

An extension to scenarios to deal with business cases for the decision-making processes in the agribusiness domain

L. Antonelli¹, G. Camilleri², C. Challiol^{1,3}, A. Fernandez^{1,4}, M. Hozikian¹, R. Giandini^{1,4}, J. Grigera^{1,4}, A.B. Lliteras^{1,4}, J. Martin¹, D. Torres^{1,4,5}, P. Zarate⁶

¹ Lifa - Fac. de Informatica, Universidad Nacional de La Plata, 50 esq 120, La Plata, Bs As, Argentina, {leandro.antonelli, cecilia.challiol, alejandro.fernandez, marian.hozikian, roxana.giandini, julian.grigera, alejandra.lliteras, jonathan.martin, diego.torres}@lifa.info.unlp.edu.ar

² SMAC group, IRIT, 118 route de Narbonne, 31062 Toulouse Cedex 9, France, camiller@irit.fr

³ also CONICET. Argentina.

⁴ also CICIPBA Pcia. Bs As. Argentina.

⁵ also Depto. CyT, Universidad Nacional de Quilmes. Bernal. Argentina.

⁶ ADRIA group, IRIT, Université de Toulouse, 2 rue du Doyen Gabriel Marty, 31042 Toulouse Cedex 9, France, zarate@irit.fr

Abstract. With the aim of pushing innovation through information and communication technology in the agri-business field, working closely with farmers is essential. It is especially important to systematically capture their knowledge in order to analyze, propose and design innovation artifacts (in terms of software applications). In this article, we use Scenarios to capture the knowledge of the experts that is elicited in early meetings previous to the definition of requirements. At those early stages, there are many uncertainties, and we are particularly interested in decision support. Thus, we propose an extension of the Scenarios for dealing with uncertainties. Scenarios are described in natural language, and it is very important to have an unbiased vocabulary. We complement Scenarios with a specific glossary, the Language Extended Lexicon that is also extended to decision support. According to V-model life cycle, every stage has a testing related stage. Thus, we also propose a set of rules to derive tests from the Scenarios. Summing up, we propose (i) an extension to Scenarios and the Language Extended Lexicon templates, (ii) a set of rules to derive tests, and (iii)

an application to support the proposed technique. We have applied the proposed approach in a couple of case studies and we are confident that the results are promising. Nevertheless, we need to perform a further exhaustive validation.

Keywords. Scenarios, Uncertainties, Decision Support, Agri-Business, LEL

1 Introduction

Agricultural processes are complex by nature because they rely on unpredictable conditions as weather or market demand, as well as human decision on biased opinions and incomplete information. Many people participate in the processes, usually with different objectives, background, experience and level of studies. Thus, it is hard to obtain a complete and accurate understanding of the whole process. This is the motivation of the RUC-APS project [9], a European funding project dealing with Enhancing and implementing Knowledge based ICT solutions within high Risk and Uncertain Conditions for Agriculture Production Systems.

Requirements engineering is one of the most important stages in software development. Errors made at this stage can cost up to 200 times to repair if they are discovered when the software is delivered to the client. There are two approaches to elicit requirements: classic and agile. Use Cases are a widely used tool to describe software requirements in a classical approach. There are different templates with different level of detail to use according to the definition of the requirements. Nevertheless, when the definition of requirements is vague, agile methods use User Stories to discover the requirements in an iterative and incremental way while the software application is developed.

In both cases, with classic and agile development, the client should have a clear idea of the role of the technology and a vision about the software artifact that he needs. Although the requirements of the software are mainly described during the requirements engineering phase, there are some early meetings previous to the requirements stage to discuss needs, wishes and expectation. In these meetings, the objectives and boundaries of the software application are defined. Following these definitions, the

requirements engineering stage can be performed to analyze and describe requirements (either with Use Cases or User Stories for example).

It is important to have tools to capture the information of the early meetings. Moreover, when stakeholders are not aware of how technology can help, it is necessary to support them. In this case, the IT team needs to learn about the domain and make proposals about innovation. We have been participating in the RUC-APS project with the aim of providing innovation in information and communication technology (ICT) to agriculture. In this period we have learned that agriculture is a field with no much integration with ICT. And it is a field with many uncertainties.

Some uncertainties are related to decision that farmers have to take and once taken, it cannot be changed. For example, the conducting system of the plants is related to make plants go upward or go down (as a bush). It should be defined before planting because both conducting systems need different distance between the plants. After the decisions made, it cannot be changed. Another decision is the training system to use. If it is decided that the plants go upward, some string is needed to help the plant to go upward. Besides a string, others elements can be used. Each element has different advantages and disadvantages. Then, the pruning system establishes how to cut the plant in order to allow it to go upward or down. All these decisions are related among them. Nevertheless, it is quite impossible to evaluate the impact of one decision on another. So, the decision has to be made one by one in a progressive process.

Others uncertainties are related to everyday situations that should be evaluated to react in consequence. For example, the temperature monitoring of a greenhouse must be set in order to keep it constant. We have seen that both types of uncertainties are captured in early meetings. Specifically, we are interested in innovation in Decision Support Systems, that is, to take a decision on the first type of uncertainties. Nevertheless, we also consider the second type of uncertainties.

In this article, we propose to capture the knowledge obtained from early meetings through Scenarios. We present an extension to the Scenarios to deal with uncertainty. We also propose to complement the description of the Scenarios with a particular glossary, the Language Extended Lexicon (LEL). An unbiased language is very important to understand the scenarios. We also propose an extension to the LEL to deal with uncertainty in order to capture business knowledge through Scenarios and complement them with LEL. According to the V-model development life cycle, the product obtained in each step (requirements, design, and

codification) should be tested. Thus, we also propose a set of rules to automatically generate tests from Scenarios. We propose a set of rules to derive tests from Scenarios (only from Scenarios, not from LEL) in order to fulfill with the V-model. These tests derived from Scenarios should be used as input to design tests for the requirements.

The paper is organized in the following way. Section 2 introduces LEL and Task/Method models we use in this work. Section 3 details our contributions, i.e. a process to capture Scenarios and LEL, as well as the extensions to the Scenarios and the LEL to deal with uncertainty. Section 3 also presents the rules to derive tests from Scenarios using the task/method model as the specification. Finally, Section 4 discusses some conclusions.

2 Background

This section describes the base elements used in our approach. It describes the original template of Scenarios and LEL that we use to capture the knowledge about business cases. And it also describes the Task/Method model, the technique used to describe tests derived from the Scenarios through the set of rules proposed. The topics of this section are the base for our approach. In the next section, the original template of Scenarios and LEL are extended to deal with business cases for the decision-making processes in the agribusiness domain. In addition, in a further section, the Task/Method model is used to describe the proposed rules of our approach.

2.1 Scenarios

Scenarios describe interactions between users and a future system [17]. It is also used to understand the context of the application because they promote the communication when there is a great variety of experts [6].

Leite [12] defines a scenario with the following attributes: (i) a title; (ii) a goal or aim to be reached through the execution of the episodes; (iii) a context that sets the starting point to reach the goal; (iv) the resources, relevant physical objects or information that must be available, (v) the actors, agents that perform the actions, and (vi) the set of episodes.

The following Scenario describes the activity of determining cultural labors for tomato production. It is important to mention that the scenario describes some task with uncertainty because many decisions have to be

taken. Although people (the farmers and their leader) take the decision, some software application can be used to support the process. Moreover, the last episode of the scenario is related to describe the definitions arrived according to some standard. This task can also be supported by an application software since the application can receive the information about the decisions, organize and present according to the standard. Moreover, the task of writing the procedure can also be tested to assure if the report produced by a future application satisfies or not the standard.

Listing 1. Scenario about cultural labors

Title: Determine cultural labors

Goal: Decide the conducting system, the training system and the pruning policies that should be used

Context: Tomato production

Resources: conducting system techniques, training system techniques, pruning policies, standard to describe procedures, procedures for the cultural labors

Actors: farmers, leader

Episodes:

The farmers and their leader decide a conducting system to use

The farmers and their leader decide a training system to use

The leader establishes the pruning policies to apply

The leader writes a procedure according to the standard describing the conducting system, the training system, and the pruning policies

2.2 Language Extended Lexicon

The Language Extended Lexicon (LEL) is a glossary used to capture and describe the domain's language [11]. Terms (also called symbols) are classified into four types: Subject, Object, Verb, and State. Subjects represent an active element that performs actions. Objects are passive elements on which subjects perform actions. A verb is used to represent the actions. Finally, States represent situations in which subjects and objects can be located. A symbol is described by two attributes: (i) notion and (ii) behavioral responses. Notion describes the symbol denotation and explains its literal meaning. While Behavioral responses describe its connotation, that is, the effects and consequences of the relationship between the defined symbol and others symbols defined in the LEL [18].

The following examples describe one term of each symbol category. It is important to remark the example in Listing 4. The verb describes the

activity of controlling the temperature of the greenhouse. This situation presents some uncertainty since the temperature can change and the farmer should act in consequence in order keep it within a specific range.

Listing 2. LEL subject: farmer

Subject: Farmer

Notion: Person that grows tomatoes in a shared lot with other farmers

Behavioral responses:

The farmer participates in the determination of the cultural labors

The farmer plant the tomatoes

The farmer grows the tomatoes

The farmer control the temperature of the greenhouse

Listing 3. LEL object: greenhouse

Object: Greenhouse

Notion: Place to grow tomatoes in a controlled environment.

Behavioral responses:

The farmer plant tomatoes in a greenhouse

The farmer control the temperature of the greenhouse

Listing 4. LEL verb: Control the temperature of the greenhouse

Verb: Control the temperature of the greenhouse

Notion: Action of monitoring the temperature in order to maintain it between certain range

Behavioral responses:

The farmers monitors the temperature

The farmers aerate the greenhouse to descend the temperature

The farmers close the windows of the greenhouse to increase the temperature

Listing 5. LEL state: Flowering

State: Flowering

Notion: Phenological state of the tomato, characterized by the appearance of leaves and flowers.

Behavioral responses:

The tomato change to fruition state after the appearance of the first fruit.

2.3 The Task/Method model

The Task/Method model is a knowledge modeling paradigm that considers the reasoning as a task [19] [21]. Its main advantage is to have a declarative form to express knowledge which can be easily processed by tools such as execution engines [4].

A Task/Method model is composed of two sub-models: (i) the domain model and (ii) the reasoning model. The domain model describes the objects of the world that are used by the reasoning model. The reasoning model describes how a task can be performed. It uses two modeling primitives: (i) task and (ii) method.

A task is a transition between two world state families (an action) and is defined by the following attributes: (i) name that describes the task, (ii) par, a typified list of parameters handled by the task, (iii) objective, the goal state of the task, (iv) method, describes one way of performing a task.

A method is characterized by the following attributes: (i) heading, the identification of the task achieved by the method, (ii) preconditions which must be satisfied to be able to apply the method, (iii) effects, consequences of a successful application of the method, (iv) control, achievement order of the subtasks, and (v) subtasks. This paper focuses on some of the attributes described before. A full description of this modeling paradigm is performed by Camilleri et al. in [4] [5].

We use the Task/Method model in order to describe the tests that should be applied to the Scenarios. Then, the Task/Method model is processed by an execution engine to finally test the Scenarios. Listing 6 shows the example of a Task/Method model to test the scenario of determining cultural labor described in Listing 1. The Scenarios are descriptions in natural language, while the Task/Method model is a computer language, that is why some changes must be done to the names. For example, while “conducting system” is valid in a Scenario, in Task/Method should be translated as “conductingSystem”. Moreover, Scenarios are too wide and abstract, not all the situation can be tested. In this example, we only define to test the method of writing the procedures in order to verify if the description of the procedures satisfies some standard. The procedures are definition from the decision of conducting system, training system and pruning policies performed previously. It is important to mention that Task/Method model is a hierarchical model, describing decompositions of tasks using other tasks. That is why the test is finally performed in method M14 (the fourth line in M1) of Listing 7.

Listing 6. Method determine cultural labor

Method: M1

Task: determineCulturalLabor

Control:

Decide (farmers, leader, conductingSystem);

Decide (farmers, leader, trainingSystem);

Establish (leader, pruningPolicies);

Write (leader, procedures, standard, conductingSystem, trainingSystem, pruningPolicies);

Listing 7. Method write procedures according to the standard

Method: M14

Task: Write (leader, procedures, standard, conductingSystem, trainingSystem, pruningPolicies)

Control:

message("procedures do not conform to writing standards, stop");

stop;

3 Proposals

This section describes the contribution of this paper: (i) an extension to Scenarios and the Language Extended Lexicon templates, and (ii) a set of rules to derive tests from Scenarios.

We propose to capture the actual knowledge obtained from early meetings through Scenarios. Nevertheless, it is very hard to collect domain information in software development. Thus, the Scenarios extended by the proposed approach help to acquire hypothetical and unclear situations, so as to convert the Scenarios in real based and concrete Scenarios. In order to deal with the uncertainty, we extend the Scenarios with 4 more attributes: key decision, variables (frozen and contextual), identified risks and factors of uncertainty. These attributes capture relevant information to make a decision. For example "determine the conduction system" is a key decision. In Listing 1 it was defined as part of the episodes, but in fact, this element should be captured as a key decision. Then, the LEL is used to complement the Scenarios, and we also proposed an extension of the LEL to deal with uncertainty. For example, the temperature of the greenhouse is a variable that change according to the weather conditions (its context) and

some actions must be done to maintain it in a specific range. This information should be captured by the LEL.

Scenarios and LEL are elements used to capture the business knowledge in early meetings. According to the V-model, this should be done on the top level of the V-model. Thus, in order to fulfill with the V-model, we also propose a set of rules to derive test from Scenarios. It is important to mention that the derivation is performed only from Scenarios, the LEL is not considered in this paper.

Thus, we provide techniques to use in the top level of the V-model (the business level): Scenarios (complemented with LEL) and business test (derived from Scenarios through a set of rules). These both elements can be used to produce the elements of the following level (requirements level). That is, Scenarios should be used to describe requirements, and business tests should be used to describe acceptance test.

The rest of the section is organized in the following way. First, a collaborative process to capture the knowledge to describe Scenarios is presented. Then, the extensions to the Scenarios and the LEL are described. Finally, the rules to derive tests described in Task/Method model from Scenarios are detailed.

3.1 A collaborative and iterative process to capture knowledge

This section describes an iterative and incremental process to capture the knowledge from the stakeholders in an early stage of software definition. In that early stage, there are many uncertainties, that is why the most critical elements to capture are the potential decision and eventual variabilities that should be considered.

The process begins with the definition of the Scenarios by a multidisciplinary team. During the description of the Scenarios, it is common that specific terms of the domain appear, that are used by the experts of the domain but are unknown to the technical team members. These terms should be described in the LEL. Scenarios are used to capture multiple views and promote communication among stakeholders [6]. Moreover, a multidisciplinary team helps to obtain a broader perspective to define scenarios.

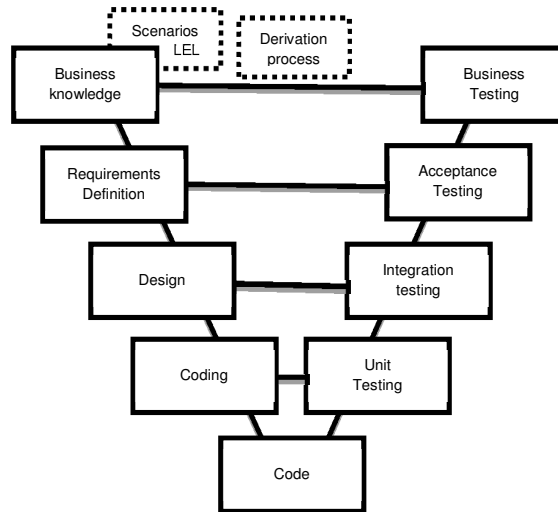


Fig. 1. Contribution located on the top of the V-model

It is important to state that the proposed process consists of describing mainly Scenarios, and describe terms in the LEL only if necessary. Traditionally, LEL is defined completely first and then Scenarios are described [12].

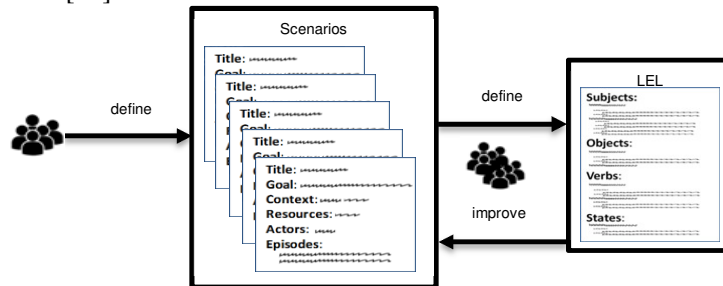


Fig. 2. Scenarios' definition

3.2 Extensions to Scenarios and LEL

The background section describes the original proposal to describe Scenarios and LEL. In order to support the capture of the knowledge in a

very early stage of software definition, with many uncertainties and lack of precision, this section describe the attributes added to Scenarios and LEL.

3.2.1 Scenarios extended to deal with uncertainty

In early meetings to discuss the incorporation of technology, it is important to add to the Scenarios information about factors of uncertainty, that is, the doubts that must be clarified later. We propose to add the following attributes: (i) key decisions, (ii) variables, (iii) risks, and (vi) factors of uncertainty.

Lupetti et al. [14] incorporate the concept of variables at an abstract level and categorize them according to their participation in Scenario design decisions. On the other hand, when a business process problem in decision support system is proposed, the work of De Maio et al. [7] stresses the importance of context-aware heterogeneous fuzzy consensus model learning from the past executions. So, there is a need to feed and maintain a knowledge base storing the associations between contextual variables, key decisions, and weights for each decision maker. Based on this reasoning, this article proposes to incorporate variables and key decisions in the description of the Scenarios.

Some uncertainties are related to decision that farmers have to take and once taken, it cannot be changed. For example, the conducting system mentioned before. Others uncertainties are related to everyday situations that should be evaluated to react in consequence. For example, the monitoring of the temperature of a greenhouse mentioned before. The first type of decisions are fixed when they are defined, and it makes possible to define more specific Scenarios considering that definition. For example the layout of the plantation can be discussed considering certain conducting system. While the second type, depends on situation that varies according the context.

Thus, we consider two types of variables: (i) frozen variables and (ii) contextual variables. Frozen variables are decisions that should be taken and they will not change from that moment. For example, the conduction system. Contextual variables refer to a decision that should be taken very often. For example, the monitoring of the temperature.

Both types of variables are important to decision support. People decide about the frozen variables, thus, they need tools to support the decision.

While contextual variables are important to design the software application considering that it must react according to the variable.

In order to analyze the key decision, more information about the variables is needed. Both types of variable, mainly frozen variables, depends on risks and uncertainties. A risk is anything that could potentially impact your plan [16]. Thus, if a frozen variable for a decision is related to defining the growing space (indoor or outdoor), although the historical temperature can support the decision to grow outdoor, according to the geographical situation, there could exist the risk of unexpected low temperature.

The uncertainty concept is closely related to risks. A risk is an event that could potentially occur, thus, there is a measurable probability of occurrence. For example, there is a chance of 0.15 of low temperature during the harvest. But the factor of uncertainties is related to elements that there is no historical information, or cannot be estimated or predicted [10].

Listing 8 describes each proposed attribute with some information related to determine cultural labor scenario.

Listing 8. Extension to the Scenarios

Key decisions:

conduction system (How many main stems each plant will have?)

training system (How to support tomato plants off ground?)

pruning policies (When to start the pruning of stems and trusses to comply with the conduction system decisions? How often to prune?)

Variables

Frozen: soil type, seed type, geographic area, growing space (outdoor, greenhouse)

Contextual: diurnal temperature, nocturnal temperature, natural lighting

Identified risks: unexpected climate phenomena: frost, stronger winds, flood risk, low temperatures

Factors of uncertainty: plant disease, market demand, water pollution

Taking decisions (give values to frozen variables and deal with contextual variables) is related to analyzing and balance risk and uncertainties. From a classic point of view, it is necessary to analyze the risk involved in the key decisions, perform a quantitative and qualitative analysis of the uncertainties in order to make the decision [22] [23]. Of

course, this decision could be made in different ways, but it is important to have documented the information to analyze in order to make the decision. That is the objective of adding these attributes to the Scenarios.

3.2.2 LEL extended to deal with contextual information

We propose to use Scenarios to capture uncertainty in order to take a decision. That is, experts and IT team analyze the scenarios and discuss alternatives to take a decision. Nevertheless, the uncertainty that the scenario represent could be a cause and effect relation, with different situations and different actions. In this case, is no need to take a decision, because all the alternatives should be considered by the application software. This, is another interpretation of decision making: a context-aware behavior, where relevant variables (context-features) take different values (representing specific situation) to trigger some decisions [7]. The context-aware behavior could present uncertainty when it is not possible to estimate or measure one or more variables' values. So, it is not possible to establish which decision should be triggered [3].

Litvak et al. [13] present an extension of LEL to provide more expressivity for Verbs (such as an action of an effect of), but, this is not enough to represent context-aware behaviors. Fortier et al. [8] model context-features and trigger decisions (handlers) as a first-class citizen due to the complexity involved in this kind of applications. Using the concepts defined in [8], this article proposes an extension to LEL that would help to define contextual information in a common vocabulary.

The LEL originally categorizes symbols in Subject, Object, Verbs, and States. This article proposes two new categories: Context-Feature and Contextual Decisions. The context-feature category allows representing each contextual feature associated to Subjects or Objects. Context-Features' values could be simple values (e.g. sensed data) or more complex elements [8]. Contextual Decision allows to represents for each situation (defined by a context-features' values) the list of triggered actions. The Given-then specification [24] can be used to describe behavioral responses for Contextual Decisions.

Let's consider the following examples. A Context-Feature "Temperature of greenhouse" which defines for some interval values trigger Contextual Decisions (Listing 9). And its Contextual Decision definition "Evaluate

Temperature Decision” (Listing 10) that determines the action to be performed.

Listing 9. LEL Context-Feature: temperature of the greenhouse

Context-Feature: Temperature of the greenhouse
Notion: Range of temperature values measured in the greenhouse
Behavioral responses:
Value < 10 °C, Evaluate Temperature Decision
Value between 10° C and 25° C, No Action
Value > 25° C, Evaluate Temperature Decision

Listing 10. LEL Context-decision: evaluate temperature decision

Context-decision: Evaluate temperature decision
Notion: When temperature is high or low, worker leader should be notified
Behavioral responses:
Given (Temperature of greenhouse < 10 °C or Temperature of greenhouse > 25° C)
Then Notify to worker leader

3.3. Test derivation from Scenarios

We propose five rules to derive test from the Scenarios [2]:

Rule 1. Tasks Identification: each verb in the Scenario’s episodes is translated into a task in Task/Method model. Each Scenario title is also a task in Task/Method model. Listing 11 shows the example.

Listing 11.

Title: Determine cultural labors → Task: DetermineCulturalLabors
Episodes
The farmers and their leader decide a conducting system to use → Task: Decide
The farmers and their leader decide a training system to use → Task: Decide
The leader establishes the pruning policies to apply → Task: Establish
The leader writes a procedure according to the standard describing the conducting system, the training system, and the pruning policies → Task: Write

Rule 2. Task’s Parameters Identification: each actor and resource used in the episodes of a Scenario is translated into a parameter in Task/Method model. Listing 12 shows the example.

Listing 12.

The farmers and their leader decide a conducting system to use → Task: Decide (farmers, leader, conductingSystem)
The farmers and their leader decide a training system to use → Task: Decide (farmers, leader, trainingSystem)
The leader establishes the pruning policies to apply → Task: Establish (leader, pruningPolicies)
The leader writes a procedure according to the standard describing the conducting system, the training system, and the pruning policies → Task: Write (leader, procedures, standard, conductingSystem, trainingSystem, pruningPolicies)

Rule 3. Episode's method: the episodes part of a scenario is translated by a method in Task/Method model. Listing 13 shows the example. Since the main Scenario is translated into the method named M1, each episode of the Scenario is translated into a method M1#.

Listing 13.

The farmers and their leader decide a conducting system to use → Method: M11
The farmers and their leader decide a training system to use → Method: M12
The leader establishes the pruning policies to apply → Method: M13
The leader writes a procedure according to the standard describing the conducting system, the training system, and the pruning policies → Method: M14

Rule 4. The Sequence of tasks: the sequence of different lines in the episodes part of a Scenario determines the sequence of tasks in the control part of a Task/Method model method. The use of expressions like "then", "after", etc... in the episodes of a Scenario determines also a sequence of tasks in the method's control part. Listing 14 shows the example.

Rule 5. Test Case Method: We assume that each test case (Test cases part of scenario) corresponds to an achievement status (succeed or fail) of the task. In a failure situation, the scenario will stop. This stop case will be represented by a method for the next task in which the precondition field corresponds to the test case failure. The example was shown in Listing 7.

Listing 14.

Episodes:
...
The leader establishes pruning policies
The leader writes a guide...

Or

... The leader establishes pruning policies, then the farmers' leader writes a guide...

Method: M1

Task: determineCulturalLabor

Control:

...

Establish (leader, pruningPolicies);

Write(leader,procedures,standard, conductingSystem, trainingSystem, pruningPolicies);

4 Conclusions

We have presented a proposal to use Scenarios at an early stage of software development, when there are many uncertainties and the software is not defined yet. Also, gathering knowledge from farmers can be difficult, since they don't always organize their ideas in a way that's useful for requirements, so the proposed extensions to Scenarios and LEL, as well as the rules to derive tests, help acquiring information from them.

We enriched the Scenarios with some attributes that capture critical information to help stakeholders to take a decision and perform a further requirements definition of a software application. We complement the Scenarios with a particular glossary, the Language Extended Lexicon that we also extended to deal with uncertainty. According to the V-Model, in which every software development phase has a related testing stage, we also propose a technique to derive test from the Scenarios. Thus, we are providing a technique to deal with uncertainty and decisions on the top level of the V-model life cycle.

We have also built software applications to manage all the information. A Media Wiki platform [15] is used as a repository of the Scenarios and LEL. A semantic media Wiki extension was also added to allow the semantic support and the creation of forms in order to make the CRUD operations (create, retrieve, update and deleted) in a more user-friendly way. Then, in order to provide support to the derivation of tests, the tool relies on a Natural Language Processor Framework [20] and on a task automation tool and administrator of configurations [1].

This work was motivated with the aim of providing decision support to the agri-business field in the context of the RUC-APS project. Although we have done some preliminary validation of the proposed strategy, we are

planning to develop some pilot project in order to conduct case studies and to obtain feedback for improvement and validation of the proposal.

Acknowledgement. Authors of this publication acknowledge the contribution of the Project 691249, RUC-APS: Enhancing and implementing Knowledge based ICT solutions within high Risk and Uncertain Conditions for Agriculture Production Systems (www.ruc-aps.eu), funded by the European Union under their funding scheme H2020-MSCA-RISE-2015.

References

1. Ansible IT automation (2018), <https://www.ansible.com/>
2. Antonelli L, Camilleri G, Grigera J, Hozikian M, Sauvage C, Zaraté P (2018) A Modelling Approach to Generating User Acceptance Tests, International Conference on Decision Support Systems Technologies (ICDSST 2018), Heraklion, Greece.
3. Borodin V, Bourtembourg J, Hnaïen F, Labadie N (2016) Handling uncertainty in agricultural supply chain management: A state of the art. *European Journal of Operational Research*, 254(2), pp 348-359.
4. Camilleri G, Soubie JL, Zalaket J (2003) TMMT: Tool Supporting Knowledge Modelling. In *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 2773, 2003, pp. 45–52.
5. Camilleri G, Soubie JL, Zaraté P (2005) Critical Situations for Decision Making: A Support Based on a Modelling Tool. In *Group Decision and Negotiation*, Springer Verlag, Vol. 14 N. 2, pp 159-171.
6. Carroll, JM (2000) Five reasons for scenario-based design. *Interacting with computers* 13.1. doi: 10.1016/S0953-5438(00)00023-0. pp 43-60.
7. De Maio C, Fenza G, Loia V, Orciuoli F, Herrera-Viedma E (2016) A framework for context-aware heterogeneous group decision making in business processes. *Knowledge-Based Systems* 102 (2016) pp 39–50.
8. Fortier A, Rossi G, Gordillo SE, Challiol C (2010) Dealing with variability in context-aware mobile software. *Journal of Systems and Software*, 83(6), pp 915-936.
9. Hernandez, J., Mortimer, M., Patelli, E., Liu, S., Drummond, C., Kehr, E., Calabrese, N., Iannaccone, R., Kacprzyk, J., Alemany, M. and Gardner, D., (2017) RUC-APS: Enhancing and implementing Knowledge based ICT solutions within high Risk and Uncertain Conditions for Agriculture Production Systems. In *11th International Conference on Industrial Engineering and Industrial Management*, Valencia, Spain.

10. Khodakarami V, Fenton NE, Neil M (2007) Project scheduling: improved approach to incorporate uncertainty using Bayesian networks. *Project Management Journal*, 38(2), PP 39–49.
11. Leite JCSdP, Franco APM (1993) A strategy for conceptual model acquisition. In *Requirements Engineering conference*. IEEE. doi:10.1109/ISRE.1993.324851, pp 243–246.
12. Leite JCSdP, Rossi G, Balaguer F, Maiorana V, Kaplan G, Hadad G, Oliveros A (1997) Enhancing a requirements baseline with scenarios. In *requirements Engineering*. Vol 2.4. doi: 10.1109/ISRE.1997.566841. pp 184-198.
13. Litvak CS, Hadad GDS, Doorn JH (2018) Nominalizations in Requirements Engineering Natural Language Models. In *Encyclopedia of Information Science and Technology, Fourth Edition*. IGI Global. pp. 5127-5135.
14. Lupetti ML, Gao J, Yao Y, Mi H (2017) A scenario-driven design method for Chinese children edutainment. In *proceedings of the Fifth International Symposium of Chinese CHI*. ISBN: 978-1-4503-5308-3 doi>10.1145/3080631.3080636. pp 22-29.
15. Media Wiki (2018), <https://www.mediawiki.org>
16. Project Management Institute (2017) *Project Management Body of Knowledge*.
17. Potts C (1995) Using schematic scenarios to understand user needs. In *proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*. ACM. doi: 10.1145/225434.225462
18. Sarmiento E, Leite JCSdP, Almentero E (2014) C&L: Generating model based test cases from natural language requirements descriptions. In *Requirements Engineering and Testing (RET), 2014 IEEE 1st International Workshop on*. IEEE, 2014. doi: 10.1109/RET.2014.6908677
19. Schreiber G, Akkermans H, Anjewierden A, De Hoog R, Shadbolt NR, Wielinga B (2000) *Knowledge Engineering and Management: The CommonKADS Methodology*, vol. 99.
20. Stanford Natural Language Processing (2018), <https://nlp.stanford.edu>
21. Trichet F, Tchounikine P (1999) DSTM: A framework to operationalise and refine a problem solving method modeled in terms of tasks and methods, *Expert Syst. Appl.*, vol. 16, no. 2, pp. 105–120.
22. Trutnevyte E, Guivarch C, Lempert R, Strachan N (2016) Reinvigorating the scenario technique to expand uncertainty consideration *Climatic Change* Volume 135, Issue 3, pp 373-379.
23. Weaver CP, Lempert RJ, Brown C, Hall JA, Revell D, Sarewitz D (2013) Improving the contribution of climate model information to decision making: the value and demands of robust decision frameworks, *WIREs Climate Change* vol 4 pp 39-60.
24. Wynne M, Hellesoy A, Tooke S (2017) *The cucumber book: behavior-driven development for testers and developers*. Pragmatic Bookshelf.