© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 7, No. 2, May-Aug 2016, pp. 49-56. ISSN: 2007-1558.

A Strategy of Requirements Engineering for Informally Structured Domains

Graduated: Karla Olmos-Sánchez
Universidad Autónoma de Ciudad Juárez
Av. del Charro 450
Chihuahua, MX
E-mail:kolmos@uacj.mx

Advisor: Jorge Rodas-Osollo
Universidad Autónoma de Ciudad Juárez
Av. del Charro 450
Chihuahua, MX
E-mail: jorge.rodas@uacj.mx

Abstract. Eliciting sufficient high-quality knowledge from individuals to build a robust and useful Artificial Intelligence or Intelligent System solution is a very time-consuming and expensive activity, especially in domains where the knowledge is informal, partial, incomplete, implicit, tacit and unstructured. Moreover, in order to develop a solution, a systematic way to incorporate the elicited knowledge into a specification is necessary. The goal of this thesis is to develop a strategy oriented to the transfer and transformation of knowledge with the aim of eliciting and structuring the most quantity of domain knowledge, either tacit or explicit; then incorporate it into a specification that covers the needs and expectations of domain specialists. The application of the strategy in real informally structured domain cases provides empirical insights about its usefulness and value.

Keywords: Software Engineering, Requirements Engineering, Informally Structured Domains, Knowledge Engineering, Thesis Summary.

1 Introduction

Artificial Intelligence (AI) and Intelligent Systems (IS) solutions are generally proposed for domains where the knowledge is informal, partial, incomplete, assumed, tacit and unstructured. In this kind of domains, named Informally Structured Domains (ISD) [1], not all concepts and their relations are formally defined, the most of the problems does not have algorithms to obtain solutions, and domain specialists use large amounts of tacit knowledge in order to solve problems. According to Polany [2], tacit knowledge is personal and context-specific knowledge, generated by experience and therefore, difficult to communicate and formalize. Therefore, eliciting sufficient high-quality knowledge from individuals to build a robust and useful AI or IS solution is a very time-consuming and expensive activity.

Eliciting knowledge implies the process of transferring and transforming knowledge, which is not a trivial task because the knowledge holder must transform it into natural language and non-verbal channels of human communication in order to transfer it to another person, who decodes this information according to their own mental model [3]. The development of AI or IS solutions should consider tree additional factors that make this process even more difficult. The first one is the presence of multiple domain specialists with different backgrounds, perspectives, interests and expectations, and whose knowledge on the domain and the characteristics of the required solution differs, depending on their familiarity and their function in it. The second one is the symmetry of ignorance, a knowledge gap, between solution-solvers and domain specialists, making it difficult the communication process. Finally, the implications of tacit knowledge must be considered [4]. According to Gacitua et al. [5], tacit knowledge can cause critical expectations, knowledge and needs of the domain specialists to remain hidden.

Currently, several methods to acquire and elicit knowledge have been proposed, such as interviews, protocol analysis, concept mapping,... However, applying these methods in isolation is not enough [6]. It is necessary to integrate and amalgamate them into a process, strategy or methodology in order to facilitate the knowledge elicitation process, make it highly effective, minimize the

Received Feb 12, 2016 / Accepted Jun 1, 2016
Editorial Académica Dragón Azteca (EDITADA.ORG)

time spend and turn into explicit the most possible quantity of tacit knowledge. Moreover, in order to develop a solution, the knowledge eliciting process is only the initial step; a systematic way to incorporate it into a specification of a product or solution closest to the needs and expectations of clients is also required. The development of this specification is related to Requirements Engineering (RE), which is a discipline aiming to elicit, analyze, evaluate, consolidate and manage the requirements of a product or solution. Although many areas relative to computer science share the challenges of RE, the major source of RE research comes from software engineering. In this area the critical role of requirements has been recognized for decades because software systems are always embedded in an application domain and their usefulness depends on the problems they can solve and on the objectives they can achieve in those domains [7]. Therefore, this work supports the hypothesis that RE research in the context of software engineering could be used as a starting point to develop a RE solution in the context of ISD.

The goal of this thesis is to develop a RE strategy, a high level plan to achieve a goal under uncertainty conditions, oriented to the transfer or transformation of knowledge, specially designed to be applied in the context of ISD, with the aim of acquiring and structuring the most quantity of knowledge, either tacit or explicit, and incorporate it into a specification that cover the needs and expectations of the domain specialists. The strategy has been named KMoS-RE (Knowledge Management on a Strategy for Requirements Engineering) and it was designed to be applied to several areas such as Artificial Intelligence or Intelligent System solutions, software system development and even other areas that shares similar challenges, such as industrial design.

The aim of this paper is to present a summary of the thesis. The remainder of this document is as follows: Section 2 explains the related works; section 3 describes the proposed solution. Section 4 describes the application of the strategy to real ISD cases. Finally, in section 5 the conclusions and future work are presented.

2 Related Works

The problem of tacit knowledge in RE is not new. Two decades ago, Goguen [8] did an extensive analysis of tacit knowledge in RE from a social perspective. More recently, several authors have studied the effects of tacit knowledge in RE. Their proposals face tacit knowledge problems for different RE tasks, such as elicitation [9], traceability [10] or specification [11]. The works have several scopes such as methodological proposals [9], specific tools [12] or techniques [13]. From the works founded, we should emphasize that of Gacitua et al. [5] because they have a different point of view: instead of proposing means to discover and manage tacit knowledge, the aim of these authors is to research the role of tacit knowledge in RE to mitigate and manage its effects. However, any of these proposals face the tacit knowledge problem from a Knowledge Management perspective.

Generally speaking, Knowledge Management is a discipline with the aim of enhancing an organization by sharing the knowledge among the people and managing its flow. Nonaka and Takeuchi [14] propose the SECI model; a model of knowledge conversion in organizations based on Polany's theory about tacit knowledge. For them, knowledge creation in an organization is the result of social interactions that involves tacit and explicit knowledge. The SECI model postulates four iterative conversion modes: 1) *Socialization*, the process of transferring tacit knowledge between individuals by sharing mental models and technical skills; 2) *Externalization*, the process of converting tacit knowledge to explicit through the development of models, protocols and guidelines; 3) *Combination*, the process of recombining or reconfiguring existing bodies of explicit knowledge to create new explicit knowledge; and 4) *Internalization*, the process of learning by task repetition. Some of these tasks could have been defined by explicit knowledge. Whatever the case, individuals will assimilate the knowledge as tacit once again.

The SECI model has been adopted by some works related to RE. For example, Pilat and Kaindl [15] propose a knowledge management perspective based on the SECI model of creation of knowledge. Although the authors are aware that the problem of sharing knowledge in RE is not new, they suggest that their perspective offers specific insights and techniques for understanding and facilitating the knowledge transfer and transformation process in RE. In other hand, Wan et al. [16] propose a model of knowledge conversion to the requirements elicitation process with the aim of minimizing the symmetry of ignorance between developers and domain specialists. The authors also base their model on the SECI model and consider the flow of knowledge between domain specialists and developers. They introduce a new agent into the elicitation process: the Requirements Specialist. This person will act as an intermediary between the domain specialists and the developers and he or she must earn the trust of those involved in the process. In conclusion, the authors argue that the proposed model can reduce the symmetry of ignorance and facilitate the elicitation of tacit requirements. By last, Vazquez-Bravo et al. [17] propose a classification of elicitation techniques to facilitate their selection based on the phases of the SECI model with the aim of minimizing the difficulties of selecting the appropriate elicitation technique. Although the works explained above share a knowledge management perspective, none of them propose a methodology, strategy or process that addresses the question of how to acquire, represent, transfer and transform the domain knowledge in order to incorporate it into a solution specification.

3 Proposed Solution

The problem that this thesis aims to solve is located in the field of *Knowledge Engineering* and *Requirements Engineering* and has the following characteristics:

– Presence of multiple Domain Specialists (DS) who have different experience, point of view, interests and expectations, and whose knowledge of the application domain varies depending on their involvement and function in the domain. In addition, those domain specialists generally have a vague idea about the product or solution.
– Presence of a group of Solution Solvers (SS) who generally are unfamiliar with the application domain. They have technical knowledge about the solution and must know the solution requirements.
– The Solution (S) has a unique design and solves or addresses a particular situation. The Solution (S) could be a product and must be developed according to a Solution Requirements Specification (SlRS).
– The SlRS is a document that contains the greatest possible amount of correct, appropriate and unambiguous requirements.
– The SlRS development will require great quantities of domain and technical knowledge about the solution.
– In order to develop the SlRS, an Arduous Negotiation Process (ANP) is necessary between the solution-solvers and the domain specialists and even among the domain specialists.

Based on the previous characteristics, the problem can be formulated as follows.

Given:

– $ISD = (DS, SS, KT, KH, K, N_c)$ a well-defined area represented as a sextuplet, where:
  1. $DS = \{ds_1 \ldots ds_m\}$ is a set of *domain specialists ds*, where ds_m represents the value taken by the variable ds in the $m-th$ unit.
  2. $SS = \{ss_1 \ldots ss_n\}$ is a set of *solution-solvers ss*, where ss_n represents the value taken by the variable ss in the $n-th$ unit. A solution-solver is an individual, generally not involved in the domain, with knowledge and experience to propose a suitable solution S to the necessity N_c. The SS members must know the features of the necessity N_c.
  3. $KT = C \cup R$ is the *union set* of concepts and relationships, namely the *knowledge that* where:
     • $C = \{c_1 \ldots c_q\}$ is a set of *concepts c*, where c_q represents the value taken by the variable c in the $q-th$ unit. A concept is knowledge about objects sharing similar properties.
     • $R_{df} = \{r_{df_1}(c_1 \ldots c_k) \ldots r_{df_r}(c_1 \ldots c_k)\}$ is a set of *relationships* r_{df} defined formally, where r_{df_r} represents the value taken by the variable r_{df} in the $r-th$ unit.
     • $R_{dc} = \{r_{dc_1}(c_1 \ldots c_k) \ldots r_{dc_s}(c_1 \ldots c_k)\}$ is a set of *relationships* r_{dc} defined by consensus, where r_{dc_s} represents the value taken by the variable r_{dc} in the $s-th$ unit.
     • $R = R_{df} \cup R_{dc}$ is the *union set* of R_{df} and R_{dc} being a *relationship* a representation of the k concepts in a relationship in the domain, with $k >= 2$.
  4. $KH = B_s \cup B_{ns}$ is the union set of B_s and B_{ns}, namely *knowing how*, where:
     • $B_s = \{b_{s_1} \ldots b_{s_u}\}$ is a set of *situated behaviors* b_s, where bs_u represents the value taken by the variable b_s in the $u-th$ unit. A behavior is a set of observable and measurable interactions; a situated behavior depends on the context and does not have solution algorithms, and therefore depends on the knowledge of the domain specialists to be accomplished.
     • $B_{ns} = \{b_{ns_1} \ldots b_{ns_v}\}$ is a set of *non-situated behaviors* b_{ns}, where b_{nsv} represents the value taken by the variable b_{ns} in the $v-th$ unit. A non-situated behavior has at least one algorithmic solution.
  5. $K = [k_{ij}{}^{\omega}]_{(m+n)t}$ a matrix, $i = \{1 \ldots m+n\}$, $j = \{1 \ldots t\}$, where $m+n$ is the sum of the number of domain specialists plus the number of solution solvers and t is the sum of the number of concepts, relationships defined formally, relationships defined by consensus, situated behaviors and non-situated behaviors, i.e. $t = q+r+s+u+v$ where:
     • $k_{ij}{}^{\omega}$ is the *degree of tacitness* of the *domain specialist* ds_i or the *solution-solver* ss_i about the *concept* c_j, the *relationship* r_j or the *behavior* b_j
     • ω a membership degree, with $\omega = f(p, pk)$, where

$$f : (DS \cup SS) \times (C \cup R \cup B) \rightarrow [-1, 0, 1]$$

is an intuitionistic membership function of the tacitness of p_i about the piece of knowledge pk_j, being p a domain specialist or a solution solver and pk knowledge about a concept, a relationship or a behavior, and

$$\forall\ (p)\ \forall\ (pk)[\omega(p,pk)=0 \rightarrow pk \square\ tacit\ \circ pk \square\ p],$$
$$\forall\ (p)\ \forall\ (pk)[\omega(p,pk)=1 \rightarrow pk \square\ explicit \circ pk \square\ p] \text{ and}$$
$$\forall\ (p) \forall\ (pk)[\omega(p,pk)=-1 \rightarrow pk \square\ p]$$

6. $N_c \subset (B \cup C \cup R)$ and N_c represents a *necessity* of the clients and users. Sometimes the necessity corresponds to a problem in the domain, but not always. In both cases, the necessity or problem demands a *suitable solution* S.

- S is defined as a suitable solution. It means an *any-time* solution that satisfies the clients and users' necessities or expectations. An any-time solution is the best current solution that generates a process at the time it stops.
- $SlRS = \{sr_1 \ldots sr_w\}$ is a set of *solution requirements sr* where sr_w represents the value taken by variable sr_w in the w*–th* unit. A solution requirement is a natural language statement to be enforced by the solution, possibly in cooperation with other system components, and formulated in terms of the application domain.
- *ANP* is informally defined as an *Arduous Negotiation Process* by which domain specialists and solution-solvers settle the features of the S while avoiding arguments.

3.1 Knowledge Management on a Strategy for Requirements Engineering (KMoS-RE)

The KMoS-RE strategy is a high level plan to achieve a solution specification through the eliciting, structuring and creating of knowledge. The strategy consists of three subsequent phases: *Domain Modeling, System Modeling and Specification Developing*, as it is proposed in [18] and structures its flow of activities according the *Knowledge Evolution Model for Requirements Engineering* (KEM- RE) (section 3.2). Furthermore, it includes transversal activities to identify and represent tacit knowledge through a record of wrong beliefs and a matrix of the tacitness level of concepts, relationships and behaviors by every participant in the project, named Piece of Knowledge (PoK) matrix. Below, a brief explanation of each phase is provided:

Domain Modeling Phase (DM). In this phase the terms, i.e. the concepts, attributes and relationships between concepts, and the basic integrity restrictions are formalized through a consensus, in order to understand the application domain without worry about the solution The terms are recorded in the Knowledge of Domain on an Extended Lexicon (KDEL), which is a lexical that classifies them into objects, subjects and verbs. The KDEL is used to facilitate the building of a graphical conceptual model. The externalization of this knowledge will enable achievement a consensus among the stakeholders; hence to minimize the symmetry of ignorance. The concepts and relationships identified in this phase will generate the first version of the Piece of Knowledge (PoK) matrix.

System Modeling Phase (SM). In this phase the current and future system processes are formalized. The current system corresponds to the system, as it exists at present. The future system represents the system after the deployment of a solution or product. The Use Cases technique was selected to model the system, both current and future, because its usefulness has been demonstrated through the time. The system model is obtained from the KDEL and the conceptual model. The behaviors identified in this phase will also change the values of the PoK matrix.

Specification Development Phase (SD). In this phase, the solution requirements are derived from the Uses Cases' scenarios of the future system and incorporated to the Solution Requirements Specification (SlRS).

3.2 Knowledge Evolution Model for Requirements Engineering

In ISD, understanding the problem and the structure of the solution are intertwined [19]. The solution-solvers, i.e. the requirements engineers, must explore different areas of the problem to find a solution; they should dialogue with the diverse domain specialists, who have their own domain knowledge or another perspective of the possible solution. By performing this task, the knowledge of the solution-solvers about the application domain increases. If necessary, they can return to previous states of the project but with additional knowledge that allows them to explore new possibilities of solution. In summary, the knowledge of the problem and its solution gradually evolves as requirements engineers gain more knowledge of the domain due to social interaction and their involvement with the business processes. In order to model that behavioral, the *Knowledge Evolution Model for Requirements Engineering* (KEM-RE) model was developed based on the SECI model.

The KEM-RE is an iterative cycle (Fig. 1) composed by stages that include the four kinds of knowledge processes in the innovation of complex problem solving [20]:

- **Knowledge Elicitation and Creation (KE&C) Stage.** Socialization mode is used in order to elicit knowledge, either from the domain specialists to the requirements engineers or vice versa.
- **Knowledge Integration and Application (KI&A) Stage.** The requirements engineers incorporate the elicited knowledge and their own experience into models. This is a complex activity in which combination and externalization modes are presented. In addition, as the requirement engineers develop models they internalize (clouds) the domain knowledge.
- **Knowledge Sharing and Exchange (KS&E) Stage.** The models developed by requirements engineers will be shared with the domain specialists. This phase takes place through socialization.
- **Knowledge Validation (KV) Stage.** The domain specialists validate the models. In order to develop this activity, an arduous negotiation process is necessary since they must internalize the knowledge behind the models. This process leads to the elicitation of new knowledge. Then the cycle starts again.

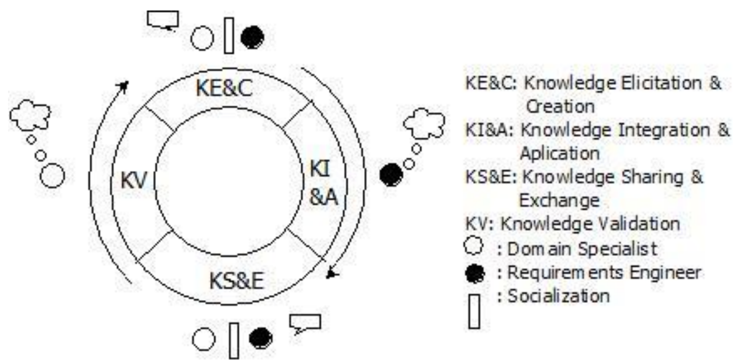

Fig. 1. Knowledge Evolution Model for Requirements Engineering

3.3 KMoS-RE Activity Flow

The KMoS-RE strategy begins with an Initialization Activity (IA) in which an initial interview is conducted. This information can be completed with formal documents such as user manuals, policies, business processes, and even legacy systems. After the interview, requirements engineers initialize the PoK matrix by identifying domain specialists, concepts, relationships and behaviors. Finally, the values of the PoK matrix are recorded according the knowledge tacitness level.

The Fig. 2 depicts the activities flow of the KMoS- RE strategy at a global level in an UML activity diagram. Every activity of the strategy correspond to one state of the KEM-RE: Model Validations (MV) is related to Knowledge Validation (KV), Knowledge Elicitation (KE) is related with Knowledge Elicitation and Creation (KE&C), Model Discussion (MD) corresponds to Knowledge Sharing and Exchange (KS&E), and Domain Modeling (DM), System Modeling (SM) and Specification Development (SD) corresponds to Knowledge Integration and Application (KI&A). The swim lanes in the figure represent the activities developed by each type of actor.

Once the IA is concluded, the requirements engineers begin to develop the artifacts to model the domain. Then, the requirements engineers discuss the models with the domain specialist in order to validate them. By doing this process, more domain knowledge is elicited, and the requirements engineers can decide to improve the previous models or to continue with the artifacts of the next phase, that is, the requirements engineers can work in parallel with several models but it is necessary to start in the established order. The above is represented in Fig. 2 with a bold line. These activities will be repeated until those involved in the project reach a consensus about the set of requirements for the solution. Each phase is composed by a set of tasks that will guide the requirements engineers to the development of the artifacts; the complete set of tasks of the KMoS-RE strategy can be consulted in [21].

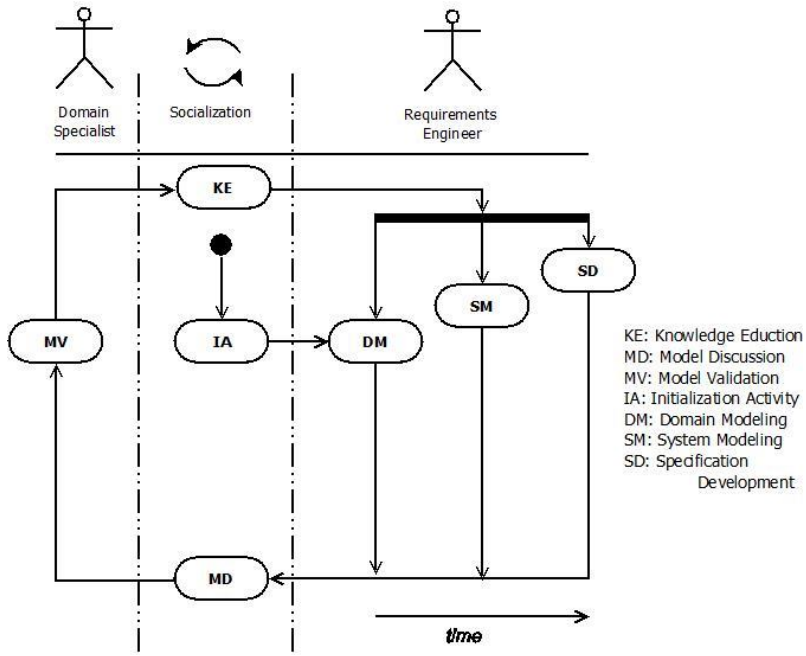

Fig. 2. KMoS-RE Activity Flow

3.4 Tacit Knowledge Identification

In ISD, most of the information is situated and depends on the context and the social interactions in order to be understood. Thus, an innovative feature of the KMoS-RE strategy is that it incorporates two techniques from sociolinguistic in order to identify tacit knowledge in a systematic way: *presuppositions* and *Bloom's Taxonomy*.

Presuppositions

According to Ma et al. [13], a way to identify the presence of tacit knowledge in a speech (spoken or written) is through the linguistic phenomena named presuppositions. A presupposition is a background belief, relating to an utterance, that 1) must be assumed by the speaker and addressee for the utterance, 2) generally, it will remain a necessary assumption whether the utterance is placed in the form of an assertion, denial, or question, and 3) can generally be associated with a particular lexical or syntactic structure, named linguistic trigger, in the utterance. Some examples of linguistic triggers to identify presuppositions are definitive descriptions, factual and implicative verbs, expressions of repetition, temporal relations, comparisons and questions.

Bloom's Taxonomy

The Bloom's Taxonomy is a framework for classifying statements of what it is expected or intended for students to learn as a result of instruction. It also provides definition for three domains of educational activities or learning: Cognitive, affective and psychomotor. Cognitive domain classifies thinking according to six cognitive levels of complexity. The lowest three levels are: knowledge, comprehension, and application. The highest levels are: analysis, synthesis, and evaluation. Each category can be related with a list of action verbs that express the measurable and observable behaviors expected of the learner. The categories are arranged from concrete to abstract; thus, the higher levels subsume the lowers. Bloom's Taxonomy has been used to elicit tacit knowledge in the development of Knowledge Management Systems [22]. Requirements engineers must analyses the discourse in order to find verbs classified in the higher order of the taxonomy, the use of those verbs by the domain specialists could suggest a reference to critical, ambiguous and abstract thinking. Therefore, requirements engineers should proceed to elicit the hidden knowledge behind the verb through well-structured questions. The KMoS-RE strategy proposes that the tasks of tacit knowledge identification be conducted in each phase in order to identify the most possibly quantity of it.

4 Experimentation and Results

The KMoS-RE strategy has been applied to several real cases in the context of ISD, which can be divided into software and non-software solutions [23]. The software applications belong to Intelligent System and are listed below:

- Design Decision System of Heating Ventilating and Air Conditioning (HVAC) modules using Case Based Reasoning.
- Business Networks Optimization System Based on Bayesian Networks.
- Cognitive Rehabilitation System for Multiple Sclerosis.
- Recommender System Specialists for Tourism Crisis.

Regarding to software applications, the KMoS-RE strategy showed its usefulness since in all cases a software product was obtained which satisfied the domain specialists' needs. The strategy also showed that it is easy-to-learn and easy-to-apply because final-year computational system engineering students applied it. Finally, the strategy showed that it did not require an excessive amount of time to be applied and that the obtained artifacts can be used to design, codify, implement and test the software; i.e. its usefulness is expanded to all stages of software development.

Another motivation of this thesis was to design a RE strategy that could be used in scopes of applicability other than software development, since the need to elicit the correct, appropriate and unambiguous requirements in ISD is not exclusive to that area. Thus, the KMoS-RE strategy was analyzed in the context of HVAC (heating, ventilation, and air conditioning) modules design. The goal of the case was to provide evidence that the KMoS-RE strategy could be implemented as the requirements elicitation process of a real company in order to obtain a HVAC module design closest to the clients' needs and expectations. The case study showed that the KMoS- RE was feasible to be implemented in this context. The strategy was also applied to formalize the domain knowledge in the scope of a project of scientometric. The outcomes allowed the domain specialist structure their domain knowledge and clarify the solution. In other words, even if a software solution was not developed, the generated artifacts were valuable for the domain specialist. In summary, these two experiences showed the openness of the strategy.

5 Conclusions

The KMoS-Re strategy was designed to confront the challenges of eliciting knowledge and incorporate it in a product or solution specification in ISD. Due to the characteristics of these domains, the strategy emerges from requirements engineering in the context of software development, but incorporates knowledge engineering techniques in order to properly manage the tacit knowledge. The Domain Modeling phase confronts the problem of formalizing the concepts and relationships, the system modeling phase deals with the problem of structuring the solutions to the problems in the domain, and the methods and techniques to manage knowledge address the problem of handling tacit knowledge. The strategy was applied to several real cases, both belonging to software and non-software development, but all of them in the context of ISD.

The solutions achieved through of the implementation of the strategy in several real cases provide evidence about the usefulness and the value of it. As future work, it is necessary to continue developing case studies to validate and improve the KMoS-RE strategy, as well as, developing software tools to automatize some activities of the strategy.

References

1. Olmos, K., Rodas, J.: Requirements engineering process model for informal structural domains. International Journal of Computer and Communication Engineering, 2:1 (2013) 75–77
2. Polanyi, M.: The tacit dimension. University of Chicago Pres (1966)
3. Distanont, A., Haapasalo, H., Vaananen, M., Lehto, J.: The engagement between knowledge transfer and requirements engineering. International Journal of Management, Knowledge and Learning, 1:2 (2012) 131–156
4. Fenstermacher, K.: The tyranny of tacit knowledge: What artificial intelligence tells us about knowledge representation. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), pp. 243--248. IEEE Press (2005)
5. Gacitua, R., Ma, L., Nuseibeh, B., Piwek, P., de Roeck, A., Rouncefield, M., Sawyer, P., Willis, A., Yang, H.: Making tacit requirements explicit. In: Proceedings of the Second International Workshop on Managing Requirements Knowledge (MARK), pp. 40--44, IEEE (2009)
6. Shadbolt, N., Smart, P.: Knowledge elicitation. In: Wilson, J., Sharples, S. (eds) Evaluation of human work. CRC Press, 163-196 (2015)
7. Broy, M.: Domain modeling and domain engineering: Key tasks in requirements engineering. In: Muïnch, J., Schmid, K. (eds) Perspectives on the Future of Software Engineering, pp. 15--30. Springer Berlin, Heidelberg (2013)

8. Goguen, J.: Formality and informality in requirements engineering. In: Proceedings of the 4th International Conference on Requirements Engineering, vol. 96, pp. 102—108 (1996)
9. Friedrich, W., Van Der Poll, J.: Towards a methodology to elicit tacit domain knowledge from users. Interdisciplinary Journal of Information, Knowledge, and Management, 2 (2007) 178–193
10. Stone, A., Sawyer, P.: Identifying tacit knowledge-based requirements. IEEE Proceedings-Software, 153:6, (2006) 211–218
11. Medeni, T., Ünsal, S., Ayas, M., Medeni, I.: Tacit knowledge extraction for software requirements specification (srs): A proposal of research methodology design and execution for knowledge visualization. In: Proceedings of the 55th Annual Meeting of the ISSS-2011, Vol. 55. pp. 1—20, Hull, UK, (2011)
12. Stoiber, R. Glinz, M.: Modeling and managing tacit product line requirements knowledge. In: Proceedings of the Second International Workshop on Managing Requirements Knowledge (MARK), pp. 60—64, IEEE (2009)
13. Mohamed, A.: Facilitating tacit-knowledge acquisition within requirements engineering. In: Proceedings of the 10th WSEAS International Conference on Applied Computer Science (ACS'10), pp. 27—32, Japan, (2010)
14. Nonaka, I., Takeuchi, H.: The knowledge-creating company: How Japanese companies foster creativity and innovation for competitive advantage. Harvard Business Review, 69: 6 (1995) 96-104
15. Pilat, L., and Kaindl, H.: A knowledge management perspective of requirements engineering. In: Proceedings of the Fifth International Conference on Research Challenges in Information Science (RCIS), pp. 1—12,. IEEE (2011)
16. Wan, J., Zhang, H., Wan, D., Huangm, D.: Research on knowledge creation in software requirement development. Journal of Software Engineering and Applications, 3:5 (2010) 487–494
17. Vásquez-Bravo, D., Sánchez-Segura, M., Medina-Domínguez, F., Amescua, A.: Combining software engineering elicitation technique with the knowledge management lifecycle. International Journal of Knowledge Society Research, 3:1 (2012) 1–13.
18. Hadad, G.: Use of scenarios in the derivation of software (In Spanish). PhD thesis, Facultad de Ciencias Exactas, Universidad Nacional de La Plata. (2008)
19. Nguyen, L., Shanks, G.: A framework for understanding creativity in requirements engineering. Information and Software Technology, 51:3 (2009) 655-662
20. Nakamory, Y.: Knowledge Science: Modeling the Knowledge Creation Process. CRC Press (2012)
21. Olmos, K., Rodas, J.: KMoS-RE: Knowledge management on a strategy to requirements engineering. Special Issue on Requirements Engineering in Software Product Line Engineering, Requirements Engineering Journal, 19:4 (2014) 421–440
22. Mitri, M.: Applying tacit knowledge management techniques for performance assessment. Computers & Education, 41:2 (2003) 173–189
23. Olmos, K.: Knowledge Management on a Strategy for Requirements Engineering. PhD thesis, Instituto de Ingeniería y Tecnología, Universidad Autónoma de Ciudad Juárez (2015)