D-CreEA: DSML for Creating Educational Analog Card Games

Rubens Anderson de S. Silva
Department of Computing
Federal University of Ceará
Fortaleza, Ceará, Brasil
0000-0001-9607-7237

Rossana M. C. Andrade
Department of Computing
Federal University of Ceará
Fortaleza, Ceará, Brasil
0000-0002-0186-2994

***Abstract*—Educational games can promote a playful learning experience by solving problems related to the subject of a discipline and assign a student as the main actor in the learning process. In addition, when using analog educational games, this experience provides a face-to-face discussion on the subject, with tactile interaction and quick feedback. However, creating an educational game is difficult, because it demands a scientific, educational basis and game design background. MDE (Model-Driven Engineering) concepts have been used in game development as a promising way to ease the dev process as it helps to develop these games in a standardized and semi-automated way. Hence, the objective of this work is to develop a Domain-Specific Modeling Language for Creating Educational Analog Card Games, called D-CreEA. With this, we provided an approach to aid the process of generating games grounded on the core ideas of card games design by the professor or the game developer, who aims to implement a playful learning methodology, with personalized educational content.**

***Index Terms*—Education, Educational Game, Analog Game, Card Game, MDE, DSL, DSML**

I. Introduction

Making a game is difficult since the difference between games and other entertainment products (such as books, music, and films) is that their consumption is relatively unpredictable. Furthermore, the sequence of events that occur during the game and the outcome of those events is, to some extent, unknown during the production phase. [1].

On the other hand, the level of effectiveness, in terms of pedagogical and playful objectives, is related to many factors, as the application context, the purpose, how it was done, if a specific content was targeted, and if the objectives were precisely defined [2], [3]. All these aspects influence the results, with broad types of outcomes: significant positive effects, mixed results (that is, instructional games facilitated specific learning outcomes, but not the others), no difference between games and conventional instruction, and yet conventional instruction can be more effective than computer games [2], [3].

One way to promote an effective game development approach is through software frameworks and meta-models. They aid the process of concept, build and prototype in a standardized and semi-automated way. These structures are defined through MDE (Model-Driven Engineering) concepts to define specific software to develop functionalities for specific domain problems.

Thus, the main research question of this work is: “How to develop an educational analog card game using a software engineering approach on game design literature?”. To answer that, this work proposes a domain-specific modeling language (DSML) to create educational analog card games, namely *D-CreEA*.

The idea is to describe the core aspects of the analog educational card games' domain and generate a metamodel. The models generated by this metamodel can serve as a base for games, since the metamodel specifies behavior, structure, and requirements of a specific class of the domain and the formal conversion between models.

D-CreEA's main target audience is Computer Science professors whose subject content can be translated into unitary questions and answers, i.e., each question has one or a set of well-defined answers. In this way, the professor can organize a specific subject and generate unitary challenges and possible answers. In this way, s/he can practice the subjects seen in the classroom and promote problem-based learning [4].

With this, the main contributions of this work is to provide an approach to generate educational games grounded on the core ideas of card games design, with personalized educational content. Also, as shown in Section III, there may exist a gap on MDE-based tabletop games development approaches, and this work promotes the discussion about the MDE adoption.

This document is divided as it follows: Background (Section II), Related Work (Section III), The D-CReEA (Section IV), Case Study (Section V), and Conclusions (Section VI).

II. Background

In this section, we discuss the *background* of this work.

A. Domain-Specific Modeling Language

The concept of Domain-Specific Modeling (DSM) can be defined as a combination of a domain-specific language (DSL), a domain-specific code generator (model-to-artifact), and a structure of domain to raise the level of abstraction beyond manual software coding (e.g., diagrams). It uses domain definitions to generate a final product in a low-level language (e.g., programming, textual) chosen from the high-level specifications [5].

Domain-specific modeling (DSMs) is primarily used in situations where similar systems are developing. This language helps increase productivity and helps the specialist to formalize the practical development of a type of application to other developers. Thus, the rules and guidelines that have been incorporated into one tool can be applied to others and can even adequately guide or alert developers to the requirements and restrictions of that domain [5].

A domain-specific language (DSL) is a high-level software implementation language that supports concepts and abstractions that are related to a particular (application) domain [6]. It is a language, therefore, it has a collection of sentences in a textual or visual notation with a formally defined syntax and semantics. A grammar or meta-model defines the structure of the sentences of the language, and the semantics should be defined using either an abstract mathematical semantics or a translation to another language with well-understood semantics. A DSL is high-level, since it abstracts from low-level implementation details and possibly from particularities of the implementation platform [6]. In its definition, a DSL not necessarily implements software but practical and usable artifacts that aid in resolving a specific domain problem. For this work, the artifacts are analog-game representations like cards, manuals, rules, etc.

The abstract syntax of a language describes the vocabulary of concepts provided by the language and how they may be combined to create models or programs. It consists of a set of provided concepts and their relationships to other concepts. It may also include rules that define whether a model is well-formed. On the other hand, the concrete syntax is the set of notations that facilitates the presentation and construction of the language constructs. The concrete syntax could either be formulated in a textual or visual manner [7]. The static semantics has the role of constraining and representing information that cannot (easily) be shown through the syntax representation. These are the rules that govern every game of an analog card game for this meta-model [5].

In the context of serious games, the use of DSMLs helps develop games in a standalone way, where the professor has the role of designing it without the help of a game developer, demanding only the required knowledge background. This ease is because all requirements and restrictions have already been formally broken down during the design of the DSML for that domain, remaining the task of game modeling and the translation into an artifact.

B. Games

A game can be defined as “a competition of physical or mental skills and strengths, requiring the participant(s) to follow a specific set of rules to achieve a goal” [8]–[10].

1) Elements of a Game: According to Prensky et al. [11], a game is a six elements structure: rules, goals and objectives, results and feedback, conflict/competition/challenge/opposition, interaction, and representation or story.

For Battaiola et al. [12], the elements of a game can also be defined by four elements: context, artifact/game, activity/game, and agent/player. This decompilation considers the environment and the need for an opportunity for the gameplay to occur.

The MDA framework (Mechanics, Dynamics & Aesthetics) [1] presents a straightforward categorization of games, fundamentally divided into three fundamental components:

- Mechanics describes the particular components of the game in terms of data representation and algorithms. They are the various actions, behaviors, and control mechanisms offered to the player within a game context - the mechanics support the game dynamics;
- Dynamics describes the runtime behavior of mechanics acting on the player’s inputs and each other’s outputs over time - Dynamics works to create aesthetic experiences;
- Aesthetics (closest term to experience) describes the desirable emotional responses evoked by the player when s/he interacts with the game system. Each game aims at various aesthetic goals to varying degrees. Taxonomy helps us to describe games to identify the required experience that the player should have for the game. The taxonomy used to describe the experience of a game is:
  - Sensation (game as sense pleasure),
  - Fantasy (make-believe game),
  - Narrative (play as drama),
  - Challenge (play as an obstacle course),
  - Fellowship (game as social structure),
  - Discovery (play as uncharted territory),
  - Expression (game as autodiscovery) and
  - Submission (play as a hobby).

2) Tabletop, Board, and Card Games: The term ”tabletop game” is the broader term covering most games generally played at a table. This term includes: 1. Board games; 2. Card games; 3. Paper & pen RPGs (Dungeons and Dragons etc.); and 4. Miniature battle games, from big war games to small melee games.

What differentiates a board game from a card game is the use of the board as an active mechanic, which is not included in card games. Thus, the table where the card game is played is not part of the gameplay but the game context.

To classify board games, the work of [13] organizes and proposes an ontology based on the BGG classification for the types of board game mechanics. The number of mechanics extends to more than 50, for example, gambling, expression, role-play, etc. Furthermore, board games can also be divided into two large families [14]: Hobby Games and Mainstream Games.

Mainstream games are widely popular and are readily available in mainstream stores. These games tend to be less rules-heavy, focusing more on ease of play. They are generally low-cost and do not include expansions.

On the other hand, Hobby Games are aimed at people who consider gaming as a dedicated hobby. They are mostly sold at specialty gaming stores, with most titles being hard to find.

These games tend to be heavier on the rules and offer a broader range of playing difficulties and target age groups than conventional games. Often, hobby games can include expansions to add to the gaming experience for established players, but they can cost a considerable investment. Therefore, Hobby Games are often intimidating or unattractive to the general public.

3) Educational Games: Educational games have the following properties:

- the facilitating factor attached to games reduces cognitive load and allows students to use their precious working memory for higher-order tasks [15].
- usable contexts such as informal learning, kindergarten/preschool, elementary school, high school, adult education, business management, military, and health [2].
- results on motivation, for example, the work of [16, p. 69] points out 13 principal components of motivation facilitated by playing: identity presentation, social relationships, play, learning, achievement, rewards, immersion, context, fantasy, uniqueness, creativity, curiosity, control, and property.
- genres include adventure games, simulation games, board games, puzzle games, business simulation games, and action games [2].
- learning areas include science education, math, language arts, reading, physics, health, natural sciences, non-content science and social skills, and general problem-solving skills development [2].

Regarding educational board games, the main advantage over digital educational games is face-to-face contact and discussion, promoting critical and experimental reflection, social interaction between students, and the rapid *feedback* promoted by the professor who may be supervising the match. In addition, educational games can bring a greater bond between professors and students during the application of the game. In a more relaxed environment, students and professors can interact more spontaneously. In a digital game scenario, this interaction would be broader, as the student interacts much more with the computer/smartphone.

III. Related Work

This section presents a discussion about the related works and some of them are highlighted to identify similarities and differences between our work and them. The related works are shown in Table I.

The work of [17] proposes a model for the development of digital board games, whose domain is formed by GameEngine, GameElement, Player, Event, Action, GameState, Goal, Sub-Goal, Non-MovableElement, MovableElement, and Rules. This work focuses on digital board games, and consequently, has elements that do not apply to card games, such as Movable Elements and GameEngine.

The work of [18] presents an extension of the GLiSMo Modeling Language [19], a language for organizing the logic and structure of an educational game. This one adds elements of Flow Theory and the Adventure game genre. It is important to mention that for the evaluation, the FQAD framework (Framework for Qualitative Assessment of Domain-specific languages) [20] was used, which has the purpose of evaluating DSMLs.

The work of [5] proposes a DSL for the development of educational digital games, focusing on the game's story (storyline). Its domain has elements like Start Element, Mission Elements, Gameplay Elements, and Stop Elements. This work differs from D-CreEA, because it does not use UML, and the professor needs to create a background story besides being for digital games.

The work of [21] proposes a DSGL (domain-specific graphical language) for Educational Digital Games of the RPG type. It boils down to a class diagram without the implementation of the model-to-artifact engine. Game developers (GD) evaluated this model. It is not clear who should develop the games, whether educators or GDs. With the same RPG theme, the work of [22] is a model for creating *sandbox* games (in which one can freely walk around a scene) with activities spread throughout the game world.

Searches of the game development literature did not identify any work proposing a model within the MDE theme for analog game development. On the other hand, the found works have various mechanics, including Quiz, Point-and-Click, Sandbox, etc., and this is positive since it does not tie the educational game's theme to a small range of games.

Some works propose a figure of the game developer as necessary in the game development, which is a correct approach. Still, it differs from the directive of our work, which is to simplify the development process for the professor, as it may have a low theoretical basis of games.

As stated in Section II, to have a DSML implemented, it is necessary to have the syntax, the semantics, a model-to-artifact transformation model, and optionally, a software that implements the DSM and simplify the translation from the model to artifact. However, many works proposed only the semantics part (some works only the syntax, such as DSGL), and others proposed a complete framework. Therefore, the evaluations/validations varied a lot. Some papers instantiates a game as evidence of effectiveness, i.e., evidence that the DSL fulfills its purpose, and others evaluate the DSL with stakeholders, e.g., professors, game developers, etc.

In conclusion, this work is similar to those listed previously as it is inserted in the context of MDE development, but it differs of them, since the type of game generated is analog. In the analog game domain, it was seen that there may have a lack of works with a DSL theme for game development. The lack of effectiveness in search coverage of this paper is kept in mind, but the gap still holds, given the difficulty of finding this type of work.

Thus, this work contributes to help to fill this gap as proposes an MDE approach through a DSML to develop educational analog card games. With this work, we hope that this type of MDE approach be adopted by other authors in order to improve the field and clarify the efficiency of this method to the development of analog educational tabletop games.

TABLE I: Comparison between our work and related works

Title	Type	Mechanic [1]	Who develops	Plataform	Validation/Evaluation
This work	DSML	Variable	Professor or GD	Analog	Case Study
[17]	DSML	Tabletop Game	Not described	digital	Case Study
[23]	framework	Card Game	Card Trading	digital	Case Study
[18]	DSML	Adventure	Not described	digital	FQAD
[21]	DSGL	RPG	Not described	digital	Evaliation with GDs
[24]	DSL	Quiz	Professor	digital	Case Study
[5]	DSML	Story telling	Not described	digital	Case Study
[25]	DSGL	Simulation	Instructor	digital	Case Study
[22]	modelo	RPG	GD ou professor	digital	Case Study
[26]	DSVL	Point and Click	Professor	digital	Case Study
[27]	DSML	Narrative	professor	digital	Case Study
[28]	DSL	Not described	GD	digital	Case Study
[29]	DSL	Not described	GDs e Health professionals	digital	Case Study
[30]	Framework	Not described	professor	digital	Case Study
[31]	Framework	Not described	professor	digital	Case Study
[32]	DSML	Adventure RPG Puzzle Games	Professor	digital	Case Study
[33]	DSL	Not described	GD	digital	Not described
[34]	Framework	Not described	Not described	digital	Not described

IV. About D-CreEA

This work aims to develop a DSML called D-CreEA - Domain-Specific Modeling Language for Creating Educational Analog Card Games. Its objective is to help Computer professors who aim to implement GBL through analog educational card games for students of the Computing area. This section presents our motivation, problem and more details about D-CreEA.

A. Motivation and Problem

Several factors influence the student's decision to abandon graduation, for example, institutional problems (i.e., lack of course structure) and personal problems (i.e., lack of affinity with the course subjects, financial problems, and lack of motivation). In this context, the professor and coordinators need to promote an engaging, motivational and efficient teaching-learning methodology. To meet this need, the use of game-based learning (GBL), a methodology that uses games as a teaching tool, can improve student motivation and contribute to a student-centered learning environment [2]. Although there is a wide range of educational games, one type of game that can be effective is the analog games, which are games that do not have digital components in the *game play*. These games involve physical pieces (e.g., board, cards, dice, paper, and pen/pencil). Their main advantages over digital educational games is face-to-face contact and discussion, promoting critical and experimental reflection, social interaction between students, and the rapid *feedback*.

However, as mentioned in Section I, the task of developing an educational game is not simple. Balancing game mechanics and dynamics to achieve a given experience is a complex activity, even when adopting a theoretical basis of game design. Even worse, creating a game without a theoretical basis can negatively impact the game's effectiveness, causing damage to the player's experience, and this is aggravated when the game is intended to teach or practice educational content [35]. Also, this difficulty is heightened in analog games, as the player's interaction with the game is multimodal.

Another difficulty is about the need to control the gameplay manually, such as for counting points or interpreting and applying rules - one of the ways to facilitate the control of *game play* is through digitally controlled inputs, and outputs [36]. There is also the added cost of development, card design, and implementation, since to create an educational board game, all physical components must be prototyped and produced.

Consequently, the following problems arise: *how to create an educational analog card game grounded on a game design approach aided by software engineering?* This problem motivates the development of this work.

A DSML-approach is valid to mitigate this problem, since the language describes the behavior, structure, and requirements of a specific class of domains and the formal conversion between models. Thus, this makes it possible to describe the transformation of information from a high level of mastery in artifacts of an analog game.

B. Target audience and content scope

The D-CreEA's main target audience is Computer Science professors whose subject of the course content can be translated into unitary questions and answers, i.e., each question has one or a set of well-defined answers. In this way, the professor can organize a specific subject and generate unitary challenges and possible answers. In this way, s/he can practice the subjects seen in the classroom and promote problem-based learning.

C. Game type and gameplay

The gameplay of the generated games will have as the essential main mechanic the Quiz, i.e., questions and answers described by the professor. However, other main and secondary mechanics can be part of the repertoire, such as Trading, Deck, Take That, Card Drafting, RPG [13], etc., since D-CreEA supports any token (e.g., dice, avatars), role interpretation, and rule description. The use of these mechanics depends only on the professor's creativity.

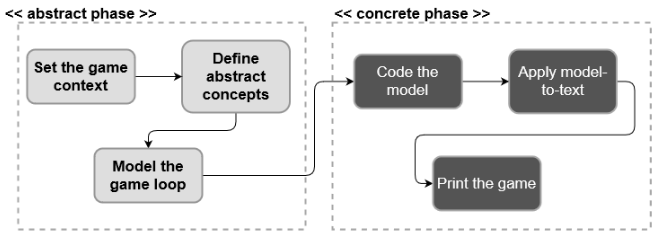


Fig. 1: Steps to instantiate a game via D-CreEA

D. How to instantiate a game

To create a game, the dev must have configured Angular[1] to execute the code of model-to-artifact. Figure 1 illustrates the process to create a game by using D-CreEA, which steps are as follows:

1) Set the game context, which is composed of the game name, description, knowledge field, and knowledge requirements;
2) Set the abstract concepts, such as type of game mechanics, objectives, roles, narrative - for this, use as basis the abstract syntax (see Figure 2);
3) Model the game loop and rules of the game, restrictions, and uses, based on the previous step - for this, the model of game loop is helpful (see Figure 3);
4) Code the model defined, implementing on a Javascript class and placing it on the model-to-text folder[2] - see the file *summary/model.ts*;
5) Execute the model-to-text by executing the Angular project and accessing the printing page; and
6) Print the game via browser.

E. Domain Analysis

The Domain analysis is the process of identifying the relevant concepts of an application domain, focusing on reusing those concepts. The products of the analysis process are reusable definitions of domain concepts that are common to all the applications of that domain.

To understand the domain of educational analog card games, we developed an exploratory research to find related works (see Section III), evaluation methods of DSMLs, and analog educational games belonging to the DSML domain.

To form the DSML domain, we use two approaches: a *Top-down* and a *Bottom-up*.

From the *Top-down* perspective, the analysis took as basis the related work (see Section III).

On the other hand, under the *bottom up* approach, the search focused on the analysis of educational analog games published in scientific game events (e.g., SBGames) and Google Scholar to frame the basis of the DSML's domain. The games were classified in terms of game type (card, board, or hybrid), the number of players, and their types of mechanics, following the BGG catalog[3]. The result is shown in Table II.

[1] https://angular.io/
[2] model-to-text: https://bit.ly/3x1qXa1
[3] BGG Game Mechanichs Catalog: https://bit.ly/30dWQ4B

It is noteworthy that the search for games was non-exhaustive. The analysis of the games found raised the following points:

- Games address a wide variety of educational topics, both inside and outside Computing. In Computing, they were: Systems Development, Design and IHC, Programming, Software (e.g., Configuration Engineering, Software Testing) and History of Computing. Outside Computing, we had: Confronting the AEDES AEGYPTY, Biology, diversity and scientific method.
- board games place the player in simulations within the educational theme proposed through RPG mechanics. This aspect is in line with the discussion presented in Section II, in which one of the requirements to achieve educational goals is through placing the student in actual situations within the educational context.
- many board games use cards as the central mechanics, and it is possible to convert some to purely card games, without the need for a board as the active mechanic. Thus, these works could be helpful for the scope of our work.
- Regarding card games, the common feature is that there are challenges involving the educational content that students must solve through mechanics simulated by the cards and their effects.
- Generally, games are based on turns or rounds for each player, but as the *Simultaneous Action Selection* mechanic suggests, opting for simultaneous actions between players is not uncommon.

With this, the domain of D-CReEA has the following entities:

- Game: represents the entity of a game, encapsulating all in-game components, as decks, game description, states, tokens, and roles.
- State: is the momentary state of the game, with its respective rules. The game must have at least one path from the initial state to the game-over state.
- Rule: according to the discussion in Section II, the rule is the game regulators defined by the professor. The Rules have three branches:
  – Statement Rule: explains a game situation, in the sense of 'how to'. A Statement Rule has its simple description and a complex one (optional), following a formal structure: me (the actor that participates in the action), given (the necessary game condition), when (a fact occur), then (do something) - all these fields are optional.
  – Effect Rule: explains a game situation, in the sense of 'player suffers/acts the action'. This entity uses the effect entity.
  – Conditional Rule: is a game situation when the decision to proceed to a state A depends on the satisfaction of a condition CA. This rule can have a set of Conditions, and if all fails, a default route is taken.
- Deck: represents the set of cards of a certain type defined

TABLE II: Base games for an Application Domain

Title	Type	Mechanics	# Players
GREaTest Card Game [37]	Cards/Híbrido	Trading, dice rolling, deck, take that, card drafting, RPG	2 a 7
Heredograma Sem Mistério [38]	Tabletop	Tile placement, dice rolling, RPG	5 equipes ou 5 pessoas
Evolução: A Luta Pela Sobrevivência [39]	Tabletop	Moviment/point-to-point, RPG	5 equipes ou 5 pessoas
ARBattle [40]	Cards	Trading, Take that, Deck, RPG	Not described
Computasséia [41]	Cards	Trading, Co-op, Tile placement, Storytelling	2 a 6
SimulES-W [42]	Tabletop e Cards	Take That, Deck, Set Collection, RPG	Not described
O Baralho das Variáveis [43]	Cards	Deck, Tile Placement	1
Desafio de Design do Goople [44]	Tabletop e Cards	Deck, Take That, Random Draw, RPG	Not described
JEEES [45]	Tabletop e Cards	Deck, Set Collection, RPG	2 a 3
Dengueside Survival [46]	Tabletop e Cards	Deck, Movement, Random Draw, RPG	1 a 6
Gente [47]	Cards	Deck, Trading, RPG	2 a 8
O Jogo do Método [48]	Tabletop e Cards	Area movement, Co-op, Collection, Dice Rolling, RPG	Not described

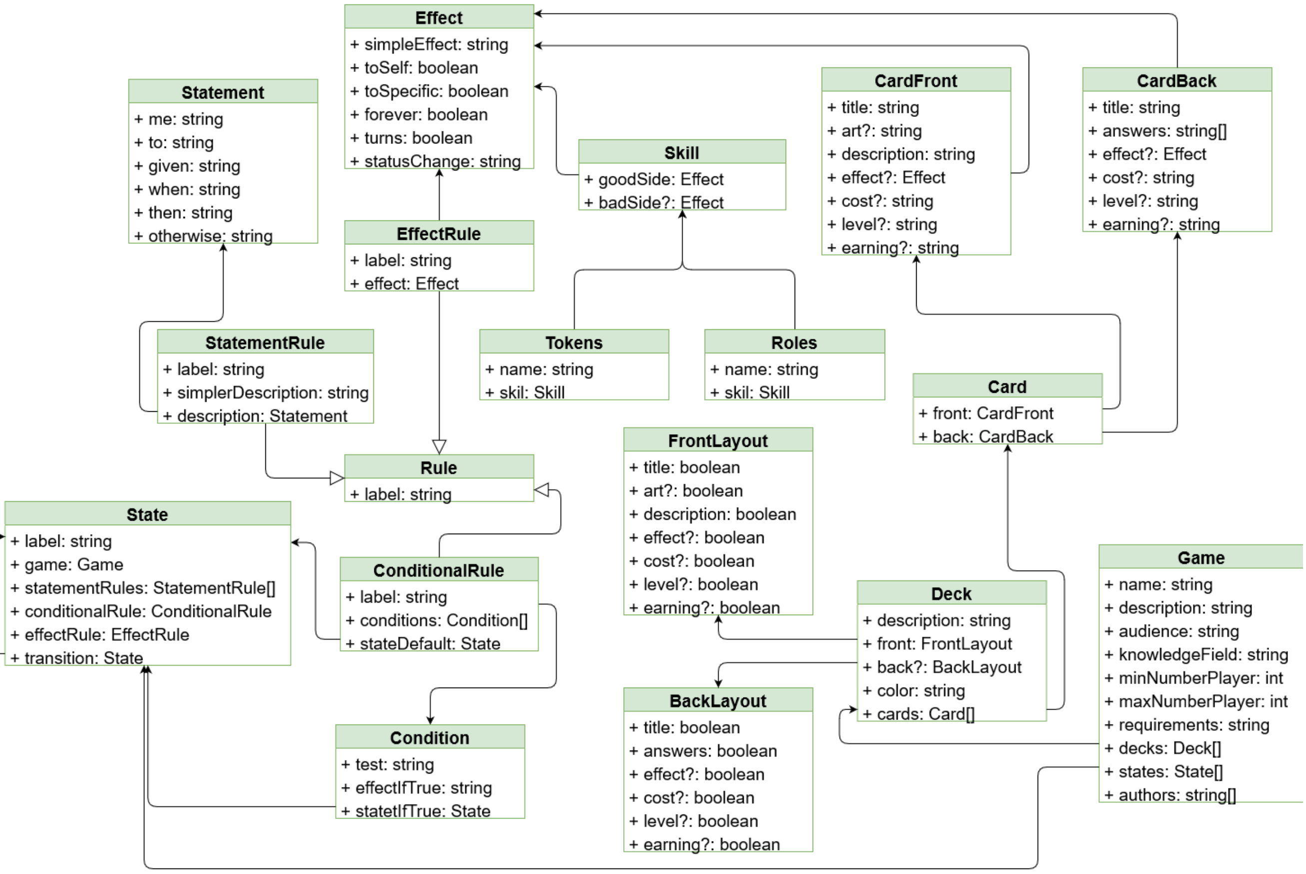


Fig. 2: D-CreEA - Abstract Syntax

by the professor. Every deck has a meaning (e.g., challenges, bonus), and it must be different from all others decks to avoid redundancy. The deck has a front layout and possibly a back layout as follows:

– Front Layout defines all fields that a card of this deck has, such as title, art, description, effect, and point types (i.e., cost, level, and earning).
– Back Layout is similar to the Front Layout but has title, answers, effect, and point types such like cost, level, and earning.

- Card: the unit of information that represents a generic game card. It follows the layout of its deck.
- Role: representation of the persona adopted to a player, which guarantees positive or negative effects.
- Token: represents palpable units that have some in-game positive or negative effects.
- Effect: is an act that changes the status of an actor during the game. A player ignites an effect, and the action affects the player itself or another one, and may last for a few turns or the rest of the game.
- Skill is an ability of an actor that has a good effect and, optionally, a bad effect.
- Statement is a description used to explain some rule. It is used in the context of Statement Rule.

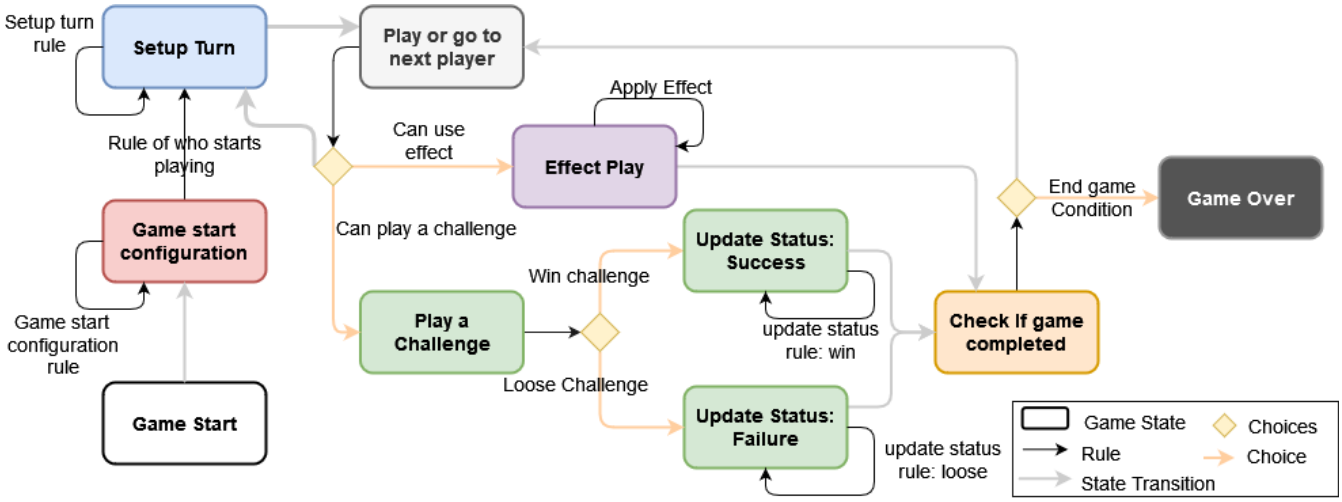


Fig. 3: D-CreEA - Game Loop

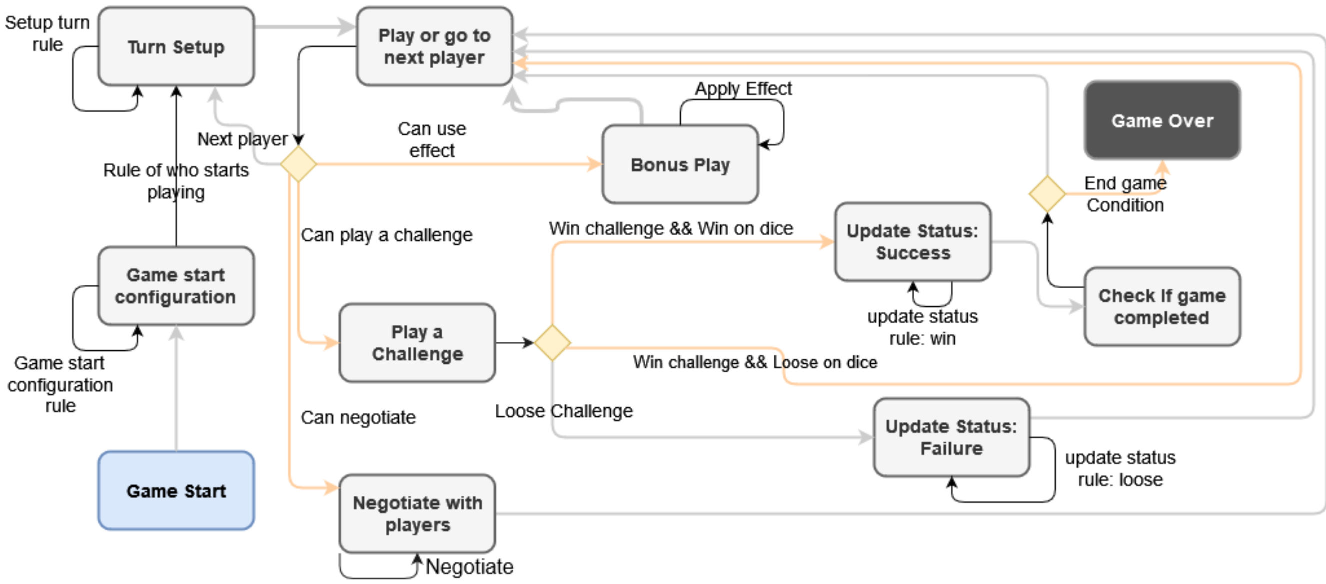


Fig. 4: GREaTest Card Game (GTCG) - Game Loop

- Condition is used to apply some effect and reach some state if an expression-test is true. It is used in the context of Conditional Rule.

F. Meta-model

The meta-model was defined from the domain and included *abstract syntax* and *static semantics*. The abstract syntax is represented by a class diagram and is shown in Figure 2. The Static semantics has the role of constraining and representing information that cannot (easily) be shown through UML diagrams. These are rules that govern every game of an analog card game for this meta-model and are implemented in model-to-artifact. In its current state, the semantic has the following restrictions:

1) Every game must have precisely one Initial and one Game Over states;
2) The initial State must have only one transition, and the Game Over state must have none. No state can transit to Initial State;
3) Every game must have at least one path from Initial State to Game Over State - defined as *game path* - with at least one state in between;
4) Every game path must have the purpose of implementing a new mechanic/type of play.

To guide the development, it is provided a game loop model with all these restrictions guaranteed. The model is defined as a game loop representing the default configuration of an educational analog card game. It is a UML activity diagram in

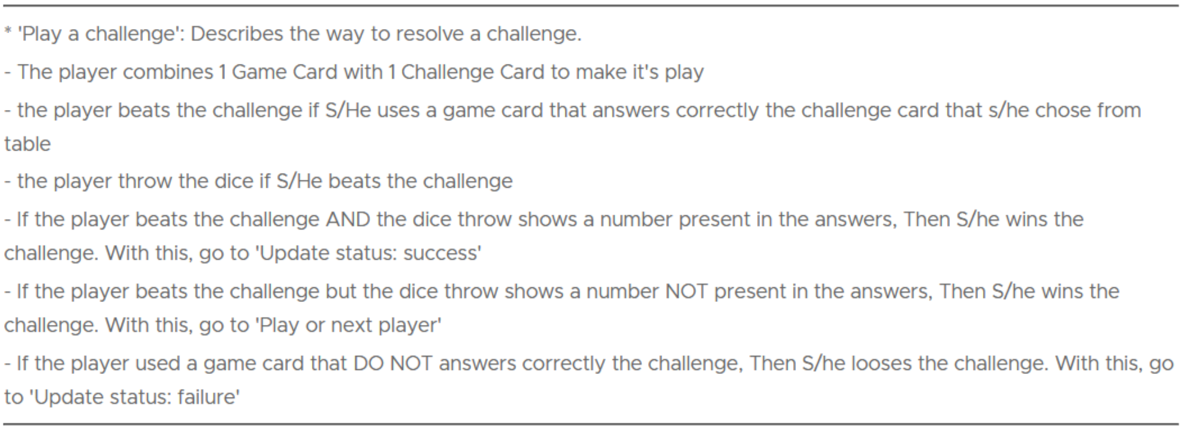
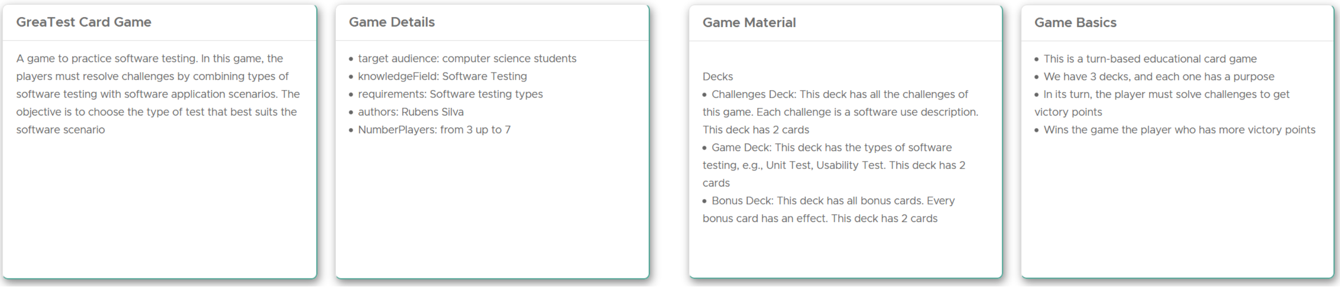
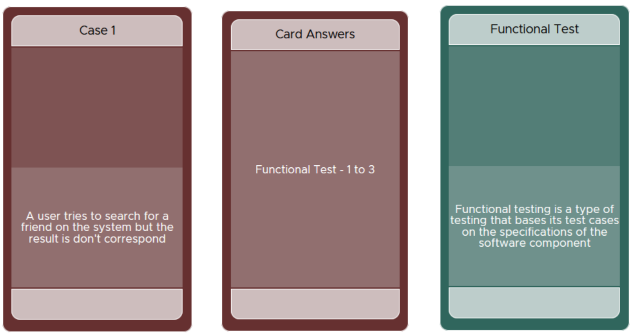

* 'Play a challenge': Describes the way to resolve a challenge.
- The player combines 1 Game Card with 1 Challenge Card to make it's play
- the player beats the challenge if S/He uses a game card that answers correctly the challenge card that s/he chose from table
- the player throw the dice if S/He beats the challenge
- If the player beats the challenge AND the dice throw shows a number present in the answers, Then S/he wins the challenge. With this, go to 'Update status: success'
- If the player beats the challenge but the dice throw shows a number NOT present in the answers, Then S/he wins the challenge. With this, go to 'Play or next player'
- If the player used a game card that DO NOT answers correctly the challenge, Then S/he looses the challenge. With this, go to 'Update status: failure'

Fig. 5: GTCG Rule Example generated by D-CreEA



(a) Game details (b) Game materials and basics

Fig. 6: GTCG info generated by D-CreEA



(a) Challenges - Front (b) Challenges - Back (c) Game Card

Fig. 7: Cards from GTCG generated by D-CreEA

which each rectangle represents a game state from the domain, and each arrow represents a rule.

The model has implemented the Quiz and the Bonus mechanics, and can (and should) be modified. In the model, every path from *Play or next player* to *Check if game is completed* represents a different game mechanic. With this, it remains to the dev to extend the model to create another types of game mechanics (see Figure 3).

G. Model-to-Artifact

The Model-to-Artifact (M2A) is a set of algorithms that receives as a parameter a model of a concrete syntax and transforms it to educational analog games artifacts, like cards, manual (with the game details and rules), tokens, etc. The chosen intermediate model is Typescript, since it can be translated to Javascript and this language can be used in any platform, such as desktop, WEB, or mobile.

M2A has four main components as follows: Rules to Text, Description to Text, Decks to Text, and Cards to Artifact.

- The Rules to Text algorithm starts at the Game Start state and walks through all game paths until all rules are presented and translated into formatted text.
- The Description to Text translates the game details, as name, description, etc., into formatted text.
- The Decks to Text translates each deck details and fields into formatted text description.
- The Cards to Artifact translates all cards from all decks into formatted card, with shape of a Tarot card. This shape was adopted due to the 'long' shape, which can fit more content.

The resulting artifacts are presented in HTML pages, customized with CSS, providing a fine looking for the cards, tokens, and manuals. Also, an HTML page can be printed by any web browser. As the implementation code is extensive, we provided an online code repository to save it and make it available for consulting[4].

V. Evaluation

The evaluation of this DSML aims to answer the following question: "Is the D-CreEA suitable for analog game development"? For this, a version of the educational game called GREaTest Card Game [37] (GTCG) was developed. GTCG is an educational analog card game that aims to practice the study of software testing. The gameplay consists in identifying the type of software test that, when used on the given scenario (i.e., the challenge), would catch bugs and help to resolve them. This evaluation can reveal important aspects of the DSML, such as suitability (i.e., if the tool really can create a game), or if there's a lack of functionality.

This game was chosen due its different types of mechanics, such as negotiation and bonus, since these types of gameplay were not included on the initial model of D-CreEA. With this, Figures 5, 6, and 5 present the game artifacts resulting from this process.

The game has three decks: Game, Challenges, and Bonus. And it uses a dice to decide if a player receives the points after it beats a challenge. Also, this game has a negotiation mechanic that provides a way to exchange cards between players directly by free negotiation in every turn - after this, the player can still try a challenge. These mechanics caused a necessary extension of the model by introducing new game paths. This game model is shown in Figure 4.

With this evaluation, it is possible to detect the strong points and the improvement points of D-CreEA. Once the full GTCG implementation is possible, there is evidence about the suitability of the DSML. Also, the base game has different types of game mechanics, such as negotiation, and a different game loop, and this is evidence for the extensibility of the DSML.

With this evaluation, the answer of the question "*Is the D-CreEA suitable for analog game development?*" is *true*. However, we could not test the efficiency of the DSML, since no comparison was possible with the original process of creating GTCG - this will be tested in future work. Also, as this is an unusual approach for this type of game dev (see Section III), the evaluation of the suitability of this kind of approach is not yet well established in the literature.

VI. Conclusions

This work presented D-CreEA, a Domain-Specific Modeling Language to aid the development of analog educational card games. In this work, the process of creating an analog game based on an existing game could be mapped to the D-CreEA, with high fidelity. About the question "Is DSML a suitable approach to generate analog games?", there is a long road ahead to answer that by showing evidence that prove or disprove the efficacy (and further beyond, the efficiency) of this type of method for this problem.

[4]model-to-text repository: https://bit.ly/3x1qXa1

However, with this work, it was possible to implement a published educational analog card game, and improve it with relative easiness. Also, we could implement a whole variety of games with different mechanics. Although, it is necessary to improve the evaluation, this is an initial step towards the direction of MDE for analog games. The D-CreEA process is still laborious, since the professor needs to implement by coding the model, demanding a software tool to provide the game editing and printing interface. However, as next steps of this work, this tool will be implemented and make available on the internet as a Service.

Thus, the main contribution of this work is to provide a software engineering approach to develop educational analog card games, represented by D-CreEA, grounded on game design elements and customized educational content. Also, since it was not found another work with this aim, this work contributes to promote the discussion about MDE approaches over the development of educational analog card games.

As threats to validity, we can cited the following:

Construction validity: as an open problem, the visual representation of the model may have been poorly implemented. However, using UML can be seen as a good choice as it can represent all relationships and entities.

Internal Validity: some personal interpretations may have occurred during data extraction and analysis of domain. In order to minimize these biases, we used a peer review.

External Validity: it was not found works presenting a process to create analog games, only digital. So either the sample of works collected is not fully complete or the field of MDE for analog games are not mature. However, improvements will be made in the DSML domain, mainly in terms of completeness (i.e., more game elements).

Conclusion validity: the conclusions achieved are partials, since there was not an instance of game created by a group of professors. Also, the game was not tested with real students audience.

As future work, we will evaluate the DSML with professors, students, and game designers in order to improve the correctness and completeness, providing a useful tool. Furthermore, we will implement an WEB SaaS application based on the DSML to ease the modeling and model-to-artifact phases.

References

[1] R. Hunicke, M. Leblanc, and R. Zubek, "Mda: A formal approach to game design and game research," *AAAI Workshop - Technical Report*, vol. 1, 01 2004.

[2] K. Fengfeng, "Chapter i a qualitative meta-analysis of computer games as learning tools," *Handbook of Research on Effective Electronic Gaming in Education*, vol. 1, 01 2009.

[3] S. de Freitas, "Learning in immersive worlds: a review of game-based learning," WorkingPaper, 2006.

[4] J. R. Savery, "Overview of problem-based learning: Definitions and distinctions," *Essential readings in problem-based learning: Exploring and extending the legacy of Howard S. Barrows*, vol. 9, no. 2, pp. 5–15, 2015.

[5] A. T. Prasanna, "A domain specific modeling language for specifying educational games," Ph.D. dissertation, Master's Thesis. Vrije Universiteit Brussel, Department of Computer Science, 2012.

[6] E. Visser, "Webdsl: A case study in domain-specific language engineering," in *International summer school on generative and transformational techniques in software engineering*. Springer, 2007, pp. 291–373.
[7] C. Hahn, "A domain specific modeling language for multiagent systems," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. Citeseer, 2008, pp. 233–240.
[8] J. G. Hogle, "Considering games as cognitive tools: In search of effective "edutainment"," 1996.
[9] J. Dempsey, K. Rasmussen, and B. Lucassen, "Instructional gaming: Implications for instructional technology," 1996.
[10] T. Malone, "What makes computer games fun?(abstract only)," in *Conference on Human Factors in Computing Systems: Proceedings of the joint conference on Easier and more productive use of computer systems.(Part- II): Human interface and the user interface-*, vol. 1981, 1981.
[11] M. Prensky, *Digital game-based learning*. New York: McGraw-Hill, 2001.
[12] A. Battaiola and D. Munhoz, "Regras e mecânicas em jogos," vol. a, pp. 22–41, 12 2018.
[13] G. X. Joshua Kritz, Eduardo Mangeli, "Building an ontology of boardgame mechanics based on the boardgamegeek database and the mda framework," in *XVI Brazilian Symposium on Computer Games and Digital Entertainment*, Curitiba, 2017, pp. 182–191.
[14] "Board game types, a quick classification," https://tableknight.com/board-game-genres/, 2019, accessed: 2020-12-20.
[15] M. Gredler, "Games and simulations and their relationships to learning," *Handbook of Research on Educational Communications and Technology*, 01 2004.
[16] A. Gulumbay and Turkey, "Affective and emotional aspects of human-computer interaction: Game-based and innovative learning approaches," *The Turkish Online Journal of Distance Education*, 01 2006.
[17] D. Altumbay, E. Cetinkaya, and M. Metin, "Model driven development of board games," 2009.
[18] A. S. Zahari, L. Ab Rahim, N. Nurhadi, and M. Aslam, "A domain specific modeling language for adventure educational games and flow theory," *International Journal on Advanced Science Engineering and Information Technology*, vol. 10, 06 2020.
[19] A. S. Zahari, L. A. Rahim, and M. Mehat, "A review of modelling languages for adventure educational games," *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, p. 495–500, 2016.
[20] G. Kahraman and S. Bilgen, "A framework for qualitative assessment of domain-specific languages," *Software Systems Modeling*, vol. 14, 01 2013.
[21] A. Tlili, F. Essalmi, L. J. B. Ayed, M. Jemni, and Kinshuk, "Towards a generic uml model to support designing educational role playing games," in *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*, 2016, pp. 153–157.
[22] F. Bellotti, R. Berta, A. De Gloria, and L. Primavera, "A task annotation model for sandbox serious games," 09 2009, pp. 233–240.
[23] G. d. O. LIMA, "Framework para construção de jogos de cartas colecionáveis com ra," 2018.
[24] C. G. García, E. R. Núñez-Valdez, P. Moreno-Ger, R. G. Crespo, B. C. P. G-Bustelo, and J. M. C. Lovelle, "Agile development of multiplatform educational video games using a domain-specific language," 2019, p. 599–614.
[25] Y. Duval, D. Panzoli, A. Reymonet, J. Plantec, J. Thomas, and J.-P. Jessel, "Serious games scenario modeling for non-experts," in *CSEDU*, 2015.
[26] E. J. Marchiori, Á. Del Blanco, J. Torrente, I. Martinez-Ortiz, and B. Fernández-Manjón, "A visual language for the creation of narrative educational games," *Journal of Visual Languages & Computing*, vol. 22, no. 6, pp. 443–452, 2011.
[27] O. De Troyer, F. Van Broeckhoven, and J. Vlieghe, "Linking serious game narratives with pedagogical theories and pedagogical design strategies," *Journal of Computing in Higher Education*, vol. 29, no. 3, pp. 549–573, 2017.
[28] M. Ziatdinov and M. Khafizov, "New automata definition of language for game development," in *Social Sciences and Interdisciplinary Behavior: The 4th International Congress on Interdisciplinary Behavior and Social Science (ICIBSoS 2015), Kazan Federal University, Kazan, Russia, 22-23 October 2015 & Arya Duta hotel, Jakarta, Indonesia, 07–08 November 2015*. CRC Press, 2016, p. 61.
[29] J. B. Mossmann, R. Rieder, C. D. Flores, and M. Pinho, "Project and preliminary evaluation of vr-med, a domain-specific language for serious games in family medicine teaching," in *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2. IEEE, 2016, pp. 663–667.
[30] F. Hamiye, B. Said, and B. Serhan, "A framework for the development of serious games for assessment," in *International Conference on Games and Learning Alliance*. Springer, 2019, pp. 407–416.
[31] R. dos Passos Canteri, L. S. García, T. A. de Felipe, L. F. O. Galvão, and D. R. Antunes, "Conceptual framework to support a web authoring tool of educational games for deaf children." in *CSEDU (2)*, 2019, pp. 226–235.
[32] S. Tang, M. Hanneghan, T. Hughes, C. Dennett, S. Cooper, M. A. Sabri *et al.*, "Towards a domain specific modelling language for serious game design," in *6th International Game Design and Technology Workshop, Liverpool, UK*, 2008.
[33] E. M. Hirdes and J. M. Leimeister, "A modeling language to describe reusable learning processes to achieve educational objectives in serious games," in *European Conference on Technology Enhanced Learning*. Springer, 2013, pp. 454–459.
[34] N. Thillainathan, "A model driven development framework for serious games," *Available at SSRN 2475410*, 2013.
[35] P. Eduardo Battistella and C. Gresse von Wangenheim, "Games for teaching computing in higher education – a systematic review," *IEEE Technology and Engineering Education*, vol. 1, pp. 8–30, 03 2016.
[36] B. Borges, R. Silva, J. Paiva, B. Aragão, I. Santos, and R. Andrade, "Design e avaliação de um aplicativo móvel complementar para um jogo de cartas educacional," in *Anais do I Workshop sobre Interação e Pesquisa de Usuários no Desenvolvimento de Jogos*. Porto Alegre, RS, Brasil: SBC, 2019, pp. 21–30.
[37] T. A. Beppe, I. L. de Araújo, B. S. Aragão, I. de Sousa Santos, D. Ximenes, and R. M. C. Andrade, "Greatest: A card game to motivate the software testing learning," in *Proceedings of the XXXII Brazilian Symposium on Software Engineering*, ser. SBES '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 298–307. [Online]. Available: https://doi.org/10.1145/3266237.3266254
[38] L. M. L. Campos, T. M. Bortoloto, A. K. C. Felício *et al.*, "A produção de jogos didáticos para o ensino de ciências e biologia: uma proposta para favorecer a aprendizagem," *Caderno dos núcleos de Ensino*, vol. 47, pp. 47–60, 2003.
[39] ——, "A produção de jogos didáticos para o ensino de ciências e biologia: uma proposta para favorecer a aprendizagem," *Caderno dos núcleos de Ensino*, vol. 47, pp. 47–60, 2003.
[40] T. Malheiros, L. Reis, and J. E. R. de Carvalho, "A utilização da realidade aumentada em jogos de cartas colecionáveis," 2012.
[41] J. C. O. Santos and K. d. S. Figueiredo, "Computasseia: Um jogo para o ensino de história da computação," in *Anais do XXIV workshop sobre educação em computação*. SBC, 2016, pp. 31–40.
[42] E. Monsalve, V. Werneck, and J. Leite, "Simules-w: Um jogo para o ensino de engenharia de software," *III Fórum em Educação de Engenharia de Software (FEES), Simpósio Brasileiro de Engenharia de Software (SBES), Salvador*, pp. 17–26, 2010.
[43] E. L. de França, R. C. Nunes, and R. C. dos Santos, "Jogo baralho das variáveis uma proposta de utilização de jogos para disciplina de programação."
[44] T. Darin, F. Rocha, D. Motta, and P. V. Angelo, "Desafio de design goople: Um jogo de cartas para apoio ao ensino do design de interação e conceitos básicos de interação humano-computador," in *Anais Estendidos do XVIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais*. SBC, 2019, pp. 100–105.
[45] K. Figueiredo, J. Ferreira, L. Murta, and E. Clua, "Um jogo de estratégia de gerência de configuraçao," *III Fórum de Educação em Engenharia de Software*, pp. 1–8, 2010.
[46] G. S. de Azevedo, L. S. Lima, and V. T. Sarinho, "Dengueside survival: Produzindo um jogo de tabuleiro estilo rpg para o combate ao mosquito aedes aegypti," 2020.
[47] F. Guerreiro, F. S. Moura, L. Rodrigues, and M. M. Fontoura, "Design case do jogo de cartas "gente"," 2020.
[48] J. Mattar, F. D. Almeida, L. M. Souza, J. d. O. Beduschi, C. C. da Silva, B. R. dos Santos, J. W. R. Amaral, and R. Sprovieri, "Gamificação e jogos para metodologia científica: proposta de jogo de tabuleiro e game," 2017.