-7695-2555-5, Minessota USA, September 2006.

Leite, J.C. (1987). *A survey on requirements analysis*. Department of Information and Computer Science, Advanced Software Engineering Project Technical Report RTP071, California, Irvine

Li, K., Dewar, R.G., & Pooley, R.J. (2003) *Requirements capture in natural language problem statements*, Heriot-Watt University. Retrieved from http://www.macs.hw.ac.uk:8080/techreps/docs/files/HW-MACS-TR-0023.pdf

Mitamura, T. & Nyberg, E. (1995). Controlled English for Knowledge Based Machine Translation: Experience with the KANT System, *Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Belgium, July 1995

Robertson, S., & Robertson, J. (2006). *Mastering the Requirements Process* (second edition), Addison Wesley, New Jersey

Ruohonen, K. (2009). Formal Languages. In lecture notes for the TUT course "Formaalit kielet". Available on http://math.tut.fi/~ruohonen/FL.pdf

Rubin, E. (2009) *Domain Knowledge Representation in Information Systems*, Thesis of Doctor of Philosophy, University of British Columbia, Retrieved from https://circle.ubc.ca/handle/2429/15229

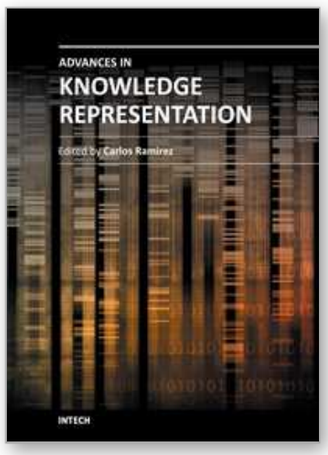
Seidewitz, Ed. (2003) What models mean. *IEEE Software*, Vol. 20, No. 5, (September 2003), pp. 26-32, 0740-7459

Wei, L., Ke-Qing, H., Kui, Z., & Jian, W. (2008) Combining Domain-Driven Approach with Requirement Assets for Networked Software Requirements Elicitation. *Proceedings of IEEE International Conference on Semantic Computing*, 978-0-7695-3279-0, California, August 2008

Zapata, C.M., Gelbukh, A., & Arango, F. (2006). Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for Requirements Elicitation, *Lecture Notes in Computer Science*, Vol. 4293, pp. 17-27, 0302-9743

Zapata, C.M., Giraldo, G.L., & Londoño, S. (2011). Executable pre-conceptual schemas, *Revista Avances en Sistemas e Informática*, Vol.7 No.3, pp. 15-23, Medellín, December 2010

Zapata, C.M., Giraldo, G.L., & Mesa, J.E. (2010). A Proposal of Meta-Ontology for Requirements Elicitation, *Ingeniare. Revista chilena de ingeniería*, Vol. 18, No. 1, August 2010, pp. 26-37, 0718-3305



Advances in Knowledge Representation

Edited by Dr. Carlos Ramirez

ISBN 978-953-51-0597-8
Hard cover, 272 pages
Publisher InTech
Published online 09, May, 2012
Published in print edition May, 2012

Advances in Knowledge Representation offers a compilation of state of the art research works on topics such as concept theory, positive relational algebra and k-relations, structured, visual and ontological models of knowledge representation, as well as detailed descriptions of applications to various domains, such as semantic representation and extraction, intelligent information retrieval, program proof checking, complex planning, and data preparation for knowledge modelling, and a extensive bibliography. It is a valuable contribution to the advancement of the field. The expected readers are advanced students and researchers on the knowledge representation field and related areas; it may also help to computer oriented practitioners of diverse fields looking for ideas on how to develop a knowledge-based application.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen