Journal of Object Technology
Published by AITO — Association Internationale pour les Technologies Objets
http://www.jot.fm/

A Collaborative Approach to Describe the Domain Language through the Language Extended Lexicon

Leandro Antonelli[a] Gustavo Rossi[a] Alejandro Oliveros[b]

a. Lifia, Fac. de Informática, UNLP, Buenos Aires, Argentina

b. INTEC-UADE and UNTREF, Buenos Airess, Argentina

Abstract Software development is a succession of descriptions in different languages in which every description is based on a previous one. Since the requirements specification is one of the first descriptions, it is important to begin software development with requirements that are as correct and as complete as possible. Although some literature holds the belief that correctness and completeness are two attributes that requirements specifications must satisfy, we know that these attributes are very difficult to meet. However, we have to find ways to diminish the level of incompleteness and deal with the possible conflicts that do arise in the requirements context. Defining the domain language before specifying the requirements is a way of coping with these problems. Nowadays, software systems involve many stakeholders and it is hard to engage all of them to produce a domain language specification. We rely on *collaboration* to foster the involvement and cooperation of the stakeholders, thus they are able to explore the differences constructively and provide a common understanding of the domain language beyond their own limited views. In this paper, we propose a strategy to capture the domain language in a collaborative way using the Language Extended Lexicon and we show a validation of the proposed strategy.

Keywords Domain Analysis; Language Extended Lexicon; Collaboration.

1 Introduction

The development of software systems is a complex activity since many actors with different concerns are involved. During the development, they perform several tasks with the aim of building diverse software products which are interrelated, since they use many others as input. Besides the interconnection between the artifacts, the nature

Leandro Antonelli, Gustavo Rossi, Alejandro Oliveros. *A Collaborative Approach to Describe the Domain Language through the Language Extended Lexicon*. In Journal of Object Technology, vol. 16, no. 3, 2016, pages 3:1–27.
doi:10.5381/jot.2016.15.3.a3

of software also makes its development a complex task. Brooks [Bro95] states that complexity is a characteristic inherent to software. Ackoff states that we commonly fail in software construction not because the solution is not technically well-built, but because it does not apply to the problem [Ack74]. Nowadays, this statement is still true, as several surveys confirm [Gro13].

We can represent the software development process as a succession of descriptions in different languages in which a previous description is necessary for the next [SV06]. So, if changes are incorporated into a description, previous and subsequent descriptions will have to be modified in order to maintain conformity. For instance, Boehm [Boe81] states that if a mistake occurs in requirements description and it is corrected in code description, the correction cost could be up to 200 times higher than the cost needed for its correction at the requirements stage. Moreover, Mizuno developed the "waterfall of errors" [Miz83], in which he states that in each stage of the software development, the possibility of occurrence of mistakes is stronger than in the previous ones, because each stage relies on products from the previous stages.

Thus, it is important to begin the software development with requirements that are as correct and as complete as possible. Although some literature holds the belief that correctness and completeness are two attributes that requirements specifications must satisfy [IEE98], other literature states that this fulfillment is unfeasible to provide in many situations [FGH^{+}94]. However, we have to find ways to diminish incompleteness and deal with the possible conflicts that tend to arise in the requirements context. Defining the domain language before specifying the requirements is a way of coping with this problem.

The Language Extended Lexicon (LEL) is a a glossary that specifies an application domain (context) language [dPLF93]. The LEL is a very convenient tool for stakeholders with no technical skills, although people with such skills will profit more from its use [dPAOdPLCC07]. The LEL effectively captures and describes the application domain language because it conforms to the mechanism used by the human brain to organize expert knowledge [Woo97]. In particular, the convenience of the LEL as a tool arises from three significant characteristics: it is easy to learn, it is easy to use and it has good expressiveness. There are several publications using the LEL in complex domains which validate these claims. Gil et al. [GFO00] state that "building a LEL in an application completely unknown to the requirements engineer and with highly complex language can be considered a successful experience, since users stated that requirements engineers have developed a great knowledge about the application". Cysneiros et al. [CdPL01] state that "the use of the LEL was very well accepted and understood by the stakeholders. As these stakeholders were non-technical experts from a specific and complex domain, the authors believe that the LEL can be suitable to be applied in many other domains". The characteristics mentioned above contribute significantly to obtaining high quality models, as they allow the actors involved in software development (experts, requirements engineers and developers with different capacities and abilities) to perform the validation of a LEL [KHDdPL00].

Nevertheless, it is very difficult to produce a domain language specification when there are too many stakeholders involved [MG12] [WTL13] [LF12]. Cleland-Huang et al. state that there exist ultra-large-scale projects that have thousands or even hundreds of thousands of stakeholders [CHM08]. According to Northrop et al. [NFG^{+}06] and Cheng et al. [CA07], the human interaction element makes requirements elicitation the most difficult activity to scale in software engineering. We rely on collaboration in order to foster the cooperation of the stakeholders, so that they are able to explore the differences constructively and search for solutions that go beyond their own limited views [How06] [QB09] [PRVPB12] [MGvdHW10] [WTL13]. In a collaborative context, all the participants co-construct together even if the task can be divided into several new subtasks. Requirements elicitation, as an interdisciplinary process, requires specific competences from all the users and the stakeholders involved. Collaboration is therefore necessary, as no person can possess all the competences from all the disciplines required for this task. In addition, good collaborative requirements elicitation produces richer, more complete and more consistent requirements [KSK14].

In this paper, we show how to capture the domain language in a collaborative way. This paper builds on the research paper 'A Collaborative Approach to Capture the Domain Language', which was published in the 18th Workshop of Requirements Engineering (WER), held at the XVIII Ibero-American Conference on Software Engineering (CIBSE) [ARO15]. The present work extends the previous by incorporating new operations to the strategy which arose from an extension of the validation and by a deeper analysis of the related works. The rest of the paper is organized as follows: Section 2 presents some background necessary to understand the strategy. Section 3 describes the strategy. Section 4 shows an experiment validating the strategy. Section 5 discusses some related works. Finally, Section 6 presents some conclusions and future works.

2 Background

This section describes the Language Extended Lexicon (LEL), a tool used for capturing the language of the application domain. An LEL is a glossary whose goal is to record the definition of terms that belong to a domain. It is tied to a simple idea: "understand the language of a problem without worrying about the problem" [dPLF93].

Terms (called symbols with an LEL) are defined through two attributes: notion and behavioral responses. Notion describes the denotation and the intrinsic and substantial characteristics of the symbol, while behavioral responses describe its connotation, i.e. the relationship between the term being described and other terms.

There are two principles that must be followed while describing symbols: the circularity principle (also called the closure principle) and the minimal vocabulary principle. The circularity principle states that the use of LEL symbols must be maximized when describing a new symbol. The minimal vocabulary principle states that the use of words that are external to the Lexicon must be minimized. These principles are vital in order to obtain a self-contained and highly connected LEL.

Each symbol of the LEL belongs to one of four categories: subject, object, verb or state. Subjects are active elements within the domain. They are similar to actors in Use Case modelling or agents in organization modelling. It is important to mention that in the LEL there is no definition of the scope of the system, thus, the whole system or even subsystems could be considered as subjects symbols. Objects are passive elements in the domain and they are resources or elements that subjects use. Verbs are actions performed by subjects using objects. And states are situations in which subjects, objects or verbs can be involved. Leite et al. provide more description about how to categorize symbols [dPLF93]. This categorization guides and assists the requirements engineer during the description of the attributes. Table 1 shows each category with its characteristics and how to describe them.

Some examples of LEL symbols from each category are presented using the classic bank application, which allows its clients to open and close accounts. If an account is

Category	Characteristics	Notion	Behavioural Responses
Subject	Active elements which perform actions	Characteristics or condition that subject satisfies	Actions that subject performs
Object	Passive elements on which subjects perform actions	Characteristics or attributes that object has	Actions that are performed on object
Verb	Actions that subjects perform on objects	Goal that verb pursues	Steps needed to complete the action
State	Situations in which subjects and objects can be found	Situation represented	Actions that must be performed to change into another state

Table 1 – LEL categories.

Subject: client
Notion
Person that operates an account.
Behavioral responses
The client can open an account.
The client can deposit money into his account.
The client can withdraw money from his account.
The client can consult his account balance.
The client can close an account.

Figure 1 – *Client* symbol's description.

activated (open), the client can deposit or withdraw money and consult the balance. The bank can also perform a cash audit.

It is important to mention that the terms that appear in the description of the symbols, and correspond with other defined symbols, are underlined in order to show the application of the circularity principle. The following examples are: subject *client* in Figure 1; object *account* in Figure 2; verb *withdraw* in Figure 3; and state *activated* in Figure 4.

3 Our approach

This section describes the proposed approach to construct a LEL in a collaborative way. We describe first its essence and, afterwards, we give details and examples about the approach. It consists of two main steps. First, a network of stakeholders must be described using the snowballing technique [LF12]. After that, the symbols of the LEL must be identified and described.

In order to describe the network of stakeholders with the snowballing technique, some key stakeholders must be identified to begin the process. Then, these key stakeholders nominate more stakeholders who, in turn, appoint some others. Thus, a network description in which nodes describe stakeholders and links describe recommendations

Object: account
Notion
The account has a balance.
Behavioral responses
The client can open an account.
The client can deposit money into his account.
The client can withdraw money from his account.
The client can consult his account balance.
The bank performs a cash audit.
The client can close an account.

Figure 2 – *Account* symbol's description.

Verbs: withdraw
Notion
Act of taking money from the account.
Behavioral responses
The bank must check that the account has enough money to perform the withdrawal.
The bank must check that the owner of the account has not withdrawn more times than the limit allows.
The bank must check that the owner of the account does not have any credit card debts.
The bank reduces the balance of the account according to the amount withdrawn.

Figure 3 – *Withdraw* symbol's description.

State: Activated
Notion
Situation where the client is ready to use an open account.
Behavioral responses
The client can close the account and he will have a closed account.

Figure 4 – *Activated* symbol's description.

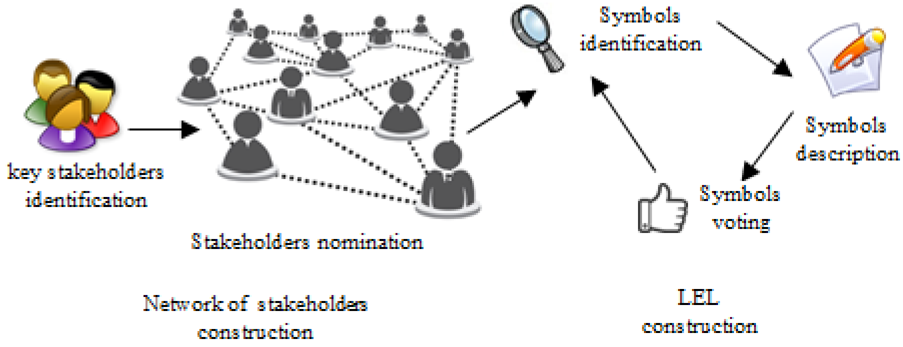

Figure 5 – Our approach in a nutshell.

is built. The node must also include the role of the stakeholder.

In order to build the LEL in a collaborative way, the stakeholders involved in the network must identify symbols, describe them, and also vote for (indicate they *like*) descriptions. First, subject symbols must be identified from the network and then described. Each role of the network must be considered a subject symbol. In order to describe symbols, different people can add different expressions to the same symbol. And they can also indicate they *like* an expression. The Symbols from the other categories must be identified from subject symbols. In general, verb symbols can be identified from the behavioral responses of subjects. Then, after describing verb symbols, object symbols can be identified.

Figure 5 summarizes the approach. It begins with the identification of key stakeholders. Then, the network of stakeholders is built using the snowballing technique. Finally, the LEL is defined in a collaborative and iterative way with the participation of the stakeholders from the network. The iteration consists of several steps: (i) identification of symbols, (ii) description of the symbols and (iii) voting.

3.1 Building a network of stakeholders

In social network analysis, the snowballing method is generally used to sample social network data for large networks in which the boundary is unknown. Snowball sampling begins with a set of stakeholders. Each of these stakeholders is asked to nominate other stakeholders. Then, new stakeholders, who are not part of the original list, are similarly asked to nominate other stakeholders. As the process continues, the group of stakeholders builds up like a snowball rolling down a hill. The process continues until no new stakeholders are identified, time or resources have run out, or when the new stakeholders being named are out of the scope set under study [LF12].

In order to build the network of stakeholders with the snowballing technique, the requirements engineer in charge of constructing the LEL with the help of the sponsor of the project must identify some key stakeholders who will initiate the nomination process. Apart from the identification of the stakeholders, it is important to describe the role they play.

Let's consider an example from the banking domain. The key stakeholders are John (an accountant) and Walter (a client of the bank). John nominates Arthur (a cashier), and Walter nominates Alice (another client) and he also nominates Arthur

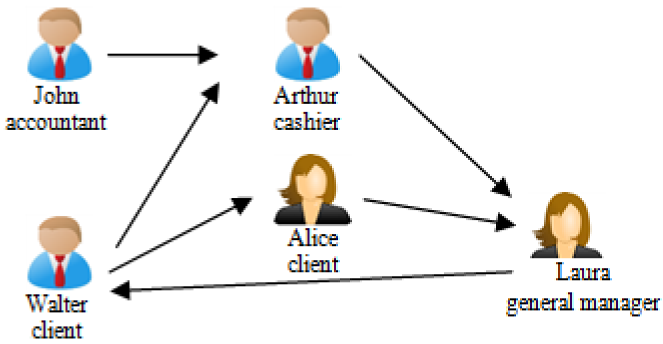

Figure 6 – Stakeholder network example.

(already nominated by John). Then, both Arthur and Alice nominate Laura (the general manager). Finally, Laura nominates Walter, but since he had already been nominated and not a single new stakeholder was nominated this time, the process ends. Figure 6 shows the network of stakeholders with their roles and nominations.

3.2 Building a LEL in a collaborative way

The traditional process to build a LEL consists of two activities: identifying and describing the symbols and both activities are performed exclusively by a requirements engineer. In our approach we propose performing both activities in a collaborative way, and we also add a social activity: expressing *like* to an expression that defines a symbol. It is important to mention that this collaborative approach includes no requirements engineer, and the activities are performed directly by the stakeholders. The identification and the description of the symbols occur in a collaborative way, and different people cooperate to define a symbol. Thus, expressing *like* is a way of validating the contribution that other people have made. For example, one person identifies a symbol and another includes an expression to describe it. Next, a different person adds one more expression to the description. And, finally, somebody else reviews the symbol and its definition and can indicate he *likes* any of the expressions. Although we could have used a rating scale from zero to five to express conformity instead of the *like* tool, the *like* tool proved to be more effective to express agreement [SN11].

Let's consider the following example: John identifies the symbol *client*. He only identifies the symbol but does not give a definition. Then, Walter adds an expression ("*The client can deposit money*") to the behavioral responses of the symbol client. After that, Arthur adds another expression (*"The client can withdraw money"*) to the behavioral responses of the symbol *client*. Finally, Alice does not wish to add any new descriptions but she indicates that she likes the description added by Walter. Figure 7 shows this collaboration.

The identification of symbols can be performed in several steps. Symbols of the subject category must be identified from the roles of the network. Each role of the network must be considered a subject symbol. Then, the description of the behavioral responses of the subject symbols will provide the source to identify symbols of the verb category. And, finally, the description of behavioral responses of the verb symbols will provide object symbols. Of course, identification and description occur in an iterative

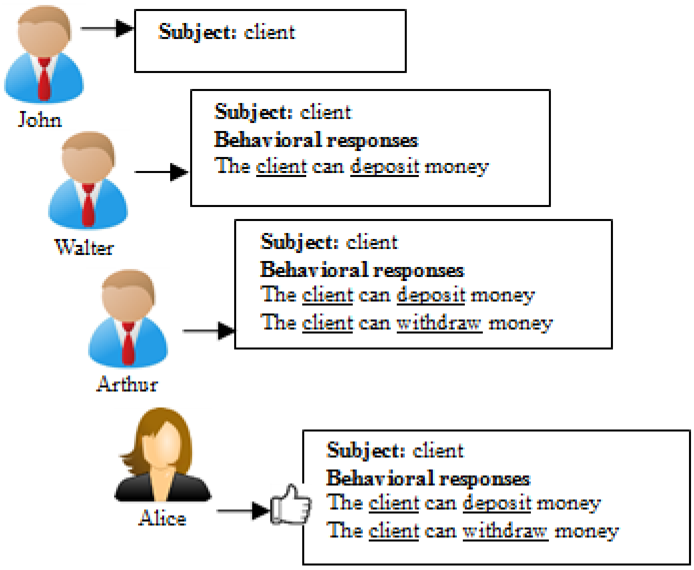

Figure 7 – Collaboration in the definition of the symbol.

and incremental way and the order of the steps mentioned above are only suggestions.

Let's consider the symbol *client* of the subject category, which includes the operation (verb) *withdraw* in its behavioral responses. Also, the verb *withdraw* must be defined and described. Then, the symbol of the object category *account* appears in the behavioral responses of the symbol *withdraw*, so the symbol *account* must be described as well. Figure 8 shows the example.

It is important to mention that the identification and the description of the symbols occur in an iterative and incremental way. There are not rigid steps to follow: it is just a recommendation to describe subjects first, then verbs from the subjects and finally objects from the verbs; but the identification can occur in any sequence. For example, it is possible that a stakeholder identifies the subject symbol *bank* from the description of the verb symbol *withdraw*. Moreover, in order to describe *bank*, the object symbol *money* (that also appears in *client*) may first be identified and described. Thus, different people collaborate in an iterative way identifying, defining and expressing *like* to symbols.

The identification and description of symbols involve two further actions. The first is removing unnecessary symbols and expressions. And the second is merging symbols. The concept of "owner" emerges to perform the operation of removal. The owner of a symbol or expression is the person who has added it. He is the sole person entitled to remove the symbol or expression, but only if nobody else has indicated *like* or added another expression to it. The removal is intended to clean the contributions that have not been supported by anyone.

Let's consider a different scenario for the definition of the symbol *client*. The first two steps are the same: John identifies the symbol *client* and does not include a definition. Then, Walter adds an expression (*"The client can deposit money"*) to the behavioral responses of the symbol client. After that, Arthur adds another expression (*"The client can put money in his account"*) to the behavioral responses of the symbol

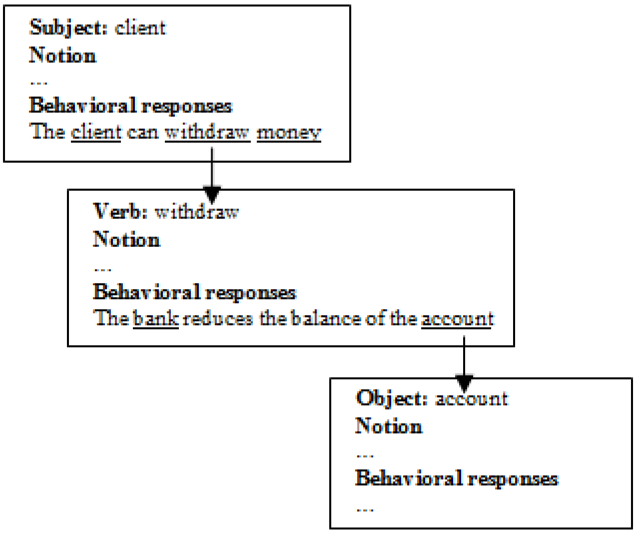

Figure 8 – Example of the identification of symbols.

client. Then, Alice does not wish to add any new descriptions but she indicates that she *likes* the description added by Walter. Finally, Arthur realizes that the expression he has added is similar to the expression added by Walter. Thus, Arthur decides to remove his expression and indicates he *likes* the description added by Walter. Figure 9 shows this collaboration.

Identifying synonyms is one of the most important and complex tasks while constructing a LEL. It is important because recurring symbols which are not identified as synonyms can cause misunderstanding. And it is complex because sometimes only a few stakeholders with a high level of expertise and a deep understanding of the domain can detect synonyms, and usually these people do not devote their time to the creation of the LEL. For this reason, we decided to include the operation of symbols merging in our approach. That means that any stakeholder participating in the description of the LEL can detect two different symbols that have similar definitions and merge them. In a merging operation, both symbols are used to identify a new one; both descriptions are juxtaposed, and the likes are preserved. The merging process does not imply the removal of any sort of elements (for example repeated descriptions). We are considering and exploring this issue for future work.

Let's consider a different scenario from the one described in Figure 8. The symbol *client* was identified, and after its description, the symbol *withdraw* was also identified. At the same time, another stakeholder identifies and describes the symbol *extract* in a similar way to *withdraw* (*"The bank reduces the amount of money of the account"*). Somebody else notices that *withdraw* and *extract* represent the same concept, and he decides to merge them in only one symbol with both descriptions. We will deal with the removal of repetitions in our future work.

4 Validation

We conducted two different experiments in order to verify the effectiveness and the applicability of our collaborative approach. In order to assess its effectiveness, we

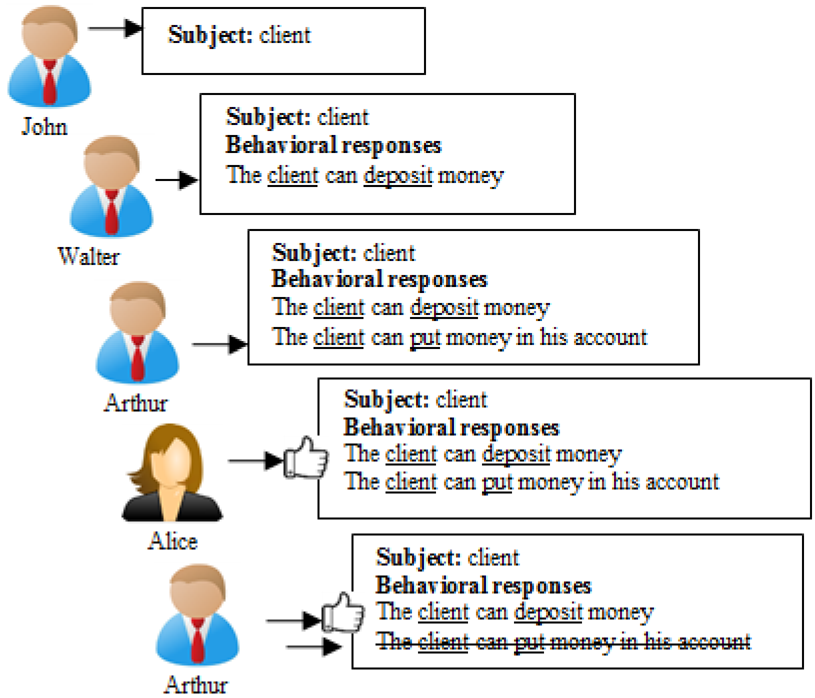

Figure 9 – Example of removal.

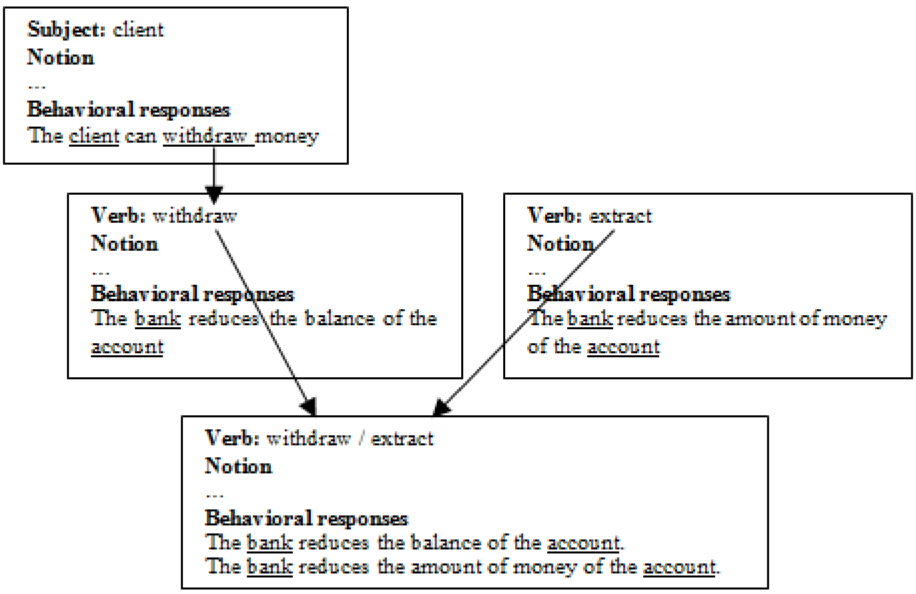

Figure 10 – Example of merging.

verified that a group of people could work in a common application domain to produce a richer LEL than the LEL produced by only one person. It is important to mention that this first experiment verified the applicability of a collaborative approach, but did not intend to verify our proposed approach. That is, in the first experiment, different groups of people worked separately in the same domain and we compared these works in order to verify that there was a core common to all the works. Then, we thought that if these people worked together collaboratively they would produce a cohesive and coherent work. And the applicability of our proposed approach was actually verified in a second experiment in which it was finally used. We trained a group of people to apply the proposed approach and all of the participants worked collaboratively to produce a unique LEL. While in the first experiment we wanted to verify that people could contribute to the construction of a LEL with a common core, in the second, we wanted to verify our approach with regard of the dynamics of many people working in an unique product.

4.1 Participants

Both experiments were part of the activities of a Computer Science postgraduate course. The participants of the first experiment were 41 students, while only nine students were involved in the second experiment. All the students had a degree in Computer Science and some experience in the software development industry: as developers, analysts or as team leaders. Some participants were also lectures at the University. Most of the participants were Argentinean but there were also people from Colombia.

4.2 Design of the first experiment

The first experiment was presented in a class. During the presentation, the participants were instructed in how to identify and describe symbols. They received three hours of training and some reference material. Then, 15 groups were formed (some groups worked remotely). Every group had to produce one LEL; thus, at the end of the experiment, we had 15 different LELs to analyze. The students had five weeks (from November 2013 to December 2013) to produce the LEL. The application domain chosen was a travel site. The participants received a brief introduction and they were assigned two websites for study. We worked on the construction in two stages so that we could evaluate the progress of their work: after two weeks, they had to present a list of the identified symbols and describe some of them. After some feedback, they had three more weeks to finish the LELs.

4.3 Analysis of the first experiment

The analysis to verify the effectiveness of a collaborative approach consisted in comparing the consolidated work of the 15 groups with the work of each single group. In particular, we verified that the consolidated work (which is in someway a collaborative work) can produce a richer LEL than any of the LELs produced by each single group. Thus, we verified that part of the LEL produced by each group was shared by all the groups and, every group also included more different symbols and different descriptions of the symbols. It is important to note that all the groups agreed in a shared set of core symbols because, although they were able to add a different point of view to the

Group	States	Objects	Subjects	Verbs	Total
1	6	14	6	20	46
2	5	15	4	12	36
3	5	13	7	16	41
4	11	10	3	17	41
5	12	6	4	21	43
6	11	6	7	12	36
7	8	11	7	21	47
8	5	4	5	15	29
9	13	16	6	13	48
10	11	34	10	18	73
11	7	10	3	8	28
12	6	9	6	16	37
13	6	8	2	9	25
14	6	10	6	17	39
15	5	9	5	8	27

Table 2 – Symbols identified by groups.

LEL, they had to agree on the essence of the domain and we wanted to verify this point before running the experiment with our approach.

Considering the fifteen LELs produced by the groups, a total of 595 symbols were identified and described. That means that an average of 40 symbols were identified and described by each group. Beyond this average, the quantity of symbols identified was quite balanced. The number of states ranged between five and 13. The number of objects ranged between five and 15 except for the group number 10, which obtained 34. The number of verbs ranged between eight and 21. And the number of subjects ranged between two and seven except for the group number 10 which obtained 10. Table 2 shows the details. It is worth noting that there were many instances of the same symbols. The number of symbols without duplication was 281. That means that there were a big number of repeated symbols among the LELs produced by each group and a small percentage of non-repeated symbols. This suggests that the collaboration of different groups of people working on the same application domain allows obtaining a richer LEL than the one obtained by each group.

We also analyzed the percentage of the average of non-repeated symbols grouped by categories and we found that every group provided 23% of non-repeated subject symbols, while they only provided between 13% and 14% of non-repeated object, verb and state symbols (Table 3). That could mean that five groups would be enough to identify and describe subjects, while we would need eight groups for the other categories. The numbers show that a collaborative approach can have benefits over an individual approach.

We identified and analyzed the most frequently repeated symbols defined by more than 11 groups. These are 13 symbols and they are enumerated in Table 4. In particular, we focused on the symbols *client* and *search* because they were the most representative of the application domain.

We analyzed the number of expressions in the notion and the behavioral responses descriptions as well as the average by group in order to identify the symbol intended for deeper analysis. All the groups defined a total of 53 expressions to describe the notion and 309 expressions to define the behavioral responses of the symbol *client*, while

Category	Average of Non replicated symbols
Subject	23%
Object	13%
Verb	14%
State	13%

Table 3 – LEL categories.

Category	Symbols
Subject	Client Company
Object	Flight Hotel Service Car
Verb	Search Cancel Reserve Buy Recommend
State	Reserved Canceled

Table 4 – Symbols most frequently repeated.

they defined 16 expressions to describe the notion and 49 expressions to describe the behavioral responses of the symbol *search*. The previous values were the total quantity of the expressions defined by all the groups, including repeated expressions. The average of symbols by groups, that is, the previous value divided by 15 (the number of groups), are detailed in Table 5. Since *client* is a symbol with more description than *search*, we decided to analyze the symbol *client*. In particular, we analyzed the behavioral responses because they have the most complete description.

The groups identified 309 expressions to describe behavioral responses for the symbol client. These 309 expressions contained only 117 different expressions and 192 repeated expressions. This is a clear clue that a collaborative work would save time and effort, since people working separately obtained mostly the same results.

	Client		Search	
Expressions	Notion	Behavioural Responses	Notion	Behavioural Responses
Total Number	53	309	16	49
Average	3.78	22.14	1.06	3.26

Table 5 – Symbols most frequently repeated.

4.4 Conclusions of the first experiment

In summary, we analyzed the repetition in the identification and in the description of symbols and we found that a group of different people working in the same application domain agree on their core description and, also, add more descriptions according to their different points of view. This suggests that a collaborative approach would have benefit, since a group of different people working separately identified and described a common core of symbols. Thus, if they had worked collaboratively, we think that they would have obtained the same results but in less time and with less effort.

4.5 Design of the second experiment

The second experiment was also presented during a class (of the same course but a different cohort). During the presentation of the experiment, the participants were instructed in different operations applying the proposed collaborative approach: (i) identifying symbols, (ii) adding expressions, (iii) indicating *like*, (iv) removing expressions and (v) merging symbols. The participants (nine people) received three hours of training and some reference material. Every participant had to collaborate in the production of a shared LEL. This time, they had four weeks (from January 2015 until February 2015) to work in the LEL. The application domain was the same as in the first experiment: a travel site. The participants received a brief introduction and they were also assigned two websites for study.

We did not test the snowballing technique because we engaged all the students to participate. The students involved constituted a suitable group for the task because many of them had experience with the travel site, either as clients or as part of the development team. The goal of this second experiment was to verify that a collaborative approach could produce a coherent and consistent LEL. These two characteristics are very important to produce a complete LEL. Every new symbol identified and described makes it possible to identify many new symbols from it. Thus, people working collaboratively help to produce a complete LEL. But, since there is no guarantee that people has an overall view of the LEL, it is important to assure its coherence. In order to assess this objective, we compared the results of the first experiment with the results of the second one, in which the approach we propose was applied.

4.6 Analysis of the second experiment

In the first experiment, 281 different symbols were identified, and considering that there were 15 groups involved, each group identified an average of 18 symbols. In the second experiment, 26 symbols were identified. This number is significantly smaller than the total of 281 different symbols from the first case, but it is bigger than the average of 18 symbols by groups. This suggests that people working in a collaborative way were focused on a core set of symbols instead of expanding to different areas of the domain as when people worked separately. Moreover, people working collaboratively produced a LEL with more symbols that the LEL produced by each group in the first experiment. This means that the collaborative LEL was more complete and also more coherent (focused). This is supported by the coherence in the identification of symbols in both experiments. In the first, 13 symbols were identified as the most frequently repeated, which were also included among the symbols identified in the second experiment. Most of them (11 out of 13) were identified with the same name

Category	Symbols of the first experiment	Symbols of the second experiment
Subject	Client Company	Client Company
Object	Flight Hotel Service Car	Flight Accommodation Service Transfer
Verb	Search Cancel Reserve Buy Recommend	Search Search flight Search accommodation Search transfer Cancel Cancel Reservation Cancel Purchase Reserve Buy Recommend
State	Reserved Canceled	Reserved Canceled

Table 6 – Symbols most frequently repeated in the first experiment and symbols defined in the second case.

in both experiments, and two were not exactly the same, but they represented the same concepts. For example, in the first experiment the symbol *hotel* was identified, while in the second, it was called *accommodation*. And in the first experiment *car* was identified whereas it was called *transfer* in the second one. Then, in the collaborative approach, some symbols further specified the types of symbols identified in both experiments. For instance, the symbol *search* was specified as *search flight*, *search accommodation*, and *search transfer*, while *cancel* was specified as *cancel reservation* and *cancel purchase*. Table 6 lists the symbols most frequently repeated in the first experiment and their equivalents in the second case.

Then, we compared the description of the symbol *client* from the first experiment with its description in the second case in order to assess the coherence of the description. In the first case, *client* had an average of 3.7 descriptions in notion, while only one description in the second, but four *likes* were indicated in the latter. Then, the average of the behavioral responses in the first experiment was 22.1 and the behavioral responses described in the collaborative approach were 24 with a total of 27 *likes*. The other symbol analyzed in the first experiment was *search*. It had an average of 1.0 notion and 3.2 behavioral responses. In the collaborative approach, one notion with three *likes* was described, and six behavioral responses with two *likes* were described (Table 7). Apart from the number of descriptions (which are not so different), in general, the descriptions are quite similar, expressing the same concepts. Then, it is difficult to include the number of *likes* in the quantitative analysis because they are not strictly new descriptions, but we are analyzing the possibility of their being considered as such in some way.

We discovered some interesting findings. The first is the low number of *likes* indicated. The average is 1.5 *likes* per description. Although the participants could

	First experiment		Second experiment	
	Notion	Average Behavioral responses	Notion (Likes)	Behavioral responses (Likes)
Client	3.7	22.1	1 (4)	24 (27)
Search	1.0	3.2	1 (3)	6 (2)

Table 7 – Comparison of the number of descriptions between the first and second experiment.

have given *like* to every description in an indiscriminate way, they only expressed *like* as a meaningful sign of approval. Another interesting finding is that the merging symbol operation was not performed. The participants contributed to the definition of the symbols revising the previous before adding a new one. Since the participants added symbols with at least one description, they facilitated the revision of the previously defined symbols to determine if a new one was needed or if those already defined could be used. The last interesting remark is related to the expression removal operation. Although no expressions were removed, in one of the symbols (the state *cancelled*) somebody added an initial description "No description" where there was in fact a previous description. The "No description" definition had two *likes*, which would suggest that the function *dislike* might be needed.

4.7 Conclusions of the second experiment

In summary, we analyzed the symbols identified in the second experiment in comparison with those identified in the first and we found that the second experiment, using the collaborative approach, identified the same symbols which were most recurrently identified in the first one. Then, a comparison of the descriptions of the two main symbols led to a similar conclusion. Thus, a LEL produced by the proposed collaborative approach is a coherent and consistent LEL which summarizes the most important symbols and descriptions obtained by groups working in the same domain.

4.8 Threats to validity and limitations

Since the designs of both experiments were quite similar, we discussed the threats to validity of both experiments together. Wohlin et al. [WRH+12] group validity threats into four categories: conclusion, internal, construct and external validity. Concerning the conclusion category, one possible threat is heterogeneity of subjects. The participants were heterogeneous in terms of experience in industry and roles, but participants were homogenous in the sense that they were mainly from Argentina and they had the experience of working in similar contexts. The second category of threats to analyze is internal validity. Instrumentation is a threat to internal validity that we intended to tackle. For that purpose, we paid special attention to use real applications in the experiment. The maturation threat was also dealt with because although the experiments lasted at most five weeks, the participants needed little time per day to describe the LEL. In this way, the subjects did not have time to get bored with or tired of the experiment. According to the construct validity category, we observed that the experiment did not suffer from such threats referred to as hypothesis guessing, evaluation apprehension or experimenter expectancies, because the people only had to produce the LEL and then, we analyzed the results. Sjoberg et al. [SAA+02] state that many threats to external validity are caused by an artificial setting of the experiment. They mention the importance of realistic tasks and realistic subjects. Realistic tasks are concerned with the size, complexity and duration of the tasks involved. Taking this into account, we set up an experiment which had the complexity of a real situation. Realistic subjects are concerned with the selection of subjects to perform the experimental tasks. In order to tackle this threat, we selected practitioners with real experience in Software Engineering. They had a wide range of experience as well as different skills. We discussed three elements in order to determine whether there were limitations or not. First of all, we used a real application to provide the knowledge to produce the LEL and we conclude this is not a limitation since reverse engineering is a common technique in software engineering. Nevertheless, constructing a LEL from a real application helped to unify the language used. Apart from that, we chose a domain well known by all the participants. Although every subject had their own interests to describe some specific part of the domain (as the first experiment showed), we think that this condition led the people to be focused on the same group of symbols in the second experiment. Finally, we think that the fact that participants knew the objectives of the experiments had no influence in the results. In the first experiment, the participants did not know what we wanted to assess, since they were only asked to produce a LEL. And in the second experiment, we think that the participants were focused and thought carefully in order to make valuable contributions, because they knew that these would be available to all the participants.

4.9 Conclusions

In conclusion, the first experiment was intended to test the results of people working separately. We wanted to know if they could obtain a common core of symbols because we were afraid that the collaborative work could produce a bias to the participants and condition their contributions. The results of the first experiment showed that the participants working separately produced a common core of symbols which was also present in the second experiment, in which the collaborative approach was actually applied. In the first experiment, there was a wider variety of symbols described, while in the second experiment people were focused. We think that there was no bias, and the participants made the effort to make valuable contributions. In these experiments the domain was known by all the participants and the source of information to build the LEL was a real application and the participants succeeded to produce a coherent and consistent LEL collaboratively.

5 Related Works

Collaboration is used in a wide range of stages in software development, e.g. in UI design [HB10], architectural design [GRW12] and coding [PBL13]. Even in the process of locating (adapting) global systems [EWBS09]. Collaboration is also related with iterative processes. For example, Sutcliffe [Sut10] proposes iterative cycles of requirements analysis and design exploration to build mutual understanding about user requirements and the space of possible solutions for those requirements. Sometimes, there is some kind of collaboration when different people work at different products independently and, after that, there is an integration of the results. For example, Ge et al. [GYYW09] propose an approach to develop a multi-functional product, in which the product is divided into different scenarios by its function, and different stakeholders are divided among scenarios to participate in the process of the requirements elicitation.

The LEL is a glossary whose goal is to capture the language of the application domain. Thus, we can consider the LEL to be a domain model different from a data model, since the LEL only captures the language of the domain needed prior to the requirements step. Thus, this section discusses some related works in requirements engineering and collaborative modeling.

Collaboration is used in different stages of requirements engineering. Sourour et al. [YWK+08] propose a collaborative approach to validate requirements. Zhou et al. [ZYL11] propose a collaborative approach to communicate requirements once they have been validated. They deal with Software as a Service (SaaS) and their approach consists in keeping each potential client of an application aware of the requirements raised by other clients or the SaaS vendor and allows a client to vote on existing requirements or raise new requirements. Thus, clients can create and evolve their proprietary requirements model.

Our approach focuses on elicitation and bears strong similarities to other authors' in the requirements engineering step. For example, Azadegan et al. [ACNY13] propose two steps: (i) identifying relevant user requirements and (ii) voting for user requirements. Our work begins with the identification of users, not considered in Azadegan et al.'s approach, but follows with the definition of the model and voting, which is similar to their aproach, because they also define a close set of operations (six tasks) performed in an iterative way. Lutz et al. [LSD12] use role play to involve different stakeholders in a use case analysis. They developed CREW-Space, i.e. a tool to support the co-located collaboration of several users to simultaneously interact through Android-enabled mobile devices with the same model displayed on a shared screen. Their work also begins with the identification and involvement of stakeholders, but they perform both things simultaneously: while we first identify stakeholders and then involve them in the construction of the LEL. Moreover, identification and involvement is the first of three different steps, each of which must be ended before the next begins. Thus, they do not provide an iterative approach like ours. Finally, although they work with different products (CRC cards), the operations they provide are similar to ours.

Bendix et al. [BE09] report their experience in working with Use Cases in a collaborative way. Their Use Cases are described in a textual way and although they have different attributes from those of the LEL, there exist some similarities between Use Cases and LEL symbols. The difference with our approach however, is that Bendix et al. propose to work first in isolation and then putting the Use Case under control. We do not consider defining LEL symbols in isolation, since all the work is done collaboratively. They also provide a merging operation to integrate the work in isolation. In our case, we perform this operation to merge synonyms. Another aspect to be pointed out is that they also rely on a traceability operation due to their isolated way of working.

De Sosa et al. [SDT11] propose an approach to define domain specific language (DSL) expressions collaboratively. DSL expressions are very similar to LEL symbols. Nevertheless, their work emphasizes the need to define dependencies between the elements and viewer-aware editors. The main reason is the need to maintain the DSL coherent and consistent.

Liu et al. [LZS+06] present the technical challenges and a solution in providing collaboration capabilities in software modeling. They define a set of primitive operations: insert, delete and update. Our operations can be reduced to theses operations but we also provide a way of verifying the contribution with the *like* tool.

Collaborative filtering is a technique for filtering large sets of data for information and patterns. This technique is used in recommender systems to forecast a user's preference. The underlying assumption is that users who have had a similar taste in the past will share a similar taste in the future [LF12]. We are not interested in predictions because, since we need to capture the language, we must not bias the creativity process.

Dheepa et al. [DVN13] introduce a novel method that uses social networks and collaborative filtering to manage the requirements elicitation process. They use the snowball method to build the user network and prioritize requirements. And they use collaborative filtering to make predictions about requirements. There are other works in the same line. For example, Lim et al. [LQF10] prioritize stakeholders on three attributes: the power to influence the project, the legitimacy and the urgency of their claims. Lim et al. [LF12] [LDF11] and Dheepa et al. [DAV13] prioritize their requirements using their ratings weighted by their project influence derived from their position on the social network. We also use the snowballing technique, but we do not consider the relevance of the users, because we consider that all the votes are equally relevant. The reason for this is that the aim of our approach is to capture the language of the domain, so the symbols identified by all the people who participate in the process must be taken into account to capture the language. Azme et al. [AMJ14] also consider users. They exploit relationships between stakeholders and past requirements to break the whole set of new requirements into meaningful categories. This distinction is not useful in our approach.

Reenadevi et al. [RD12] propose a strategy to identify malicious stakeholders, since requirements rated by malicious stakeholders affect the quality of the product. We believe that malicious stakeholders will not affect the LEL description, because there is no prioritization similar to requirements prioritization. We think that the biggest challenge in the description of a glossary is involving the participants. Duarte et al. [DFdSdS12] faced a similar problem and they propose a technique to involve stakeholders during requirements elicitation through the support of online collaboration and the usage of visualization techniques to stimulate stakeholders and increase their awareness about requirements. We should try to incorporate the techniques they use to stimulate stakeholders in a LEL definition. Damian et al. [DMK07] [DKM10] use social network analysis to study collaboration, communication and awareness among stakeholders. Social network measures, such as degree centrality and betweenness centrality, were used to analyse the collaboration behaviour. We do not perform this analysis, but we have this information, and shall include it in a future work.

Some works propose tools instead of a specific technique. For example, Kukreja [Kuk12] propose Winbook a tool for collaborative requirements elicitation and management. Winbook combines tools provided for the most common social network applications and integrates tools of the WinWin negotiation framework. This proposal is mainly oriented to negotiation while our work relies on collaboration. Nevertheless both approaches agree on the *like* tool and we think that our approach will benefit from the *dislike* tool that they propose. Sateli et al. [SAW13] present Reqwiki, an open source semantic wiki that includes natural language processing (NLP) assistants, which work collaboratively with humans on the requirements specification documents. This work shows the importance of semantic and natural language; that is why we propose beginning with the glossary before writing requirements, and we consider it is important to enrich our approach with some NLP technique to help stakeholders to identify and describe symbols. Wen et al. [WLL12] also rely on semantics. They propose a collaborative requirements elicitation approach which produces a requirements-semantics concept by collective participation. This approach is similar to ours since they also provide operations to acquisition and verification. During the acquisition, stakeholders define requirements and the verification occurs when they tag the requirements to semantic description performed by the requirements engineer. In our approach, we only deal with the semantics defined by the stakeholders and we let the stakeholders verify definitions by themselves. Wen et al. also use semantic wikis technology as requirements authoring platform. Yang et al. [YWK+08] propose the EasyWinWin, a tool that was proven to be very good at capturing initial requirements involving heterogeneous stakeholders. However, it was less easy to use in updating requirements and related information as a project proceeds. Thus, they developed an initial version of a WikiWinWin [WYB10] since people find that wikis are easier to learn and use, and are able to organize information in a flexible and updatable manner. The LEL also has the advantage of being easy to learn and use, and allows the stakeholders to organize the information in a flexible and updatable manner. Moreover, the LEL could be recorded in a wiki but there are some consistencies check required from a specific tool. Ajmeri et al [ASG10] found that community-oriented social software is very useful in the context of software engineering in general and requirements engineering in particular. They emphasize that while the benefits of social platforms are valuable, they are necessary but not sufficient for making the exercise effective. Semantics is an important issue to enable knowledge-assisted agile requirements definition. Their ease of use, transparency of communication, user orientation, self-organization and emergent nature resulting from a continual social feedback are particularly relevant to an agile requirements definition exercise. We agree with the importance of semantics and we are working on extending our approach to support agile requirements.

6 Conclusions and Future Works

Capturing knowledge from an application domain can be very disappointing if we do not involve the necessary number of stakeholders. But involving too many people to construct the LEL can be difficult to manage using the traditional technique, in which one or a group of requirements engineers must analyze, organize and describe the information elicited from a group of people. Thus, in order to cope with this problem, we developed an approach to capture the language of the application domain in a collaborative way. This approach allows involving many stakeholders who work collaboratively. We made some preliminary experiments that showed that the LEL produced by our approach was a richer and more complete LEL than each of the ones produced separately. The experiments also showed that the LEL obtained by our approach was smaller and focused more on the most relevant symbols. Thus, although the results of the preliminary experiments are positives we have to continue our experimentation. We are designing a new experiment intended to corroborate our findings considering different application domains, and also to study further issues such as the inclusion of the role of the moderator [Vuk09] [DRR+07] in order to focus even more on the collaborative work, the social function *dislike* and the possibility of removing both *likes* and *dislikes*. The application domain we used in our experiments was well known by all the participants. We think that in complex domains, the repetition of symbols and the identification of synonyms could be an issue. In this situation, we think that the role of the moderator is important since he plays the role of a person with a global vision of the application domain description through the LEL and can suggest that two symbols should be analyzed by the participants of the collaborative construction to determine whether they are really synonyms or not. In this kind of complex domain, the moderator would also help to determine if the lack of *likes* on a symbol means that people disapprove of those definitions or they simply do not know the symbol. In these situations, we think that the *dislike* tool would be important to differentiate disapproval from lack of knowledge. Finally, we believe that it is important to build a specific tool to support the process proposed. Although there are some general tools like wikis that can be used to support this collaborative process, we are designing another one for this specific purpose. We think that the tool must check some consistency. For example, subject and object descriptions of behavioral responses must be coherent. Then, incorporation and removal of symbols also demand the revision of all the description to maintain consistency.

References

[Ack74] Russell L. Ackoff. *Redesigning the Future, a Systems Approach to Societal Problems*. Inc, John Wiley and Sons, 1974.

[ACNY13] Aida Azadegan, Xusen Cheng, Fred Niederman, and Guopeng Yin. Collaborative requirements elicitation in facilitated collaboration: Report from a case study. In *HICSS*, pages 569–578. IEEE, 2013. URL: `http://dblp.uni-trier.de/db/conf/hicss/hicss2013.html#AzadeganCNY13`, `doi:10.1109/HICSS.2013.136`.

[AMJ14] Zeina Azmeh, Isabelle Mirbel, and Laila El Jiani. A tool to improve requirements review in collaborative software development platforms. In *IEEE 8th International Conference on Research Challenges in Information Science, RCIS 2014, Marrakech, Morocco, May 28-30, 2014*, pages 1–6, 2014. URL: `http://dx.doi.org/10.1109/RCIS.2014.6861057`, `doi:10.1109/RCIS.2014.6861057`.

[ARO15] Leandro Antonelli, Gustavo Rossi, and Alejandro Oliveros. A collaborative approach to capture the domain language. In *Anais do WER15 - Workshop em Engenharia de Requisitos, Lima, Peru, April 22-24, 2015*, 2015. URL: `http://wer.inf.puc-rio.br/WERpapers/pdf_counter.lua?wer=WER15&file_name=WER15-antonelli.pdf`.

[ASG10] Nirav Ajmeri, Riddhima Sejpal, and Smita Ghaisas. A semantic and collaborative platform for agile requirements evolution. In *Managing Requirements Knowledge (MARK), 2010 Third International Workshop on*, pages 32–40, Sept 2010. `doi:10.1109/MARK.2010.5623810`.

[BE09] L. Bendix and P. Emanuelsson. Collaborative work with software models - industrial experience and requirements. In *Model-Based Systems Engineering, 2009. MBSE '09. International Conference on*, pages 60–68, March 2009. `doi:10.1109/MBSE.2009.5031721`.

[Boe81] Barry W. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.

[Bro95] Frederick P. Brooks. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, anniversary edition, 1995.

[CA07] Betty H. C. Cheng and Joanne M. Atlee. Research directions in requirements engineering. In Lionel C. Briand and Alexander L. Wolf, editors, *FOSE*, pages 285–303, 2007. URL: `http://dblp.uni-trier.de/db/conf/icse/fose2007.html#ChengA07`, `doi:10.1109/FOSE.2007.17`.

[CdPL01] Luiz Marcio Cysneiros and Julio Cesar Sampaio do Prado Leite. Using the language extended lexicon to support non-functional requirements elicitation. In *WER*, pages 139–153, 2001. URL: `http://dblp.uni-trier.de/db/conf/wer/wer2001.html#CysneirosL01`.

[CHM08] Jane Cleland-Huang and Bamshad Mobasher. Using data mining and recommender systems to scale up the requirements process. In *Proceedings of the 2Nd International Workshop on Ultra-large-scale Software-intensive Systems*, ULSSIS '08, pages 3–6, New York, NY, USA, 2008. ACM. URL: `http://doi.acm.org/10.1145/1370700.1370702`, `doi:10.1145/1370700.1370702`.

[DAV13] V. Dheepa, D.J. Aravindhar, and C. Vijayalakshmi. A novel method for large scale requirement elicitation. *International Journal of Engineering and Innovative Technology*, 2(7):375–379, January 2013.

[DFdSdS12] Diogo Duarte, Carla Farinha, Miguel Mira da Silva, and Alberto Rodriguez da Silva. Collaborative requirements elicitation with visualization techniques. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*, pages 343–348, June 2012. `doi:10.1109/WETICE.2012.14`.

[DKM10] Daniela Damian, Irwin Kwan, and Sabrina Marczak. Requirements-driven collaboration: Leveraging the invisible relationships between requirements and people. In Ivan Mistrík, André van der Hoek, John Grundy, and Jim Whitehead, editors, *Collaborative Software Engineering*, pages 57–76. Springer, 2010. URL: `http://dblp.uni-trier.de/db/books/daglib/0023900.html#DamianKM10`, `doi:10.1007/978-3-642-10294-3_3`.

[DMK07] Daniela Damian, Sabrina Marczak, and Irwin Kwan. Collaboration patterns and the impact of distance on awareness in requirements-centred social networks. In *Requirements Engineering Conference, 2007. RE '07. 15th IEEE International*, pages 59–68, October 2007. `doi:10.1109/RE.2007.51`.

[dPAOdPLCC07] Antonio de Pádua Albuquerque Oliveira, Julio Cesar Sampaio do Prado Leite, Luiz Marcio Cysneiros, and Claudia Cappelli. Eliciting multi-agent systems intentionality: from language extended lexicon to i* models. In *SCCC*, pages 40–49. IEEE Computer

Society, 2007. URL: http://dblp.uni-trier.de/db/conf/sccc/sccc2007.html#OliveiraLCC07, doi:10.1109/SCCC.2007.8.

[dPLF93] Julio Cesar Sampaio do Prado Leite and Ana Paula M. Franco. A strategy for conceptual model acquisition. In *RE*, pages 243–246. IEEE, 1993. URL: http://dblp.uni-trier.de/db/conf/re/icre1993.html#LeiteF93, doi:10.1109/ISRE.1993.324851.

[DRR+07] Björn Decker, Eric Ras, Jörg Rech, Pascal Jaubert, and Marco Rieth. Wiki-based stakeholder participation in requirements engineering. *IEEE Software*, 24(2):28–35, Mar/Apr 2007. doi:http://doi.ieeecomputersociety.org/10.1109/MS.2007.60.

[DVN13] V. Dheepa, C. Vijayalakshmi, and E. R. Naganathan. A novel method for large scale requirement elicitation. *International Journal of Advanced Computational Engineering and Networking*, 1(1):7–12, March 2013.

[EWBS09] Chris Exton, Asanka Wasala, Jim Buckley, and Reinhard Schaler. Micro crowdsourcing: a new model for software localization. *The International Journal of Localisation*, 8(1):81–89, 2009.

[FGH+94] Anthony C. W. Finkelstein, Dov Gabbay, Anthony Hunter, Jeff Kramer, and Bashar Nuseibeh. Inconsistency handling in multi-perspective specifications. *IEEE Trans. Softw. Eng.*, 20(8):569–578, 1994. URL: http://portal.acm.org/citation.cfm?id=631143&dl=&coll=GUIDE&CFID=54331563&CFTOKEN=81075447, doi:http://dx.doi.org/10.1109/32.310667.

[GFO00] Gustavo D. Gil, Daniel Arias Figueroa, and Alejandro Oliveros. Producción del lel en un dominio técnico. informe de un caso. In *WER*, pages 53–69, 2000. URL: http://dblp.uni-trier.de/db/conf/wer/wer2000.html#GilaFO00.

[Gro13] Standish Group. The chaos report, 2013. URL: http://blog.standishgroup.com/.

[GRW12] Phil Greenwood, Awais Rashid, and James Walkerdine. Udesignit: Towards social media for community-driven design. In Martin Glinz, Gail C. Murphy, and Mauro Pezzé, editors, *ICSE*, pages 1321–1324. IEEE, 2012. URL: http://dblp.uni-trier.de/db/conf/icse/icse2012.html#GreenwoodRW12, doi:10.1109/ICSE.2012.6227089.

[GYYW09] Chang Ge, Suihuai Yu, Gangjun Yang, and Weiwei Wang. A collaborative requirements elicitation approach based on scenario. In *Computer-Aided Industrial Design Conceptual Design, 2009. CAID CD 2009. IEEE 10th International Conference on*, pages 2213–2216, Nov 2009. doi:10.1109/CAIDCD.2009.5375171.

[HB10] Jeffrey Heer and Michael Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In Elizabeth D. Mynatt, Don Schoner, Geraldine Fitzpatrick, Scott E. Hudson, W. Keith Edwards, and Tom Rodden, editors, *CHI*, pages 203–212. ACM, 2010. URL: http://dblp.uni-trier.de/db/conf/chi/chi2010.html#HeerB10, doi:10.1145/1753326.1753357.

[How06] Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6), 06 2006. URL: `http://www.wired.com/wired/archive/14.06/crowds.html`.

[IEE98] IEEE. Ieee recommended practice for software requirements specifications. *IEEE Std 830-1998*, 1998. `doi:10.1109/IEEESTD.1998.88286`.

[KHDdPL00] Gladys N. Kaplan, Graciela D. S. Hadad, Jorge Horacio Doorn, and Julio Cesar Sampaio do Prado Leite. Inspección del lexico extendido del lenguaje. In *WER*, pages 70–91, 2000. URL: `http://dblp.uni-trier.de/db/conf/wer/wer2000.html#KaplanHDL00`.

[KSK14] Jacqueline Konate, Abd El Kader Sahraoui, and Gwendolyn L. Kolfschoten. Collaborative requirements elicitation: A process-centred approach. *Group Decision and Negotiation*, 23(4):847–877, 2014. URL: `http://dx.doi.org/10.1007/s10726-013-9350-x`, `doi:10.1007/s10726-013-9350-x`.

[Kuk12] N. Kukreja. Winbook: A social networking based framework for collaborative requirements elicitation and winwin negotiations. In *Software Engineering (ICSE), 2012 34th International Conference on*, pages 1610–1612, June 2012. `doi:10.1109/ICSE.2012.6227227`.

[LDF11] Soo Ling Lim, Daniela Damian, and Anthony Finkelstein. Stakesource2.0: using social networks of stakeholders to identify and prioritise requirements. In Richard N. Taylor, Harald C. Gall, and Nenad Medvidovic, editors, *ICSE*, pages 1022–1024. ACM, 2011. URL: `http://dblp.uni-trier.de/db/conf/icse/icse2011.html#LimDF11`, `doi:10.1145/1985793.1985983`.

[LF12] Soo Ling Lim and Anthony Finkelstein. Stakerare: Using social networks and collaborative filtering for large-scale requirements elicitation. *IEEE Trans. Software Eng.*, 38(3):707–735, 2012. URL: `http://dblp.uni-trier.de/db/journals/tse/tse38.html#LimF12`, `doi:10.1109/TSE.2011.36`.

[LQF10] Soo Ling Lim, Daniele Quercia, and Anthony Finkelstein. Staknet: Using social networks to analyse the stakeholders of large-scale software projects. In *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ICSE '10, pages 295–304, New York, NY, USA, 2010. ACM. URL: `http://doi.acm.org/10.1145/1806799.1806844`, `doi:10.1145/1806799.1806844`.

[LSD12] Rainer Lutz, Sascha Schafer, and Stephan Diehl. Using mobile devices for collaborative requirements engineering. In *Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on*, pages 298–301, Sept 2012. `doi:10.1145/2351676.2351729`.

[LZS+06] S. Liu, Yang Zheng, Haifeng Shen, S. Xia, and Chengzheng Sun. Real-time collaborative software modeling using uml with rational software architect. In *Collaborative Computing:*

Networking, Applications and Worksharing, 2006. CollaborateCom 2006. International Conference on, pages 1–9, Nov 2006. doi:10.1109/COLCOM.2006.361897.

[MG12] Nilofar. Mulla and Sheetal Girase. A new approach to requirement elicitation based on stakeholder recommendation and collaborative filtering. *International Journal of Software Engineering and Applications*, 3(3):51–60, May 2012. doi:10.5121/ijsea.2012.3305.

[MGvdHW10] Ivan Mistrík, John Grundy, André van der Hoek, and Jim Whitehead. Collaborative software engineering: Challenges and prospects. In Ivan Mistrík, André van der Hoek, John Grundy, and Jim Whitehead, editors, *Collaborative Software Engineering*, pages 389–403. Springer, 2010. URL: http://dblp.uni-trier.de/db/books/daglib/0023900.html#MistrikGHW10, doi:10.1007/978-3-642-10294-3_19.

[Miz83] Y. Mizuno. Software quality improvement. *IEEE Computer*, 16(3):66–72, March 1983. doi:10.1109/MC.1983.1654331.

[NFG+06] Linda Northrop, Peter Feiler, Richard P. Gabriel, John Goodenough, Rick Linger, Tom Longstaff, Rick Kazman, Mark Klein, Douglas Schmidt, Kevin Sullivan, and Kurt Wallnau. Ultra-Large-Scale Systems - The Software Challenge of the Future. Technical report, Software Engineering Institute, Carnegie Mellon, June 2006. URL: http://www.sei.cmu.edu/uls/downloads.html.

[PBL13] Luca Ponzanelli, Alberto Bacchelli, and Michele Lanza. Leveraging crowd knowledge for software comprehension and development. In Anthony Cleve, Filippo Ricca, and Maura Cerioli, editors, *CSMR*, pages 57–66. IEEE Computer Society, 2013. URL: http://dblp.uni-trier.de/db/conf/csmr/csmr2013.html#PonzanelliBL13, doi:10.1109/CSMR.2013.16.

[PRVPB12] Javier Portillo-Rodríguez, Aurora Vizcaíno, Mario Piattini, and Sarah Beecham. Tools used in global software engineering: A systematic mapping review. *Information and Software Technology*, 54(7):663–685, 2012. URL: http://dblp.uni-trier.de/db/journals/infsof/infsof54.html#Portillo-RodriguezVPB12, doi:10.1016/j.infsof.2012.02.006.

[QB09] Alexander J. Quinn and Benjamin B. Bederson. A taxonomy of distributed human computation, 2009.

[RD12] R. Reenadevi and P. Dugalya. Identifying malicious stakeholders using algorithm for large scale requirement-elicitation. *International Journal of Computer and Communication Technology*, 3(6):106–108, 2012.

[SAA+02] Dag I.K. Sjoberg, B. Anda, E. Arisholm, Tore Dyba, M. Jorgensen, A. Karahasanovic, E.F. Koren, and M. Vokac. Conducting realistic experiments in software engineering. In *Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n*, pages 17–26, 2002. doi:10.1109/ISESE.2002.1166921.

[SAW13] Bahar Sateli, Elian Angius, and Rene Witte. The reqwiki approach for collaborative software requirements engineering with integrated text analysis support. In *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual*, pages 405–414, July 2013. doi:10.1109/COMPSAC.2013.68.

[SDT11] Josune De Sosa, Oscar Diaz, and Salvador Trujillo. Defining dsl expressions collaboratively in multidisciplinary embedded engineering. In *Proceedings of the 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, SEAA '11, pages 217–220, Washington, DC, USA, 2011. IEEE Computer Society. URL: http://dx.doi.org/10.1109/SEAA.2011.41, doi:10.1109/SEAA.2011.41.

[SN11] Yasuaki Sakamoto and Jeffrey V. Nickerson. Evaluating design solutions using crowds. In Vallabh Sambamurthy and Mohan Tanniru, editors, *AMCIS*. Association for Information Systems, 2011. URL: http://dblp.uni-trier.de/db/conf/amcis/amcis2011.html#SakamotoN11.

[Sut10] Alistair Sutcliffe. Collaborative requirements engineering: Bridging the gulfs between worlds. In Selmin Nurcan, Camille Salinesi, Carine Souveyet, and Jolita Ralyte, editors, *Intentional Perspectives on Information Systems Engineering*, pages 355–376. Springer Berlin Heidelberg, 2010. URL: http://dx.doi.org/10.1007/978-3-642-12544-7_20, doi:10.1007/978-3-642-12544-7_20.

[SV06] Thomas Stahl and Markus Völter. *Model Driven Software Development: Technology, Engineering, Management*. Wiley, 2006.

[SZ11] M.D. Sourour and N. Zarour. A methodology of collaborative requirements validation in a cooperative environment. In *Programming and Systems (ISPS), 2011 10th International Symposium on*, pages 140–147, April 2011. doi:10.1109/ISPS.2011.5898877.

[Vuk09] Maja Vukovic. Crowdsourcing for enterprises. In *SERVICES I*, pages 686–692. IEEE Computer Society, 2009. URL: http://dblp.uni-trier.de/db/conf/services/services2009-1.html#Vukovic09, doi:10.1109/SERVICES-I.2009.56.

[WLL12] Bin Wen, Ziqiang Luo, and Peng Liang. Distributed and collaborative requirements elicitation based on social intelligence. In *Web Information Systems and Applications Conference (WISA), 2012 Ninth*, pages 127–130, Nov 2012. doi:10.1109/WISA.2012.14.

[Woo97] Larry E. Wood. Semi-structured interviewing for user-centered design. *Interactions*, 4(2):48–61, 1997. URL: http://dblp.uni-trier.de/db/journals/interactions/interactions4.html#Wood97, doi:10.1145/245129.245134.

[WRH+12] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer, 2012. URL: http://dx.doi.org/10.1007/978-3-642-29044-2, doi:10.1007/978-3-642-29044-2.

[WTL13] Wenjun Wu, Wei-Tek. Tsai, and Wei Li. Creative software crowdsourcing: from components and algorithm development to project concept formations. *International Journal of Creative Computing*, 1(1):57–91, 2013. doi:DOI:10.1504/IJCRC.2013.056925.

[WYB10] Di Wu, Da Yang, and Barry Boehm. Finding success in rapid collaborative requirements negotiation using wiki and shaper. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10, Jan 2010. doi:10.1109/HICSS.2010.210.

[YWK+08] Da Yang, Di Wu, S. Koolmanojwong, A.W. Brown, and B.W. Boehm. Wikiwinwin: A wiki based system for collaborative requirements negotiation. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 24–24, Jan 2008. doi:10.1109/HICSS.2008.502.

[ZYL11] Xin Zhou, Li Yi, and Ying Liu. A collaborative requirement elicitation technique for saas applications. In *Service Operations, Logistics, and Informatics (SOLI), 2011 IEEE International Conference on*, pages 83–88, July 2011. doi:10.1109/SOLI.2011.5986533.

About the authors

Leandro Antonelli is professor of Project Management at Universidad Nacional of La Plata. And he is also Teaching Assistant in Data abstraction and algorithm analysis and Object oriented programming courses. His interest area is Requirements Engineering. Contact him at leandro.antonelli@lifia.info.unlp.edu.ar.

Gustavo Rossi is Full Professor at Universidad Nacional de La Plata, Argentina. He holds a PhD from PUC-Rio. His interest areas are: Web and Requirements Engineering and Human Computer Interaction. Contact him at gustavo@lifia.info.unlp.edu.ar.

Alejandro Oliveros is Professor at Universidad Nacional de Tres de Febrero and Universidad Argentina de la Empresa, both of Argentina. His interest area is Requirements Engineering. Contact him at aoliveros@gmail.com.