Computational Linguistics for helping Requirements Elicitation: a dream about Automated Software Development

Carlos Mario Zapata J.
Leader of the Computational Language Research Group, School of Systems, Mines Faculty, Universidad Nacional de Colombia
Cra. 80 No. 65-223, of. M8A-310
Medellín, Colombia, South America
cmzapata@unal.edu.co

Abstract

Requirements elicitation is one of the first processes of software development and it is intended to be hand-made by means of analyst-stakeholder interviews. As a natural-language-based activity, requirements elicitation can take advantages of Computational Linguistics techniques, in order to achieve better results looking for automation in this field. In this paper we survey some of the work related to software development automation, guided by Computational Linguistics techniques, and performed by the Computational Language Research Group from the Universidad Nacional de Colombia. We aim the definition of future trans-national effort to be made in this research line.

1 Introduction

When stakeholders need to solve their information problems, they commonly search for the development of software applications (Pressman, 2005). At the beginning of this process, a set of analyst-stakeholder interviews take place, in order to capture the requirements belonging to the domain in which future software application must work. After that, in a hand-made process called "requirements elicitation", the analyst transforms the captured information into formal and semi-formal artifacts, mostly diagrams. At this stage, software application is specified by means of such diagrams (Leite, 1987).

Since interviews are the most used techniques for collecting software requirements, they experiment some of the most common problems of natural language (NL) communication: misunderstanding, ambiguity, and lack of clarity (Christel and Kang, 1992). However, as an NL-based process, requirements elicitation can use some of the Computational Linguistics (CL) and Natural Language Processing (NLP) techniques, as a way to solve such problems. The main goal of using CL and NLP techniques in this particular problem is related to the search for automation in the software development process.

This is the strategy we (the Computational Language Research Group—CLRG) choose to follow for clarifying requirements elicitation process and, therefore, for trying to automate the first phases of software development process. In this paper, we summarize some of the CLRG effort invested in helping requirements elicitation process with mostly CL techniques, but searching for strong NLP techniques, for instance, syntactical and discourse parsers, and named entity recognition systems, among others. We aim to show how we try to solve our problems in this field (recognizing the existence of too much effort from other groups in the world, but focusing on our own work), as a way to motivate the definition of trans-national projects searching for the same goals as us. Because our native language is Spanish, some of the examples we provide in this paper are encoded in this language.

The structure of this paper is the following: in section 2, we discuss our solutions to common problems of requirements elicitation process; in section 3 we propose some possible joint projects in this field of knowledge; finally, in section 4 we present conclusions and future work.

2 Solutions to common problems of requirements elicitation process

Figure 1 gives us the overall software engineering process envisioned by this research. This is a kind of "big picture" about the way we are creating CL- and NLP-based tools for helping automated software development process. In the following subsections, we discuss a more detailed view of every tool.

2.1 Pre-conceptual schemas

The first gap we needed to bridge in this process was related to the knowledge representation of requirements. In this context, the UML (Unified Modeling Language, OMG, 2010) is the *de-facto* standard for representing requirements, but it is a language directed to technical readers, and stakeholders are not usually technical people. For this reason, we explored the possibilities to use a graphical language closer to the stakeholder discourse, and we created the pre-conceptual schemas (Zapata, 2007) by adapting some previous effort made by Sowa's Conceptual Graphs (Sowa, 1984). Figure 2 shows an example of the pre-conceptual schemas, manually created by an analyst during the software elicitation process of one software application.

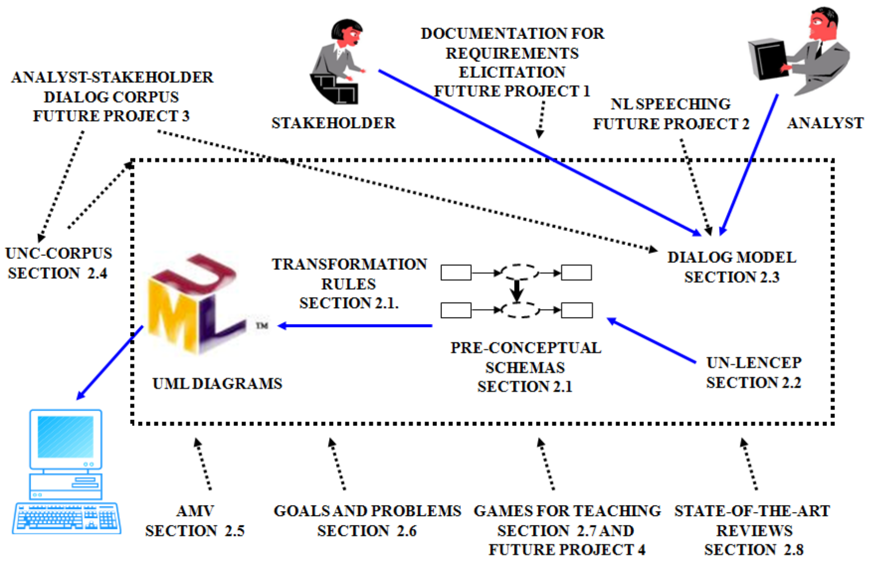

ANALYST-STAKEHOLDER DIALOG CORPUS FUTURE PROJECT 3
STAKEHOLDER
DOCUMENTATION FOR REQUIREMENTS ELICITATION FUTURE PROJECT 1
NL SPEECHING FUTURE PROJECT 2
ANALYST
UNC-CORPUS SECTION 2.4
TRANSFORMATION RULES SECTION 2.1.
DIALOG MODEL SECTION 2.3
UML DIAGRAMS
PRE-CONCEPTUAL SCHEMAS SECTION 2.1
UN-LENCEP SECTION 2.2
AMV SECTION 2.5
GOALS AND PROBLEMS SECTION 2.6
GAMES FOR TEACHING SECTION 2.7 AND FUTURE PROJECT 4
STATE-OF-THE-ART REVIEWS SECTION 2.8

Figure 1. Overall view of CL- and NLP-tools for automated software development.

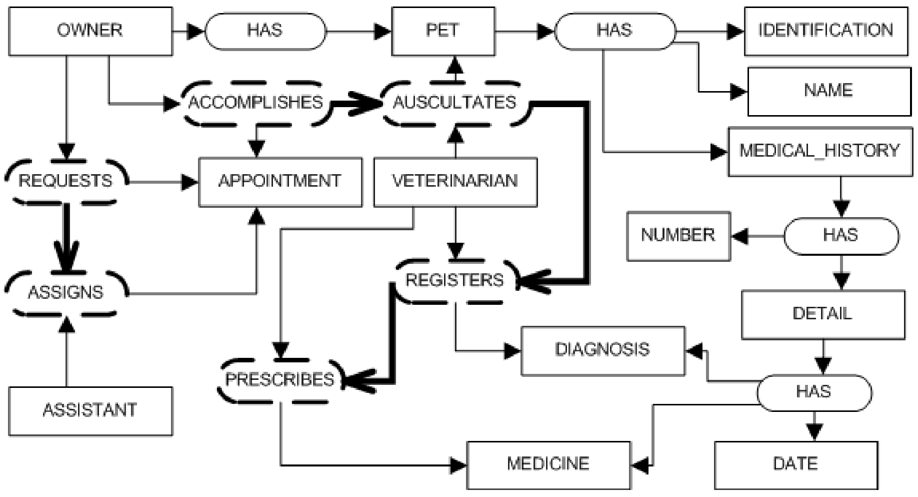


Figure 2. An example of Pre-conceptual Schemas (Zapata, 2007).

Pre-conceptual schemas have provided a new way to validate the stakeholder discourse, in order to clarify and understand what stakeholder has to say about the domain information related to the software application to-be-made.

2.2 UN-Lencep: Specifying pre-conceptual schemas

Pre-conceptual schemas gave us a new way to communicate with stakeholders in the requirements elicitation process, but their usage was limited to analysts. However, if we are interested in creating a pre-conceptual schema, we need the involvement of both kinds of actors in such action. In this case, we need to communicate each other in an NL-like way.

The solution to this problem came from two of the several techniques from Computational Linguistics: Information Extraction (IE) and Controlled Languages. In first place, we use a set of templates, in the same sense of IE templates, for matching in a stakeholder discourse the same features of a pre-conceptual schema. Then, we constrained the NL discourse, and we created UN-Lencep (*Universidad Nacional de Colombia—Lenguaje para la especificación de esquemas preconceptuales*, Zapata *et al.*, 2008). By combining both techniques, we had the possibilities to create a textual discourse in UN-Lencep. In the case of the pre-conceptual schema in figure 2, the UN-Lencep discourse could be something like this:

A pet belongs to an owner.
The pet has identification, name, and medical history.
The medical history has a name and one detail.
The detail has a date, a diagnosis, and a medicine.
When the owner requests an appointment, the assistant assigns an appointment.
When the owner accomplishes the appointment, the veterinarian auscultates the pet.
When the veterinarian auscultates the pet, the veterinarian registers the diagnosis.
When the veterinarian registers the diagnosis, the veterinarian prescribes the medicine.

Note that UN-Lencep phrases can be made by non-technical people, like stakeholders. The task of capturing requirements is now under the responsibilities of the analyst-stakeholder team, instead of the analyst alone. Again, the UN-Lencep discourse is manually created by the analyst with the help of the stakeholder. We have developed a tool called UNC-Diagrammer, for helping the software elicitation process in creating UN-Lencep discourses and pre-conceptual schemas. This tool has some minimal NLP processing, because UN-Lencep is a template-based controlled language.

2.3 Dialog model

UN-Lencep and pre-conceptual schemas provided the partial solution to our requirements capture problems. However, the fact that requirements elicitation was initiated by a set of stakeholder-analyst interviews reminded us the rest of the task. If we could discover a way to obtain the UN-Lencep discourse from something like an interview, we could link the beginning of the process to our partial solution.

The answer, again, came from previous experiences in Computational Linguistics. The work made on dialog models provided us an environment to prove our hypothesis about stakeholder-analyst interviews. We found some previous work on dialog models related to train reservations, and we employed it to discover the structure of dialog, as sets of tagged utterances and turnovers. With these ideas in mind, we propose a structure for requirements elicitation dialog (Zapata and Carmona, in press), as shown in figure 3. We are, also, exploring the close relationship between dialog models for requirements elicitation and ontologies (Zapata *et al.*, in press).

We are currently working on some projects for obtaining UN-Lencep discourses from a dialog with the structure provided by figure 3. Also, we are working in proving the utilities of such conversion in order to diminish software costs and development time in Latin-American software companies, and we select the COMPETISOFT model for promoting such improvement.

2.4 UNC-Corpus

Modeling is the center of requirements elicitation activities. We need models to understand the structure, the behavior, and the interaction among the concepts belonging to some domain. Traditionally, analysts make models by using their own knowledge and understanding of the world domain, in a subjective way. But, is it possible to simulate such activity? How can we represent the knowledge acquired about modeling by an analyst in creating models? The work in Corpus Linguistics provided us some useful ideas about these questions. A corpus is a collection of proved uses of a language. If we considered UML as a graphical modeling language (but, finally, *a language*), we could gather several "proved" uses of this language in the shape of computationally readable files. We employed these files to create UNC-Corpus (Zapata *et al.*, 2008), a UML-diagram corpus. Also, we used UNC-Corpus for "completing" diagrams, as analysts actually does, by reviewing the contents of the corpus as shown in figure 4.

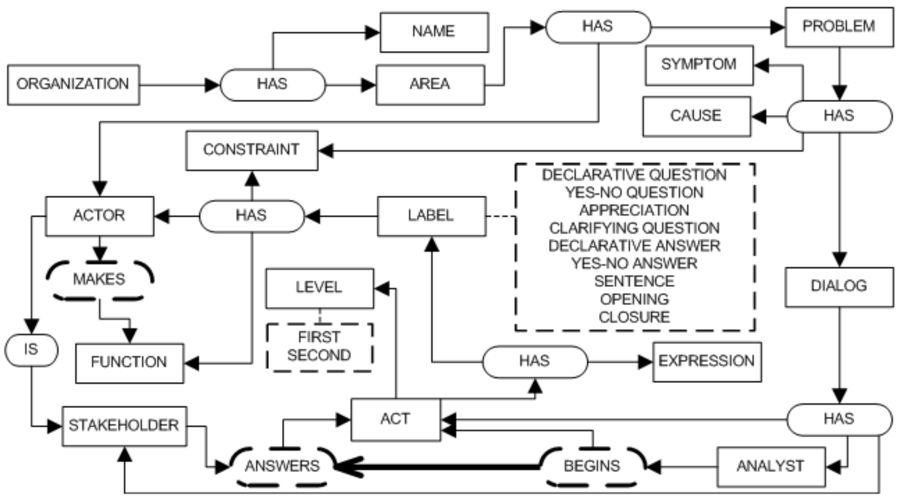


Figure 3. Requirements elicitation dialog model (Zapata and Carmona, in press).

2.5 AMV: a solution for conjugating and lemmatizing Spanish verbs

Spanish is one of the most difficult languages for tasks related to conjugate and lemmatize verbs. Our language has a complex structure when we need to use a verb.

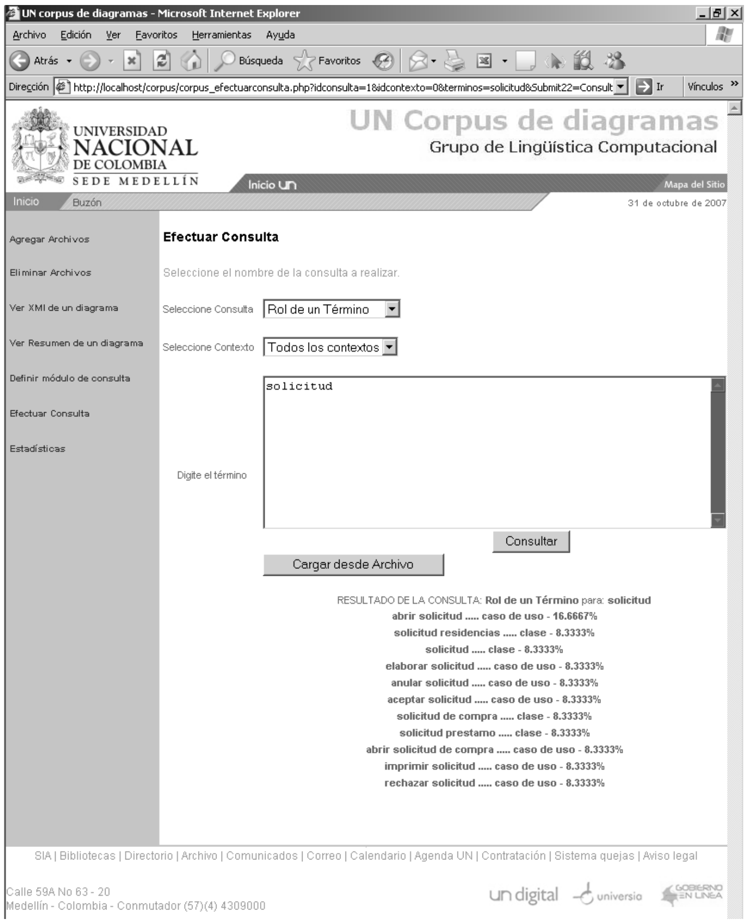
CLRG have assumed these difficulties and, after exploring state of the art in Spanish conjugators, decided to create AMV (*Analizador Morfológico de Verbos*, Zapata y Mesa, 2009), an application that recognize the internal structure of the vast majority of Spanish verbs. AMV can be shown in figure 5.

2.6 Goals and problems

AMV gave us some insight about the structure of Spanish verbs, so we could discover some differences about these verbs. For example, we discovered state verbs, action verbs, and goal verbs. Goal verbs are slightly different from the other kinds of verbs, because they express activities with no duration, generally associated to states to be reached. Three kinds of goal verbs can be identified: improvement, maintenance, and achievement verbs. Goal verbs are not recognized by most of the people, and their usage tends to be misunderstood along the software development process.

CLRG devoted some effort to identify goal verbs from NL discourses, and then represent them into pre-conceptual schemas (Zapata *et al.*, 2007). For completing this task, we used previous work of Antón (1997) for gathering some verbs in the above mentioned categories, and then we employed a lexicon from Maryland University in order to discover the internal linguistic features of such verbs. With this information in hand, we increased the number of available verbs for expressing goals. After that, we define a new set of symbols to be used in pre-conceptual schemas for representing goal verbs and then translating them into goal diagrams (Lezcano, 2007). Figure 6 shows an example of pre-conceptual schemas including goal verbs.

We are currently exploring the relationships among goals and problems. In our theory, problems are seen either as negative goals or obstacles for a goal to be reached. So, we are trying to define a set of structures for representing goals and another set for representing problems. Also, we are defining some rules for obtaining goal expressions from problem sentences and *viceversa*. The first step of the process was the state-of-the-art review of such structures (Zapata and Vargas, 2009), and we are delivering a Master's Thesis with the structures and the heuristic rules for proving such relationship.


UN corpus de diagramas - Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Atrás Búsqueda Favoritos
Dirección http://localhost/corpus/corpus_efectuarconsulta.php?idconsulta=1&idcontexto=0&terminos=solicitud&Submit22=Consult
Ir Vínculos
UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN
UN Corpus de diagramas
Grupo de Lingüística Computacional
Inicio UN
Mapa del Sitio
Inicio Buzón
31 de octubre de 2007
Agregar Archivos
Eliminar Archivos
Ver XMI de un diagrama
Ver Resumen de un diagrama
Definir módulo de consulta
Efectuar Consulta
Estadísticas
Consultar
Cargar desde Archivo
SIA | Bibliotecas | Directorio | Archivo | Comunicados | Correo | Calendario | Agenda UN | Contratación | Sistema quejas | Aviso legal
Calle 59A No 63 - 20
Medellín - Colombia - Conmutador (57)(4) 4309000
un digital universia GOBIERNO EN LÍNEA

Efectuar Consulta

Seleccione el nombre de la consulta a realizar.

Seleccione Consulta: Rol de un Término

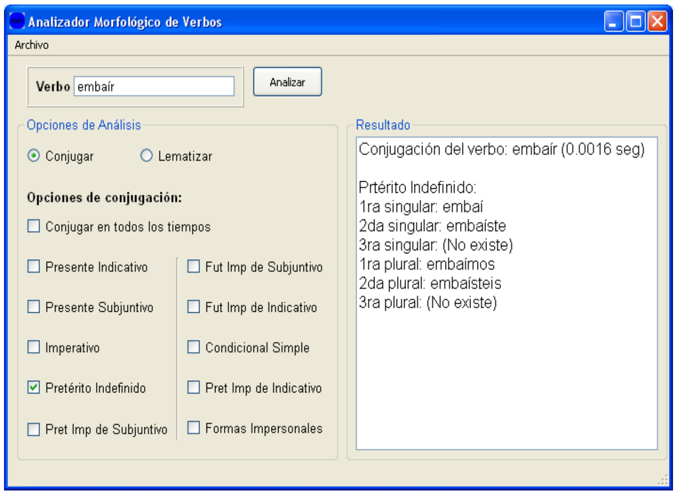
Seleccione Contexto: Todos los contextos

Digite el término: solicitud

RESULTADO DE LA CONSULTA: **Rol de un Término** para: **solicitud**

abrir solicitud ..... caso de uso - 16.6667%
solicitud residencias ..... clase - 8.3333%
solicitud ..... clase - 8.3333%
elaborar solicitud ..... caso de uso - 8.3333%
anular solicitud ..... caso de uso - 8.3333%
aceptar solicitud ..... caso de uso - 8.3333%
solicitud de compra ..... clase - 8.3333%
solicitud prestamo ..... clase - 8.3333%
abrir solicitud de compra ..... caso de uso - 8.3333%
imprimir solicitud ..... caso de uso - 8.3333%
rechazar solicitud ..... caso de uso - 8.3333%

Figure 4. A snapshot of the use of UNC-Corpus (Zapata *et al.*, 2008).



Verbo: embaír

Conjugar [✓] Lematizar [ ]

Opciones de conjugación:

[ ] Conjugar en todos los tiempos
[ ] Presente Indicativo
[ ] Presente Subjuntivo
[ ] Imperativo
[✓] Pretérito Indefinido
[ ] Pret Imp de Subjuntivo
[ ] Fut Imp de Subjuntivo
[ ] Fut Imp de Indicativo
[ ] Condicional Simple
[ ] Pret Imp de Indicativo
[ ] Formas Impersonales

```
Conjugación del verbo: embaír (0.0016 seg)

Prtérito Indefinido:
1ra singular: embaí
2da singular: embaíste
3ra singular: (No existe)
1ra plural: embaímos
2da plural: embaísteis
3ra plural: (No existe)
```

Figure 5. Snapshot of AMV (Zapata and Mesa, 2009).

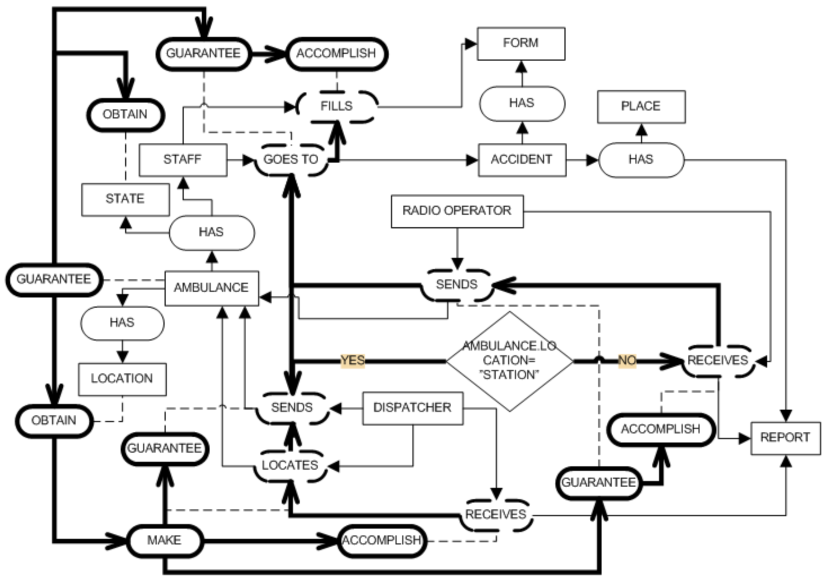


Figure 6. An example of pre-conceptual schemas including goal verbs (Zapata *et al.*, 2007)

2.7 Games for teaching

As a part of our research and teaching strategy, we use games to show and reinforce some concepts about our knowledge area. For example, we are currently developing an on-line game—called "Software Boulevard"—for understanding how software industries make their intangible products. In this game, we intend to simulate the real behavior of this kind of companies, but making the actors answer questions about software development process in several phases. Another example of our strategy is "Requirements elicitation dialog game" (Zapata and Giraldo, 2009), which is based on the importance of dialog inside the software development process. This game is like a word puzzle in which players must fill in the blanks a set of words previously acquired by answering questions related to software development. The blanks are located inside a simulated analyst-stakeholder interview and also as parts of a pre-conceptual schema. The main goal of the game is make conscious the players about the importance of good answers in requirements elicitation, in order to adequately "translate" the given information into diagrams that consistently reflect such information.

2.8 State-of-the-art Reviews

The definition of several projects requires the extensive search for papers and journals related to the topics we need to incorporate in the process. In addition to the mentioned review on goals and problems (Zapata and Vargas, 2009), we conducted some other state-of-the-art reviews on Controlled Languages (Zapata and Rosero, 2008), Dialog Models (Zapata and Mesa, 2009b), and the Wizard-of-Oz experiment (Zapata and Carmona, 2007). Also, we made a review on Interlinguas (Zapata and Benítez, 2009), and we are preparing some other reviews on Computational Dialog and Code Generation.

3 Joint projects on requirements elicitation and computational linguistics

Our final goal—and probably "dream"—is the automation of software development process from early stages related to speech discourses. We strongly believe this goal is so big enough to be reached by only one research group. We made now some part of the task, but we need help to complete it. For this reason, we want to create some transnational projects related to this field of knowledge to be executed by several research groups in Latin America, for example the Computation Research Centre from the Instituto Politécnico Nacional in Mexico, the Linguistic Engineering research group from the Universidad Nacional Autónoma de México, the Working Group 2.9 (Software Requirements Engineering) from IFIP (International Federation for Information Processing), and the Human-Computer Interaction Research Group from the Pontificia Universidad Católica de Valparaíso. We have contacts inside these research groups and we are willing to initiate joint research projects related to Computational Linguistics and Requirements Engineering.

The first project in which we are concerned is the use of technical documentation for requirements elicitation. In almost every organization in the world, technical documents define the way such organization must behave. If we were capable to understand the surrounding information in these documents, we could elicit many concepts to be validated in the analyst-stakeholder interviews, making too much work before the interviews take place. In this project, we need groups with expertise in analyzing and processing some kind of technical documents (for instance, technical reports, law sentences, instructions, and so on).

The second project we need to propose have natural language speeching as the main issue. The way a stakeholder-analyst interview is conducted suggests that some expressions are repeated once and again in the context of the dialog. These expressions are guidelines to gather important information about the domain. In this case, we need groups with larger experience in recording, retrieving, and analyzing speech dialogs.

A computational corpus of stakeholder-analyst interviews is the main product of the third project we want to execute. Corpus linguistics can offer many techniques for analyzing such corpus, in order to discover meta-information about the process of requirements elicitation by means of interviews. The common uses of expressions can lead to predictive information concerning one domain. Consequently, we need to gather as many recorded interviews as we can, and research groups with this kind of information.

Finally, games are needed for understanding and simulating the entire process of requirements elicitation as a result from stakeholder-analyst interviews, and this is the goal of the fourth project we need to propose. Our group has been using games in the shape of teaching strategies and we plan to keep using this strategy, because we think we are successful on it. Here, we need research groups with the intention to co-create and use games as teaching strategies. Also, we need people with some experience in evaluating the impact of games as teaching strategies.

The above mentioned projects have some CL- and NLP-based techniques as good offerings to find a solution. Also, for achieving the goals of every project, we need to interact with experts in software elicitation process. We hope this cross-functional and trans-national effort will give the necessary tools to make true the dream about automation in software development process.

4 Conclusions and future work

The Computational Language Research Group has been developing some projects for helping requirements elicitation process by means of Computational Linguistics, and we shown some of this work in this paper. We tried to summarize the most important of our projects concerning this issue, because our aim is to propose and develop trans-national projects in searching for automated software development.

The "big picture" of this work exhibits joint projects for making requirements elicitation closer to natural language dialog and speech. We look for a dream in which software development will be a simpler task, developed by common people by using natural language speeching interfaces.

Some work has still to be done:

- Eliciting requirements from technical documents belonging to an organization.
- Incorporating speech recognition to the requirements elicitation.
- Building a computational corpus of analyst-stakeholder interviews.
- Creating new games as teaching strategies for understanding the entire requirements elicitation process.

All of these projects are intended to be made by trans-national groups with some concern about software development process, computational linguistics, and natural language processing.

Acknowledgment

This work is founded by the *Vicerrectoría de Investigación* from *Universidad Nacional de Colombia,* under the project: “Software Boulevard un juego de estrategia en Web para la enseñanza de competencias de gestión en Ingeniería de Software”, Project number 9766.

References

Annie Antón. 1997. *Goal Identification and Refinement in the Specification of Software-Based Information Systems*. PhD Thesis, Georgia Institute of Technology, Atlanta, USA.

Michael Christel and Kyo Kang. 1992. *Issues in Requirement elicitation.* Technical Report, CMU/SEI-92-TR-012, ESC-TR-92-012. Software Engineering Institute, Carnegie Mellon University, Pittsburg.

Julio Cesar Leite. 1987. *A survey on requirements analysis*. Department of Information and Computer Science, University of California, Irvine, Advanced Software Engineering Project Technical Report RT-P071.

Luis Lezcano. 2007. *Elaboración semiautomática del diagrama de objetivos.* M.Sc. Thesis, Universidad Nacional de Colombia, Sede Medellín.

Object Management Group (OMG). 2010. *UML Superstructure.* Available at: http://www.omg.org/uml.

Roger Pressman. 2005. *Software Engineering: a practitioner's approach, 6th ed*. McGraw Hill, New York.

John Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Publishing Co., Reading, MA.

Carlos Zapata. 2007. *Definición de un esquema preconceptual para la obtención automática de esquemas conceptuales de UML.* Ph.D. Thesis, Universidad Nacional de Colombia, sede Medellín.

Carlos Zapata and Servio Benítez. 2009. Interlingua: Análisis crítico de la literatura. *Revista Facultad de Ingeniería Universidad de Antioquia*, 47:117–128.

Carlos Zapata and Nicolás Carmona. In press. Un modelo de diálogo para la Educción de Requisitos de Software. *Dyna.*

Carlos Zapata and Nicolás Carmona. 2007. El experimento Mago de Oz y sus aplicaciones: Una mirada retrospectiva. *Dyna*, 74(151):125–135.

Carlos Zapata and Gloria Giraldo. 2009. El juego del diálogo de educción de requisitos de software. *Avances en Sistemas e Informática*, 6(1):105–114.

Carlos Zapata, Gloria Giraldo, and John Mesa. In press. Una propuesta de Metaontología para la Educción de Requisitos de Software. *Ingeniare.*

Carlos Zapata, Alexander Gelbukh, and Fernando Arango. 2006. UN-Lencep: Obtención Automática de Diagramas UML a partir de un Lenguaje Controlado. *Memorias del VII Encuentro Nacional de Computación ENC'06*, San Luis Potosí, México, 254–259.

Carlos Zapata, Juan Hernández, and Raúl Zuluaga. 2008. UNC-Corpus: corpus de diagramas UML para la solución de problemas de completitud en ingeniería de software. *Revista EAFIT*, 44(151):93–106.

Carlos Zapata, Luis Lezcano, and Paula Tamayo. 2007. Validación del método para la obtención automática del diagrama de objetivos desde esquemas preconceptuales. *Revista Escuela de Ingeniería de Antioquia*, (8):21–35.

Carlos Zapata and John Mesa. 2009. Una propuesta para el análisis morfológico de verbos del español. *Dyna*, 76(157):27–36.

Carlos Zapata and John Mesa. 2009b. Los Modelos de Diálogo y sus Aplicaciones en Sistemas de Diálogo Hombre-Máquina: Revisión de la literatura. *Dyna*, 76(160):305–315.

Carlos Zapata and Roberto Rosero. 2008. Revisión Crítica de la Literatura especializada en Lenguajes Controlados. *Avances en Sistemas e Informática*, 5(3):27–33.

Carlos Zapata and Fabio Vargas. 2009. Una revisión de la literatura en consistencia entre problemas y objetivos en Ingeniería de Software y Gerencia Organizacional. *Revista Escuela de Ingeniería de Antioquia*, 11:117–129.