Hindawi
International Journal of Computer Games Technology
Volume 2018, Article ID 9834565, 16 pages
https://doi.org/10.1155/2018/9834565

Research Article

An Assessment of Serious Games Technology: *Toward an Architecture for Serious Games Design*

Walid Mestadi,[1] **Khalid Nafil**,[2] **Raja Touahni**,[1] **and Rochdi Messoussi**[1]

[1]*Ibn Tofail University, Faculty of Sciences, Kenitra, Morocco*
[2]*Mohamed V University, Faculty of Sciences, Rabat, Morocco*

Correspondence should be addressed to Walid Mestadi; mestadi.walid@gmail.com

Received 21 February 2018; Revised 8 June 2018; Accepted 9 July 2018; Published 1 August 2018

Academic Editor: Hanqiu Sun

The design of an engaging and motivating serious game (SG) requires a strong knowledge of learning domain, pedagogy, and game design components, which are hard to be found and restrained by an individual or one entity. Therefore and in the light of this statement, the collaboration between domain content, pedagogical, and playful experts is required and crucial. Despite the fact that the existing models that support SG design are intended to have a combination of learning and fun, the design of SG remains difficult to achieve. It would then be appreciated to propose means and guidelines that facilitate this design. To do so, this paper proposes a taxonomy, which classifies models that treat SG design, and then presents an opening as a functional architecture for supporting SG conception, which promotes the separation during the design, the collaboration between different involved experts, and the reuse of prior expert productions.

1. Introduction

The human evolution depends on how much importance is given to knowledge inter-generation transfer by taking into consideration individual characteristics as well as environment changes for each generation (i.e., digital or. com social network, web 2.0, and gaming). Students need a wide education in various fields which is essential for economical outcomes [1]. Researches in educational field require several researchers with different backgrounds. The challenges remain on student's engagement and motivation in traditional education, where engaged and motivated ones persist and investigate in understanding rather than only receiving the educational material. In contrast, disengaged students react to the education offer with less importance and without excitement or commitment. Thus, student's engagement is fundamental and critical for educational success [2–4].

The targeted engagement and motivation factors in education field are naturally available in gaming activities. Video games success and popularity are similar to those of books, movies, television, and other forms of media [5]. Video games are designed to engage players in an interactive environment, which makes them different from those media; also, they are played cooperatively or competitively, alone, with other physically present players, or with thousands of other online players, and they are played on various devices (consoles, computers, and cell phones) [6]. Games and their characteristics [7] have the potential to engage young and adult players naturally. As reported by [6], 91% of children between the ages of 2 and 17 play video games and up to 99% of boys and 94% of girls play these games. In addition, video games brought over $25 billion in 2010, more than Hollywood's 2010 box office. Educational and gaming experts are interested in designing games, which combine the fun factor and the educational content to engage and motivate students while learning. Such games are called serious games (SGs) [6–8].

Serious game is defined as a game [5], a mental contest [9], an interactive computer application [8], a digital game, a simulation, a virtual environment, and a mixed reality/media [10], applied in serious context such as education [5], government or corporate training, health, public policy, and strategic communication objectives [9]. Moreover they are used to impart skills/knowledge/attitude [8] or to deliver information [10], using the fun elements to engage learners [5, 8–10].

More specifically, serious game (SG) for educational purpose works on addressing the engagement issue by using the fun factors to immerse learners in an active learning environment [7, 11] and pushing learners to compete and overcome challenges by actions with immediate feedback [8]. However, because of the lack of standards on how to design SG in the field of education, it will be difficult to judge if the SG design results really meet the purpose for which it is designed. For example, if the domain content is designed by an individual or organization that is unfamiliar with the educational field, which is the case of a game designer, it may slip content errors if it is not validated by domain content and educational expert [12, 13].

However, the major issue relies on combining and balancing game elements (game characteristics, game mechanics, and gameplay) with learning factors (domain content or knowledge, skills, and learning mechanics) while keeping SGs potential promises. Designing and delivering a SG that engages and affects learners require a strong understanding of its filed and theories: game design, learning theories, and domain content. This can be achieved only by investigating how to mix and balance game potential in concert with learning outcomes and pedagogical objectives and how to manage the collaboration of experts with diverse creative and scholarly backgrounds [5, 8, 14, 15].

The current paper provides the main research works in SG design reported in the literature. The study of those proposals aims to understand SG design field, the challenges faced, and the proposed solutions in order to draw a classification and then propose an adequate solution for SG design. Our starting point for going into this in depth is exploring research's works in SG design field. In fact we will explore how SG model can be classified according to the nature of the model proposed, in order to facilitate the field's understanding, and then propose a solution that can help to design SG. As a first step we give an overview on notable works relevant to SG designs, in order to underline what they are trying to improve. Then, a discussion of SG model is given. Furthermore a comprehensive taxonomy of SG model is presented. Next an opening is presented and discussed, and we explain how it could be a best solution and useful for designing SG. Moreover, we study its implementation on two existing serious games. Then a conclusion is presented as a synthesis.

2. State of the Art

Several researches work on modeling their framework to deal with SG challenge from different views, where the challenge is seen as the difficulty in integrating the domain content into the game structure, the technological complexity introduced by game development, the mapping of learning theories and game mechanics, the reusability of prior production of SG, the organizational aspects, and the collaboration between the education and game experts or as framework which proposes a coherent set of structural components, which is used to create the foundations and outlines of the whole or a part of SG. However, designing SG which integrates and balances education and game requires solid bases to start with, such as a theoretical framework [30, 31].

From the view of integrating the domain content into the game structure, authors in [16] aim to teach software engineering processes by proposing a card game design under the name of "Problems and Programmers", which is a competitive game in which each student plays the role of a project manager; they lead the same project and the player who finishes first is the winner. For this, players have to manage their budget and resources to produce high quality software while meeting software project requirements. In order to achieve this goal, players should use their knowledge of software engineering to avoid any problems. The game imposes the use of the Waterfall development model that requires passing through the following steps: analysis, design, development, integration, and testing. Based on the "Problems and Programmers" design, authors in [25] propose a card game called "PlayScrum" which aims to teach the Scrum Agile Framework. Also, authors in [17] aim to teach programming and computational thinking concepts by proposing a platform's game called "Program your robot", in which player controls a game object which represents a "robot". The player should write an algorithm to order the "robot" to avoid obstacles faced in the platform. From an abstract level the "Program your robot" design is similar to the SG proposed by [18, 20, 22] in terms of "drag and drop" and writing a code by using preprogramming commands. In the same context, authors in [21] propose a SG design that uses domain content and analogical representations, where the learner is asked to complete a task such as comparing between real cases and analogical ones that are presented side-by-side. Authors in [29] focus on domain content integration into SG design by proposing a model, which is composed of two components. The first one represents a main game while the second one represents a set of separated "mini-games", "puzzles", and "quizzes" which are related to the main game. The main game is composed of a set of quests/missions, in which the educational objectives are represented by Non-Player Characters (NPC) in addition to the resources to be collected. Those resources are required to interact with "mini-games", in which we incorporate domain content to teach. Based on their model, the authors design a game called "Clean World" which aims to teach "Recycling and Renewable Energies".

From an organizational aspect and the collaboration between the education and game experts view, authors in [12] propose a structure of SG design composed of six phases under the name of "facets": "pedagogical objectives", "domain simulation", "interactions with the simulation", "problems and progression", "decorum", and "conditions of use". Additionally the authors provide a process model that specifies how pedagogical experts (domain content, pedagogy, knowledge, etc.) and playful experts (game designers, level designers, game producers, storyboard writers, artistic directors, actors, graphic designers, sound managers, etc.) collaborate according to the proposed structure. Each phase underlines a specific problem related to SG design and the needed expertise. However, in pedagogical objectives phase, pedagogical experts collaborate in order to represent domain content, educational objectives, and domain errors interpretation ("define problems where players can fail") into a valid domain model. Afterward, the pedagogical experts build a domain simulation with a game engine in compilation with the valid domain model. The aim of the simulator is to validate the domain model and to enumerate the player's action by testing the simulator responses to the player's actions including the domain errors interpretation. Then, the playful experts should represent simulator interactions by playful interactions (analogical representation), in addition to other interactions that are pure for game world. In the problems and progression phase, the pedagogical and playful experts collaborate to design problems (based on domain errors interpretation) challenging player and how he progresses in each level. The decorum phase is the complement of the previous phase, where the playful experts envelop or represent playful interactions with art, graphics, and also the introduction of music, avatar, etc., in addition to calling advice from pedagogical experts. In the last phase, the conditions of use represent how and when to use the targeted SG. Similarly, authors in [13] propose a knowledge management approach, which focuses on early stages of conception and development of a SG aiming to satisfy the requirement of customers from financial institutions. Those customers are interested in teaching new hired employees about the organization, the group structure, its functions, and corporate values. The knowledge management approach goes through four main stages. The first one consists of classifying knowledge according to a structured typology: (a) nature of the knowledge which refers to knowledge characteristics (e.g., confidentiality and degree of relevance) used to communicate with customers for validation and prioritization, (b) type of knowledge which refers to knowledge classification (e.g., factual, declaratory, and procedural), and (c) pedagogical objectives specification which refers to hierarchical view of knowledge type with its pedagogical objectives. The second step deals with game design decision making about the pedagogical objectives integration into game story or gameplay; as suggested by those authors, factual knowledge type should be integrated into game story and procedural knowledge type into gameplay or game mechanics; also this step works on guiding game designer during the definition of game rules. In addition, the creation of a cognitive model step represents domain objects, actions, properties, rules, and their internal interactions to build an expert system. The expert system will be used by the game designer in the final stage to make decision on which variables the player is going to control, the consequences of good/bad actions, rewards, conditions of failure/victory, and frequency of messages or cut-scenes.

From the theoretical framework view, authors in [7] propose an "Input-Process-Output" model underlining the relation between game features (fantasy, curiosity, competitiveness, control, and visual and sound effects), instructional content, and the player, from a motivational and engagement perspective. The model underlines the fact that game features should be mixed with instructional content in order to design an instructional program. These features trigger game cycle to keep the player motivated and engaged while achieving a learning objective, which should be maintained by feedback of player's progression. Also, the model highlights the need for matching game features or game events with learning outcomes, in the cases that game features do not represent directly the instructional content. Finally, the "Input-Process-Output" indicates that SG should propose the mechanics to be played several times. This model could be considered as a metamodel [32] for SG design and has been extended by [24] into a conceptual framework. The proposed framework highlights two perspectives for SG design. The first one focuses on the structure which requires the definition of pedagogical objective and instructional content to design intended learning outcome mixed with game attributes that support learning (i.e., feedback) in order to design learning activities. Based on these activities, the authors believe that the game genre with its mechanics will be easy to choose; while the second one focuses on SG behaviors in order to maintain player motivated and engaged. In addition, authors in [33] propose an iterative cyclic process focusing on leisure game called the "Mechanics, Dynamics, and Aesthetics" framework MDA. The MDA framework aims to design a game from two perspectives; the first one deals with the three components of the game design and their relations, and the second one focuses on how player's interactions are affecting those components. However, the MDA framework focuses only on gameplay. Authors in [30] proposes "Design, Play, and Experience" DEP framework by extending the MDA framework to be explored in SG design context. The DEP framework is defined by five layers: "learning", "storytelling", "gameplay", "user interface", and "technology". Each layer is structured by three interconnected components: "design", "play", and "experience" similar to MDA framework. Also, each layer targets a specific objective in SG design, which goes through design, play, and experience in an iterative way. Also, DEP framework is proposed to underline the influences among those layers.

From mapping of learning theories and game mechanics view, aiming to support educational theories with SG design, authors in [34] propose a highest level mapping between games and learning mechanics called "SG mechanics". The SG mechanics is structured according to bloom's taxonomy cognitive functions, namely, remembering, comprehending, applying, analyzing, synthesizing, and evaluating. In more specified level, authors in [32] propose a taxonomy which links learning functions and game design patterns considered as a technical solution; they investigate inter-patterns relations such as strong dependence "instantiation, composition", a low dependence "modulation, aggregation", and the logical presence in game genre "conflict". On the other hand, authors in [35] aim to use SG design patterns to support communication between various experts during SG game design. Moreover, authors in [36] discussed a matching between existing games genre such as "First-Person-Shoot" (FPS) and learning outcomes. They stated that FPS characteristics are suitable for improving learning function [6].

According to the reusability of prior production of SG view, authors in [27] attempt to standardize prior production into a central repository. They propose a technical architecture to adapt existing and new components in serious context in order to be reused during the SG design. Also,

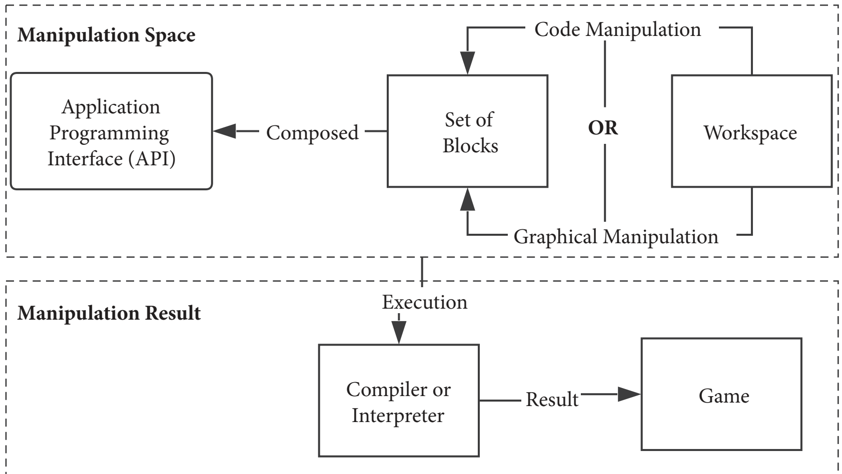


Figure 1: Abstract presentation of the adopted design for teaching computational thinking.

the architecture provides a mechanic that allows the communication (information exchange) between components. In contrast, authors in [28] intend to allow game designer building SG which can interact with existing virtual learning environment without being distracted by standards related to learning via middleware which unifies the communication procedure.

From a complexity technological side, authors in [26] propose high-level tool which provides guidance to educators to design SG via a graphical interface. The tool generates a game for specific platforms (flash, XNA console, etc.) by using a Role-Playing Game (RPG) model.

3. State of the Art Discussion

Although those research works are not exhaustive, their multiple views may quite well represent the SG design aspects relevant to our context. Also, for the classification perspective there is the desire for finding a reduced terminology that represents those and similar works.

Going deeper into the direction of classifying SG design approach, the card game "Problems and Programmers" designed for teaching specific domain content with a deep look, gives clues about how to explore the concepts used in its structure and rules in order to integrate new content; from an abstract view, the programmer card represents a human resource with its personal data such as name and professional data, namely, skills, personality, and salary; based on these data, the players can take decision about what this resource can produce and with what level of quality, the personality to manage collaboration conflict, and the salary to manage budget. For example, the concept presented by the programmer, problems, and concepts cards could embed domain content like manufacture or similar one, and if there is domain content differentiation, it could make an adaptation as presented by "PlayScrum" card game which extends "Problems and Programmers" from teaching "Waterfall model" to teaching "Scrum Agile Framework". In the game "Program your robot", as well as the game proposed by [18] and the web platforms in [37, 38], the authors aim to teach computational thinking; they adopt the same design as that presented in Figure 1, where the Manipulation Space contains specific tools for programming such as Application Programming Interface (API), which exposes a set of functions (preprogramming commands such as "move" and "jump") forming a set of blocks. The player manipulates those functions in order to elaborate an algorithm to complete missions and achieve goals by designing an algorithm into the Workspace, either by code manipulation (e.g., programming language) or by graphical manipulation (drag and drop). The player's algorithm is interpreted or compiled (depending on the proposed solution), and then the result is reflected into the game (see, for example, [18]).

Also, in the "Clean World" game, the mini-game "garbage collection" represents the concept of "items collections" which means that we can collect any items from any domain content (for example, coins, words, potatoes, etc.) and then separate them according to given features, and the mini-games can be changed by any other game only if its game mechanics embedded items collection and items separation concepts. Such a design simplifies the SG design and also investigates the domain content integration into game and it could be considered as architecture; in contrast, it limits the field of use or application and targets one game genre only. Moreover, designing a technical framework is not always investigating domain content integration into game [27, 28].

Mapping between learning theories and game in SG design, that is, the mapping procedure investigating which game aspects can support learning, is quite common. The investigation underlines links between learning theories and game from two views, technical, functional, or both:

(i) functional: abstract level, such as learning function [32, 34, 36], game attributes [24], and game mechanics [34];

(ii) technical: concrete level, such as teaching practices [35], game design pattern [32], and game genre [36].

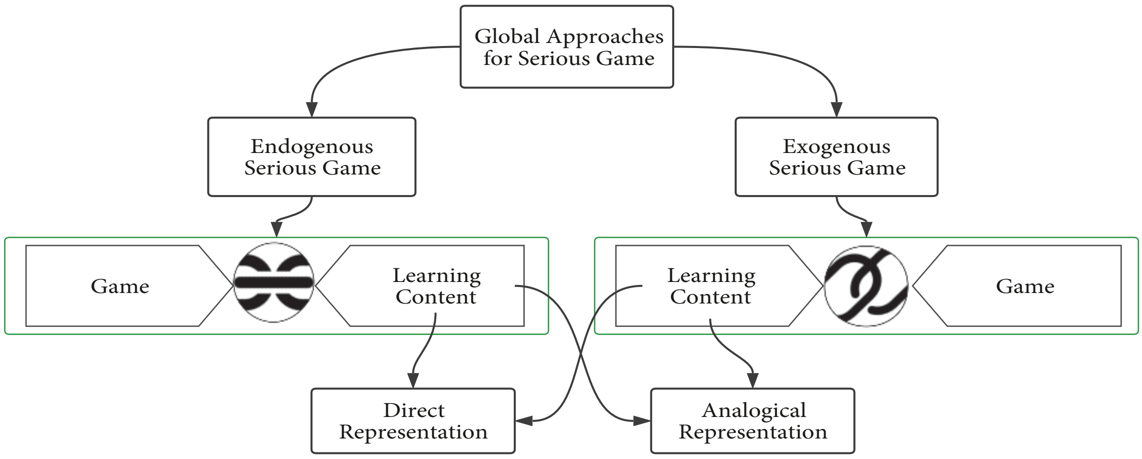


Figure 2: The hierarchical classification of serious games.

However, the mapping of learning theories and game does not consider domain content and, more importantly, they need contextualization where they could be explored efficiently such as a process model [35] besides a functional or a technical architecture (Figure 3):

(i) A process model defines the way of collaboration between different experts, in which we focus on external structure of SG design (analogically, car production process); see, for example, a comparative study of process model of serious game design [39]. However, the presence of all various experts (heavy and high cost) is mandatory at every time it is needed to design a SG, which limits the reusability of the proposed structure and organization.

(ii) A technical architecture aims to simplify the SG design and limits the field of application—in most cases targeting one game genre.

(iii) A functional architecture consists of describing in a symbolic and schematic manner the different elements of the system, their interrelations, and their interactions in order to meet the system specifications, emphasize intention and objectives, provide functional decomposing, influence the process model, limit the intervention of stakeholders by providing guidance, simplify translating into a technical architecture, and most importantly improve the reusability; in contrast, it has a high complexity in realization [40–42].

4. Toward a Comprehensive Taxonomy to Classify Research Works Dealing with Serious Game Design

Relying on the work of [43], SG can be classified into the following principal categories:

(i) market: the kind of market that uses them such as healthcare;

(ii) purpose: the purpose they are designed to serve.

Also, a combination of criteria can be adopted to categorize SG, providing more specified hybrid taxonomy based on market and purpose of the SG, which cannot be clearly classified. Moreover, the authors introduced their own model combining the gameplay, purpose, and scope, believing that SGs are composed of both "serious" and a "games" dimensions.

According to researchers in the field, designing SG could be done by two global approaches, exogenous and endogenous [7, 12, 13, 30, 44, 45]. The endogenous or intrinsic approach claims that domain content and game should be naturally embedded or tied. It consists of integrating the domain content into game structure and rules, where the gameplay represents the learning content which is necessary for game goal achieving. In addition, as the game is interesting, the content becomes interesting, and it is considered as a good approach to create better SG requiring to start by a blank board and to make the domain content in the centre of the design process; also it may target SG with more complex learning goals. In contrast, the exogenous or extrinsic approach considers that the domain content and the game are unrelated, which means that the game represents a simple wrapper for domain content. The gameplay does not represent the domain content and it is separated from the learning content. Such games use preexisting game structure and rules [7, 12, 13, 30, 44, 45].

Exploring the two general approaches discussed earlier (the exogenous and the endogenous) seems to be a good start for going into the proposition of a new classification; they split approaches addressing SG design into two distinct categories (Figure 2). The exogenous category represents SG in which the learning content is separated from game representing a heterogeneous entity—the gameplay is indirectly related to the learning content. In contrast, the endogenous category represents SG in which the learning and the game representing a homogeneous entity—the gameplay is directly related to the learning content.

Both categories (the exogenous and the endogenous) definitions do not specify learning content representation into the game world, which leads to a freedom in learning

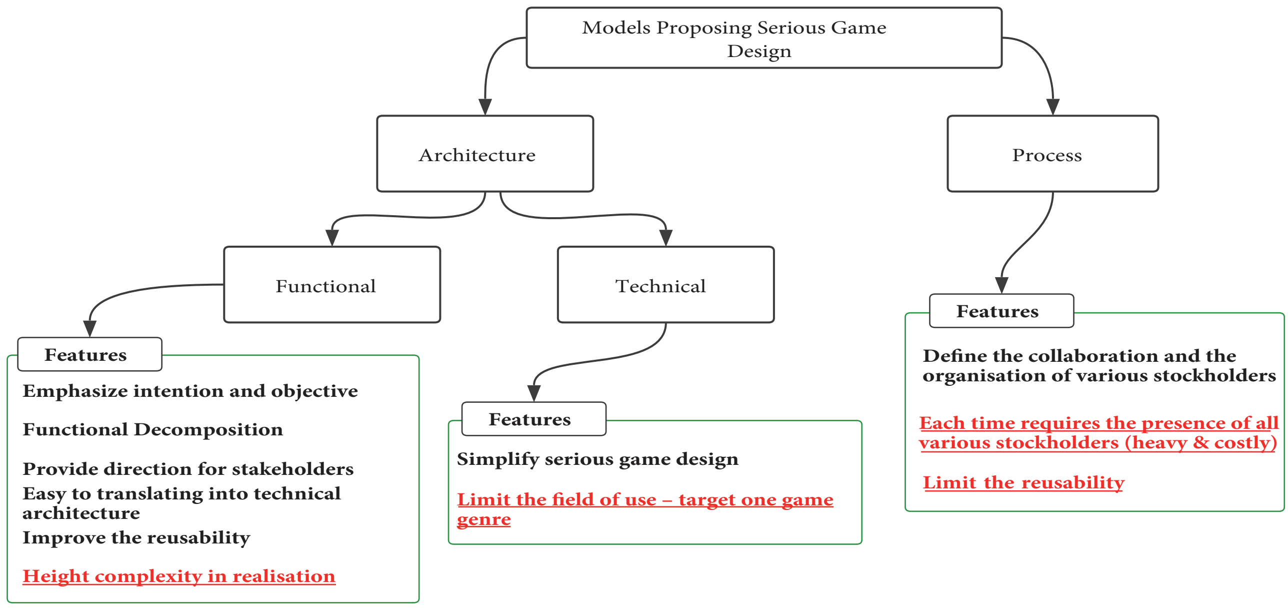


Figure 3: The hierarchical classification of models approaching SG design.

content design; as stated by [7] "students may learn about physics by piloting a spaceship on reentry to earth's orbit"; from this statement it is difficult to claim that the learning content is directly represented in the game world or it is represented differently. In contrast, authors in [12] state that "the students must defend a territory (the metaphor of the body) by adjusting the defences (metaphor of the immune system)"; it is clear that the learning content is represented by an analogical representation, as both statements refer to endogenous approach. In the same way of the exogenous approach, authors in [7] state that "children may learn fractions and by doing so slay a dragon in an enchanted forest", which refers to an analogical representation. In contrast, authors in [13] report an experience which consists of using a SimCity-style building game accompanied by a knowledge base; in order to play the game the learner should earn points by responding to questions proposed outside the game; also, authors in [19] propose how to explore the Angry Birds game with video analysis and modeling software (Tracker) for teaching kinematics and dynamics in physics field. This fact leads us to subdivide learning content element into two subcategories: direct representation and analogical representation.

As stated before, exogenous and endogenous approaches both represent a general solution for SG design, but for exogenous approach, the solution is limited to the reuse of preexisting successful games and the learning content could be either inserted or related by another means to the game world. By contrast, for endogenous approach, the solution defines the intention of designing an engaged and an effected SG, which means that the SG design challenge is ongoing. As presented earlier in the State of the Art Discussion, several models emerged approaching the challenge by the proposition of either of three categories: a process model, a technical architecture, or a functional architecture. In order to complete our taxonomy, SG design should be classified also according to the three categories as a second level classification (Figure 3).

In order to classify existing research works dealing with serious games design, we propose a grid which examines these works according to some questions as summarized in Table 1. The grid is divided into two levels; the first one concerns works proposing serious game implementation, and the second one concerns those who are proposing model to design serious game accompanied or not by implementation.

As mentioned before, the endogenous approach is considered as a good one to create better SG and proved by models presented in the State of the Art targeting such category, except the works that support SG design such as the mapping approach. The proposed taxonomy will be adopted to classify those models according to endogenous approach (Table 2). The sequences of this table can be explained as follows: The first column represents models presented in the State of the Art which discuss only SG design. The second column identifies the domain content representation directly or analogically or both into the game world ("Yes" or "No" and "Yes!" or "No!", probable). The third column shows the nature of the models which could be a process model or a functional or a technical architecture or all of them ("Yes" or "No" and "Yes!" or "No!", probable).

The solution we want to present to design serious games (see the next section) is entirely based on the concept of the functional separation of the design of playful aspects from serious aspects; instead, in the authors' proposal [24] the definition of pedagogical objective and instructional content to design intended learning outcome are designed separately, but these last are mixed with game attributes in order to design learning activities. Moreover, we found in the authors' proposal [12] that the definition of problems and progression phase requires the presence of pedagogical

Table 1: The grid of questions used during the classification.

Questions	Clues	Examples	Criteria
Level one: works proposing serious game implementation			
How to play the Serious Game?	Playing the Game requires knowledge of the teaching content.	Playing the Game proposed by the authors in [16–18] requires knowledge of the teaching content.	Endogenous
	Playing the Game is the reward after completing a serious activity where the game is used to teach content.	Playing the game ("SimCity-style building" [13]) is the reward after completing a serious activity (exploring the knowledge base) or the game ("Angry Birds" [19]) is used to teach content (kinematics and dynamics in physics field).	Exogenous
How teaching content is represented in the serious game?	The serious game clearly presents the teaching content.	The games presented in [16–18] clearly represent the teaching content, except the case of two games [17, 18] which incorporate symbolic representations (the reader is invited to consult the experience on the advantage of symbolic representations for learning [20]).	Direct Representation
	The serious game represents the teaching content with different elements from another domain.	As suggested by the authors in [12], to teach the immune system (the source domain) is like teaching the defence of the territory (the target domain); also, the authors in [7] mention that when fighting a dragon the children can learn fractions; more precisely the authors in [21] clearly specify the use of analogies in serious game design. However, in order to gain insight into the importance of using analogical representations in the learning process, the reader is invited to read our recent work on analogical representations [22].	Analogical Representation
Level two: model to design serious game accompanied or not by an implementation			
How does the model structure the serious game design?	The model proposes a set of **organized activities** that interact to achieve a result (often refers to a product)	The model proposed by the authors in [12, 13] presents a set of organized activities to conceive a serious game.	Process
	The model proposes an **implementation** that targets a single type of game that accepts the integration of a single content type.	As we pointed out in the Discussion, the game structure "Problems and Programmers" [16] can be used to integrate content from the manufacturing domain. Other examples can be found in Google Play Store at the web address referenced in [23] in which we can find two serious games that have the same structure and mechanics, where the first one aims to help children memorize the "Basic Music Notes" and the second one for memorizing the "Tifinagh Alphabet", which is the writing used by Berbers in North Africa.	Technical architecture
	The model proposes a set of **functionality** for designing serious games.	The model proposed by the authors in [24] presents a set of functionality for designing serious games.	Functional Architecture

and playful experts. Strong separation reduces constraints on the concern of both aspects during SG design. In order to achieve this goal, we will first separate the playful and serious aspects; then we propose functional decomposition of the serious aspects independently of the playful aspects; and then we will present how the two aspects interact with each other.

5. Toward an Architecture for Serious Games Design

5.1. The Architecture Description: Overview. All the works discussed so far lead to an important conclusion: education is a sensible field and SG design (if not well-defined) will not solve all of its challenges. However, researches and

Table 2: Synthesis which presents the classification of serious game and models addressing SG design discussed in state of the art according to the endogenous approach.

Works	Content representation into the game		Models proposed for serious game design		
			Process	Architecture	
	Directly	Analogical		Functional	Technical
An experimental card game for teaching software engineering processes [16].	Yes	No	No	No	Yes
PlayScrum - A Card Game to Learn the Scrum Agile Method [25].	Yes	No	No	No	Yes
A serious game for developing computational thinking and learning introductory computer programming [17].	Yes	No	No	No	Yes
A Platform Independent Game Technology Model for Model Driven Serious Games Development [26].	Yes	No	No	No	Yes
RAGE Architecture for Reusable Serious Gaming Technology Components [27].	Yes	No	No	No	Yes
A general architecture for the integration of educational videogames in standards-compliant virtual learning environments [28].	Yes	No	No	Yes	Yes
A New Methodology of Design and Development of serious game [29].	Yes	No	No	No	Yes
The design of an analogical encoding tool for game-based virtual learning environments [21].	Yes	Yes	No	No	Yes!
The six facets of serious game design [12].	Yes!	Yes	Yes	No	No
Knowledge Management Approach to Support a Serious Game Development [13]	Yes	Yes!	Yes	No	No
Games, motivation, and learning: A research and practice model [7].	Yes	No	No	Yes	No
A conceptual framework for serious games [24].	Yes	No	No	Yes	No
The design, play, and experience framework [30].	Yes	No	No	Yes	No

professional working in SG design agree that SG is composed of three main components: domain content, game, and learning theories. Each component requires several stakeholders or experts from different background/creativity. Therefore, the design of a SG with respect to expert concerns makes a SG process as hard as a complex system, or even more. However, due to the heterogeneousness fields of knowledge implied in the design of SG, communication may influence the collaboration between these experts. And, it is obvious that when the extent and the complexity of the domain content become important, each expert sees the game from a given angle and communicates with others using a specific language of his own field of expertise; therefore confusion in the achievement of the objectives is likely to happen and, hence, results may be dominated by either the learning aspect or the playful one. However, there is a necessity to develop a process model or architecture oriented application (a functional architecture) or technical one which is not highly recommended. The architecture oriented application or functional architecture is the adequate base to start with. It provides a functional decomposing by defining clearly which functionalities are required to conceive a SG, and each functionality represents a container delimited by a clear edge, which naturally limits the intervention of all various stakeholders/experts most importantly it emphasizes intention and objectives by defining clearly how SG could be made (global vision), based on the right decisions and questions that could be drawn. The architecture should contain functionalities (most presented in literature and not exhaustive) like domain content and learning theories, pedagogy, learner profile, and game (Figure 4).

However, the domain content and learning theories, the pedagogy, and the learning profile have a common context representing serious sides which make them grouped into one entity called educational robot; on the other hand, all game aspects will be represented into one entity called game.

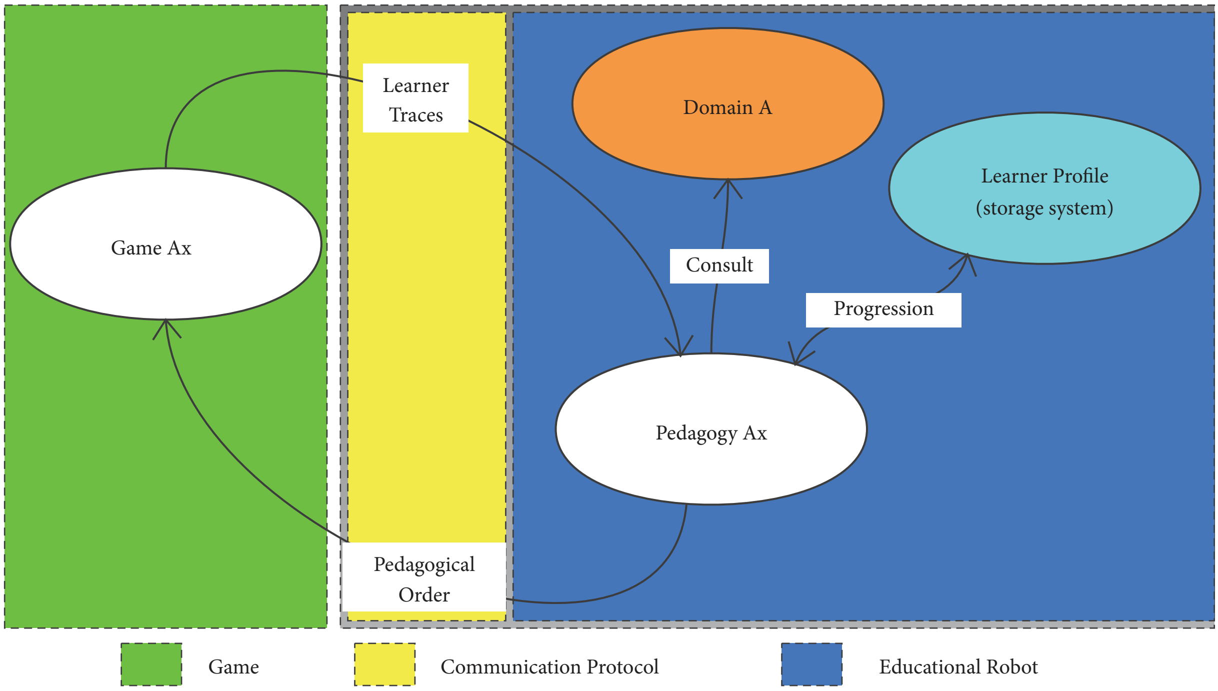


Figure 4: The proposed architecture for serious games design.

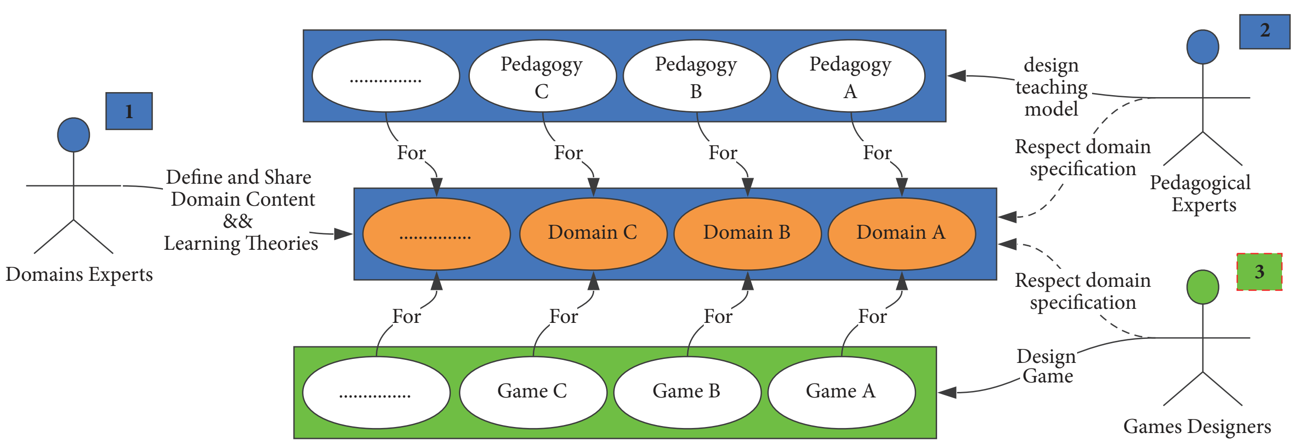


Figure 5: Presentation of serious games production process according to the proposed architecture; the numbers in the squares represent intervention order.

The educational robot and the game should exchange data or information; the game knows what the learner/player is doing, and by communicating relevant information (learner traces) to the educational robot, the last one knows what the player has done, and by analyzing learner's traces a decision is communicated back to the game (pedagogical order) to be applied such as repeating the same level with less or higher complexity, going to next level, and going back to previous level. The proposed architecture promotes the reusability, meaning functionality can be replaced or improved without or with miner influence on other functionalities, and provides guidance and influences the process by specifying how experts collaborate, Figure 5, as follows:

(i) The domain content and learning theories are the responsibility of domain expert, in which he defines domain content specifications (an example is presented in Figure 6), and then the domain content can be shared with pedagogical and playful experts.

(ii) The pedagogy is the responsibility of pedagogical expert. Based on the shared domain content specifications, he can draw a learning model which takes into consideration how to teach content in addition to all decisions required to help learners to overcome the faced difficulty to validate each part from domain content and to keep track of learner progression during several sessions, for example, by designing a

```xml
<?xml version="1.0" encoding="UTF-8"?>
<structure name="optional_name" id="unique_identifier" version="1.0.0" domain="Domain_Title">
 <level id="unique_identifier" desc="Chapter_Title">
  <layer id="unique_identifier" desc="Section_Title">
    <block id="unique_identifier" desc="Content_Title" >
      <cognitive>
       <remember complexity="1 ... 5" score="1 ... 20" type="conceptual;...;principle">
            Content Description.
       </remember>
       <understand complexity="1 ... 5" score="1 ... 20" type="conceptual;...;principle" >
            Content Description.
       </understand>
       <apply complexity="1 ... 5" score="1 ... 20" type="conceptual;...;principle">
            Content Description.
       </apply>
      </cognitive>
    </block>
    <block id="unique_identifier" desc="Content_Title">
      <cognitive>
        <analyze complexity=" 1 ... 5" score="1 ... 20" type="conceptual;...;principle" >
            Content Description.
        </analyze>
        <evaluate complexity="1 ... 5" score="1 ... 20" type="conceptual;...;principle">
            Content Description.
        </evaluate>
        <create complexity="1 ... 5" score="1 ... 20" type="conceptual;...;principle">
            Content Description.
        </create>
      </cognitive>
      <dependances>
        <level id="Level_Id" />
        <layer id="Layer_Id" />
        <block id="Bloc_Id" />
      </dependances>
    </block>
  </layer>
 </level>
 <level id="unique_identifier" desc="Chapter_Title" >
       .......................................
 </level>
</structure>
```

Figure 6: A proposal for domain content specifications, designed using XML (Extensible Markup Language) to represent domain content specifications, in which the hierarchy is structured by level close to chapter, layer to section, and block to subsection. The block is composed of a cognitive tag that contains the required learning function and its attributes in addition to the dependences tag which refers to the previous block, layer, or level; in the case that the block depends on a given layer or level, it is necessary to review all blocks that contain these dependences.

pedagogical expert system or a pedagogical agent [11], or see our example in Figure 7.

(iii) The game is the responsibility of playful expert or game designer. Based on the shared domain, he can design a game that respects content specifications.

5.2. The Architecture Description: Conceptual Level. In order to illustrate how SG game could be designed according to the proposed architecture, we propose an example (varying depending on the involved experts) of domain content specifications, Figure 6, and then we present how educational robot works and how it communicates with the game by using flowchart (a type of diagram that represents an algorithm), Figure 7.

The design of domain content specifications should make the domain content meaningful, understandable, and easy to read, in which the domain expert defines the structure of the content to be taught (e.g., the hierarchy) and then indicates for each content the learning function that is required [46], the content type (e.g., conceptual, fact, procedural, and principle) [46], the complexity or difficulty according to a specific standard scale (e.g., from 1 to 5), dependences between domain content hierarchy components, and finally a min score (e.g., from 1 to 20). The domain content hierarchy represents how learner should progress into the domain content, the learning function and the content type underline how content should be learned, the min score represents the minimal score to validate the associated part of content, the complexity or difficulty highlights the associated part of content useful for content representation, and finally the dependences refer to prerequisite activities for a given activity in the domain content hierarchy (e.g., activity N depends on activities N-1, N-4,...). The term activity is used to represent a position in the domain content hierarchy. The design of game should respect the domain content specification such as the hierarchy (e.g., level design or game activity such as

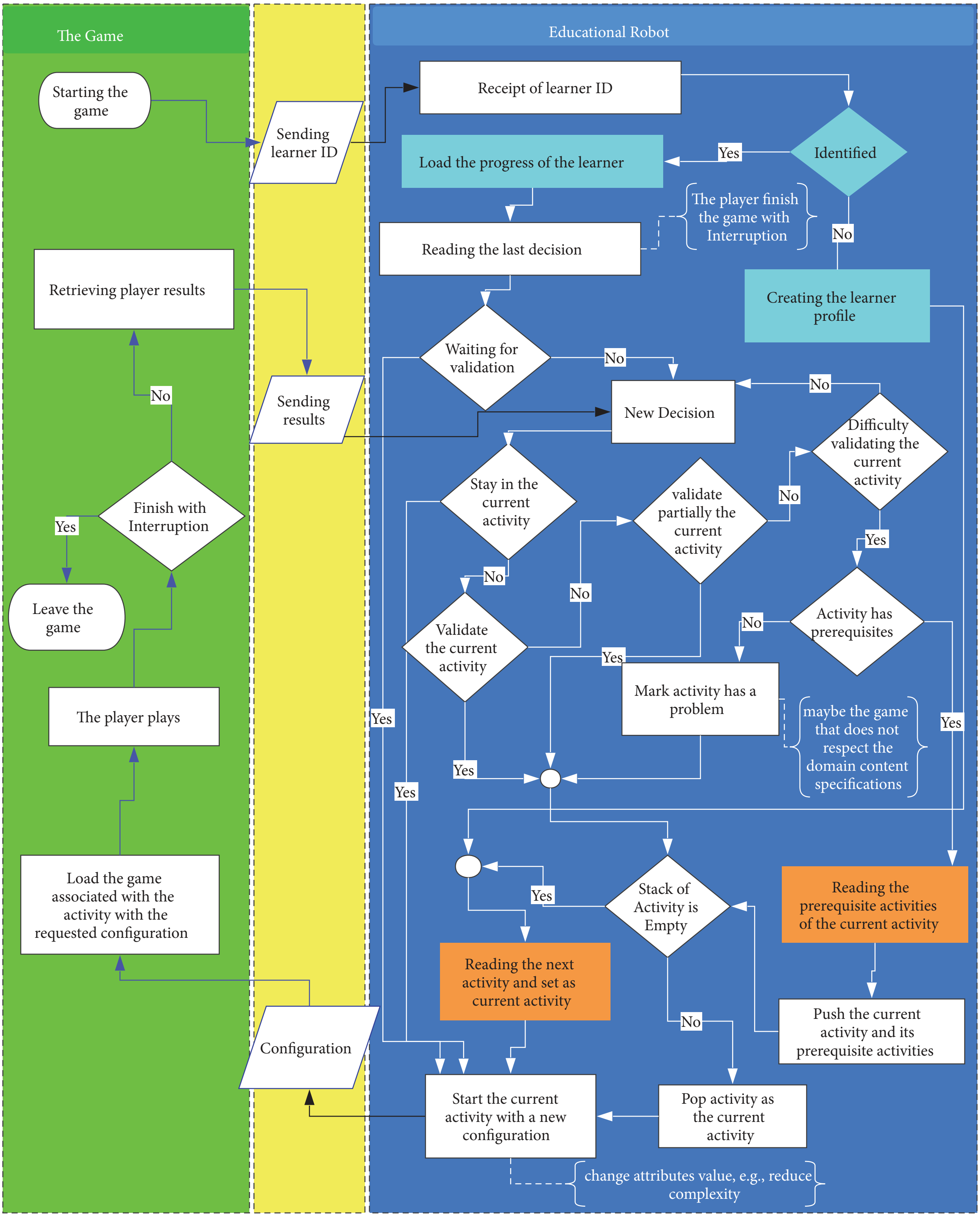


Figure 7: An example that shows implementation of the educational robot and how it communicates with the game.

mini-game), the learning function required and the content type (e.g., game mechanics), and the complexity (e.g., game adaptation).

5.3. The Architecture Description: Practical Level. The proposed architecture offers many advantages and benefits. In order to demonstrate that it is relevant and can handle real situations related to the serious game design, we proposed to use a case from two developed serious games which are "Problems and Programmers" [16] and "SimSE" [47] as a running example. Therefore, the objective relies on implementing these two games based on the proposed architecture. The two games which are described below share the same educational objective, teaching the software project management and the life cycle model, specially Waterfall. However, each game uses a different presentation and gameplay (game card and game based simulation). As pointed out in State of the Art, the "Problems and Programmers" [16] game represents a competitive game in which each student plays the role of a project manager, but the "SimSE" [16] game aims to teach project management in a single player mode. Moreover, the player drives the process by, among other things, hiring and firing developers, assigning tasks to them, monitoring their progress, and purchasing tools. At the end of the game, the player receives a score indicating how well they performed. Based on this description, we brief in the following points the main elements which describe how the domain content is represented in these two games.

At the start of SimSE game, the player reviews the resources and assesses what they have to work with. All resources have predefined values which constitute the current context or configuration. The resources are listed below:

(i) The project is characterized by description, budget, money spent, allotted time, and time used.

(ii) The customer is characterized only by name.

(iii) The purchased tools are characterized by name and cost and organized by categorizing their use.

(iv) The employee is characterized by name, energy, mood, pay rate, and expertise's years in requirement, design, coding, and testing.

In the first step of "Problems and Programmers" game, the player starts with picking up a card from the project deck and then five cards from the main deck which contains cards representing programmers, concepts, and problems. Then, in each turn the player is required to take two cards from the documentation deck which contains code cards. All the picked cards and those in the decks have predefined values which constitute the current context or configuration. The card types are listed below:

(i) The project card is characterized by name, budget, quality, length, and complexity.

(ii) The problem card is characterized by name, condition, and description to show consequences.

(iii) The concept card is characterized by name and description showing the effect, decisions which a player can make.

Table 3: Example of three mission configurations.

Difficulty level	Time	Quality	Budget	Length
Simple	Max	Min	Max	Min
Intermediate	[Min, Max]	[Min, Max]	[Min, Max]	[Min, Max]
Difficult	Min	Max	Min	Max

(iv) The programmer card is characterized by name, short expertise description, salary, personality, and skill.

The architecture underlines the fact that domain expert will be the first to design the teaching content. We use two existing serious games as a running example; we will provide generalized example of the structure of teaching subject (project manager), in which the project to be managed will be represented by a mission structured by several phases following the Waterfall model. Each mission is characterized by several constraints (time, quality, budget, etc.) and it requires managing several resources represented by collaborators. The collaborators also are characterized by expertise regarding each phase, degree of collaboration, and other factors. Each phase contains obstacles referring to the problems that may occur. Note that the subject represents an atomic entity, meaning that we cannot teach an element (e.g., that collaborators chose) without its relations with other elements (e.g., project). As presented by those two games, all elements have predefined values, which represent the current context or configuration based on which the player/learner make decisions/actions.

The atomic aspect necessitates that the learner progresses from a simple configuration to a more difficult one. Tables 3, 4, and 5 represent our example of domain design, which shows some basic configurations of the mission, collaborators, and obstacles. Each characteristic could accept a value within a [Min, Max]. Figure 8 represents a basic design of learner progression from simple to more difficult configuration, by using combination from Tables 3, 4, and 5.

By applying the architecture on the two games, we separate the design of the serious aspect, which is in our case the educational robot, from the playful aspect (the game). The educational robot uses the hierarchical domain content design (Figure 8) to identify how learner should progress from a simple configuration to a complicated one. The educational robot establishes the communication with the game as follows: (1) Select a configuration and ask the game to apply it. (2) The game applies the configuration and sends back the result at the end of the activity. (3) The educational robot checks the result and then makes decision (selecting the next configuration or retrying the previous one) as shown in Figure 9(a). This process is repeated until the end of the content hierarchy.

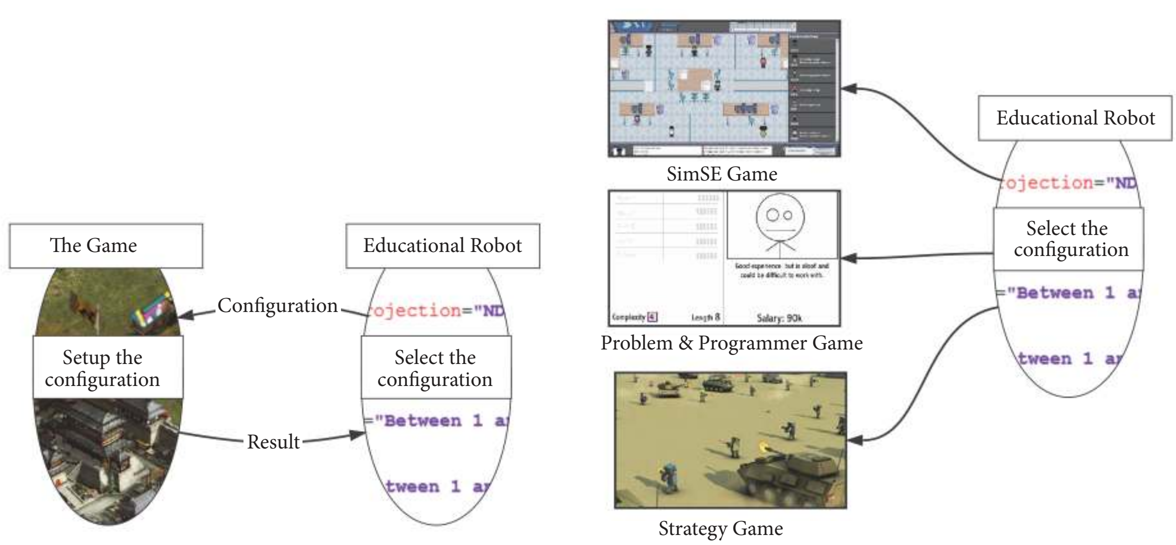
The same domain content will be integrated into different games: card game "Problems and Programmers", game based simulation "SimSE", and strategy games that share the same concept of domain design (analogical representation) as illustrated in Figure 9(b). In this running example of applying the proposed architecture in two existing serious games, the implementation was backward contrarily to the

Table 4: Example of collaborators configurations.

Quality level	Name	Collaboration degree	Consumption	Power	Productivity			
					Phase a	Phase b	Phase c	Phase d
Low	Coll 1	Min	Min	Min	Min	Min	Min	Min
Intermediate	Coll 2	[Min, Max]	[Min, Max]	[Min, Max]	[Min, Max]	[Min, Max]	[Min, Max]	[Min, Max]
Hot	Coll 3	Max	Max	Max	Max	Max	Max	Max

```
<?xml version="1.0" encoding="UTF-8"?>
<structure name="optional_name" id="1" version="1.0.0" domain="Project_Manager">
  <level id="1" desc="Chapter_Title">
   <layer id="1" desc="overall educational objective">
     <block id="1" desc="educational objective" complexityLevel ="simple" "Validationscore="" >
          Difficulty level simple mission, obstacles
                              and Quality level low collaborators
       </block>
       <block id="2" desc="educational objective" complexityLevel ="intermediate" " Validationscore="" >
          Difficulty level intermediate mission, obstacles
                              and Quality level intermediate collaborators
          <dependances>
                 <block id="1" />
          </dependances>
       </block>
       <block id="3" desc="educational objective" complexityLevel ="complexes" " Validationscore="" >
          Difficulty level complexes mission, obstacles
                       and Quality level hot collaborators
          <dependances>
                <block id="2" />
          </dependances>
       </block>
   </layer>
  </level>
</structure>
```

Figure 8: An example of the hierarchical domain content design.



(a) (b)

Figure 9: (a) Communication between the educational robot and the game. (b) Integration of the educational robot into different games.

Table 5: Example of problems/obstacles configurations.

Difficulty level	Number	Complexity Phase a	Complexity Phase b	Complexity Phase c	Complexity Phase d
Simple	Min	Min	Min	Min	Min
Intermediate	[Min, Max]	[Min, Max]	[Min, Max]	[Min, Max]	[Min, Max]
Difficult	Max	Max	Max	Max	Max

architecture process. We started by identifying the elements which constitute the domain content and then we designed an abstracted domain content that was encapsulated in the educational robot. This robot mapped successfully with the two games. This success demonstrated the feasibility of the proposed architecture. Based on the reusability aspect of this architecture, the educational robot can be also integrated into other games respecting the elements of the abstracted domain content. The proposed architecture is designed to work according to the presented process, Figure 5, in which the domain expert designs domain content and shares it (in public repository). The pedagogical expert designs a teaching model for the shared domain content encapsulated into an educational robot, which can be implemented as an API (Application Programming Interface) or as a Web Service. The playful experts design a game for the shared domain content and integrated educational robot. The main advantages can be briefed in the following points:

(i) modifying decisions of the educational robot without affecting the game;

(ii) responding to new educational robot updates by simple game improvements;

(iii) reusability of the educational robot with new designed games.

6. Conclusion

The presented taxonomy represents a guide to new SG researchers; they will be capable of analyzing state of the art of proposed SG design models, overcoming the lack of standards in classifying these models. The other advantage realized by the proposed taxonomy is helping SG designers to choose the adequate category/model according to their perspective (endogenous/exogenous). Also, the proposed architecture indicates how stockholders collaborate and how SG game is made, allowing domain experts to focus on defining their knowledge without worrying about the playful aspects, and the game designers to focus on building the game without having deep knowledge of the domain content and more importantly the pedagogical aspects. Such architecture presents the following benefits:

(i) Separation during the SG design reduces the dependences and the overlaps of the various expert's concerns and encapsulates expert's intellectual production in a computer component such as an expert system, which replaces the intervention of the human expert. When we need specific domain knowledge, we use the expert's product without its presence.

(ii) Collaboration during the use with less constraints reduces the effort and integration time of game and serious expert's production.

(iii) Reuse of both serious and game expert's production develops several serious games for one domain content, allowing the change or the improvement of a serious component with less changes in the game component and vice versa.

All these features (separation, collaboration, and reuse) lead to an open market of SG based on the reusability without implying expert's presence "commercial off-the-shelf" [48, 49]. In the open market places, in which we can find that SG components encapsulate expert's knowledge specific to the learning theories, the domain content, and the game, the SG is designed as separate components and in each one specific concern is encapsulated (expertise production), so that the SG can be produced by combining those components. In future work we will design a simple serious game that presents implementation of the proposed architecture.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] J. M. Allen, S. Wright, N. Cranston, J. Watson, K. Beswick, and I. Hay, "Raising levels of school student engagement and retention in rural, regional and disadvantaged areas: is it a lost cause?" *International Journal of Inclusive Education*, pp. 1–17, 2017.

[2] F. M. Newmann, *Student engagement and achievement in American secondary schools*, vol. 1234, Amsterdam Avenue, New York, NY, USA, Teachers College Press, 1992.

[3] J. A. Fredricks, P. C. Blumenfeld, and A. H. Paris, "School engagement: Potential of the concept, state of the evidence," *Review of Educational Research*, vol. 74, no. 1, pp. 59–109, 2004.

[4] D. J. Shernoff, M. Csikszentmihalyi, B. Schneider, and E. S. Shernoff, "Student engagement in high school classrooms from the perspective of flow theory," *Applications of Flow in Human Development and Education: The Collected Works of Mihaly Csikszentmihalyi*, pp. 475–494, 2014.

[5] D. R. Michael and S. L. Chen, *Serious games: Games that educate, train, and inform*, Muska & Lipman/Premier-Trade, 2005.

[6] I. Granic, A. Lobel, and R. C. M. E. Engels, "The benefits of playing video games," *American Psychologist*, vol. 69, no. 1, pp. 66–78, 2014.

[7] R. Garris, R. Ahlers, and J. E. Driskell, "Games, motivation, and learning: a research and practice model," *Simulation & Gaming*, vol. 33, no. 4, pp. 441–467, 2002.

[8] B. Bergeron, *Developing serious games*, game development series, 2006.

[9] M. Zyda, "From visual simulation to virtual reality to games," *The Computer Journal*, vol. 38, no. 9, pp. 25–32, 2005.

[10] T. Marsh, "Serious games continuum: Between games for purpose and experiential environments for purpose," *Entertainment Computing*, vol. 2, no. 2, pp. 61–68, 2011.

[11] D. A. Lieberman, "What can we learn from playing interactive games?" *Playing Video Games: Motives, Responses, and Consequences*, pp. 447–469, 2006.

[12] B. Marne, B. Huynh-Kim-Bang, and J.-M. Labat, "Articuler motivation et apprentissage grâce aux facettes du jeu sérieux," in *EIAH 2011-Conférence sur les Environnements Informatiques pour l'Apprentissage Humain*, pp. 69–80, Editions de l'UMONS, Mons, 2011.

[13] B. C. Ibáñez, V. Boudier, and J.-M. Labat, "Knowledge management approach to support a serious game development," in *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies, ICALT '09*, pp. 420–422, Latvia, July 2009.

[14] J. Kirkley, S. Kirkley, and J. Heneghan, "Building bridges between serious game design and instructional design," *The design and use of simulation computer games in education*, vol. 2, p. 74, 2007.

[15] S. De Freitas and S. Jarvis, *A framework for developing serious games to meet learner needs*, 2006.

[16] A. Baker, E. O. Navarro, and A. Van Der Hoek, "An experimental card game for teaching software engineering processes," *The Journal of Systems and Software*, vol. 75, no. 1-2, pp. 3–16, 2005.

[17] C. Kazimoglu, M. Kiernan, L. Bacon, and L. Mackinnon, "A Serious Game for Developing Computational Thinking and Learning Introductory Computer Programming," *Procedia - Social and Behavioral Sciences*, vol. 47, pp. 1991–1999, 2012.

[18] M. Muratet, P. Torguet, J.-P. Jessel, and F. Viallet, "Towards a serious game to help students learn computer programming," *International Journal of Computer Games Technology*, vol. 2009, Article ID 470590, 12 pages, 2009.

[19] M. Rodrigues and P. Simeão Carvalho, "Teaching physics with angry birds: Exploring the kinematics and dynamics of the game," *Physics Education*, vol. 48, no. 4, pp. 431–437, 2013.

[20] V. M. Sloutsky, J. A. Kaminski, and A. F. Heckler, "The advantage of simple symbols for learning and transfer," *Psychonomic Bulletin & Review*, vol. 12, no. 3, pp. 508–513, 2005.

[21] D. Williams, Y. Ma, S. Feist, C. E. Richard, and L. Prejean, "The design of an analogical encoding tool for game-based virtual learning environments," *British Journal of Educational Technology*, vol. 38, no. 3, pp. 429–437, 2007.

[22] W. Mestadi, K. Nafil, R. Touahni, and R. Messoussi, "Knowledge Representation by Analogy for the Design of Learning and Assessment Strategies," *International Journal of Modern Education and Computer Science*, vol. 9, no. 6, pp. 9–16, 2017.

[23] https://play.google.com/store/apps/developer?id=Maxware.

[24] A. Yusoff, R. Crowder, L. Gilbert, and G. Wills, "A conceptual framework for serious games," in *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies, ICALT '09*, pp. 21–23, Latvia, July 2009.

[25] J. M. Fernandes and S. M. Sousa, "PlayScrum - A card game to learn the scrum agile method," in *Proceedings of the 2nd IEEE International Conference on Games and Virtual Worlds for Serious Applications, VS-GAMES 2010*, pp. 52–59, Portugal, March 2010.

[26] S. Tang, M. Hanneghan, and C. Carter, "A platform independent game technology model for model driven serious games development," *Electronic Journal of E-Learning*, vol. 11, no. 1, pp. 61–79, 2013.

[27] W. Van Der Vegt, W. Westera, E. Nyamsuren, A. Georgiev, and I. M. Ortiz, "RAGE Architecture for Reusable Serious Gaming Technology Components," *International Journal of Computer Games Technology*, vol. 2016, 3 pages, 2016.

[28] Á. Del Blanco, J. Torrente, P. Moreno-Ger, and B. Fernández-Manjón, "A general architecture for the integration of educational videogames in standards-compliant Virtual Learning Environments," in *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies, ICALT '09*, pp. 53–55, Latvia, July 2009.

[29] A. F. S. Barbosa, P. N. M. Pereira, J. A. F. F. Dias, and F. G. M. Silva, "A New Methodology of Design and Development of Serious Games," *International Journal of Computer Games Technology*, vol. 2014, Article ID 817167, 8 pages, 2014.

[30] B. M. Winn, "The design, play, and experience framework," in *Handbook of research on effective electronic gaming in education*, pp. 1010–1024, IGI Global, 2009.

[31] P. Rooney, "A theoretical framework for serious game design: exploring pedagogy, play and fidelity and their implications for the design process," *International Journal of Game-Based Learning*, vol. 2, no. 4, pp. 41–60, 2012.

[32] S. Kelle, R. Klemke, and M. Specht, "Design patterns for learning games," *International Journal of Technology Enhanced Learning*, vol. 3, no. 6, pp. 555–569, 2011.

[33] R. Hunicke, M. Leblanc, and R. Zubek, "MDA: A formal approach to game design and game research," in *Proceedings of the 19th National Conference on Artificial Intelligence*, pp. 1–5, USA, July 2004.

[34] N. Suttie, S. Louchart, T. Lim et al., "In persuit of a 'serious games mechanics' : A theoretical framework to analyse relationships between 'game' and 'pedagogical aspects' of serious games," in *Proceedings of the 4th International Conference on Games and Virtual Worlds for Serious Applications, VS-GAMES 2012*, pp. 314-315, Italy, October 2012.

[35] B. Marne, J. Wisdom, B. Huynh-Kim-Bang, and J. Labat, "The Six Facets of Serious Game Design: A Methodology Enhanced by Our Design Pattern Library," in *21st Century Learning for 21st Century Skills*, vol. 7563 of *Lecture Notes in Computer Science*, pp. 208–221, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[36] J. Sherry and A. Pacheco, "Matching computer game genres to educational outcomes," *Electronic Journal of Communication*, vol. 16, no. 1, p. 2, 2006.

[37] https://scratch.mit.edu.

[38] https://code.org/learn.

[39] L. Bennis and S. Benhlima, "Comparative study of the process model of Serious Game Design through the generic model DICE," in *Proceedings of the 1st International Conference on Intelligent Systems and Computer Vision, ISCV 2015*, Morocco, March 2015.

[40] M. Mancona Kandé and A. Strohmeier, "On the role of multidimensional separation of concerns in software architecture,"

in *OOPSLA Workshop on Advanced Separation of Concerns in Object-Oriented Systems*, Minneapolis, Minnesota, USA, 2000.

[41] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Addison-Wesley Professional, 2003.

[42] I. Jacobsson, M. Christerson, P. Jonsson et al., *Object-Oriented Software Engineering*, 1992.

[43] D. Djaouti, J. Alvarez, and J. Jessel, "Classifying Serious Games," in *Handbook of Research on Improving Learning and Motivation through Educational Games*, Advances in Game-Based Learning, pp. 118–136, IGI Global, 2011.

[44] M. P. J. Habgood, S. E. Ainsworth, and S. Benford, "Endogenous fantasy and learning in digital games," *Simulation & Gaming*, vol. 36, no. 4, pp. 483–498, 2005.

[45] C. Fabricatore, *Learning and videogames: An unexploited synergy*, 2000.

[46] L. W. Anderson, D. R. Krathwohl, P. W. Airasian et al., *A taxonomy for learning, teaching, and assessing: A revision of Blooms taxonomy of educational objectives , abridged edition*, Longman, White Plains, NY, 2001.

[47] E. O. Navarro and A. Van Der Hoek, "Software process modeling for an educational software engineering simulation game," *Software Process: Improvement and Practice*, vol. 10, no. 3, pp. 311–325, 2005.

[48] E. Folmer, "Component based game development—A solution to escalating costs and expanding deadlines?" in *Component-Based Software Engineering*, H. W. Schmidt, I. Crnkovic, G. T. Heineman, and J. A. Stafford, Eds., vol. 4608 of *Lecture Notes in Computer Science*, pp. 66–73, Springer Berlin Heidelberg, Berlin, Germany, 2007.

[49] T. Susi, M. Johannesson, and P. Backlund, *Serious games: An overview*, 2007.