I need a title

Daniel Paredes

FH Dortmund,

Abstract. Interviews are the most common technique in requirements elicitation. However, the presence of tacit knowledge have an important influence on interviews success. In this paper, we describe a tacit knowledge classification. Then, we make use, with examples, of a theoretical framework to detect and minimize the effects of tacit knowledge. Finally, we analyze how likely is to have a successful elicitation interviews based on the tacit knowledge classification.

Keywords: interview, tacit knowledge, elicitation

1 Introduction

Requirements engineering is the subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction [1]. The first step of requirements engineering is requirements elicitation which is defined as the process of finding and formulation requirements [2], and this process has many barriers due to stakeholders are not able to express their needs, explain what they do and why, conflicting demands, or new demands once other are met, just to mention some examples.

One of the most used techniques to elicit requirements is open interviews, and also is considered the most effective way to transfer domain knowledge [3]. However, the success of interviews depends on mainly two factors, analyst set of interviewing skills and stakeholders domain knowledge; nonetheless, statistically analyst domain knowledge is important but not as relevant as the other two[4]. Therefore, analyst should make use of tools and techniques to detect and elicit tacit knowledge form stakeholders. Tacit knowledge is in requirements engineering is defined as the knowledge that a stakeholders has, but does not pass to the requirements analyst [5].

In this paper we describe a classification of tacit knowledge in elicitation interviews based on [6]. Then we apply the theoretical framework described in [3] in examples to show how to deal with tacit knowledge. The examples were selected in order to match the each scenario of the classification. Finally, we use the result from [4] to identify in which scenario is most likely to perform a successful interview.

The remainder paper is structured as follows. In section 2 we provide a summary of the papers used to write this paper. In section 3 we present our approach. Finally, in section 4, we present our conclusions.

2 Background

2.1 Requirements elicitation: Towards the Unknown Unknowns [6]

Elicitation in requirements engineering is still problematic because missing or mistaken requirements are hard to elicit, and this produce projects delays and have financial implications. In this paper, the authors propose an elicitation review framework (ERF) in order to explore the different challenges related to the "unknowness" of the domain knowledge. Therefore, proposing a road map of research to tackle the different levels of the "unknowness".

The authors made use of the properties of *expressible* (known knowledge), *articulated* (documented domain knowledge), *accessible* (need memory recall), *relevant* (to the project). With these four properties, authors defined the following levels of unknowness in domain knowledge.

- Known knowns: knowledge that is expressible, articulated, and relevant.
- Known unknowns: knowledge that is not expressible or articulated, but accessible and potentially relevant.
- Unknown knowns: knowledge that is potentially accessible but not articulated.
- Unknown unknowns: knowledge that is not expressible, articulated nor accessible but may be relevant.

This levels of knowledge implies problems of different perspective of the elicitation process that involves analysts and stakeholders. The *known knowns* is a simple scenario since the counterparts are aware of the domain problem which is expressible and articulated. In the case of the *known unknowns*, the analysts has a domain knowledge and they should be able draw out the information from the stakeholders that they might be unaware of. In contrast, in the *unknown knowns* scenario the stakeholders are the ones that are aware of the domain knowledge but for some reason it is no articulated, and therefore, the analysts once they notice any glimpse of the stakeholders information they should able to discover and elicit that. On the other hand, in the case of *unknown unknowns* both counterparts are unaware of the missing information that might be relevant to the system.

Based on the perspectives described before, the authors identified three challenges to requirement elicitation. The first one is to identifying tacit knowledge (unknown knowns), even in the case the analyst presume it exist (known unknowns). The second challenge is that the analyst should know what is relevant and should be

articulated. And the last one is the articulation of the knowledge. Thus, in the worst case scenario the goal is go from the unknown unknowns perspective to known knowns.

Finally, the authors suggest four research directions to deal with the unknowness problem. Starting with the known unknown in which the analyst should challenging assumptions, implications, and relaxing domain constrains in order to increase the probability to elicit the unaware knowledge of the stakeholders. In the same direction, in the design discovery, which is a variant of the latter case, the analyst should deal with the the following statement "I'll know what I want when I see it". Thus, there is a need of prototyping or simulations as part of the elicitation process. In the case of the unknown knowns in which the analyst can elicit tacit knowledge by taking into account the background (political, cultural, emotional) of stakeholders, and use that background as an "emotional guidance" in order to draw out the relevant information. Finally, in the case of unknown unknowns the authors suggest to main approaches. The first one is based on the "over-the-horizon" knowledge, in which authors proposed a socio-technical approach based on social media and e-communities. And the other one is using counter examples that can challenges the boundaries of the idea under development.

2.2 Using Argumentation to Explain Ambiguity in Requirements Elicitation Interviews [3]

One of the major causes of ambiguities in elicitation interviews is the presence of tacit knowledge. In some cases ambiguities can't be always be explained as separated term, sometimes it is required to have a context. Even under these circumstances, the analyst must be able to identify and alleviate them in order to elicit relevant information of the system. For this purpose it is necessary to provide the analyst with proper tools. In this paper Yehia Elrakaiby et al. proposed a theoretical framework to overcome ambiguity during interviews in the elicitation phase. The framework is based on the "Argumentation theory".

In that sense, Elrakaiby et al. focus on one type of ambiguities, the "acceptance unclarity". An acceptance unclarity occurs every time the analyst is able to assign an interpretation or meaning to the speech fragment of the stakeholder, the interpretation matches the intended meaning of the stakeholder, but the interpretation is not acceptable or justified. It could be either because it seems to be inaccurate to comprehend the problem, or analysts identify inconsistencies with their current understanding of the problem or domain knowledge. By using argumentation theory framework, statements and ambiguities can be characterized as "arguments" and "attacks" respectively.

Argumentation theory models a type of human dialog based on arguments and conclusions. It makes explicit attacks between arguments and the argumentation flow that leads to conclusions. A basic model in this framework is a pair (A, D),

where A is a set of arguments and D is a set of attacks among those arguments. For example, a set A is defined as $A = \{A1, A2, A3\}$, and a possible set of attacks could be $D = \{(A1, A3)\}$. Which means that if A1 is realizable then A3 can't be realizable.

In the paper, Elrakaiby et al. models statements given by the stakeholders, analysts domain knowledge and analysts inferences as arguments, and ambiguities between them as attacks. For example, let say the analyst listens the following statement *the professor will upload the task description within three days* (A1), but the analyst know (domain knowledge) that *the professor may be on a business meeting* (A2), so the analyst think (inference) that *it may be possible that it will be take longer to upload the task description* (A3). In this scenario the set of attacks D is given by $D = \{(A1, A3)\}$. Thus, since there is an attack it is possible to ask for clarifications or details.

The theoretical framework proposed by Elrakaiby et al. allows analysts to detect and minimize ambiguities during elicitation interviews, while most of the methods that focus on ambiguities analyze written texts. On the other hand, this framework focuses in more complex ambiguities that cannot be view as single terms.

2.3 Effect of Domain Knowledge on Elicitation Effectiveness: An Internally Replicated Controlled Experiment [4]

The effectiveness of elicitation interviews may be influenced by analyst skills or characteristics. In these high intensive oral communication scenario the analyst must be able to draw out relevant information and needs from the stakeholders. It has been reported that the effectiveness of the interviews has a direct relation with the domain knowledge of the analyst. However, there are also studies suggesting that in some cases the domain knowledge have negatives effects in the effectiveness of interviews.

In this paper Aranda et al. studied the influence of the analyst domain knowledge on the effectiveness of elicitation interviews. The main question they tried to answer was

> *Does analyst domain knowledge influence (either positively or negatively) the effectiveness of the requirements elicitation activity?*

For this purpose, the authors performed the study in two stages. They performed an initial baseline experiment with two domain problems, and then they performed an internal replication with two other domain problems. Furthermore, the authors divided the elicitation process in two phases. The elicitation phase which is the actual interview with the stakeholder, and the reporting phase in which the analyst understands and documents the information gathered in the elicitation phase.

One remark of the study is that the authors chose students because of their lack of experience in elicitation interviews, isolation of the domain knowledge, and to analyze, in the internal replication, the influence of the training in requirements engineering in elicitation interviews. In the study participated post-graduated students as interviewers and two professors as interviewees. The students should make open interviews and elicit the information afterwards. Moreover, for each domain problem the students were separated in two groups based on their level of domain knowledge, *domain-aware* and *domain-ignorant*. On the other hand, the effectiveness of the elicitation's was based on the comparison between the number of concepts, processes and requirements elicited by the students and the benchmarks.

The results of the baseline experiment suggest that the domain knowledge of the analysts has no significant influence in the effectiveness of the elicitation interviews. However, the results also suggest that the domain knowledge of the interviewees has statistically significant influence. Supplementary, the results of the internal replication also suggest that the domain knowledge of the stakeholders is more relevant than the analysts', in term of effectiveness of the elicitation interviews. Nevertheless, these results also show that the positive effects of the training in requirements engineering of the interviewers, with these effects being as relevant as interviewees' domain knowledge.

3 Tacit knowledge in elicitation interviews

3.1 Types of tacit knowledge

Many analyst consider interviews the most important elicitation technique, and usually elicitation process starts with interviewing the stakeholders. Interviews allow analysts to check their understanding about the problem domain immediately and ask for clarifications. Moreover, this technique is versatile in the sense of it is possible that new and unexpected issues come up and they can be attacked at that moment.

However, elicit tacit knowledge in interviews is still a hard task to do. This tacit knowledge can be classified in four groups as describe in [6]. The **known knowns**, **known unknowns**, **unknown knowns** and **unknown unknowns**. The known knowns is the knowledge that is documented, expressible, and relevant to the system. In order words, is the scenario in which it is possible to elicit since the beginning all the requirements, there is no tacit knowledge. The known unknowns is the knowledge that cannot be expressible because the stakeholders are not aware of their lack of domain knowledge, their might forget it. In this case analyst are aware of the domain knowledge and therefore, their job is to challenge assumptions or implications. The unknown knowns is the knowledge that stakeholders hold but it is not documented by some reason. Thus, analysts should use their interviewing skill to elicit when they spot any glimpse of that

knowledge. The unknown unknowns is the knowledge that neither the analysts nor stakeholders are aware.

3.2 How to deal with tacit knowledge

The unknowness clarification presented in above suggest the presence of three problems. The goal is to go from any scenario in which tacit knowledge exists to known knowns. In order to achieve that goal and take the advantages presented in elicitation interviews (immediately clarification of assumptions and debate of implications) we are going to use the theoretical framework described by Elrakaiby [3]. In this framework we model the interview as a set of arguments and attacks between them. In that sense, we will able to spot ambiguities and tacit knowledge in terms of inconsistencies and insufficiency of information. We will adapt some examples from [3] to show how each tacit knowledge scenario can be addresed.

known unknowns: In the following example the domain knowledge of the analyst is used to detect inconsistencies (attacks), because the stakeholder didn't take into account relevant information about domain application (location of trash bins).

A customers wanted a recycling-support system and one of the goals was *to avoid fines from the municipality for incorrect recycling*. According to the domain knowledge of the analyst, *trash bins are placed along the streets*. Thus, the goal was inconsistent with the domain knowledge of the analyst because *there is no way to trace back person who brakes the recycling rule*. This scenario can be modeled using arguments and attacks as follows:

The arguments: (A1)trash bins are placed along the streets, (A2)since trash bins are placed along the streets, the trash owner cannot be traced back. (B1)people who do not recycle should be fined (B2)to fine people, the municipality must be able to trace products in trash bins back to their owners.

Attacks: A2 attacks B1, and vice versa.

unknown knowns: In the following example analyst challenges the stakeholder statement, as a result the stakeholder reveals new information (reply).

A customers wants to develop a system to allow patients to measure the amount of glucose in their blood, and then send the result to their general practitioner. If the glucose level is above a certain threshold, the practitioner pays a visit to the patient. The customer said: *in one-two days the doctor sees the notification in the system*. Then analyst asked: *What happens if the doctor is not available for any reason?*. The customer replied: *The general practitioner is substituted by another doctor who accesses the same system*.

The arguments: (A1)in one-two days the doctor sees the notification,(B1)the doctor might be unavailable,(B2)since the the doctor might be unavailable, the notification might be delayed.

The attacks: A1 attacks B2, and vice versa

unknown unknowns: In the following example the stakeholder didn't notice the inconsistency between his statements, but the analyst did and asked for clarification. However, the stakeholder couldn't give an answers due to his ignorance.

One of the customers wants to develop an app to manage medical-related reservation in Tuscani. The costumer mentioned that the *current reservation system, based on phone calls, was centralized*. During the interview, the customer said that *depending on where the examination/visit will be, the patient has to call Nottola or Siena*. The analyst could not understand how this realization was possible since it means the reservation system is not centralized. Therefore the analyst asked for clarification and the stakeholder couldn't give any acceptable answer.

The arguments: (A1)the current reservation is centralized, (B1)depending on where the examination/visit will be (B2)the patience has to call Nottola or Siena, (B3) since it is possible to call either Nottola or Siena the system is not centralized.

The attacks: A1 attacks B3, and vice versa

3.3 Limits of elicited knowledge

In the last section we present how to use a theoretical framework based on arguments and attacks to addressed the different levels of tacit knowledge. In this section we will compare in which scenario of tacit knowledge the elicitation interview would be more effective based on Aranda et al. work [4] about the effect of domain knowledge on elicitation process.
Effectiveness is described in [4] as the total percentage of problem domain elements identified compared to a benchmark. However, since we have no access to any requirement data, we will use the conclusions of [4] to compare the different scenarios of tacit knowledge (known unknowns, unknown knowns and unknowns unknowns).

The works presented in [4] concludes that: (1)analyst's domain knowledge(ADK) has small but statistically significant effect on the effectiveness, (2)stakeholders' domain knowledge(SDK) has a big positive and significant influence and (3)analyst's interviewing skill and general training in requirement activities have also a big positive and significant influence. In this paper, we will focus only in comparing based on the domain knowledge holder since that knowledge can be measured.

Table 1: Influence of domain knowledge in the different tacit knowledge scenarios

scenario	ADK	SDK
known unknowns	+	0
unknown knowns	0	++
unknown unknowns	0	0

We can observed in Table 1 that it is more likely to be effective in the scenario of *unknown knowns* because the stakeholders' domain knowledge have more influence than analyst's domain knowledge. On the other hand, the analyst's interviewing skills will have different approaches in the case of *known unknowns* and *unknown knowns*. In the first case, the interview will be more focused on challenging assumptions and implications, and in the latter case on spotting glimpses of stakeholders' hidden knowledge.

4 Conclusions

Elicitation interviews is not an easy task. Like in any intensive oral communication there are misunderstandings related to tacit knowledge. In our work we focus on analyzing the different levels of how tacit knowledge[6] would affect the effectiveness of elicitation interviews. Moreover, we use the theoretical framework described in [3] to identify and manage tacit knowledge during interviews. And finally, we analyze based on the results of [4] in which scenario of tacit knowledge it is more likely to have a successful elicitation interview.

References

[1] J. Dick, E. Hull, and K. Jackson, *Requirements engineering*. Springer International Publishing, 2017.

[2] S. Lauesen, *Software requirements: Styles and techniques*, 1st ed. Pearson Education, 2001.

[3] Y. Elrakaiby, A. Ferrari, P. Spoletini, S. Gnesi, and B. Nuseibeh, "Using argumentation to explain ambiguity in requirements elicitation interviews," in *Requirements engineering conference (re), 2017 ieee 25th international*, 2017, pp. 51–60.

[4] A. M. Aranda, O. Dieste, and N. Juristo, "Effect of domain knowledge on elicitation effectiveness: An internally replicated controlled experiment," *IEEE Transactions on Software Engineering*, vol. 42, no. 5, pp. 427–451, 2016.

[5] A. Ferrari, P. Spoletini, and S. Gnesi, "Ambiguity as a resource to disclose tacit knowledge," in *Requirements engineering conference (re), 2015 ieee 23rd international*, 2015, pp. 26–35.

[6] A. Sutcliffe and P. Sawyer, "Requirements elicitation: Towards the unknown unknowns," in *Requirements engineering conference (re), 2013 21st ieee international*, 2013, pp. 92–104.