Capítulo 3
Formalization of domain problems described in the cause-and-effect diagram in the context of the software development process

Fabio Alberto Vargas Agudelo - Fvargas@tdea.edu.co
Tecnológico de Antioquia
Carlos Mario Zapata Jaramillo - Cmzapata@unal.edu.co
Universidad Nacional de Colombia, Medellín

I. INTRODUCTION

The system analyst is responsible—in the early requirements elicitation process—for identifying the problems to be solved with a software application. Such identification is based on: (i) the analyst experience and knowledge about the domain area; and (ii) the stakeholder support. Problems are collected by the system analyst during the domain analysis for using them as the main input of the requirements specification. Such specification is consistent with the problems identified and the stakeholder needs.

The system analyst detects and describes problems from the domain discourse recognition and eventually presents them in a formal way by using a representation diagram. Some methods of the software development process, goal-oriented software engineering (GORE) [1], and organizational analysis have diagrams for representing problems. For instance, UNC-Method [2] includes the cause-and-effect diagram for specifying problems and relating them to the system and the stakeholder goals; logical framework [3] has objective trees and problem trees for linking goals and problems during the project formulation and the decision-making process; Business Modeling with UML [4] has a goal schema for graphically linking goals and problems; NFR Framework [5] has problem frameworks for specifying problems in the non-functional requirements elicitation process.System analysts specify the problems in diagrams as textual descriptions, based on their experience and knowledge. However, no formal process is driven for guaranteeing the problems described are actual problems, and such problems are clearly understood by the stakeholders. The aforementioned facts lead to some gaps still remaining in the early software requirements elicitation process: relation among early and late requirements is poor [6]; requirements and business goal are commonly unrelated to each other [7]; information systems use to misrepresent the requirements captured from the business model [8]; functionality expected from business processes is unrelated to the software system functionality [7]; traceability among expectations, needs, and their representation in a goal diagram is poor [9]; business goals are misused for assuring the requirements specification completeness and sufficiency [10] [11]; an initial company model is insufficient for getting relevant information based on the context of the company [6]; problems are poorly described, making difficult to link and trace them [12]; problems identified during requirements elicitation are described in a positive way [12]; no formal methods are established for goal and problem definition [13]; problem definition is a hand-made process, since analysts commonly draw up goal and problem diagrams in a subjective way [14]; analyst experience and stakeholder knowledge are useful for determining goals, but no validation against problems is included in such a process [15].

In this chapter, we propose a method to formalize the problems of the cause-and-effect diagram drawn up by the analyst in the domain context where the software application will be used. We define a set of syntactic rules for specifying problems, and we allow analysts and stakeholdersto understand the problems better. Improved traceability and consistency related to the business goals, the system goals, and the requirements can be achieved by using the formalized problem schemas.

This chapteris organized as follows: in Section II, we present the theoretical framework with some definitions; in Section III, we present some methods for specifying problems in the requirements elicitation process and the organizational analysis; in Section VI, we propose a method to formalize the problems of the cause-and-effect diagram during the software development process; in Section V, we provide an example for applying our proposal into a lab study; finally, in Section VI, we discuss some conclusions and future work.

II. THE ORETICAL FRAME-WORK

Goal-Oriented Requirement Engineering (GORE) is an approach for promoting the use of goals as the basis of the software requirements elicitation. GORE includes a point of view related to the purpose of the system—intentional in nature. The introduction of an intentional point of view allows stakeholders to expresstheir needsmore naturally, focusing on what they want—their goals—*versus* the way to achieve them—conventional requirements. Requirements can be derived from goals [16]. Some approaches to GORE differ in two main factors: the focused requirements engineering activity—*e.g.*, requirements elicitation, modeling, and analysis—and the supported degree of formalism. For example, KAOS [17] is focused on formal requirements modeling. NFR Framework [5] is also focused on modeling but targeted on non-functional requirements with a less formal approach. I* is a methodology based on the NFR Framework [18] focused on the initial phases of the requirements elicitation—particularly, the business modeling. GBRAM [19] is intended to integrate scenarios into the context of goal modeling.

Cause-and-effect diagram is used by organizations during the requirements elicitation process in order to think about the actual causes of the potential problems and then to establish corrective actions. In this context, the cause-and-effect diagram can be used to guide the analysis and reflection about the problem understanding, the identification of root causes and possible solutions, and the decision- making process.

Problems are specific issues requiring a solution in a specific domain. Perales [20] defines a problem as any planned/spontaneous situation producing, on the one hand, a certain degree of uncertainty and, on the other, a behavior aimed at finding a solution.

Formalization is a set of rules, expressions, and meanings intended to characterize a language, allowing for a step-by-step interpretation of a particular process [21].

Pre-conceptual schema is a conceptual-graph-like knowledge representation for requirements elicitation (see the main elements in Fig. 1). Such a representation can be obtained from controlled natural language discourses, and it can be then converted into standard UML diagrams. Pre-conceptual schemas are intermediate models for obtaining UML diagrams from natural language discourses [22].

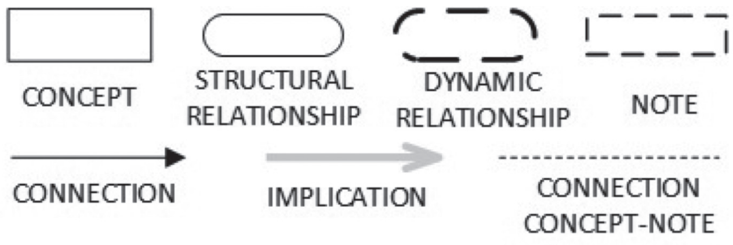


Fig. 1. Syntax of the pre-conceptual schemas. Source [22]

III. BACKGROUND

The UNC-Method [2] includes—in the software requirements elicitation process—the cause-and-effect diagram as a tool for identifying the domain problems and their relation to both organizational and system goals. Such identification is hand-made by the analyst and the stakeholder for specifying problems and creating a textual representation of the cause-and-effect diagram. Traceability and consistency among the organizational goals and the functional requirements of the software application are difficult to achieve at this level.

Business Modeling with UML [4] has a goal/problem diagram to relate the organizational goals to the problems identified in the domain. Problems are intended to be obstacles to the achievement of goals. Problem specification is manually written by using textual descriptions, and relationships of problems and the different actors involved in the process are difficult to achieve. Also, syntactic and semantic structures to formalize the problems are underspecified.

In the NFR Framework [5], problems are specified by using problem frameworks in the non-functional requirements elicitation process. Informal representation of the problem domain can lead to traceability and consistency problems when reviewed against the original goals.

Zapata, Acevedo, and Moreno [13] make a representation of the semantic relations among problems and goals by using predicate logic. The authors propose formal methods as a way to plan goals and domain problems.

Lin and Zhi [15] establish I* techniques for combining goal analysis and problem analysis. They argue goal decomposition requires domain knowledge and, consequently, problems can be used for generating requirements. However, no structures are used for specifying either goals or problems.

Vargas [12] defines grammar rules for specifying goals and problems, and he applies them to the cause-and-effect diagram. He also establishes an approximation to a consistency relation based on such rules, but he uses syntactic structures with explicit common words among goals and problems. This fact leads to some drawbacks since words used to express problems are different from those used to express related goals.

In terms of organizational analysis, Sanchez [3] argues the logical framework method includes an objective tree—future outcomes you want to accomplish—and a problem tree—what you want to solve. The problem tree is generated from the objective tree by using some rules. Basically, problems are expressed as negative states related to the objectives. When the central objective of the process is expressed in a negative form, you can have the central problem and so on. Even though the rules are defined, the whole process should be manually carried out by the organizational analyst.

IV. RESULTS

During the early software requirements elicitation, the specification of the problems to be solved by the software application is crucial. The analyst and the stakeholder are directly involved in this task, and they complete it based on their experience and organizational knowledge. In some methods for eliciting requirements, a diagram is often used for drawing the problems. The formal method we propose in this section for expressing the problems in the cause-and-effect diagram is based on a set of syntactic and semantic rules and pre-conceptual schemas for graphically representing context information. We use such rules and schemas due to their proximity to the natural language of the stakeholder and the technical language of the analyst. The steps of the formal method for expressing problems in the cause-and-effect diagram are the following:

Step 1. Recognizing the syntactic-semantic structure for specifying problems. In Fig. 2 we propose a preconceptual-schema-based structure for problems. An organizational analyst should be familiarized with this structure after the first step.

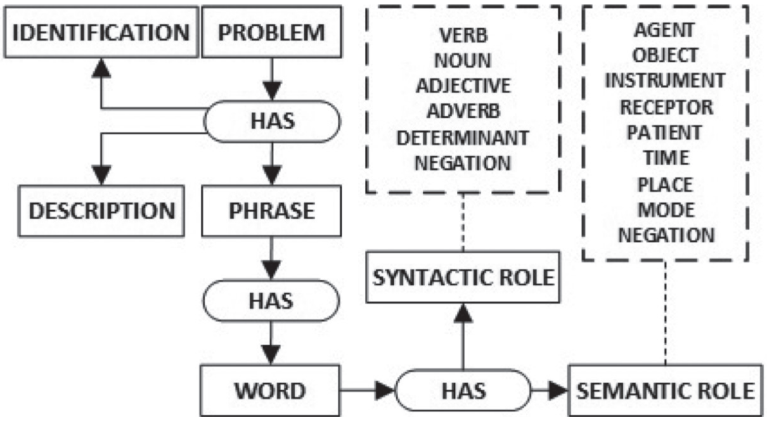


Fig. 2. Problem structure based on pre-conceptual schemas. Source: the authors

Step 2. Using a set of syntactic-semantic rules for formalizing problems. We define three rules to be formally used for specifying the problems described in the cause-and-effect diagram; we allow the analyst to specify problems in a clear and easy way to be understood by the stakeholders. By using the rules, relationships among problems, and organizational and system goals are guaranteed by traceability and consistency. In the rules, the syntactic structures of the problems are adapted from Vargas [12], while the graphical schemas are proposed in this paper. We introduce the red slash as a new symbol for denoting the negation of an element of the pre-conceptual schemas. Red slash is supposed to be put over the element we want to negate. The abbreviations used in the rules are: S=Sentence, V=Verb, Ad=Adjective, NP= Noun Phrase, Adv=Adverb, N=Noun.

Rule No. 1. Enunciating the problem by using a negative connotation adverb. The structures are shown in Tables 1 and 2.

Rule No. 2. Enunciating the problem by using a negative connotation adjective. The structures are shown in Table 3.

Rule No. 3. Enunciating the problem by using a negative connotation noun. The structures are shown in Table 4.

V. LABORATORY EXAMPLE

Zapata and Arango [2] present a cause-and-effect diagram related to the selling process of a company (See Fig. 3).

Table 1. Negative-connotation adverb type 1. Source: the authors

Description	Restrictions	Example

S=NP1+Adv1+Adv2+V+NP2	Adv1=Not; V = Action verb;
	Adv2 = positive connotation adverb
		The laboratory technician does not efficiently deliver samples

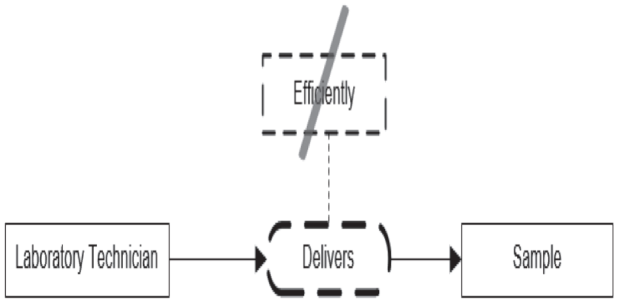
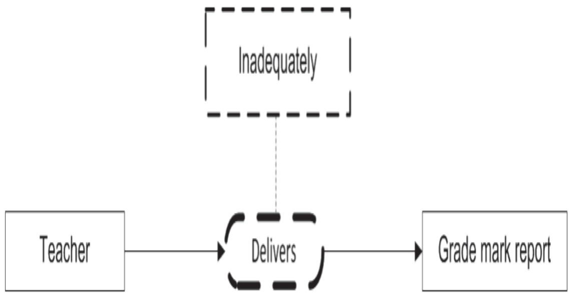
Graphical schema for formalizing problems		
		
S=NP1+Adv+V+NP2	Adv = negative connotation	The Teacher inadequately delivers the grade mark report
Graphical schema for formalizing problems		
		

Table 2. N egative-connotation adverb type 2. Source: the authors

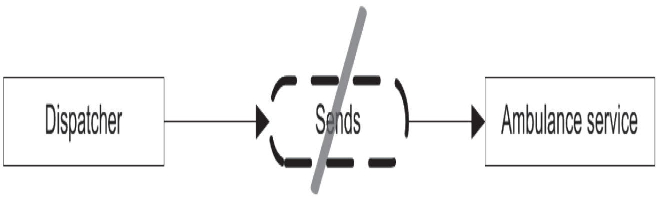
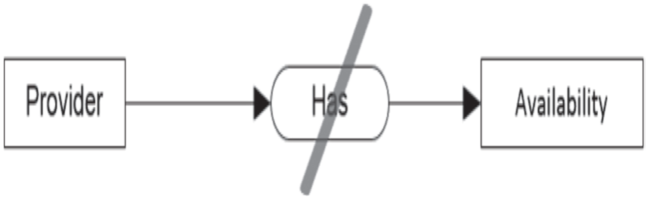
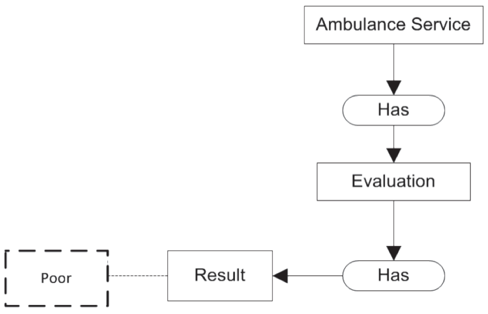
Description	Restrictions	Example
S=NP1+Ad-v+V+NP2	Adv1=Not; V = action verb	The dispatcher does not send the ambulance service
Graphical schema for formalizing problems		
		
S=NP1+Ad-v+V+NP2	Adv = not; V= verb to have	The provider does not have availability
Graphical schema for formalizing problems		
		

Table 3. Negative-connotation adjective type 1. Source: the authors

Description	Restrictions	Example
S=NP+V+Ad	Ad= negative connotation	The Ambulance service is poor
Graphical schema for formalizing problems		
 Ambulance Service Has Evaluation Has Result Poor		

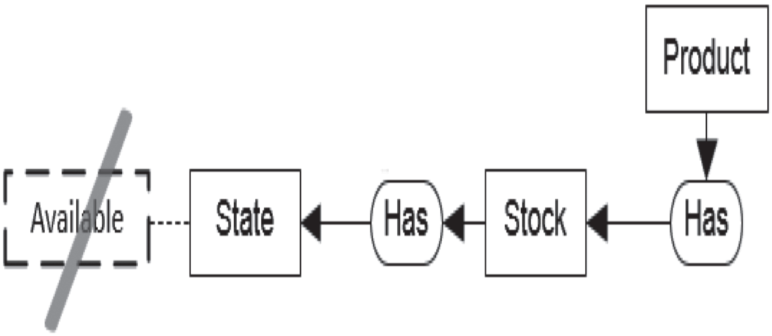
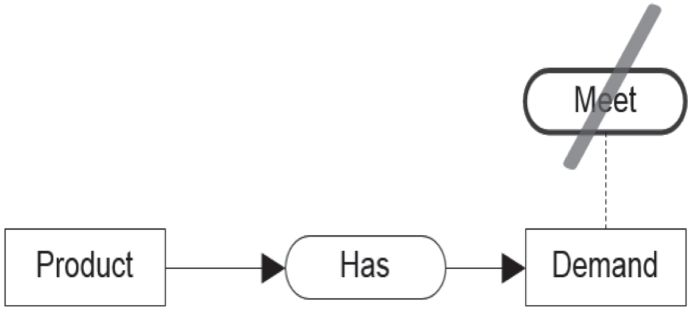
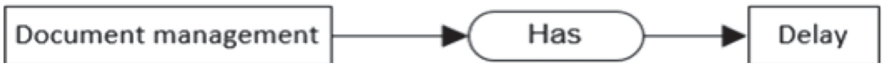
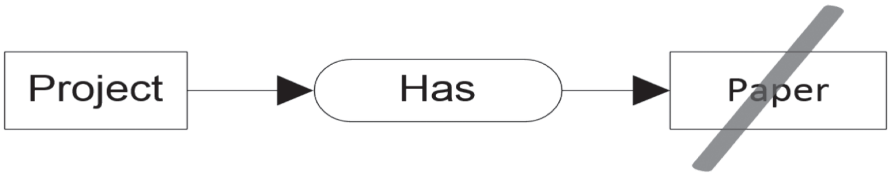
Description	Restrictions	Example
S=NP+V+Adv+Ad	Ad = positive connotation Adv = Not V= verb to be	The Stock product is not available
Graphical schema for formalizing problems		
		
S=NP+V1+Adv+V2	V1= verb to be Adv = not V2= achievement verb	The product demand is not met
Graphical schema for formalizing problems		
		

Table 4. Negative-connotation noun type 1. Source: the authors

Description	Restrictions	Example
S=NP1+V+NP2	NP2= negative connotation	Document management has delays
Graphical schema for formalizing problems		
		
S=NP1+V+NP2	NP2 = positive connotation; V= verb to have in negative form	The Project has no papers
Graphical schema for formalizing problems		
		

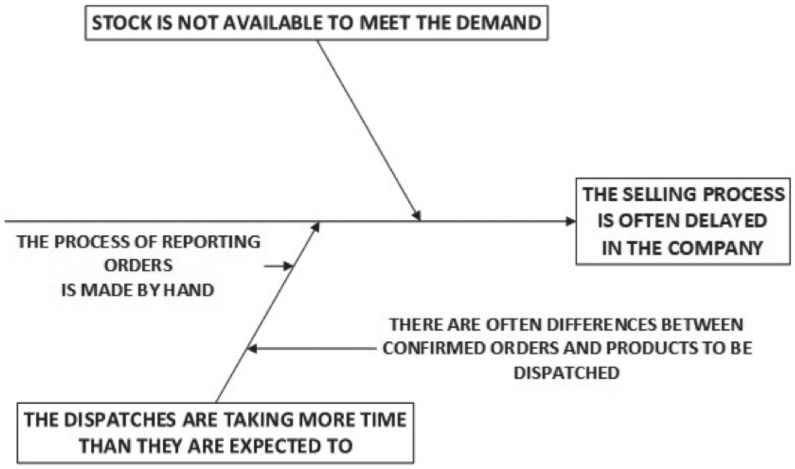

STOCK IS NOT AVAILABLE TO MEET THE DEMAND
THE SELLING PROCESS IS OFTEN DELAYED IN THE COMPANY
THE PROCESS OF REPORTING ORDERS IS MADE BY HAND
THERE ARE OFTEN DIFFERENCES BETWEEN CONFIRMED ORDERS AND PRODUCTS TO BE DISPATCHED
THE DISPATCHES ARE TAKING MORE TIME THAN THEY ARE EXPECTED TO

Fig. 3. Cause-and-effect diagram related to the selling process of a company. Source [2]

Problems described in the diagram (see Fig. 3.) are ambiguously generated since both their structure and specification are unclear. In Table 5, we summarize some of the detected ambiguities.

According to our proposed method, we need to re-write and formalize each problem by using the rules defined in this paper. In Table 6, we present the problems formalized. The modified cause-and-effect diagram is presented in Fig. 4.

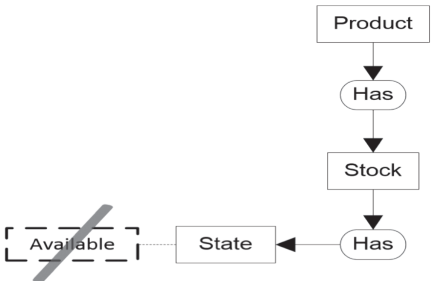
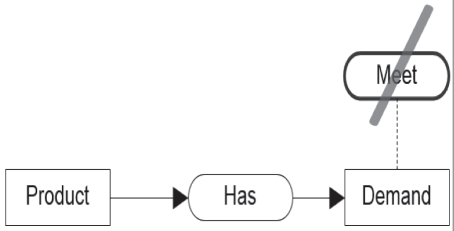
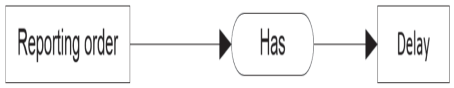
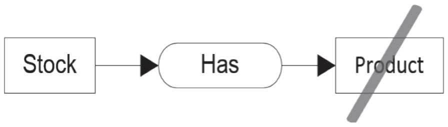
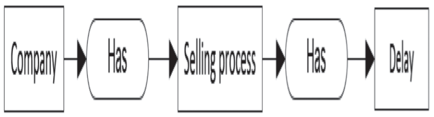
Table 5. Ambiguities detected in the cause-and-effect diagram. Source: the authors

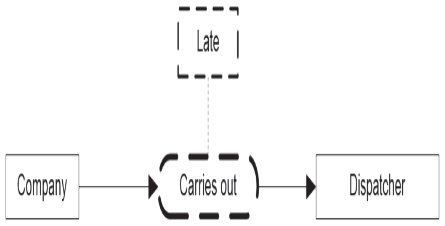
Problems	Ambiguity detected
Stock is not available to meet the demand	The problem specification does not refer to a single problem, but rather many problems can be deduced.
The selling process is often delayed in the company	The problem specifi cation is clear, but the syntactic structure of the problem can be improved
The dispatches are taking more time than they are ex-pected to	The speci fication is not conducive to the clear statement of the problem since the sentence does not contain a negative connotation
The process of reporting or-ders is made by hand	The specifi cation is not conducive to the clear statement of the problem, since the sentence does not contain a negative connotation.
There are often differences between confi rmed orders and products to be dispatched.	The specifi cation does not defi ne the problem clearly. Again, no negative connotation is detected.

VI. CONCLUSIONS AND FUTURE WORK

Many of the problems detected and further specified in software development processes are exposed in an unclear way, and in some cases, they avoid the expression of a problem by themselves. This fact implies misinterpretations made by analysts and stakeholders, causing reprocesses and irrelevant software requirements discovery to an organization.

Table 6. Problems formalized. Source: the authors

Problems	Problems formalized
Stock is not available to meet the demand	The Stock product is not available (Rule No 2)  The Product demand is not met (Rule nro2) 
The process of reporting orders is made by hand	The reporting order has delays (Rule No 3) 
There are often differences between confi rmed orders and products to be dispatched	The Stock has no products (Rule No 3) 
The selling process is often delayed in the company.	The company selling process has delays (Rule No 3) 

The dispatches are taking more time than they are expected to do.	The company carries out the dispatches late (Rule No 1) 

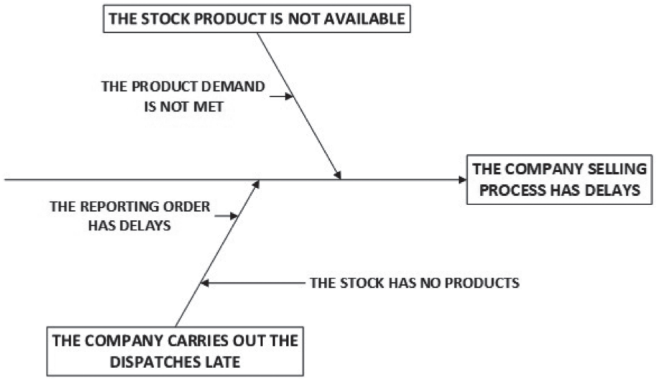

THE STOCK PRODUCT IS NOT AVAILABLE
THE PRODUCT DEMAND IS NOT MET
THE COMPANY SELLING PROCESS HAS DELAYS
THE REPORTING ORDER HAS DELAYS
THE STOCK HAS NO PRODUCTS
THE COMPANY CARRIES OUT THE DISPATCHES LATE

Fig. 4. Final cause-and-effect diagram. Source: the authors

The formalization of the problems allows for generating traceability and consistency during the initial software requirements elicitation, leading to a less ambiguous representation of problems, so thestakeholders and analysts can understand the problems more easily .

In this chapter we proposed structures for formalizing problems in order to support system analysts in the early software requirements elicitation process. We expect to generate a higher degree of reliability in the software solutions raised. In fact, problems should be aligned to the organizational context, and they should be relevant for understanding the stakeholder needs.

As future work, we propose the extension to the ruleset used for formalizing problems as a way to generate traceability with different actors in processes of organizational and requirements analysis. We also propose the automation of the process with the generation of ontologies and syntactic and semantic relationships with the organization goals for validating the problems detected.

VII. REFERENCES

[1] P. P. Negri, V. E. S. Souza, A. L. de Castro Leal, R. de Almeida Falbo, G. Guizzardi. "Towards an Ontology of Goal-Oriented Requirements". *In CIbSE*. 2017. pp. 469-482.

[2] C. M. Zapata, F. Arango. "The UNC-Method: a problem based software development method". *Ingeniería e Investigación*. Vol. 29(1). 2009. pp. 69-75.

[3] N. Sánchez. "El marco lógico. Metodología para la planificación, seguimiento, y evaluación de proyectos". *Revista Visión Gerencial*. Vol. 6(2). 2006. pp. 328-343.

[4] H. Eriksson, M. Penker. "Business modeling with UML: Business patterns at work". *OMG Press*, 2000.

[5] M. Marinho, D. Arruda, F. Wanderley, & A. Lins. "A systematic approach of dataset definition for a supervised machine learning using NFR framework". In 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC). 2018. pp. 110-118.

[6] A. Martínez, O. Pastor, J. Mylopoulos, P. Giorgini. "From Early Requirements to Late Requirements: A goal-based approach". *Eight International Bi-Conference Workshop on Agent-Oriented Information System (AOIS)*. 2006. pp. 1-12.

[7] D. F. Mendonça, G. N. Rodrigues, R. Ali, V. Alves & L. Baresi. "GODA: A goal-oriented requirements engineering framework for runtime dependability analysis". *Information and Software Technology*. Vol. 80. 2016. Pp. 245-264.

[8] H. Estrada, A. Martínez, O. Pastor, J. Sánchez. "Generación de especificaciones de requisitos de software a partir de modelos de negocios: un enfoque basado en metas". *V Workshop de engenharia de requisites*. 2002. pp. 179-193.

[9] J. Jureta, S. Faulkner, P-Y. Schobbens. "Clear Justification of Modeling Decisions for Goal-Oriented Requirements Engineering". *Requirements Engineering*. Vol. 13(2). 2008. pp. 87-115.

[10] C. Banerjee, A. Banerjee & S. K. Sharma. "Estimating influence of threat using Misuse Case Oriented Quality Requirements (MCOQR) metrics: Security requirements engineering perspective". International Journal of Hybrid

Intelligent Systems. Vol. 14(1-2). 2017. pp. 1-11.

[11] U. Zafar, M. Bhuiyan, P. W. C. Prasad & F. Haque. "Integration of Use Case Models and BPMN Using Goal-Oriented Requirements Engineering". *JCP*. Vol. 13(2). 2018. Pp. 212-221.

[12] F. Vargas. "Método para establecer la consistencia de los problemas en el diagrama causa-efecto con el diagrama de objetivos de KAOS". *Tesis de maestría*. Universidad Nacional de Colombia. 2010.

[13] C. Zapata, J. Acevedo, D. Moreno. "Representación de relaciones semánticas entre problemas y objetivos mediante lógica de predicados". *Revista EIA*. 2011. Vol. 8(15). pp. 61-72.

[14] C. Zapata, L. Lezcano. "Caracterización de los verbos usados en el diagrama de objetivos". *Dyna*. Vol. 76(158). 2009. pp. 219-228.

[15] C. Menghi, P. Spoletini & C. Ghezzi. "Integrating goal model analysis with iterative design". *In International Working Conference on Requirements Engineering: Foundation for Software Quality*. pp. 112-128. 2018.

[16] L. Olsina & P. Becker. "Linking Business and Information Need Goals with Functional and Non-functional Requirements". In CIbSE. 2018. pp. 381-394.

[17] A. Dardenne, V. Van Lamsweerde, S. Fickas. "Goal-directed requirements acquisition". *Science of computer programming*. Vol. 20(1). 1993. pp. 3-50.

[18] E. Yu. "Modelling Strategic Relationships for Process Reengineering". *Ph.D. Thesis*. University of Toronto. 1995.

[19] A. I. Antón, C. Potts. "The use of goals to surface requirements for evolving systems". *20th IEEE International Conference on Software Engineering*. 1998. pp. 157-166.

[20] F. Vargas Agudelo. "Modelo para la especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas". *Tesis de doctorado*. Universidad Nacional de Colombia. 2016.

[21] Y. Sermet & I. Demir. "Towards an information centric flood ontology for information management and communication". *Earth Science Informatics"*. Vol. *12*(4). 2019. Pp. 541-551.

[22] C. M. Zapata, A. Gelbukh, F. A. Arango. "Pre-conceptual schema: A conceptual-graph-like knowledge representation for requirements elicitation". *5th Mexican International Conference on Artificial Intelligence*. 2006. pp. 27-37.

[23] B. Manrique-Losada, C. M. Zapata-Jaramillo & D. A. Burgos. "Re-expressing business processes information from corporate documents into controlled language". *In International Conference on Applications of Natural Language to Information Systems*. 2016. pp. 376-383.