

Towards a new requirements' definition methodology using ontologies for Pervasive Games Based Learning Systems

Yemna MEJBRI^{1,4}, Maha KHEMAJA^{2,4} and Kaouther RAIES^{3,4}

¹ National School of Computer Sciences, University of Manouba

² Higher Institute of Applied Sciences and Technology, University of Sousse

³ Faculty of Economic Sciences and Management, University of Sfax

⁴ PRINCE Research Group, ISITCom, Sousse

Abstract

Since a Pervasive Games Based Learning System (PGBLSs) is considered as technology enhanced learning system, it becomes important to enhance the development process. Despite the growing presence of mobile devices and the wireless network communication technologies, users needs satisfaction is particularly challenging due to problems arising from the highly dynamic environments in which services will operate. We propose, in this paper, a semantic model driven requirements engineering process in order to improve the development of PGBLSs. This model is based on an ontology of requirements (RO) as a powerful formalism to assist requirements' analysts for fulfilling changing requirements in PGBLSs dynamic contexts. In such environments, analysts have to establish the relative priorities of requirements for resolving conflicting requests. For this issue, a requirements analysis technique is also proposed.

Keywords: Pervasive Games based Learning Systems, Ontology, Requirements Engineering, Analysis technique, Game Theory.

Received on 19 November 2016, accepted on 5 December 2017, published on 4 January 2018

Copyright © 2018 Yemna MEJBRI *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/_____

1. Introduction

PGBLSs were born in the interplay of several subsystems. As such, a PGBLS includes the pervasive computing trend which aims to open up the design space of games by expanding the contractual magic circle of play spatially, temporally or socially [1]. Thus, pervasive games (PG) constitute the central stone of a PGBLS. Within a PG, a player may have effective and interactive experiences that motivate and engage her/him in the learning process by putting him/her into virtual environment of learning.

PGBLSs demand operating in highly dynamic environments and adapting to the current and changing requirements of the user's context. In general terms, context can be seen as being "any information used to characterize the situation of an entity. "An entity is a person, place or object that is considered relevant to the interaction between

a user and an application, including the user and the application themselves"[2]. Context information does not only describe static invariants aspects, but also it is characterized by frequent changes and dynamic information. As we move into a new era of dynamic context, more attention is needed to support the requirements of PGBL systems and allow their evolution at runtime. Once a set of new PGBLSs requirements are identified, it is difficult to implement all of them due to time, resource and budget constraints. PGBLSs requirements should often be implemented in stages, and their analysis helps determining which ones should be implemented first. A function can always be added and the user interface enhanced. Some requirements are critical for the success of the system. Hence, requirements should be analyzed so that the ones that are most likely to achieve customer's satisfaction can be selected for implementation.

Due to the emergence of studies on technology enhanced learning systems, the capacity to build pervasive games

based learning systems that process dynamic context information and adapt the knowledge to meet the requirements of users, has attracted the attention. This paper intends to use an ontology of requirements within the requirements engineering process which is suitable for continually updating the user's changing requirements.

The rest of this paper is organized as follows: the second section presents background and motivation. Section three presents the related works. Section four presents the application of RE process. Finally the fifth section reports on the progress and the future works

2. Background and Motivation

2.1. Requirements Engineering (RE)

A requirement prescribes a property judged necessary for a system. The IEEE Standard Glossary of Software Engineering Terminology [3] defines a requirement as: (1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2). The term requirement has a strong meaning in the Requirements Engineering literature (RE). Requirements Engineering (RE) is the first fundamental step of any project of system development. According to GTIE (Working Group on Requirements Engineering) of the AFIS (French Association of Systems Engineering) [4], RE means all activities to discover, analyze, validate and evolve a set of a system's requirements. RE is essentially based on communication between different participants. The main objective is to enable a common understanding by the various project stakeholders, to design the system. Stakeholders are people, organizations, or objects that are directly or indirectly involved with the project. They can either affect or be affected by the project or the system, which means they are important to identify. In general, this understanding is represented in the form of written and graphical models, corresponding to the requirements specification.

The elicitation of the requirements consists on the collection, capture, discovery and development of requirements from a variety of resources including human stakeholders. The analysis focuses on reviewing and the understanding of the elicited requirements and their appraisal for quality in terms of accuracy, completeness, clarity and consistency. The result of this step is a set of identified requirements but that are not yet formal. The specification is the registration and the documentation of requirements so that they are usable by stakeholders, in particular, by developers who need to design and build the system. It is to establish the final list of requirements by organizing them according to categories. The validation is the confirmation of the quality of requirements and their

compliance with the needs and desires of stakeholders. The requirements are tested using a prototype. Finally, a validation of the quality attributes of those requirements (consistency, accuracy, completeness, wholeness) must be conducted.

Since PGBLSs requirements often evolve, then the only way to guarantee the success of the requirements' definition process is to embrace these changes throughout the different stages of the development life cycle. Unfortunately, traditional RE process takes place only once prior to the start of development. Thus, changing requirements risk to increase the cost of development or to be undoable. For this reason, Agile Software Development has attracted our attention. Hence, surveying the potential of bringing them together will undoubtedly benefit PGBLSs requirements' definition.

2.2. Agile RE

Agile RE is able to embrace continuous changes in requirements and accommodate the growing context evolution. Agile RE supports the principal ideas behind traditional requirements engineering but it differentiates in a preference for various techniques. The techniques used in the elicitation phase are much like those of the traditional development. However in agile, the elicitation process is performed iteratively and continuously, before each development iteration, accentuating the communication of the elaborated requirements with the customer. Similar to the traditional methodology, the analysis phase in Agile checks the requirements for completeness, consistency, essentiality, and feasibility. This is achieved through conducting JADS [5] for all the involved stakeholders to prioritise the already made user stories and to sort out any conflicts/ omissions, if any, in the requirements. In Agile, prioritisation takes place before each iteration, and not just one time, as in the traditional approach, before the development starts. Unlike traditional development that prioritizes requirements based on many criteria (i.e. risks, cost, implementation dependencies, business value, and time), the agile approach prioritizes requirements based mostly on the highest business value they deliver.

The basic idea of Agile specification phase is to guide the developing team to build models that resolve the problems of design without the need to overbuild these models [6]. However, traditional development keeps all the models, at the different levels of abstractions, to become a part of the system documentation, and that needs to be kept up-to date with any future alterations. Agile methods [7] use frequent review meetings and various tests to validate requirements. However, the main difference between traditional and agile development in requirements validation is the strong emphasis placed on testing in the agile methods.

2.3. Agile Requirements modelling techniques

Generally requirements can be regarded “as a specification of what should be implemented” [8]. They indicate what players really want, the functionalities that a system is supposed to provide to satisfy the customers, the constraints of the system etc. Such a clear understanding of the player's priorities helps the development team better meet player needs. The study of literature shows different requirement analysis methods [9]. As an example, we can list:

Numerical Assignment Technique (NAT)/ MoScow: Numerical Assignment is fundamental technique for requirement analysis in which several groups of requirements analysis are made and then requirements are assigned to one of these groups on the bases of their priority. The number of groups can vary, but in practice, three groups are very common (e.g. critical, standard, optional). When using numerical assignment, it is important that each group represents something that the stakeholders can relate to (e.g. critical, standard, optional), for a reliable classification. The result of numerical assignment is requirements analyzed on an ordinal scale. However, the requirements in each group have the same priority, which means that each requirement does not get a unique priority.

In MoScow technique, the requirements are classified into four groups depending on the importance of the functional requirements [10]:

M: MUST have this. It is the highest priority and without it the project considered a failure.

S: SHOULD have this requirement if possible. Customer satisfaction depends on this requirement. But we cannot say its absence causes a project to fail.

C: COULD have this requirement if it doesn't affect anything else.

W: WON'T have the requirement this time but WOULD like to in the future.

This technique helps understanding customer needs. The problem with this method is the difficulty of distinguishing the terms “Must” and “Should” as they both express a customer preference or desire.

Analytical Hierarchical Process (AHP): In AHP the candidate requirements are compared pair wise, and to which extent one of the requirements is more important than the other requirement. The total number of comparisons to perform with AHP are $n \times (n-1)/2$ (where n is the number of requirements) at each hierarchy level, which results in a dramatic increase in the number of comparisons as the number of requirements increases.

Value Oriented Analysis (VOP): The first step in setting up a value oriented analysis process is to establish a framework for identifying the business's core values and the relative relationships among those values. VOP uses the relationships that exist between core business values to assess and prioritize requirements and ensure their traceability. The VOP framework establishes a mechanism for quantifying and ordering requirements for an application increment, a prototype, or a software requirements specification. Company executives identify the core business values and use a simple ordinal scale to weight them according to their importance to the organization.

Hundred Dollar Method/ Cumulative Voting (CV): each stakeholder is given a constant amount (e.g. 100, 1000 or 10000) of imaginary units (for example monetary) that he or she can use for voting in favor of the most important issues [11]. In this way, the amount of money assigned to an issue represents the respondent's relative preference (and therefore analysis) in relation to the other issues. The points can be distributed in any way that the stakeholder desires. Each stakeholder is free to put the whole amount given to him or her on only one issue of dominating importance. It is also possible for a stakeholder to distribute equally the amount to many of, or even to all of the issues.

Planning Game (PG): In planning game (also called PG) requirements are first elicited from the users, and written on story board. Then these requirements are analyzed by stakeholders into three different piles: 1. those requirements without which the software will not work/perform 2. Those requirements that are less important but give noteworthy business worth, and 3. Those requirements that would be good to have [12]. At the same time engineers calculate the time needed to develop each requirement and hence distribute the requirements on the bases of risk into three groups 1. Those that they can approximate accurately, 2. Those that they can approximate logically fine and 3. Those that they cannot approximate at all.

Considering specific aspects of PGBLSs domain is necessary. Among those aspects, learning and entertainment outcome is the most important one. Our research focus on Agile requirements modeling techniques but the principal issue is how to associate the value in order to prioritize the requirements. For this aim we consider using Game theory in the analysis phase.

2.4. Ontological Engineering

Ontology is among the concepts of the proposed solution in this research. The word ontology comes from the Greek *ontos* (being) and *logos* (word). It denotes the science of being and the descriptions for the organization, designation and categorization of existence [13]. Carried over to computer science in the field of artificial intelligence and information technology, an ontology is understood as a representational artifact for specifying the semantics or meaning about the information or knowledge in a certain domain in a structured form [14]. In fact, ontologies provide a formal representation of knowledge and the relationships between concepts. It can be used for both, describing requirements specification documents [15, 16] and formally representing requirements knowledge [16, 17]. In contrast to traditional knowledge-based approaches, e.g. formal specification languages, ontologies seem to be well suited for an evolving approach to the specification of requirements and domain knowledge [17]. Moreover, ontologies can be used to support requirements management and traceability [15, 17]. Automated validation and consistency checking are considered as a potential benefit compared to semi-formal or informal approaches providing no logical formalism.

3. Related works

In this section, the most promising RE related areas are explored. However, there has been little work in requirements engineering for game development. Callele's research [18] is considered one of the first works to address these issues. He suggests that the majority of problems found in game development are due to inadequate requirements engineering between the pre-production and production phases. Alves discussed requirements engineering challenges and concluded that RE is highly relevant for mobile games industry [19]. She reported a study conducted at Meantime, a mobile game developing company based in Recife, Brazil.

Applying RE in the pervasive systems field is still full of plenty rich research areas. Miguel et. al. [20] studied the applicability of three Requirement Engineering (RE) techniques (Use Cases [21], Viewpoints [22], and Goal-Oriented [23]) for the specification of collaborative systems and paying special attention to the awareness requirements. In order to carry out their study, they specify some awareness requirements of a real system (Google Docs) [24]. Geisser and al. [25] focus on the distributed software engineering tasks. Traditional methods of requirement engineering support collocated scenarios. The paper presents a cost-effective, adaptable and evaluation concept for requirement engineering in distributed environment. They develop a tool "TraVis" for visualizing and analyzing requirements in distributed settings. The experimented design explains and discusses many aspects of data collection. The data analysis phase covers a statistical analysis of the quantitatively measured data.

Sitou and Spanfelner [26] presented a model based on requirements engineering to analyze and specify the basic behavior of the system and the adaptive behavior based on the needs of the customers. The approach is based on the elicitation, analysis and specification of different parts of the context adaptive systems. The model enriches the context with aspects of participants, activities, changing behavior change, etc. This model consists of a user model, a task model, a domain model, a platform model, a model of dialogue and a presentation template.

Alharthi and al. [27] aimed to identify country and/or culture specific as well as common requirements for e-Learning systems, and to construct a framework for analysis of the diversity aspects such as culture and technical as well as sustainability aspects (environmental, technical, educational, social, etc). The framework contributed to the RE process for development and improvement of e-Learning systems, which might improve the overall sustainability of online and on campus teaching and learning activities.

These works, among others, have contributed to the definition of the requirements either in game development, in pervasive services or in learning systems by applying some or all RE concepts'.

The main contribution of our proposal is to apply RE process i.e., elicitation, analysis, specification and validation on one PGBLS which relates those three fields.

4. Proposed RE process

We aim to take advantage of ontologies and propose mechanisms and techniques to use them in a guided approach in order to define and analyze PGBLSs requirements. Our ontology of requirements is intended to reduce the ambiguity of needs and avoid incomplete requirements definition. In this context, Castaneda et al [28] describe the benefits and challenges of using ontologies in the process of requirements engineering (RE). This is exactly the basis of our approach. Indeed, RE imposes a systematic series of activities to be conducted on requirements. It contains the activities of elicitation, analysis, specification, validation and management [29]. Our approach takes those activities [30] but adapts them for the definition of PGBLSs requirements.

4.1. Requirements elicitation

The elicitation of requirements consists on the collection, the capture, the discovery and the development of requirements from a variety of resources including human stakeholders. To do this, we suggest reviewing the literature to study the areas of learning systems and pervasive games. The result of this step consists of a first set of textual PGBLSs requirements. Since the PGBLSs is characterized by the overlapping of two areas: Pervasive games and learning, we have broken the requirements into three categories of requirements which are pervasiveness requirements, games requirements and learning systems requirements. Literature review was conducted to get requirements for the development of PGBLSs. Unfortunately, no publications dealing with requirements for PGBLSs could be found in scientific literature. Instead, a list of PGBLSs influencing factors could be identified, which have an effect on the learning success and motivation of players like the works of [31], [32], [33], [34], [35], [36]. To automatically document the elicited requirements, we have used a corpus. A corpus is a collection of documents [37]. Our corpus is collected from the aforementioned researches. Figure 2 presents the PGBLSs knowledge corpus.

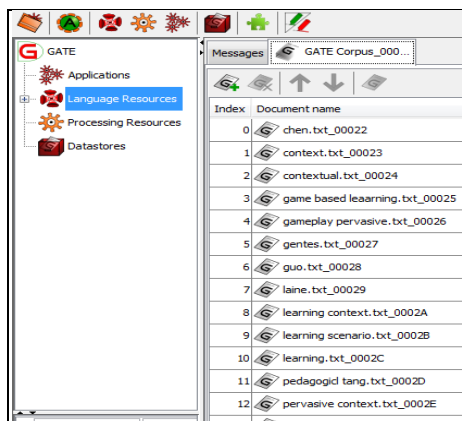


Figure 1. PGBLSs knowledge corpus

4.2. Requirements analysis

Some requirements are critical for the success of the system. Hence, requirements should be analyzed so that the ones that are most likely to achieve customer satisfaction can be selected for implementation. PGBLSs development organizations often must deal with requirements that tend to evolve quickly. Rapid changes in competitive threats, player preferences and development technology make pre-specified requirements inappropriate. Agile methods that seek to address the challenges in such dynamic contexts have gained much interest among practitioners and researchers. Table 1 outlines a number of different analysis techniques. We, then, categorized the requirement analysis techniques according to a number of criteria such as ease of use, required completion time, reliability of results, etc. Those ranks accorded to the various techniques are based on a literature study [9] [10] [11] [12].

Table 1. Comparison of different analysis techniques

Techniques	Rank according				Total Number of Comparisons
	Ease of use	Total Time Taken	Scalability	Accuracy	
Numerical Assignment Technique (NAT)	6	6	5	6	1
Analytical Hierarchic	5	5	6	5	3

Table 2. Manual analysis of PGBLSs requirements with Game Theory concepts

al Process (AHP)					
Value Oriented Analysis (VOP)	2	1	1	2	1
Cumulative Voting (CV)	4	3	4	3	1
Planning Game (PG)	1	2	3	4	1

Outcome of the table says that VOP is supposed to be the best method for prioritizing software requirements. It is an easy method, it gives one of the most accurate results, and it is rather comfortable to handle even if there are many more requirements. In most questions' PG was located in the middle, neither the best nor the worst techniques. The worst candidate is NAT. However, this order of the requirement analysis techniques obtained is not a global one as rankings can be reordered if criterion weights are assigned differently. Nevertheless, the technique and formulae used here to compare among different analysis techniques can be used in any scenario with appropriate criterion weights suitable for that scenario. In general, requirements can be analyzed taking many different aspects into account. An aspect is a property or attribute of a project and its requirements that can be used to prioritize requirements. Common aspects are importance, penalty, cost, time, and risk [6]. Often, the aspects interact and changes in one aspect could result in an impact on another aspect [7]. However, in the field of PGBLSs, requirements are changing following the highly dynamic environments. In the user context, users can specify their preferences along with their context. For example, a player may specify that whenever she/he looking around in the museum and trying to locate exhibits (Museum Scrabble game), she/he would like that the query processor takes into account the chosen topic and the available hints. Besides, Game Theory [38] is a mathematical tool that describes and analyzes behaviours in strategic situations. It is usually used to predict the outcome of complex interactions among rational entities. We propose a game theory based approach to resolve Users Requirements conflicts in the context of PGBLSs. Table 2 illustrates a manual analysis of three different requirements with Game Theory concepts.

Requirements list definition	Elements of Game Theory			Assumptions of Game Theory		
	Players	Strategies	Information	payoffs	Rationality	Common knowledge
<p>Learning requirement</p> <p>Learners perform activities to create an outcome using an environment in which there are learning objects and services. (requirement conforms IMS LD standard)</p>	Learners	Learning activities	A complete description of the possible activities, the outcomes for each activity and the available objects and services in the environment.	Learning outcomes	Calculate which activities will maximize their learning outcomes.	Each learner/user/player knows the set of learner/user/player, strategies and payoffs from all possible combinations of strategies: call this information 'X'. Common knowledge means that each learner/user/player knows that all learner/user/player know X, that all learner/user/player know that all learner/user/player know X, that all learner/user/player know that all learner/user/player know X, and so on...
<p>Pervasiveness requirement</p> <p>Users need to expand the contractual magic circle of the system socially, spatially or temporally.</p>	Users	Social strategies, spatial strategies, temporal strategies	A complete description of the social, spatial and temporal aspects.	Services outcomes	Calculate which strategies will maximize their services outcomes.	Common knowledge means that each learner/user/player knows that all learner/user/player know X, that all learner/user/player know that all learner/user/player know X, that all learner/user/player know that all learner/user/player know X, and so on...
<p>Game requirement</p> <p>Players, need a set of strategies available to them, and specification of players' payoffs for each combination of such strategies (possible outcomes of the game)</p>	Players	Options specific to the game	A complete description of the information available to players at each decision made.	Entertainment outcomes	Calculate which options will maximize their entertainment outcomes.	Common knowledge means that each learner/user/player knows that all learner/user/player know X, that all learner/user/player know that all learner/user/player know X, and so on...

The analysis table shows that Game theory seems a good starting point for analyzing strategic interactions in PGBLSs. Throughout this analysis, feasible conflicts in PGBLSs would appear because end users seek to maximize gains and minimize losses. From the outcome point of view, the payoff depends on the choices of all players and it distinguishes many activities choices as non cooperative (have less priority) and others activities choices are non cooperative (more priority). However, as the context of PGBLSs changes needed information changes or new information appeared.

Game theory is rich of concepts to sets out, first, broad patterns of resource allocation (activity assignments). Then, it shows how conflict arises from the allocations. Finally, it highlights potential solutions to those conflicts.

Among those concepts, we can list “The Pareto-optimal outcome of the game which is the outcome as a result of cooperation amongst the players involved. While the Nash Equilibrium is the outcome of a game whereby given the

options of another player, no player can do any better (improve his payoff) by changing his strategy. Nash stability is about what is good for an individual without considering what is good for the whole system while Pareto-optimality is about what is good for the system without considering the interests of the individuals within the system” [39]. The Dominant strategy equilibrium of a game is a strategy which any agent in the game would use regardless of what strategies the other agent uses. All these outcomes of the game are called ‘Solution concepts’ which lead to the best strategies of every agent or player (the more important strategies).

The process of analysis should be automated because game theory is difficult to be processed automatically when users are not Game Theory skilled.

In fact, Game Theory concepts are expressed in a natural pseudo language based on mathematics and could not be used in an automatic processing. This requires a formalism that must be flexible and adaptable to the dynamic aspect of the context.

For that aim, we propose a Semantic Web technology based solution. More specifically our proposal, in this paper, will focus on a unified semantic modeling for both Game Theory and Requirements (more detailed in section 4.3). Consequently, we have proposed in [40] a consistent and comprehensive domain ontology of game theory (GTO). Figure 2 shows an excerpt of the GTO.

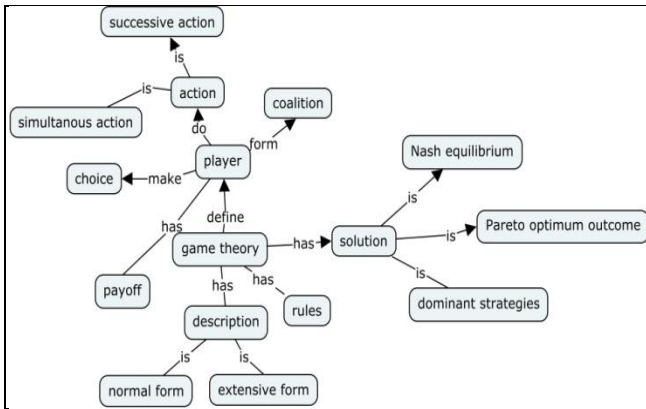


Figure 2. An extract of the proposed PGBLSs ontology of requirements

The Game theory ontology (GTO) will be used for PGBLSs needs analysis in order to allow learning/gaming objectives achievement. In fact, the role of GTO in the process of designing PGBLSs is to help the designer to automatically analyze his/her strategic situation by game theory. Our proposed automatic analysis process is more explained by the next activity diagram.

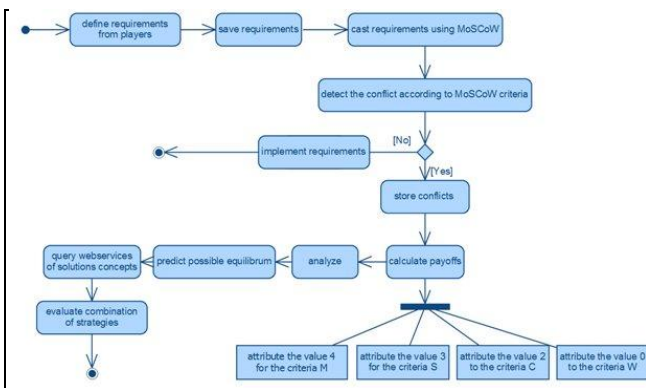


Figure 4. The activity diagram of the automatic analysis process

The Analyst must be able to save details of user requirements gathered from different requirements elicitation techniques in an ontology of PGBLSs requirements (result of specification phase). Where a requirements conflicts is identified and verified, the Analyst must be able to store details of this conflict.

In the next step, the GBL scenarios designer wants to predict possible equilibrium of the game. He/she should, in this step, query the GTO for identifying the kind of algorithms to execute for equilibrium prediction. The Analyst cast requirements for what they think should be the payoffs for each requirement. The requirements are analyzed using the MoSCoW criteria (Must, Should, Could, Would) to describe the importance of each requirement to the increase of learning outcome. Aggregate sum of their votes are calculated to get the actual payoffs for the game.

When context change, The Analyst, instead of evaluating all possible combinations of players strategies, he only concentrate on Nash equilibrium.

4.3. Requirements specification

In this phase, we suggest an ontology of requirements representing the requirements, the relations between them and the relationships with the system. Figure 3 illustrates the idea of the proposed ontology.

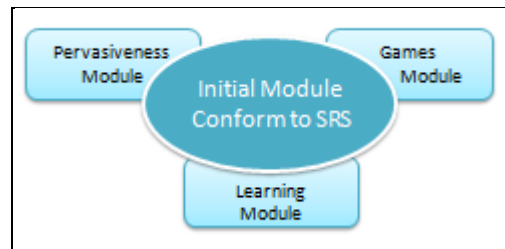


Figure 3. The proposed modules for the ontology of requirements

There are a number of methods and methodologies one can employ to develop their own ontology in a given domain. Among those methods [41] we find CyKB, Uschold and King's method, Grüninger and Fox's method, etc.. In this paper, we have adopted Noy & McGuinness's method to build our ontology of requirements. Noy & McGuinness's seven steps method requires one to [42]: 1. Determine the domain, scope and purpose of the ontology; 2. Consider reusing existing ontologies; 3. Enumerate important terms in the ontology; 4. Define the classes and the class hierarchy; 5. Define the properties of the classes – slots; 6. Define the facets of the slots; and finally 7. Create the instances classifications.

a. The scope, the domain and the purpose of the PGBLSs requirements ontology

The ontology covers the PGBLSs domain in its high level aspects (features and functionalities) as well as its organizational ones (learning strategies, gameplay scenario, and pervasive services).

The main objective of the target ontology is to provide a generic model containing knowledge about the core concepts related to PGBLSs (players, entities, requirements,

etc.). This ontology will be a support for the definition of PGBLSs requirements.

The ontology will be a meta-view for the different PGBLSs knowledge in the literature. It should harmonize the PGBLSs terminology spread in these researches and help requirements engineers in their development task.

b. Reuse of existing ontologies

The acquisition of the PGBLSs requirements knowledge started from standards (e.g. IEEE Guide for Developing System Requirements Specifications). Other knowledge acquisition sources were the different ontologies and researches that exist in the literature [43][44][45]. Moreover, ontology reuse will present an ontology modular’s development as illustrated in Figure 3.

c. The concepts

Based on the knowledge elicitation step, we have made concepts and relationships between them in a conceptual model. The concepts and the relationships of the PGBLSs requirements ontology proposed in this paper were chosen according to the number of repetition of concepts in the various researches related to PGBLS domain.

The concepts were organized around three main aspects which are game aspect, pervasiveness aspect and learning aspect. Various concepts are described briefly, in the following.

SRS module: this module includes the concepts related to the Software Requirements Specification [46]

Reference: List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

Function: Summarize the major functions the product must perform or must let the user perform.

System feature: The functional requirements for the product by system features

Pervasiveness module: this module includes the concepts related to the pervasive aspects.

Environment: Captures the entities that surround the user. Attributes, services, temperature, noise, networks, etc.

Social context: Describes social aspects of the user context. Attributes – information about friends, relatives, role in society. For example, social context at home is different from social context at work.

Spatio-temporal context: Describes aspects related to the time and space. Attributes - time, location, speed, direction, etc.

Information: Global and personal information, software, databases.

Games module: this module includes the concepts related to the game development field.

Action: Describes what the player is doing. Attributes – goals, tasks and actions.

Emotional requirements: The fun and enjoyment part.

Game mechanics: such as basic rules , themes, characters, environment and story are finalized.

Learning module: this module includes the concepts related to the learning systems development field.

Consistent learner information: Learner information shall be consistent throughout the platform.

Groups and roles: It shall be possible for users be allocated to one or more groups and assigned roles.

Load content objects: It shall be possible to load, store and make sharable content objects available to users. Run-time interactions with content objects should be supported. This includes being able to load bundled resources (content packages) and unpack them.

d. The relationships

High-level relationships between those concepts were defined. They were categorized into four kinds: Functional, Inverse Functional, Symmetric, SubClassOf and Transitive.

Relationship is functional if, for a given individual it can be in relation with individual, it can be in relation (through such property) with only another individual. For example, Aline (player1) hascompetitor Bob (player2). Inverse Functional relationship is the inverse of functional relationship. Symmetric relationship can be briefly described as “Aline hascompetitor Bob” allows to infer also “Bob hascompetitor Aline”. Transitive relationship means that if Aline hascompetitor Bob and Bob hascompetitor Amelie, then we can infer that Aline hascompetitor Amelie. SubClassOf implies a superclass-subclass hierarchy.

e. The axioms

In addition to concepts and relationships, an ontology contains axioms and attributes. Formal axioms are assertions accepted as true about abstractions of a field. The axioms allow us to define the meaning of concepts, put restrictions on the values of attributes, examine the conformity of specified information, or derive new concepts [47].

Table 3 illustrates some axioms with their descriptions and the related concepts.

Table 3. Part of the Table of axioms

Description	Expression	Concepts
A world can be either real or virtual.	$\forall x: \text{world} \Rightarrow \text{realworld}(x) \vee \text{virtualworld}(x)$	World Realworld virtualworld
players belong to different groups are competitors.	$\text{player}(?x) \wedge \text{has_group}(?x, ?y) \wedge \text{has_group}(?x, ?z) \wedge \text{differentFrom}(?y, ?z) \rightarrow \text{competitor_of}(?y, ?z)$	player

4.4. Requirements validation

The validation is the confirmation of the quality of requirements and their compliance with the needs and desires of stakeholders. The requirements are tested using a prototype. Indeed, a validation of the quality attributes of those requirements (completeness, validity, usability) must be conducted.

The completeness criterion is achieved by mapping the ontology of requirements and other Ontologies of requirements related to PGBLSs from literature in order to detect the level of covering knowledge comparing to other research.

The validity criterion is evaluated by requests of RO using its terminology and its capacity to answer. Table 4 summarizes some of these questions.

Table 4. Informal and formal questions to the requirements ontology

Queries	Part of result
What are the locations available for playing the game? ----- Location (?L) → sqwrl: select (?L)	Location x Location y
What are the impacts of such an action on the learning outcome? ----- Implies (outcome, ? I) → sqwrl: select (?I)	Enhancing-impact
What are the exhibits to be collected in the museum X? What is the location of each exhibit? ----- Has_item(Museum X, ?a) ∧ Has_Location(?a, ?l) → sqwrl:select(?a, ?l)	Exhibit x Exhibit z Located in the Hall H1 Located in the Hall H4

Finally, the usability criterion is validated by using the RO in a real project. Currently, we aim to test the integration of ontologies in the process of defining PGBLSs requirements.

References

- [1] Montola, M., Stenros, J., and Waern, A. Pervasive Games: Theory and Design. Experiences on the Boundary Between Life and Play. Morgan Kaufmann, Amsterdam et al., (2009).
- [2] ANIND K. DEY, Understanding and Using Context, Future Computing Environments Group College of Computing & Gvu Center Georgia Institute of Technology, Atlanta, GA, 30332-0280, USA (2001),
- [3] IEEE Computer Society "IEEE Standard Glossary of Software Engineering Terminology". IEEE Standard (1990)
- [4] AFIS, Modèle de données AFIS, version 2.0, groupe de travail Méthodes et Outils, (2005)
- [5] L.A. Macaulay, *Requirements Engineering*, Springer-Verlag, (1996)
- [6] S.W. Ambler, Agile Modeling, John Wiley & Sons, (2001)
- [7] Heba Elshandidy, Sherif Mazen, Agile and Traditional Requirements Engineering: A Survey, International Journal of Scientific & Engineering Research, Volume 4, (2013)
- [8] Boehm BW, Ross R. 1989. Theory-W software project management: Principles and examples. IEEE Transactions on Software Engineering 15(7):902F916
- [9] Berander P. 2004. Using students as subjects in requirements analysis. In: Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04). IEEE Computer Society, Los Alamitos, pp.167F176
- [10] The MoSCoW Analysis Technique LMR Technologies.2012. Agile Practices: Scrum, XP, Lean, Kanban
- [11] K. Beck, Extreme Programming explained, Reading, MA:Addison Wesley,1999

We focus on a case study in order to validate the feasibility of our proposed approach.

5 Conclusion

We presented in this paper our vision regarding an explicit classification of PGBLSs requirements following the three dimensions which are pervasiveness, games and learning systems. The main idea of our approach is an ontology of requirements following the requirements engineering process. The guidelines of the ontology of requirements help developers to capture user requirements, to facilitate the updating of dynamic requirements due changing context and allow reuse of the ontology when the learning environment varies.

Following that, an ontology of requirements is developed, focusing on different kinds of context changes. We aim, in the future work, that our proposal could be used as a foundation for the relating domains.

- [12] J.karlsson, C.Wolin and B. Regnell, An evaluation of methods for prioritizing software requirements, information and software technology, pp 939-947,(2007)
- [13] Grüber, T.: A translation approach to portable ontology specification. *Knowledge Acquisition* 5(2) (1993)
- [14] Allemang, D., Hendler, J.A.: *Semantic web for the working ontologist: Modeling in RDF, RDFS and OWL*. Elsevier, Amsterdam (2008).
- [15] Guarino, N., Carrara, M., and Giaretta, P. An Ontology of Meta-Level Categories. In J. Doyle, E. Sandewall and P. Torasso (eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference (KR94)*. Morgan Kaufmann, San Mateo, CA: 270-280. (1994)
- [16] Wouters, B., Deridder, D., Van Paesschen, E.: The Use of Ontologies as a Backbone for Use Case Management. In: "European Conference on Object-Oriented Programming (ECOOP 2000), Workshop : Objects and Classifications, a natural convergence", (2000)
- [17] T. R. Gruber. A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2):199–220, (1993)
- [18] D. Callele, E. Neufeld and K. Schneider, "Requirements Engineering and the Creative Process in the Video Game Industry", 13th IEEE International Conference on Requirements Engineering, 2005.
- [19] Carina Alves, Geber Ramalho and Alexandre Damasceno, "Challenges in Requirements Engineering for Mobile Games Development: The Meantime Case Study" 15th IEEE International Requirements Engineering Conference, 2007
- [20] M. A. Teruel, E. Navarro, L. J. Víctor, F. Montero, and P. González, "An empirical evaluation of requirement engineering techniques for collaborative systems," in *Proceedings EASE 2011*, Darham, 2011, pp. 114-123.
- [21] A. Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2000.
- [22] A. Finkelsetin, *et al.*, "Viewpoints: A framework for integrating multiple perspectives in system development," in *International Journal of Software Engineering and Knowledge Engineering*, 1992, vol. 2, no. 1, pp. 31-57.
- [23] L. M. Cysneiros and E. Yu, "Non-Functional requirements elicitation," in *Perspectives on Software Requirements*, J. C. Sampaio do Prado Leite, and J. H. Doorn, Ed., Springer, 2003, ch. 6.
- [24] Google Inc., 2010. Google Docs. (April 2013). [Online]. Available at: <https://docs.google.com>.
- [25] M. Geisser, T. Hildenbrand, F. Rothlauf, C. Atkinson, "An Evaluation Method for Requirements Engineering Approaches in Distributed Software Development Projects," *Software Engineering Advances*, ICSEA 2(007)
- [26] W. Sitou and B. Spanfelner, *Towards Requirements Engineering for Context Adaptive Systems*, Technische Universität München, Department of Informatics, Boltzmannstr.3, 85748 Garching/Munich, Germany, (2007)
- [27] Ahmed D. Alharthi, Maria Spichkova, Margaret Hamilton, *Requirements Engineering Aspects of ELearning Systems*, RMIT University, Melbourne, Australia, (2015)
- [28] Verónica Castañeda, Luciana C Ballejos et Maria Laura Caliusco : Improving the quality of software requirements specifications with semantic web technologies. (2012)
- [29] Decker, B., Rech, J., Ras, E., Klein, B., Hoecht, C.: Selforganized Reuse of Software Engineering Knowledge supported by Semantic Wikis. In: Proc. of Workshop on Semantic Web Enabled Software Engineering (SWESE). November, (2005)
- [30] Kotonya, G. and I. Sommerville. *Requirements Engineering: Processes and Techniques*. John Wiley & Sons, Chichester, (1998)
- [31] G. Hong, H. Trætteberg, A. I. Wang, and M. Zhu, TeMPS: A Conceptual Framework for Pervasive and Social Games, in Third IEEE Information Conference on Digital Game and Intelligent Toy Enhanced Learning Kaohsiung, Taiwan, (2010)
- [32] K. Jegers, "Pervasive Game Flow: Understanding Player Enjoyment in Pervasive Gaming, *ACM Computers in Entertainment*, vol. 5, April (2007)
- [33] A. Gentes, A. Guyot-Mbodji, and I. Demeure, Gaming on the move: urban experience as a new paradigm for mobile pervasive game design, *Multimedia Systems*, Vol. 16, pp. 43-55, doi:10.1007/s00530-009-0172-2 (2010)
- [34] B. K. alther, Notes on the methodology of pervasive gaming in *Entertainment Computing*, Vol. 3711, F. Kishino, Y. Kitamura, H. Kato & N. Nagata, Eds, pp. 488-495 (2005)
- [35] Quintin, J.-J., Depover, C., & Degache, C. Le rôle du scénario pédagogique dans l'analyse d'une formation à distance. Paper presented at the Actes de la conférence EIAH , Paris, INRP, 335-340.(2005)
- [36] Paquette, G. *Ingénierie Pédagogique*. Québec: Presses de l'Université du Québec. (2002)
- [37] Module 1: Introduction to GATE developer, University of Sheffield
- [38] R. Myerson. *Game Theory: Analysis of Conflict*, Harvard University Press, Cambridge (1991)
- [39] Madani, K. 2010. Game theory and water resources. *Journal of Hydrology*, 381(3), pp.225—238.
- [40] Y. Mejbri, M. Khemaja, 2013. A Domain Ontology of Game Theory Applied to Game Based Learning, the European Conference on Games Based Learning
- [41] O. Corcho, M. Fernandez-Lopez, and Asuncion Gomez-Perez, Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, vol. 46, no. 1, pp. 41-64, (2003)
- [42] N. F. Noy, and D. L. McGuinness, *Ontology development 101: A guide to creating your first ontology*, Stanford Medical Informatics (2001)
- [43] E. Serral, *A MDD Strategy for developing Context-Aware Pervasive Systems*, Universidad Politecnica de Valencia, (2008)
- [44] G. Hong, H. Trætteberg, A. I. Wang, and M. Zhu, "TeMPS: A Conceptual Framework for Pervasive and Social Games," in Third IEEE Information Conference on Digital Game and Intelligent Toy Enhanced Learning Kaohsiung, Taiwan, (2010)
- [45] Laine, T.H., Islas Sedano, C., Joy, M. and Sutinen, E. Critical factors for technology integration in game-based pervasive learning spaces, *IEEE Transactions on*

- Learning Technologies, vol 3, no 4, pp. 294 – 306. (2010)
- [46] IEEE Guide for Developing System Requirements Specifications, Software Engineering Standards Committee of the IEEE Computer Society, IEEE Std 1233, 1998 Edition (1996)
- [47] Staab, S., Maedche, A.: Axioms are Objects, too – Ontology Engineering beyond the Modeling of Concepts and Relations. In: Workshop on Applications of Ontologies and Problem-Solving Methods, ECAI 2000, Berlin (2000)