Knowledge-based Development of Games Using Design Patterns Ontology*

Barbara Giżycka[0000−0001−7540−3662], Krzysztof Kutt[0000−0001−5453−9763], and Grzegorz J. Nalepa[0000−0002−8182−4225]

AGH University of Science and Technology
{bgizycka,kkutt,gjn}@agh.edu.pl

Abstract. Tools for automatization of knowledge on game mechanics and their interrelationships are still lacking. Game design patterns, as proposed by Björk and Holopainen, seem promising in this area, as they can be represented formally as an ontology. This paper presents our proposal of such a representation, developed using OWL2. We discuss the design of the ontology, and demonstrate how it can be used to conceptualize the design of a classic video game. In the future, the ontology will provide a knowledge base for a new tool for game developers, in order to enable more complex, interesting and emergent game design.

Keywords: ontologies · knowledge-based automation · design process · game design · design patterns

1 Introduction

Modern game designers have a multitude of tools for game design and development at their disposal. Nevertheless, a proper and consistent way of capturing their knowledge on game mechanics has not been established yet. Nowadays, there is a persistent need for formally (or at least semi-formally) defined knowledge base for game design. A partial solution has already been suggested by S. Björk and J. Holopainen [2] in the form of *game design patterns* (GDPs).

A pattern is defined as a description of a part of the interaction that is possible in the game. The idea refers to the patterns in software engineering [4]. In game design, a pattern can be a description of a design solution, but also of game mechanics, game elements, or general game design approaches. Two important features of the game design patterns are that (1) they form hierarchies, and (2) they are connected within a complex web of interrelationships. Patterns organized in a hierarchy help to reflect various aspects of the game play, while different relationships mirror complex dependencies between game elements. The structure of patterns closely resembles a structure of an ontology, as understood from the viewpoint of knowledge engineering [7].

This paper provides a description of the GDPs' ontology, developed as a basis for a future ontology-based tool for game design and development. The GDPs are more thoroughly described in Sec. 2. The description of the ontology itself is provided in

* Supported by the AGH UST grant.

Sec. 3. To demonstrate how such a tool would work, a design of a classic video game is described in terms of the patterns in Sec. 4, and a sample use case is provided in Sec. 5. Related works are discussed in Sec. 6 and the paper is concluded in Sec. 7.

2 Game Design Patterns

The Game Design Patterns constitute a framework developed by [2]. Each pattern describes an aspect or element of game play. Each of the patterns is linked with at least several others by means of 3 forms of specified relationships.

In this framework, *Instantiating* means that one pattern is referring to an area of game play that is more general or abstract to the other: for example, a pattern ***Score***, which refers to the numerical representation of player's performance, often might instantiate ***Collecting*** – as it is an action of acquiring game elements that add points to the score. *Modulating* describes a reverse, in a sense, situation, where a "sub-pattern" fine-tunes the more abstract one – a pattern ***Privileged Movements*** modulates a pattern ***Movement***. *Potentially conflicting* patterns are ones that possibly form a contradiction. For example, a pattern ***Perfect Information***, where the player is provided with complete information about the game state, potentially conflicts with ***Randomness***.

There is a demand for a representation of knowledge more structured than a loose collection provided by [2]. Knowledge-based tools developed thanks to these representations could simplify the design process. In order to reflect the concept of game design patterns, it seems that ontologies provide a promising solution.

3 Knowledge representation for game design – ontology

The ontology[1] consists of a one global GDP class, 38 subclasses and 296 instances. The subclasses are all disjoint. Each subclass comprises of a group of patterns that are related to a certain area of game play. The patterns themselves are instances in the ontology. Each pattern possesses three string-type annotations: a label, a comment with a brief description of the pattern, and an example of a specific application of it (see Fig. 1). Fig. 2 provides a glimpse of a small fragment of the visualization of the ontology.

The relationships between the patterns are described as object properties. *Instantiates* and *Instantiated_by* as well as *Modulates* and *Modulated_by* are properties inverse one of another. These pairings have 3026 and 2864 uses in the ontology, respectively. The *Potentially_conflicting_with* relation is symmetrical. 1036 uses (potential conflicts between patterns) are specified in the ontology. For all of the object properties, the domain and range are provided by the global GDP class.

4 Pattern-based design conceptualization of a video game

The game design patterns were developed with two main purposes in mind: for game designing, but also for game analysis. Setting out from a single pattern, and following

[1] The full ontology is available on request at author's website, see: `https://www.affcai.eu/doku.php?id=sci:ontology`.

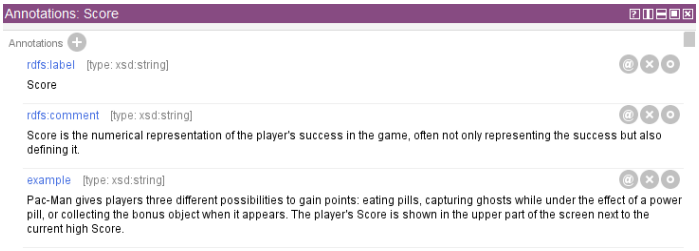

Fig. 1. An example of a pattern instance, as presented by Protégé interface.

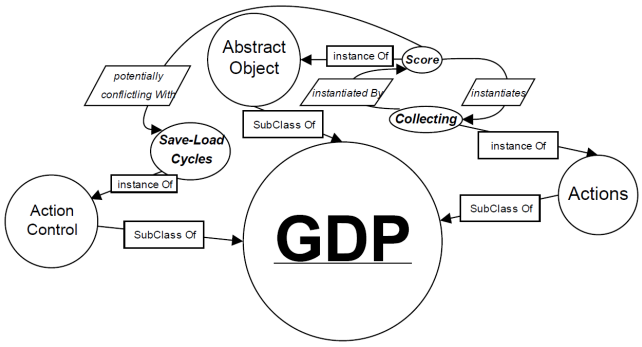

Fig. 2. A zoom on the fragment of the GDP ontology.

the suggested relationships, one is able to describe each aspect of the game play. This section provides an exemplary analysis of a classic arcade game, *Pac-Man* (see Fig. 3).

Fig. 3. A screenshot of *Pac-Man* remake by Neave Games from 2016.

Briefly speaking, *Pac-Man* is a game where the player navigates through a maze and collects pills. The goal of the game is to 'eat' all of the pills on the level. The player is impeded by four ghosts that chase the player's avatar. For collecting the pills and fruits, the player is rewarded with points that are added to their score.

In this basic description of the game, one can already identify numerous design patterns. Patterns ***Movement*** and ***Collecting*** reflect the core of the *Pac-Man*'s

game play. ***Score*** instantiates ***Collecting***, which is instantiated by, i.e., ***Pick-Ups*** that can be collected. On the other hand, ***Collecting*** modulates ***Character Development***. For a short duration, eating power pills in *Pac-Man* enables eating ghosts, which can be perceived as a temporary development of the character. ***Score*** in *Pac-Man* will stand in opposition to ***Save-Load Cycles*** – the player cannot create save points for any game states to load them later. For a graphical aid of *Pac-Man*'s pattern relations, refer to Fig. 2.

5 Practical application – example use cases

The Game Design Patterns Ontology was prepared using the OWL 2 Web Ontology Language, and as such can be further processed, e.g. with the use of SPARQL. This allows (semi-)automatic processing of the developed game project. To provide an intuition on how the design process can be improved by the proposed ontology, a sample query is presented in this section (see Listing 1.1). The use case relates to a situation when a game designer wants to check for potential conflicts in the prepared game design.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX gdp: <https://geist.re/gdp#>
ASK {
?subject  gdp:potentially_conflicting_with  ?object .
FILTER(?subject  IN(gdp:score, gdp:collecting, gdp:save-load_cycles) )
FILTER(?object  IN(gdp:score, gdp:collecting, gdp:save-load_cycles) )
}
```

Listing 1.1. Check if there are any conflicts in a given set.

The query in 1.1 will check if, in a given set of patterns (here ***Score*, *Collecting*, and *Save-Load_Cycles***), any two of them are in a *Potentially conflicting* relationship, either as a subject or an object (mind the symmetry of the relation). In this case, the query will return **TRUE** (see Sec. 4 for explanation).

6 Discussion and Related Works

Several similar works regarding ontologies were selected and discussed in the light of the one proposed in this paper. For starters, MOUDIL [3] is a framework for building patterns in Human-Computer Interaction. The authors provide a tool for pattern edition, as well as building a pattern ontology. Appropriate visualization of the ontology might be beneficial in terms of gaining an understanding by the ontology user [8]. In [5], it is suggested that ontology-based tools have to be subjected to utility assessment. A more systematic approach to ontology-based tools development, using user-centered scenarios is proposed. A need for consideration of guidelines developed in the area of Human-Computer Interaction, as well as for more automation of the design process [11] is expressed. In the area of video games, The Video Game Ontology is proposed [1]. However, compared to GDP ontology, it is focused on structural elements of games, game events and player behavior, and lacks insight into the mechanics and design solutions.

7 Conclusion and Future Plans

In this paper, a new ontology for game design patterns has been described. Following the explanation of the underlying concept of patterns in game design, an exemplary analysis of a classic arcade game has been provided. Finally, a detailed characterization of the ontology has been presented.

In order to bring the ontology to the developers, future works include creating an ontology-based tool. Our main focus are complex adaptive systems, with a special emphasis on engaging the user's emotion to be the motor of the adaptation. For a designer of such affective video games, an ontology extended with information on emotional change's caused by pattern implementation could help to develop an emotion model of either the player, or the game world itself [6]. As such, the designer would be aided in modeling the emotion that he intends to evoke in the player, by using specific patterns to, i.e., raise the player's arousal level [10]. We recognize possibility to use other ontologies [9,1]. Processing the ontology and improving the game design, as presented in the article, will therefore not only concern the mechanics itself, but also emotions, which will definitely form a new approach towards video game design.

References

1. The video game ontology. `http://vocab.linkeddata.es/vgo/`, accessed: 2019-06-20
2. Björk, S., Holopainen, J.: Patterns in Game Design. Charles River Media (2005)
3. Gaffar, A., Sinnig, D., Javahery, H., Seffah, A.: Moudil: A comprehensive framework for disseminating and sharing hci patterns. In: CHI03 Workshop on Perspectives on HCI patterns: Concepts and Tools (2003)
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns. Addison-Wesley Pub Co., 1st edn. (1995)
5. Giboin, A., Gandon, F., Corby, O., Dieng, R.: Assessment of ontology-based tools: a step towards systemizing the scenario approach. In: EON2002: Evaluation of Ontology-based Tools Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002. vol. 62. CEUR http://ceur-ws. org/Vol-62/ (2002)
6. Giżycka, B., Nalepa, G.J.: Emotion in models meets emotion in design: Building true affective games. In: IEEE Games, Entertainment, Media Conference, GEM 2018, Galway, Ireland, August 15-17, 2018. pp. 1–5 (2018). https://doi.org/10.1109/GEM.2018.8516439, `https://doi.org/10.1109/GEM.2018.8516439`
7. Guarino, N.: Formal ontology and information systems. In: Proceedings of the First International Conference on Formal Ontologies in Information Systems. pp. 3–15 (1998)
8. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou, E.: Ontology visualization methodsa survey. ACM Computing Surveys (CSUR) **39**(4), 10 (2007)
9. Matheus, C.J., Kokar, M.M., Baclawski, K.: A core ontology for situation awareness. In: Proceedings of the Sixth International Conference on Information Fusion. vol. 1, pp. 545–552 (2003)
10. Nalepa, G.J., Kutt, K., Giżycka, B., Jemioło, P., Bobek, S.: Analysis and use of the emotional context with wearable devices for games and intelligent assistants. Sensors **19**(11), 2509 (2019). https://doi.org/10.3390/s19112509, `https://doi.org/10.3390/s19112509`
11. Sheriyev, M., Atymtayeva, L.: Automation of hci engineering processes: System architecture and knowledge representation. Advanced Engineering Technology and Application **4**(2), 41–46 (2015)