A System for Statistical Contextual Structure of Representation for Game Studies

Rodrigo G. Domingues¹, Alexandre Cardoso¹, and Rafael D. O. Venâncio²

¹Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia – MG, 38400-902, BR
²Faculdade de Educação, Universidade Federal de Uberlândia, Uberlândia – MG, 38400-902, BR

Corresponding author: Rodrigo G. Domingues (e-mail: rodrigod@hiperlogic.com.br).

This study was supported by the Centro de Aperfeiçoamento de Pessoal no Ensino Superior (CAPES).

ABSTRACT Several theoretical frameworks and structures have been proposed to contribute to the field of Game Studies or game design. Despite providing some guidance, they lack methods or processes of data manipulation to provide additional information in order to compose a knowledge base and further improve the field. This paper presents a system focused on distributed data gathering, introducing a structure that relies on the theoretical game design frameworks and theories of both the communication and structuralism fields. The structure was designed focusing on mathematical validity based on set and fuzzy set theories, and organized according to theories of illocutionary logic in terms of meaning. The system proposed provides methods to statistically quantify qualitative data and perform operations in order to gather information for applications in game design processes, game design education, game studies research and several related fields.

INDEX TERMS Educational technology, Fuzzy sets, Distributed systems, Big data applications, Data collection, Human factors, Adaptive Estimation, Knowledge management

I. INTRODUCTION

Although specific processes of game design date at least from the late 18th century [1], research regarding general processes of game design is quite recent, and it is considered a relatively underdeveloped field [2]. The proposals for game taxonomy or categorization, presented in section 2, provide subjective or inconsistent information. Additionally, the industry and specialized media maintain the genre taxonomy [3], a proposal from 1985 for video games, as the de facto standard.

The popularization of the act of playing, promoted via electronic games, built momentum for the study of this new media; what was once viewed simply as an object of study within society [4]-[6] now attracted academic interest for the field itself. However, “games as a general category still resist an objective definition” [1], pg. 987.

The processes proposed to define games usually comprise some components perceived during gameplay, for instance: ‘objective’, ‘obstacles’ and ‘agency’, often accompanied by the word ‘fun’ or ‘entertainment’. The ‘Game Design Patterns’ (GDP) [8], an attempt to provide a methodology similar to the ‘Software Design Patterns’ [7], assessed many situational elements that can be present in games. The proposal of a game ontology [9] was intended as a hierarchical classification of elements to be used in a collaborative descriptive system. In this system, users would be able to post their impressions regarding games they have played, using the ontology as a reference to construct their texts.

Regarding frameworks, the Mechanics, Dynamics, Aesthetics (MDA) [10] model proposed a new approach to understanding games by considering aesthetic elements directed towards the player. The Cognitive Behavioral Game Design framework (CBGD) [11] suggests inquiries that should be made or pursued by the game designers in order to design a game. None of the frameworks offer an objective definition regarding games.

The closest definition of a term proposed subdividing games into two categories: ‘act of play’ and ‘game as form’ [12], asserting that the “game as form” promotes the manifestation of “acts of play”.

Considering the aforementioned definitions, the studies by Caillois [5], the proposal that games are a media element [13], the studies regarding the historical development of games [1], and the proposed frameworks or structures presented in section 2, games, as a general category, can be considered to be an emulation of a mimesis [14].

Therefore, games, as a general category, relate to illocutionary acts; they are not only elements to emulate or simulate, depending on the desired degree of fidelity; they also represent situational happenings. The fidelity of these happenings depends on the degree of abstraction intended by
the designer. This demands a system that can capture not only the designer’s intentional representation of the product, but also the player’s perceptive representation.

The system must aggregate the objective elements manipulated and the subjective values produced within a well-defined structure. We labeled this structure Statistical Contextual Structure of Representation (SCSR), and it is assessed considering games the main subject.

The system, named Statistical Contextual Structure of Representation System (SCSRSys), was designed using the Internet infrastructure to reach the interested parties and gather data. Its design also intends to provide a meaningful representation according to the rules of illocutionary logic [15].

The data structure relied on mathematical foundations based on set theory and fuzzy set theory, providing a way to not only aggregate and perform set operations, but also to quantify them in a valid statistical information structure.

The system aims to provide information to and be usable by either game designers, game design students, researchers or players in general.

The remaining sections of the paper discuss the design of the SCSRSys. In section 2, we provide an overview of the illocutionary logic and fuzzy set theories, as well as the frameworks and tools we considered in order to aid the structural design. In section 3, we present the system proposal and discussions regarding its design. Section 4 is dedicated to the system architecture, how data is provided or gathered and how it can be used. Section 5 presents discussions regarding the use of the structure, with some use cases and future developments. The conclusion contains final remarks regarding the overall accomplishments and further research.

II. THEORETICAL FOUNDATIONS

Games have been considered activities or objects that provide entertainment. That definition was the focus of several attempts to assess their concept or structure, seeking elements or practices that comprise them. But a new research approach for Game Studies unfolded when the discussion about games as narrative elements [16] took place. Studies that reference fun [17], [18] define games as elements of communication, relating them to illocutionary acts.

However, illocutionary acts, despite having foundations within logic [15], are described according to subjective concepts. This requires understanding and information extraction within fields that must consider ambiguities and uncertainties, usually demanding a method of classification for such concepts.

The mathematical field of fuzzy sets considers such concepts and provides a set of theories and tools that allow the assessment, quantification and aggregation of data. These tools take into account methods of classification that represent information with degrees of certainty, proving useful for the structure researched and proposed in this paper.

Furthermore, the field of Narratology faced similar situations in the process of developing a system to classify subjective information [19]-[21]. Both Narratology as well as the field of Communications [22] provided useful understanding of the proposed frameworks for game design. As a form of guidance, these fields aided the assessment of their contributions to the field of Game Studies and the identification of their gaps or flaws that were unnoticed or disregarded.

A. COMMUNICATIONS AND ILOCUTIONARY LOGIC

Journalists convey information or messages. Storytellers use messages as an envelope to intentionally evoke feelings or emotions. Games, however, require a process of message delivery or revelation controlled by voluntary, interactive actions performed by the player. Those actions, the message they convey, and the emotions they evoke are intended by the game designer.

This process considers ‘intentionality’ [22], a concept that represents the interpretation of someone’s reality, satisfied by results of voluntary or involuntary intentional acts, regardless of skill.

This concept manifests in the game designer as the desire to create a game, and in the player, the moment he or she desires to play. It is the designer’s responsibility to manifest in the player at least a variant subset named ‘folk concept of intentionality’ (FCI) (Fig. 1) [23], which is characterized by a set of desires and beliefs whose satisfaction depends on the awareness of such desires and on the belief in having the skills to fulfill them.

![FIGURE 1. The folk concept of intentionality. Source:[23], pg. 112.](Image)

After establishing that the game designer’s main goal consists in manifesting the FCI in the player, it must now be satisfied. That is achieved by overcoming the designed challenges, which are communicated to the player considering his or her skill set. The game designer’s intentionality is satisfied by both the game production and its acceptance by the act of play.

Therefore, the game prompts the FCI, whose purpose is to reveal messages and evoke feelings or emotions in the player through his or her voluntary actions.
This process relates to illocutionary acts – the atomic elements of communication [24] –, a broader and more abstract concept that encompasses the speech acts, the theories of intentionality and the aspects of expression and meaning, with their own set of rules for validation [15].

An illocutionary act occurs when a speaker emits an utterance with propositional value to at least one listener. Such utterance occurs within a time and place. Its propositional value is valid in the speaker’s universe of interpretation.

Utterances can assume the form of assertions in order to validate the interpretation in the listener’s universe of interpretation, compose other utterances, or refute an utterance in the listener’s universe of interpretation. They can also assume the form of orders to be carried out by the listener within the speaker’s universe of interpretation. The context that defines the forms of utterance is called ‘illocutionary force’.

In this context, six elements compose the processes for the illocutionary logic: speaker, listener, time, place, speaker’s universe of interpretation and listener’s universe of interpretation. The illocutionary act, consisting of the illocutionary force of a proposition, intends to mold the listener’s world of values and interpretations.

A result of an illocutionary act is valid when the listener’s universe of interpretations matches the universe of interpretations intended by the speaker, taking into account the time and place where the illocutionary act is performed. This dynamic is illustrated in Fig. 2.

A discussion can be considered a dynamic process carried out by at least two persons. The persons alternate their roles as speaker and listener, issuing utterances or providing confirmations within an ordered set of illocutionary acts as illustrated in Fig. 3.
object, with its inherent behavior, for a purpose other than the one it was initially designed for, as presented in section 3.

B. FUZZY SETS

In mathematics, the set theory presents methods and operations to systematically structure and organize information in well-defined categories. Such categories belong to a given context, defining a collection that connects elements according to rules that determine membership [26].

Four main operations can be applied to such collections: union, intersection, negation and pertinence (\(+, -, \sim, \cup\)). Other operations, such as difference and mutual exclusion (\(-^\vee\)), can be constructed combining the main ones. Union and intersection are considered connectives, because they create a new set from two or more sets according to a specified procedure.

Negation creates the complement of the set and pertinence returns a truth value regarding the pertinence of an element to a set or the pertinence of a set of elements, in its totality, to a set. This last process provides the concept of subsets, that is, sets whose elements belong to another set, and supersets, that is, sets that contain all elements that belong to another set.

The set theory comprises the basis of the information field, contextualizing data within the collections that represent meaning, providing information. These are the concepts needed in order to propose a structure to produce knowledge.

But games, as elements that provide illocutionary acts according to the players' responses, which can also be considered illocutionary acts, rely on interpretation, introducing uncertainty regarding elements, meanings and pertinences.

The proposed structure cannot rely on uncertainty or discreet values. It must offer means to identify degrees of validity or success, something related to the degree of illocutionary force [15]. This degree provides a spectrum of possible values to evaluate in analyses that allow subjectivity.

Classical set theory is often inadequate to deal with uncertainty in the rule that assigns the objects to sets, mainly considering empirical objects, which often cannot be defined as such. This limits the use of such elements to a small set of applications, excluding mainly those regarding the fields of arts or humanities, such as communication theory, social sciences or game studies.

To overcome such limitations, mathematicians proposed the membership function [26], adding an extra function to the classical set theory that quantifies an element's degree of pertinence in a collection. Instead of a binary discrete interval defined by the set (0,1), the membership function returns a value within the continuous interval [0,1], where 0 indicates no inclusion and 1 indicates full membership, maintaining the main operations with minor modifications, as follows.

Considering two fuzzy sets A and B:

- A+B: returns a set containing all elements that are in set A or in set B. Considering the membership function and its degree of pertinence, if an element is in both sets, its degree of pertinence will be represented considering the one with the greatest value;
- A.B: returns a set containing only the elements that are in both sets A and B. Considering the membership function and its degree of pertinence, the degree of pertinence will be represented considering the one with the lowest value;
- \(~A\): returns a set containing all the elements not belonging to set A. Considering the membership function and its degree of pertinence, the returned set may contain elements whose pertinence in A is partial, with a value subtracted by 1. Thus, an element with degree 0.3 in A will have a degree of 0.7 in ~A.

The other operations can be obtained following similar logical procedures.

This provides a tool to measure how effectively the information is interpreted by others, considering the membership function. The ones interested in the information can establish a limit value to obtain a discrete structure similar to the usual collections of the set theory.

Therefore, the representation of a piece of information created by an individual is a usual collection. Its membership function relates only to its interpretation of whether the object belongs to the set.

However, considering sets in the same context of information, a quantification aggregation function can result in a set indicating how much the element was mentioned as belonging to it. We can obtain a membership function by using statistical methods when the amount of individuals who informed the data is known.

For each individual \(i\) of a total \(n\), the interpreted set \(S_i\) is a traditional collection, with a discreet binary membership function indicating that, if the element is in the collection, it is a member; otherwise, it is not.

The collective representation \(S_c\) represents the fuzzy set whose membership function is statistically calculated considering the number of individuals \(n\) and the number of times the element was considered a member of \(S_i\), i.e.,

\[
S_c = \left( \sum_{i=0}^{n} S_i \right) / n
\]

Where:

\[
\sum_{i \in \text{inf, sup}} S_i \Rightarrow e: +1 \text{ if } e \text{ belongs to } S_i; e: +0 \text{ otherwise.}
\]

\(S_c\) is the result of the union of all \(S_i\), with each element also carrying its quantification value divided by the number of sets used in the quantification process.

This provides the necessary tools for the designer to assess the players' perception regarding elements in games. The tools present a method in which non-intended elements with few perceptions can be discarded; designed elements with high perception value corroborate the designer's intentions.
Additionally, designed elements with low perception value or non-intended elements with high perception value inform that something within the design process was missing or misunderstood.

Such elements must be within the context of game design, whose assessment, discussions and proposals are presented next.

C. GAME DESIGN TOOL PROPOSALS

“The Art of Computer Game Design”, by Chris Crawford [3], is amongst the first publications to present processes and guidance in the field of video game design, including the author’s personal experiences, which he considered to be essentially artistic. This publication establishes the origin for the comparative reasoning model used in the development of the field of video game design and production.

Although Crawford emphasizes the artistic context, he also considers mechanical concepts, such as hand-eye coordination, resource management activities, puzzle solving and spatial reasoning, to be the “real substance” of games. These concepts influenced the author to propose the genre taxonomic system. Such classification was valid when video games were simple. However, due to the complexities and interdisciplinary composition in modern games, this classification is essentially subjective and ambiguous. Despite that, the industry and specialized media made this taxonomical classification the de facto standard, using it to this day.

Several studies were conducted to assess and enumerate game elements. We will now present their attempts to define a structure that could describe a game or help the process of game design, considering the most recent, relevant and cited, followed by a comparison chart (table 1) of each contribution.

PATTERNS IN GAME DESIGN

The field of software engineering successfully applied concepts proposed in the field of architecture, adapting its design patterns to that field’s specific needs. This led game researchers to also seek answers by following the same procedure, resulting in the Game Design Patterns (GDP) [8].

They propose a set of elements, labeled ‘patterns’, organized in categories according to situational occurrences. Each element indicates how the pattern works, in which situation it is used, and to which patterns it relates (either directly or in conflict).

The GDP presents a basic structure consisting of a set of information such as:

- **Name:** the name of the pattern; a short, specific and idiomatic sentence;
- **Description:** a description of the pattern, where it was identified and how it behaves in the game;
- **Consequences:** tradeoffs and consequences of using the pattern or of its presence;
- **Using the Pattern:** description of how to use the pattern in relation to past projects;
- **Relations:** which patterns relate to this one and how they are related.

This set of information describes the pattern’s behavior, situations when it is used, how to use it and to which other patterns it is related.

In the proposal [27], the patterns were distributed and placed among several collections organized hierarchically. Each hierarchy level represents an abstract context about a behavior, with specific definition in the endpoints, that is, the patterns themselves. This organization attempts to explain why and how the ‘pattern’ presents its proposal within mechanical contexts.

The GDP relies on the designer’s skill to realize how or if a situational pattern unfolds, how it fits within a situation or how to construct or apply it. The patterns are used as assets, and the GDP suggests that classification can be obtained or identified only by assessing or enumerating the patterns present in a game.

The GDP is valuable as a source of information, either to help understand situations where a pattern can fit or to be used as a reference to identify, assess or enumerate patterns in a game to build its structure.

However, the collections proposed to contextualize the patterns do it referencing the game as an object, not the game as an act. They also focus only on the pattern’s inherent behavior, not on how the pattern acts upon the player to satisfy both intentions: the player’s and the designer’s.

This proposal does not consider the use of the pattern as an intentional act, but as an atomic one, stating that if such pattern is present, a specific situation unfolds.

- **Pros:** this proposal differs from the usual enumeration that analyzes games considering the quality of their visual or auditory assets or their narrative and gameplay. The proposed elements – the patterns – are enumerated according to situational occurrences, also indicating how they relate to each other;
- **Cons:** the identification of specific situations is difficult and subjective. The use of game design patterns is one-dimensional, a set that indicates only whether the pattern is present in the game. It disregards the pattern’s use considering the designer’s intention for the player. The proposal does not consider quantification to provide information regarding whether that is a situation; it provides only enumeration, not a proper structure.

The SCSR uses the qualities of this proposal and provides extra layers of context to overcome the gaps, as we will present in the next section.

MDA

Considering the player as one of the factors in the game design process, it is pertinent to assert that subjective elements take part in the game’s composition and must be accounted for in the game design process. While the Crawford model [3] and the game design patterns only presented elements that
compose games in a production perspective, the Mechanical Dynamics Aesthetics framework (MDA) [10] proposes game elements according to their consumption, perception and experience.

It asserts that the fun provided by games is produced via interaction with a system governed by rules, arguing that a game is a collection of mechanical elements. Those elements behave in a dynamic process to produce aesthetic values in the player. This relationship offers a reference point to seek information; the subjective element ‘fun’ is related to the well-studied concept of ‘aesthetics’, providing means to identify which elements trigger which aesthetic responses to use in games.

The MDA splits the game processes into the consumption process or space (Fig. 4) and the design process or space (Fig. 5), whose stages can relate to each other.

The answer to the first question can be found in the processes described by design thinking [28], although it still lacks a formal method of comparison. The answer to the second question depends on how professionals of the other competences take part in the design and decision processes.

- Pros: the MDA considers the player’s experiences as the focus in the design process; it also suggests how the aesthetic values relate to the use of the object assessed in the mechanic elements;
- Cons: it is an abstract theoretical framework focused only on the conceptual design process; its use does not rely on consulting the “player’s space” and does not provide a structure that allows comparison or quantification.

Despite its usefulness, the MDA takes only the “design space” into consideration, referring to the “player’s space” only in terms of the goals the designer wishes to achieve; the players have only a passive role in this framework, despite acting as subjects within it.

Also, although there is plenty of room for contribution by researchers, the framework does not inform how researchers can contribute to its development; similarly to the GDP, it is specifically directed to game designers.

The SCSR considers contexts of behavior within collections that contain the elements, grouping the behaviors according to contextual information in another set of collections, representing intentions. This proposal allows for the contribution of all participants in the gaming field, namely: designers, students, players and researchers. The structure also offers proper methods of quantification and comparison, not only to classify, but also to provide information and contribute to the Game Studies knowledge base.

GAME ONTOLOGY

The approach of the Game Ontology Project (GOP) [9] proposed an ontology to identify important elements in games. The ontology organizes the elements in a hierarchical structure, contextualized according to behavioral attributes.

The GOP differs from the GDP because it does not intend to use situational references as rules to create games. The assessed elements are categorized according to the identification of their commonalities and differences, analyzed in a range of concrete examples.

This proposal does not intend to classify games, but to catalogue elements within a hierarchical structure. This catalogue can be used to describe the game according to its design space. It can also work as a framework to explore research questions related to games and gameplay, as proposed by the gamebricks classification [29], or to construct a vocabulary for describing, analyzing and critiquing games.

Such vocabulary is proposed as the set of ground rules of the gamelog project [30], a system that attempts to develop the concept of ‘ludoliteracy’ in players, suggesting that they...
write essays about the games they are playing according to those rules.

The GOP disregards the process of game design in its methodology; its primary intention was to focus on what can be perceived or experienced in games as players. This aspect provides some fuzzy boundaries about the elements assessed and the games that exemplify them. Elements were gathered from gameplay activities without a proper system for validation of the empirical process.

The top level of the ontology consists of five categories: interface, rules, goals, entity and entity manipulation. Each category consists of: a description, the child elements, the composite elements and their description and behavior. The leaves represent the supposedly concrete elements, such as ‘push button’ or ‘joypad’, and provide, as their parent nodes, examples of which games the node can be applied to, and the form of application: ‘strong’ or ‘weak’.

Each element of the ontology consists of a set of attributes: a succinct title or name, such as the name of a pattern; its description; strong and weak examples of games that apply the element or where it was identified; its parent element and possible child or part elements. The parent-child relationship provides the notion of type and subtype, with the part elements consisting of compound elements constructed from the combination of others.

The study does not offer guidelines about how it can be used in the design process, nor about how researchers can effectively contribute to the process; however, the author provides a website created with a collaborative documentation system. This system presents the elements assessed so far, indicating where in the behavioral hierarchy they are placed and what their attributes are.

- **Pros**: the study proposes an organization of elements present or perceived in games, categorized according to their perceived inherent behavior;
- **Cons**: although the study mentioned having been constructed with players’ participation, it does not detail how others can contribute; it does not mention the possibilities of quantification nor any details about the validation of the data. The classification considers only the element’s behavior, not its intended use. The ‘weak’ and ‘strong’ examples denote that there are degrees in the perception of the element.

Like the GDP, the GOP provides useful data enumerating several elements that comprise games; however, unlike the GDP, the GOP relies only on the element’s behavior for its classification system. This approach presents more concrete concepts, resulting in a less ambiguous categorization than the classification by situation used in the GDP. Both provide resources to be used as elements or assets in the SCSR.

**DISTINCTIVE FEATURES**

The representation structure is filled differently among individuals, whether they are designers or players. This generates ambiguity, usually due to describing the same representation with a different name or providing the same name for different concepts. That is why, to avoid inconveniences, the elements assessed must be simple: a word or compound word.

The Distinctive Features framework (DF) [2] proposes such a protocol. It suggests compound words such as “theme present” and “theme relevant” or “strategy possible” and “strategy variable”, among others, in a form that indicates whether the element is intentionally present or absent in the game. A game like chess, for instance, where the theme is not relevant and strategy is possible and variable, would be represented as [+theme present] [-theme relevant] [+strategy possible] [+strategy variable]. If an element is intentionally absent, it is indicated by the minus sign instead of the plus sign. Adventure games, for example, initially lacked complex graphics due to device limitations, but nowadays lack graphics only by design. In that case, the feature [-graphics] would be part of the structure.

The sequence representation presented by the DF is easy to recognize and allows for some comparison. For instance, considering a digital and an analog version of chess, some elements will be present in both, and some will not; however, the framework does not allow the designer to perform set operations such as union or difference in a process to properly extract information. A union operation of a graphical version of a text-based adventure, with the same gameplay and the same story, would result in a set containing the features [+graphics] and [-graphics], creating, at the very least, a logical absurdity.

In addition, the context represented by the sequence only acts in one dimension, the intentional presence or absence of the element in the game. The proposal regarding the dimension of meaning is to provide additional descriptions of what the presence or absence means, still allowing the possibility of ambiguity, a flaw the authors suggest can be mitigated using heuristics.

Heuristics are proposals made by researchers according to experiments and empirical data. However, players only take part indirectly, in the process of the data gathering, which is orchestrated and supervised by researchers. The DF is the closest to the SCSR, but still has flaws.

- **Pros**: it is a discreet and concise framework to describe game elements and their meaning; it provides some form of comparison to perform analysis and extract information from games;
- **Cons**: it does not provide methods for quantification or mathematical models for comparison; the player participates indirectly in the process of element assessment or game classification, assisted by researchers, not in a free process; common set comparisons may generate representations that are logically absurd.
Although in a different form, the DF provided the related contexts to construct the SCSR, a structure that provides means of mathematical manipulation. The manipulation process considers the set theory and also implements methods for quantification and validation of data by comparison according to the fuzzy set theories [26].

CHARACTERISTICS CHART
The frameworks and tools presented contributed to some aspects of the proposed structure, which we will present in the next section. Some of these frameworks, such as the GDP and GOP, provide data to compose assets for the SCSR structure in games.

Table 1 displays the characteristics that are present or absent in each of the frameworks analyzed and that the SCSR aimed to provide, namely:

- **Mathematical representation:** The structure must be able to be represented mathematically, indicating the viability of a system at least in terms of storing and retrieving information in a given context;
- **Mathematical operation:** The structure must provide means to perform mathematical manipulations, at least in order to enable comparison;
- **Closure:** the mathematical operations provided must result in a structure with the same characteristics of the structures operated. This indicates not only that a system to store and retrieve such data can be designed, but also that it can be designed to manipulate such data and provide information within the same context;
- **Data quantification:** Considering that the data will be mostly retrieved empirically, the structures must provide methods for data quantification;
- **General participation:** the methods to provide empirical data must depend only on the participant’s will and on valid ethical rules of participation; the participant must not depend on others to provide information;
- **Context identification:** the structure must contextually identify the element assigned to it, in the place it is assigned;
- **Intentionality towards the player:** the structure must indicate what role the element was intended for, or realized by, the player, indicating that an intent in the context of illocutionary acts can be described;
- **Possibility of game categorization:** The structure must provide the possibility to, once assessed, identify one game or one category of games;
- **Easy element identification:** The structure must employ concisely named elements to facilitate identification and understanding of the element’s meaning and context.

### III. STATISTICAL CONTEXTUAL STRUCTURE OF REPRESENTATION IN GAMES

Despite the attempts made by all proposals, an appropriate representation requires a proper understanding of the context analyzed. In games, this understanding is called ‘ludoliteracy’ [30] and this is the intention of gamelog, a tool for game description considering some ground rules. Gamelog also allows the study of how games are perceived, but it presents the analysis in a descriptive form. Each entry for a game would need to be read and synthesized in order to identify common elements or possible structures, a time-consuming process.

The SCSR must be easy to recognize or produce in relation to the game played or observed. The player must adequately interpret the amount of conceptual aspects the game presents; how he or she interacts with the game; what is expected of him or her in the interactions; what are the elements inherent behavior; to what end they were intended and how they are organized.

The most important aspect is the possibility to perform operations within representations to allow comparisons or quantification. That means the represented data must present meaningful information, be easy to read, and provide methods of operation in the mathematical context.

The collection-like operations and the synthesis of elements are an important contribution made by the DF; the GOP presents a collection of elements hierarchically organized according to the concept of behavior, relating their descriptive role to their behavior, but with no space to specify intentions.

For instance: a screw, an element that has specific behavior, can perform the role of holding together two pieces of wood, but it can also act as a gear in a guitar tuning mechanism or create a hole in a car tire.

The behavior alone is not enough to contextualize the element in the whole concept. The role represents how such behavior acts in the game, creating an additional layer of classification, something the proposals lack. This layer relates to one concept: the role assumed by the element or the role that was realized in the interpretation.

The representation for the information protocol demands two layers: the external one representing the role intended for

<table>
<thead>
<tr>
<th>Study/Characteristic</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game Design</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patterns</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDA</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game Ontology</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gamelog</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distinctive Features</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCSR</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
the elements or perceived by the player, and the internal one corresponding to behaviors that take part in that role.

A. ROLES AND BEHAVIORS

The behaviors can be abstracted from the GOP but refined to better represent the context. They were extracted from the hierarchical structure, with some hierarchical levels considered as elements themselves, either with abstract or conceptual representation. The same can be applied to the patterns of the GDP. Those elements can be used without effort in the assessment process. The proposed behaviors are:

- Interactivity: Interface or entity manipulation elements. All elements that allow or indicate possibilities or require the user to interact in order to get a response. Concrete elements (such as buttons), affordances (such as ‘move’ or ‘shoot’), and visual indications (such as an object shining) are examples of such a set of elements;
- Mechanics: Rules, that is, goals or algorithms to illustrate the behavior concept of mechanics. They are responsible for processing the responses to the player’s interactions (or lack thereof), and represent the phenomenological aspect of the game. Examples of elements in this category are: kill, reach, subtract, match or even the abstract element ‘puzzle’;
- Gamification: Elements that store values, data or states. If an element needs to be verified by the rules, this is the behavior to categorize it. E.g.: high score, energy, life, running (state);
- Ludic: We decided to use the Latin concept of ludus to represent behavior related to concepts of psychology, pedagogy, sociology or philosophy. This category can be further expanded to represent more detailed behavior. Some examples of elements in this set are theme, music, story and main character. Even the abstract concept ‘puzzle’ can be part of this behavioral category;
- Device: This represents the physical behavior, the pieces needed to represent a specific game. A chessboard and its 32 respective pieces, or a console, cartridge and TV set, for instance, are devices.

Games compel or dissuade players to perform actions through insinuations or prohibitions, using artifacts to mask such manipulations. Both concepts, compel and mask, can be defined, respectively, as acts of persuasion or aesthetics, indicating two of the functions the elements may perform. Both functions comprise what can be called ‘Paidia’ [5] pg. 27.

In contrast, the concept of ‘Ludus’ excludes the components of exuberance or fun in games, maintaining only components of orchestration “to bind with arbitrary, imperative, and purposely tedious conventions” [5] pg. 13.

Those are the three main functions, or roles, the elements perform, but a game also needs form, that is, elements that make it real and possible to interact with, not just a concept in the player’s mind. This represents the last function: reification.

The four functions, persuasion, aesthetics, orchestration and reification, can synthesize the episodes described in the second section, with each function containing a set of behaviors to categorize the elements:

- Aesthetics: Manipulates affective responses, relates to the field of arts and aims to reach the aesthetic behavior concept [31]. Used by the game designer to manipulate feelings or emotions, or to mask persuasive situations. Interactivity and ludic are the main behaviors that perform in this role;
- Persuasive: Aims to guide the players’ behaviors, compelling them to or dissuading them from performing actions, molding their perception according to a designed or intended situation. Follows methods contained in the field of psychology [32] with correspondence to intentionality [22]. Interactivity, ludic and gamification are the main behaviors that perform in this role;
- Orchestration: Defines the affordances of elements, the cause and effect relationships, how the act of playing progresses or escalates, and the victory or defeat conditions. Establishes a relationship between objects and their behavior in terms of the aesthetic or persuasion functions. Interactivity, mechanical and gamification are the main behaviors that perform in this role;
- Reification: The set of elements that form the game as an object, such as specific data sets or specific algorithms to be manipulated by the orchestration function, for instance: specific software libraries or game engines to be used in the production or execution of the game. Interactivity, mechanics and device are the main behaviors that perform in this role.

B. THE FUNCTIONAL STRUCTURE

Behavior and roles characterize the Statistical Contextual Structure of Representation Function (SCSR Function or SF), a two-layer representational structure, with the outer layer – the functions – representing the roles performed by the elements and the inner layer corresponding to their behavior.

With the aesthetics (A), persuasive (P), orchestration (O) and reification (R) functions composing its body, the SF can be represented graphically or in mathematical form as follows:

Each component function composed by a set of behavior, defined as interactivity (I), ludic (L), gamification (Q), mechanics (M) or device (D), is represented as:
A = \{I, L\} \tag{3}

P = \{I, L, Q\} \tag{4}

O = \{I, M, Q\} \tag{5}

R = \{I, M, D\} \tag{6}

The behavior sets and the functions are not mutually exclusive. An element categorized in a behavior set in a function can also be categorized in another. For instance, the ‘high score’ element can belong to the (Q) behavior set of the (O) function (Q_o), since it records who earned the highest score, but also to the (P) function (Q_P), inciting players to ‘beat the high score’.

Another example is the ‘puzzle’ element, which can be an (M) element of the (O) function (M_o), as a phenomenological behavior, but also an (L) element of either the (P) or (A) function (L_P, L_A), a ludic behavior representing challenges or invitations to the player.

Therefore, if a game has a theme, but it is not relevant, it is simply an aesthetic element, categorized within the ludic behavior in the aesthetic function. If the theme is relevant, such as the medieval setting in a war-game [1], it would also belong to the ludic behavior of the persuasive function. This characteristic minimizes the need to assign extra descriptive attributes to the element.

The final SF representation is as follows:

\[ SF = \{I, L\}_A, \{I, L, Q\}_P, \{I, M, Q\}_O, \{I, M, D\}_R \] \tag{7}

The SF can be considered within the field of discrete mathematics, with its set of operations such as union, intersection and subtraction, resulting in a data set that is still an SF, denoting closure. This provides the game designers and researchers in game studies with a tool to generate or represent a source of information to help them in the process of game design.

The representation of games with the SF can be obtained by statistical quantification of elements provided by players and designers by means of the computational tool described in the next section.

\section*{C. OPERATIONS AND VALIDATIONS}

When considering a design process, the designer may start it with abstract and subjective conceptual elements, or with concrete, objective ones. These types of process respectively represent the top-down and bottom-up design strategy. Both approaches need documentation, categorization and correlation of elements regarding their behavior and meaning. The designer’s intention plays a part in this assessment, defining what role the concept or the element should perform, and this must be communicated to the team.

The SF offers a concise way of structuring elements regarding their behavior and the role intended for the end product. The featured operations allow the elements to be assessed considering different contexts, starting either from the conceptual meaning towards a concrete element or from a concrete element seeking its conceptual meaning.

The union operation allows the designer to describe independent contextual SF’s. For example, the assessment of the SF_c for elements of context such as narrative elements; of the SF_m for the ones related to mechanics; of the SF_p for the ones related to interactivity, affordances or players’ agency; and of the SF_p representing aesthetic or persuasive elements to evoke emotions from the player. Each team can assess the representation considering activities of their competence or field of knowledge and apply the union.

\[ SF_{game} = (SF_c + SF_m + SF_p) \] \tag{8}

This results in an SF representing what has been assessed up to that time for the game. The same can be done considering game levels as an example; each SF associated with each game level, and the union of all of them representing the game.

Furthermore, the intersection provides a method to identify common elements, concrete or abstract, in different games. Considering three games A, B and C, the designer can construct the SF of each and apply the intersection.

\[ SF_{genre} = (SF_a.SF_b.SF_c) \] \tag{9}

This results in an SF containing only the elements along with their role and behavior present in all games. This represents the minimal set of elements belonging to that kind of game, a useful approach to identify minimal sets for genres, for instance.

Similarly, the difference operation provides the results of which elements (and their respective roles) belong to a game, but not to another, a useful method to compare games or different game assessments, or even identify characteristics that differentiate one genre from another according to the minimal sets representing each genre.

Such operations provide opportunities for designers to assess their users’ feedback, realizing what elements or concepts the players identify, how easily they identify them, and whether they identify them correctly.

Additionally, it helps players to develop the ‘ludoliteracy’ concept [30], providing means for them to understand the relations regarding the elements, their behaviors and roles, without the descriptive process of gamelog.

The SF acts as a game dictionary, enumerating the elements that were assessed for the game and categorizing them according to their behavior and role, and provides information about how the designer’s SF and the players’ SF differ. The result of this difference can indicate how successful the designer was in presenting the intended universe of interpretation to the player, as described in the illocutionary logic [15].

Even so, there are lots of elements or contexts to assess, enumerate and categorize. In order for the information to be valid, the SF number for a given game must be large, which demands a system to collect, classify and operate such a volume of data.
IV. THE DESIGNED SYSTEM: SCSRSys

The genre taxonomy was first mentioned by Crawford [3] in 1985, a time where everything was simpler due to lack of resources. Processing and storage were costly, graphics were virtually inexistente, the computers were slow and expensive and it was rare to know someone who had one in his or her home at that time, a situation quite different from today.

Today, almost everyone has a computer that is much more powerful than the ones in 1985, smaller, with more resources and able to transfer data quickly. The situation is favorable for the development of systems able to easily gather a vast amount of data, processing it in seconds and providing valid and useful results.

This section describes our intention in creating the system to retrieve, process and store data considering the SCSR (SCSRSys): the architectural requirements and how they were satisfied; the system’s topological architecture; the data structure; the implementation issues and the tests performed to guarantee the reliability of the operations.

A. MAIN INTENTION

In addition to the intention to represent games, there is a need to rely on a large volume of data in order to provide statistically satisfactory results. Obtaining this large volume of data can be resolved using current methodologies of pervasive data gathering.

Users are constantly connected, interacting with some technological artifact, all the time. Since the SCSR considers the free participation of the general public, a system accessible via a public channel, working without interruptions, is not only desirable, but mandatory.

This guided the decisions for the architecture and development, as well as the data structure’s evolutional design. This design is now only in its first version, but it is able to improve at each iteration, in a continuous developmental project, maintaining the concepts and meaning proposed.

It is our intention that future developments of the structure can be mapped from one broad category, the ludic behavior, for instance, to other specific behavioral categories derived from it, allowing users to convert the data from older to more recent versions to properly represent the game.

In addition, it is the intention of the team to provide data for other researchers, students or enthusiasts to study, process and extract useful information, which demands an open model.

Therefore, the system must provide data access, with data gathering performed by the general public, considering a high-availability profile, storing and processing information within the context of the SCSR, demanding the architectural requirements and the design of an architecture in order to be properly developed.

B. ARCHITECTURAL REQUIREMENTS

To provide services uninterruptedly, wherever the user requires them, the system must be accessible remotely; therefore, it must be executed in a server, and client access should be as independent as possible.

The current best practices of systems engineering focus on RESTful API’s [33], served through HTTP channels to a browser application, able to be executed and manipulated within desktop or mobile browsers.

The proper software design approaches advise the division of the system into two layers: the Frontend System (FS), responsible for providing the data visualization and some operations for the client to request data, and the Backend System (BS), responsible for providing data manipulation, retrieval, storage, validation and security features.

The database must consider distributive architecture, not only due to the expected or desired number of users, but also to provide mechanisms of redundancy to preserve data. In addition, it is desirable that the database be distributed, as well as the database system, providing synchronization mechanisms to guarantee the validity of the data.

The programming language must easily provide means to develop the requisitions and represent the SCSR structure, with its attributes and operations, so that the maintenance processes do not demand excessive cognitive effort from the developers or engineers.

Considering these requirements, it became clear that the best decision was to deploy the system using cloud services. Such services provide the high-availability required by the system, guarantee and allow continuous integration practices within a distributed environment, without the need to reach out to the users for updates. In addition, it allows the hosting of both the frontend system and the backend system; the former, developed to provide HTML pages that request services from the latter, which provides responses via RESTful API and communicates with a distributed database server architecture, designed with NoSQL technologies to better represent the required structure and quickly provide the data requested.

Regarding the software requirements, the system needs to receive, send and store the SCSR representing a game in a user’s view. Therefore, it needs features to store information about games and about users in order to map their individual interpretation of the SCSR.

In addition, since the genre taxonomy is considered the de facto standard, it is possible to provide a statistical method to indicate to which genres the game can be associated.

We also considered the need of storage for each game-quantified SCSR and the fuzzy set representation of the game SCSR, to be computed on a scheduled basis in order to provide quick responses, as well as a method for users to add contacts. This would work similarly to a social network, providing a feature for a user to compare his own game SCSRs with the ones assessed by his or her friends.

Several features were disregarded in this stage due to time constraints, but since the architecture was designed considering future enhancements within continuous integration, they can be implemented in future iterations. We
describe the details of architectural topology, data structure and implementation in the remainder of this section.

C. ARCHITECTURE

The basic architectural topology subdivides the system in two layers: a) the backend, executed in the server, awaiting requests for data or operations, and b) the frontend, hosted in the server and delivered to the user upon request, but executed in the user’s computer.

The backend is subdivided into two subsystems. The first one provides a RESTful API [33] endpoint to interface with the frontend system by means of an HTTP server, which performs the communication roles and the computational requests. The second one provides the data storage and retrieval features. This is a distributed database system, managed by a synchronization service in order to guarantee data sanity and availability, all depicted in Fig. 6 and Fig. 7.

FIGURE 6. The architectural topology of the SCSRys frontend and API. Source: The Author.

The frontend is provided using the one-page application (OPA) methodology, using modern application development frameworks for execution in browsers. This methodology is focused on providing the features of data visualization and the user experience methods for a set of operations, such as: data comparison, friendship requests, user search, game consulting, user log in or sign up, the messaging system and the methods for data retrieval and storage.

The frontend is the application where users will provide the data for the system. It communicates with the backend via HTTP requests to perform the following operations: logging in, suggesting a game, starting or modifying an SCSR, consulting the consolidated SCSR of a game or a friend’s SCSR and performing comparisons or the operations mentioned in section III.

Other applications may also request access to the services. The backend is a RESTful application; the applications only need to implement the code according to the data structure documentation and the services provided. For all purposes, the Internet-connected applications are the interfaces for empirical data gathering.

The backend system is subdivided into three main systems: a) validation and security, b) ‘semantics processing’ and c) social representation, described as follows.

VALIDATION AND SECURITY

The validation and security subsystem (Fig. 8 and Fig. 9) consists of the features to identify who is using the system and what his or her roles are, and to maintain the session identification in order to guarantee the privacy of the data, the availability and the sanity of the system.

FIGURE 7. The architectural topology of the SCSRys backend. Source: The Author.

In order to provide such features, the system must store user identification in the database and, in case of a valid user log in, generate a valid token with an expiration date, to be part of the messages that indicate a valid request or transaction.

This token is sent to the frontend system and communicated to the overall system in the HTML headers of the HTTP requests. If the token is invalid or the date is expired, a new login request is issued with a new valid token generated upon successful operation.

The identification of user roles indicates the responsibilities of the user within the system, defining three categories:

- The creator role allows the user to manipulate every type of data within the system. It is intended only for the system designers and is to be used within the development and test stages, not within the production stage;
- The admin role allows the user to manage some
information, verify the use of abusive language, and remove such data from the system to maintain the sanity of the information. This type of user is also able to manage users’ status, blocking, unblocking or banning them from the system. The admin also has the ability to consult, upon request, the transactions performed within the system in order to identify possible gaps or flaws;

- The user is the role assigned to the common individual that uses the system. It allows him to log in, consult, provide or update his SCSR for games catalogued in the system; request a game to be added in the catalog; consult the publicly available basic user information from the database; request, block or accept friendship requests, send messages and consult friends’ SCSRs for games they have assessed.

**FIGURE 8. Basic Security features within the system. Source: The Author**

**FIGURE 9. Security features considering user roles within the system. Source: The Author**

**SEMIANTICS PROCESSING**

The semantics processing consists of the features related to the SCSR data to be manipulated by storage and retrieval or manipulation requests made via the API endpoint.

The storage is a direct feature; the SCSR request sent to the system informs which user is requesting its storage and to which game it refers, requiring that both the user and the game exist in the system and that the user be logged in.

At each SCSR storage request, a history transaction is made to identify the data of the request, the elements added or removed and to which function and behavior such elements were added or removed, creating a historical record to provide information for users to perform historical analysis.

Three main features compose the retrieval of an SCSR: the retrieval of the user SCSR for a game, the retrieval of the consolidated SCSR of a game, or the retrieval of related SCSRs regarding a game.

The first one is a simple request where the user and the game must exist in the system, the user must be logged in, and he or she must have provided the SCSR for that game. This will return the SCSR data the user assessed for the game or a notification response in case the user does not have an SCSR assessed for that game.

The request of consolidated SCSR game data will return the statistically quantified data for that game by retrieving the SCSR assessed for that game from all users that assessed it, and performing the quantification operation. This will only be performed if the time limit of that consolidated data for that game expired, otherwise it simply returns the requested information already stored in the system. In order to encourage users to contribute, the consolidated SCSR of a game can only be available for the user if he or she contributed to the SCSR for that game.

The retrieval of a related SCSR regarding a game consists on retrieving an SCSR that does not belong to the user. This operation is restricted to data from other users that are in the user’s friends list or to users with administrative roles.

Finally, the manipulation operations requests consist of comparison, union, intersection and difference operations as described in the fuzzy sets section.

**SOCIAL REPRESENTATION**

The social representation was designed to provide means for users to compare their assessments with the assessments made by other people in a controlled manner, requiring the agreement of both parties to provide access to their SCSR assessments.

The SCSR assessments of friends can only be consulted if both parties assessed the game SCSR. A user cannot consult a friend’s SCSR representation of a game he or she has not assessed. This limitation encourages the user to provide his or her understanding of the game in order to compare it with friends.

The social features consist of the usual ones present in the most common social networks, such as: friendship request, blocking, unblocking and friendship cancelation. It also
focuses on a communication component, allowing users to exchange messages within the system in order to provide means for collaborative assessments.

D. STRUCTURE

The SCSR function must be transmitted to and from the backend and frontend and implemented to allow easy data identification for the debugging process. It must also be possible for third parties to develop their own applications using the provided API.

The representation must consider the two-layered structure, as mentioned in section 3, with the idea that such structure is able to evolve, meaning that a behavior set can be split into two, for instance, or a new behavior set can be represented via operations considering two new sets.

Thus, the structure is organized into three contextual elements:

- The schema version: a JSON structure that indicates the current version of the structured data transiting in the system. The first version is indicated by the value 1, although the current use case project considers the value 0. Future SCRS versions can only be applied via researches providing mathematical validity related to the current version, that is, there must be a conversion system from one to another. An example of the schema version is depicted in Fig. 10;

- The version correlation: this schema indicates how the function or behaviors from the previous version relate to the current version. For instance, it was mentioned that the ludic behavior set represents elements whose behavior fits within the fields of pedagogy, philosophy, psychology or sociology. Thus, considering that such behavior set unfolds into two new ones in the newer version, say ‘set A’ and ‘set B’, the version correlation must inform that ‘ludic behavior’ => (‘set A’, ‘set B’), providing a method to map elements of previous versions into the current ones and update the data accordingly. An example of the version correlation structure is depicted in Fig. 10;

- The structure itself: this is the structured data itself, containing the data to provide information for the schema validation and its respective data collection, which consists of the semantic and contextual information within the two-layered structural proposal. The depiction in Fig. 11 does not inform the elements.

E. IMPLEMENTATION

The delivery of the system represented by the architectural topology, with the features intended and data structures proposed, is possible due to current languages and technological tools.

We required a development language with data representation similar to the JSON structure, allowing for the easy development of abstract structures that implement the set theory operations, such as union, intersection, pertinence and difference, among others.

```json
SCSRSchema = {
    "version": 2,
    "SCSR": {
        "User": "",
        "Game": "",
        "Functions":{
            "Aesthetic_function": {
                "Behaviors":{
                    "Interactivity",
                    "Ludic"
                }
            },
            "Persuasive_function": {
                "Behaviors":{
                    "Interactivity",
                    "Ludic",
                    "Quantification",
                    "State"
                }
            },
            "Mechanical_function": {
                "Behaviors":{
                    "Interactivity",
                    "Mechanical",
                    "Quantification",
                    "State"
                }
            }
        }
    }
}

SCRSRSchemaMapping={
    "Previous": 1,
    "Current": 2,
    "mapping": {
        "Persuasive_function":{
            "Gameification": {
                "Quantification",
                "State"},
            "Mechanical_function":{
                "Quantification",
                "State"
            }
        }
    }
}
```

**FIGURE 10.** The schema indicates the current version of the structure and how the current and previous versions relate to each other. Source: The Author

In addition, we needed a development language whose documentation set, support and easy-to-apply coding standards, alongside an ample library of computational resources, helped improve development speed.

Considering these requirements, we chose the Python language due to its dictionary data type closely matching the JSON structure, its enforced indentation programming style (which guarantees easy readability), and its object-oriented features – more specifically, the abstract base classes of mutable data sets, which implement the necessary features to develop the description classes for the types of structures needed. That means it aggregates elements and provides the same features and operations as the mathematical sets.

The Python codebase, contributed to by millions of users, offers a large set of libraries, including the ones that perform object relation mapping with databases, as well as a communications framework that uses several communication
channels – HTTP included –, such as Flask\(^3\). This provides an easy way to develop RESTful API systems. The language features that provide easy installation and deployment also contributed in the decision.

The tools used to implement the system consisted of a framework for web applications, the database system and the frontend technology as well as some services, namely:

- **Flask**: a micro framework for Python intended to offer easy steps to develop and deploy web applications. Its features allow developers to code tasks based on the pattern of the requested URL, a useful choice for development of RESTful API systems;
- **Mongo DB\(^4\)**: a database management system designed considering the NoSQL architecture. It provides methods of data storage considering several structural approaches consisting mainly of a JSON structure-based approach that is useful for the system’s needs;
- **VUE.js\(^5\)**: a javascript framework to develop webpages considering user interactions and reactive agents, providing easy methods to design user friendly webpages;
- **Cloud Services**: In order to guarantee the system’s high-availability and scalability, a cloud service is needed to host, at least, the backend subsystem in order to provide the requested features;
- **Container infrastructure**: The container infrastructure is used in the development stage to provide a clean environment for development and testing among developers. It avoids the effect of “it works in my computer”, common in development environments.

Some of the tools mentioned required hiring a service provider.

In addition to the architecture definition and the tools assessed, the system design adopts the object orientation paradigm [34], which provides an appropriate mapping relationship between conceptual and concrete structures. The main system is comprised of 23 classes divided into the following packages:

- **Installation**: the package that implements the routines to start a SCSR system with the initial data. It creates the database and populates it with an admin user, the basic game genres and three initial games and elements retrieved from proposals of the GOP and the GDP, preparing the system to serve the API;
- **Administrative**: administrative tasks and user representation to provide the basic operations to use the system, storing user information and credentials and controlling his or her actions. This context contains the classes: User, AdminGroup and Admin;
- **Home**: main classes to represent and operate the system such as Login, SignUp, GameRetrieve and GameRequest;
- **Social**: classes to represent and operate in the context of a social network such as FriendsList, FriendsRequest, Transactions and Messages;
- **Game**: classes to represent and operate the context of the game objects to be assessed. The classes composing this package are:
  - The Game class, representing the games within the system;
  - The Genre class, representing the genres to be assigned to games by users;
  - The UserGameGenre class, indicating that a user has assigned a given genre to the game;

\[\text{FIGURE 11. The data structure representation for the schema in the current version with the contextualized data. Source: The Author}\]

\(^3\) www.flask.pocoo.org - accessed on May 23rd, 2018  
\(^4\) www.mongodb.com - accessed on May 23rd, 2018  
\(^5\) www.vuejs.org - accessed on May 23rd, 2018
o The GameGenreQuantification class, used to quantify how many times a genre has been assigned to a game and what the genres assigned to that game are;

o The GenreGame class, used to provide which games are assigned to a respective genre;

• SCSR: classes to operate in the main context of the system. The base classes provide the topology of the SCSR structure. Composed of:
  o Element: the atomic element to be assigned to a behavior within a role in the SCSR;
  o ElementReference: a reference to consult details about the element;
  o Behavior: a specialization of an abstract mutable set to aggregate a set of elements in a collection, providing mathematical operations and contextualizing them regarding the base behaviors as discussed in section 3;
  o Function: a specialization of an abstract mutable set to aggregate a definite number of behaviors in a collection, providing mathematical operations and contextualizing the behaviors according to their roles, as discussed in section 3;
  o SCSR: a specialization of an abstract mutable set to aggregate a definite number of functions in a collection representing the SCSR, providing mathematical operations and assigning it to a user and a game present in the system;
  o ConsolidatedSCSR: a specialization of an abstract mutable set to aggregate a definite number of functions in a collection, with data quantification to represent the membership function of the elements within the SCSR, providing methods to extract valid SCSR structures according to how much the element must pertain to the consolidation. The ConsolidatedSCSR provides the representations described in the fuzzy set theory.

The simplified classes diagram is represented in Fig. 12, representing version 0 of the system, a fixed structure implemented to provide the use case tests in order to validate the mathematical operations. It does not implement the processes described considering the continuous evolution of the structure, such as the version schema, for instance.

F. TESTS
The validation of the system and of the structure consisted of a series of automated tests. They were created following the unit tests methodology [35] in order to verify the mathematical validity of the structure with the system proposal. They were subdivided into atomic and related tests, with the atomic test operating in each class only and the related tests operating considering the relations between classes, the former guaranteeing persistence and data validity, the latter to guarantee systemic and mathematical validity.

The atomic tests consisted of:
  • Element: create, retrieve, remove element;
  • ElementReference: create, retrieve, remove reference;
  • Behavior: create, retrieve, remove behavior;
  • Function: create, retrieve, remove function;
  • SCSR: create, retrieve, remove SCSR;
  • ConsolidatedSCSR: create, retrieve ConsolidatedSCSR
  • Game: Create, retrieve, update, delete;
  • Genre: Create, retrieve, update, delete;
  • User: Create, retrieve, update, delete.

FIGURE 12. The simplified class diagram of the SCSRSys classes and their relations. Source: The Author.

The related tests consisted of:
  • Element: retrieval by behavior, function, SCSR, consolidated SCSR and element reference;
ElementReference: assignment via element addition, retrieval by element;

Behavior: retrieval by function, adding element, consulting element, removing element, union, intersection, difference, addition (quantification), history production, history reversion;

Function: retrieval by SCSR, adding element to behavior, adding behavior, consulting element, removing element, union, intersection, difference, addition (quantification), history production, history reversion, reset;

SCSR: retrieval by user, retrieval by game, add user, add game, adding element to behavior within function, adding function, adding behavior to function, consulting element, removing element, union, intersection, difference, addition (quantification), history production, history reversion, reset;

ConsolidatedSCSR: quantification, membership cut, comparison with SCSR

UserGameGenre: creating a representation of a user assigning a genre to a game and removing the assignment of a genre to a game validating the result in the GameGenreQuantification data;

All tests were successful, indicating that the core system to operate the SCSR is ready to be used and it is possible for third parties to verify it by requesting to contribute to the project.

V. DISCUSSIONS

The SCSR benefits from the existing game elements already assessed, such as the GDP, GOP, Machinations [36], gamebricks [29] and even offers the possibility of contribution of new elements by users, provided they inform references to validate the elements.

In addition, the class representing elements contains a mechanism to assign the element to another persisted element, indicating that the current element cannot be considered valid. When the element is retrieved, via behavior, the system updates the reference where the deprecated element was placed, keeping the meaning updated.

The SCSR can be used for other purposes besides game classification and structure representation. It can provide a method to identify minimal sets of elements according to user preferences or according to genres; for instance, to retrieve the consolidated SCSR for each game that has the genre ‘action’ assigned to it by more than 85% of the evaluators. With each consolidated cut set, one can perform the intersection operation to extract the common structured elements for action games, providing the minimal representation for that genre.

The minimal representation obtained, for example, for two different genres could then be combined to represent an action-adventure multi-genre game. This representation can be used to design a game that contains the elements and can later be validated by submitting the game to be assessed by users. Its acceptability can be evaluated within the market, providing a useful tool for production researches.

The SCSR structure can contribute acting as a resource at the research stage of the process of design thinking [28], providing trustworthy analytical data for game designers, improving the field of game studies and also offering a tool to transmit information in a structured manner, contributing to the process of game design education.

VI. CONCLUSION

This paper presented a system to implement the statistical contextual structure of representation (SCSR), proposed in order to store, retrieve and manipulate data regarding games, relying on formal mathematical concepts within the set and fuzzy set theories.

The proposal was built via analysis of previous tools or frameworks, identifying their qualities, flaws and gaps to construct a representation.

This representation was intended to enumerate elements perceived, witnessed or considered present within games, providing means to classify them according to their behavior or situational context.

The representation aimed to refer to the elements in a simple and concise manner, categorizing them regarding not only their inherent behaviors, but also their intended uses for the player or considering the acts perceived by the player, resulting in a two-layered structure with contextualized meaning in each layer.

In addition, the proposal only requires from researchers the responsibility to determine the form of the structure. However, to the general public, tapping the collective mind, it presents the role of assigning the place of a recognized or perceived element.

This allows each individual representation to map the individual understanding of the game; a quantification operation represents the collective understanding of the game, offering information to the designer that can be used to determine whether the game reached its design intentions.

Although there are still features to be developed in order to properly deliver the system to end users, the test cases validated the data operations and manipulations, providing valid results according to the desired mathematical concepts.

The representation also presents the possibility of further enhancements, refining the abstract behavioral sets, subdividing them into more concrete and meaningful contexts in order to minimize ambiguities or misinterpretations. This is an ongoing project that can benefit from the contributions made by game studies researchers and further contribute to the game studies field.

ACKNOWLEDGMENT

The authors thank the company Hiperespaço Logicware Consultoria LTDA for providing the infrastructure to perform the tests and the services that allowed the creation of a communication channel for interested parties.
This study was funded in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) Finance Code 001.

REFERENCES

RODRIGO G. DOMINGUES is MSc. in Computer Science by the Universidade Federal de São Carlos in the field of Computer Graphics and a PhD. candidate in Electrical Engineering at Universidade Federal de Uberlândia, researching game design and game studies.

ALEXANDRE CARDOSO has a BSc. (1987) and MSc. (1991) in Electrical Engineering by Universidade Federal de Uberlândia and a PhD in Electrical Engineering (Digital Systems Engineering) by Escola Politécnica at Universidade de São Paulo (2002). He has experience in the fields of Electrical Engineering and Computer Engineering, with emphasis in Software Engineering and Computer Graphics, acting mainly in the following fields: Virtual and Augmented Reality, Education, Virtual Environments, Human Computer Interfaces and Information Visualization.

RAFAEL DUARTE OLIVEIRA VENANCIO. is a PhD. in Media and Audiovisual Processes by the University of São Paulo. He is a professor in the course of Journalism, a professor in the Post Grad Program in Technology, Communication and Education and the Director of the Faculty of Education of Universidade Federal de Uberlândia.