Multimed Tools Appl
DOI 10.1007/s11042-016-3552-6

An ontology for videogame interoperability

Janne Parkkila[1] · Filip Radulovic[2] · Daniel Garijo[2] · María Poveda-Villalón[2] · Jouni Ikonen[1] · Jari Porras[1] · Asunción Gómez-Pérez[2]

Received: 10 December 2015 / Revised: 5 March 2016 / Accepted: 21 April 2016

Abstract During the last 20 years, video games have become very popular and widely adopted in our society. However, despite the growth on video game industry, there is a lack of interoperability that allow developers to interchange their information freely and to form stronger partnerships. In this paper we present the Video Game Ontology (VGO), a model for enabling interoperability among video games and enhancing data analysis of gameplay information. We describe the creation process of the ontology, the ontology conceptualization and its evaluation. In addition, we demonstrate the applicability of the Video Game Ontology in action with three example games that take advantage of the created ontology. Also, we demonstrate the use of the VGO in enabling interoperability among the example games.

Keywords Video game ontology · Video game interoperability · Ontology evaluation

1 Introduction

During the last decade, the video game industry has changed greatly due to growth of mobile games and digital distribution channels [12], such as Steam[1] and Google Play Store.[2] With introduction of casual games like Angry Birds[3] and Candy Crush Saga[4] the video game industry has gained a significant growth boost even further [6]. Playing video games has

[1] http://store.steampowered.com/

[2] https://play.google.com

[3] https://www.angrybirds.com/

[4] http://candycrushsaga.com/

✉ Janne Parkkila
janne.parkkila@lut.fi

[1] School of Business and Management, Innovation and Software, Lappeenranta University of Technology, Lappeenranta, Finland

[2] Ontology Engineering Group Madrid, Universidad Politecnica de Madrid, Madrid, Spain

Published online: 17 May 2016

become an accepted form of entertainment and video games are played among all ages and types of people [4]. It is estimated that the total video game market will grow to be over 100 billion US dollars by 2018 [8].

This growth has also lead to a fierce competition between game companies, leading into problems such as visibility of new games in digital marketplaces [11] and grown budget requirements for marketing. Similar problems are familiar in other business fields and marketing professionals have recognized co-branding [1, 17] and cross-promotion [5, 14] as possible solutions to address these issues. In co-branding, the goal is to form a partnership between two or more companies and to market products together. In cross- promotion, closely related products are marketed together, such as teleoperators offering their customers discounts for music streaming services. Taking into account the perspective of co-operation among the game developers, the possibility of connecting games together may enable new benefits for the game companies [2], such as improving player engagement and boosting player discovery of new games based on their previously played games. However, this requires technological solutions to enable interoperability among separate video game products. To this date, exchanging information between video games has been achieved in an ad-hoc manner among single products, but there are no general solutions for interoperability between video games.

Exchanging game-related information between two games may enable players to use common items that can be carried over from one game to another. In addition, completing an achievement in one game may unlock a special level or equipment in another game or even grant the player a completely new game. Such approach would enable game companies to exchange players between games, by creating incentives to visit other games through such “hyperlinks” of games. The games may even share characters or have decisions in one game that trigger events in another one. This hyperlinking of games would allow developers to create more immersive game experiences [2] and may help with the problem of game discovery and marketplace visibility.

In this paper we address the issue of interoperability between games. We do so by introducing Video Game Ontology (VGO) for allowing games to exchange information between each other. The ontology is available for anyone to use as open-source at http://purl.org/net/VideoGameOntology. The VGO enables mapping of characters, items or events between games. The benefit of the ontology is twofold. First, it enables capturing deeper knowledge of player behavior in video games than what is currently being gathered with more traditional approaches. Second, it allows representing relevant metadata and statistics from different games by using the same model, and also enabling more user-friendly and unified experience for the exchange of data between games. We demonstrate the use of Video Game Ontology through three example video games. These example video games, their source code and the ontology extension are all made available as open-source at www.gamecloudgames.com.

The purpose of the ontology is to capture the knowledge in the field of video games. The presented ontology brings semantic modeling of video game knowledge a step closer to reality, which can lead to better and more immersive player experiences. The ontology enables developers to create video games that are connected to each other, thus improving human-computer interaction and lowering development barriers for new paradigms in video games, such as transferring player items from one game to another. The focus of this research is the modelling of video game data and it does not take into consideration how the data transfer is done, giving the developer the freedom to choose how to handle data exchange. However, a common approach in the field of semantic web is to enable access to modeled ontology through standard HTTP protocol and web URIs, as has been done with the publication of the Video Game Ontology and the ontology extensions for our test games.

The rest of the paper is organized as follows: Section 2 discusses the related work on linking games together and existing work on ontologies in the field of video games. Section 3 continues to present the process of creating the Video Game Ontology, describing the main concepts of the ontology and evaluating it. Section 4 shows example use cases of Video Game Ontology with existing video games and discusses how they benefit from the ontology. Finally, Section 5 draws together a conclusion of the work done and discusses the results, presenting also some future research directions.

2 Related work

Connecting video game products together is not a completely new idea, even though it is not widely spread among game developers. A good large-scale example was developed by Crowd Control Productions (CCP),[5] who linked two of their games, Dust 514[6] and Eve Online,[7] with each other. The two games share the same universe and story, where the gameplay takes in place. The games exchange data that affects them respectively in different ways. Even though Dust 514 is a first person shooter and Eve Online a massive multi-player game set in space, the two games manage to exchange information in a meaningful manner. For example, the Eve Online space pilots can bomb the locations where Dust 514 players are fighting their wars. In return, the key points captured by Dust 514 players generate valuable minerals also for Eve Online players to use. Thus, both of the games are linked together through the exchange of information and achievements. However, this implementation is only done within the games of the same company, in an ad-hoc manner.

Transferring game content, such as virtual items, has been explored by Van Buskirk et al. [16]. In their research, Linden Lab (creator of Second Life[8]) and IBM demonstrated the interoperability between different virtual worlds by transferring avatars between Second Life and an OpenSim[9] virtual world. Also Van Buskirk et al. [16] designed a seamless connection between an e-commerce web store and a virtual world: In their system a user would be able to drag items from the e-store as 2D images and drop them into the virtual world as 3D objects. However, these efforts are more of an ad-hoc approaches to solving interoperability, than being a general solution.

The Game Ontology Project (GOP) [19] aims at creating a common vocabulary of game design concepts. The goal of the GOP is to provide a framework for exploring, dissecting and understanding relationships of different game elements. The ontology does not take into account game content, but focuses on capturing knowledge of game design and its rules and is directed towards creating a common vocabulary for research, rather than game interoperability. Zagal et al. [18] also researched the use of Game Ontology Project to support learning in game studies. The use of GOP helped students to better understand video game concepts. However, the project is targeted towards more abstract concepts (e.g. rules specifying that a

[5] http://www.ccpgames.com/

[6] http://dust514.com/

[7] https://www.eveonline.com/

[8] http://secondlife.com/

[9] http://opensimulator.org/

game has no ending or that a game has a camera-based worldview) than actual contents of a video game.

Another project related to ontology of games is the Digital Game Ontology [3]. It aims to combine a Music Ontology Project[10] with the Game Ontology Project to produce a comprehensive ontology for describing a game as a piece of media, its production and player activities. However, at the time of writing, the ontology is not available to the public and we have not been able to evaluate it further.

Analyzing gameplay data of video games is a common method for game developers to enhance their games. The goal of gameplay analysis is to create better experiences and to raise the profits by redesigning parts of the games according to the gameplay data. Using an ontology to improve analysis of gameplay data was presented in a form of a framework called GameGuts [7]. The framework used the Game Ontology Project vocabulary to model game information, but it is limited to only analysis of gameplay information and does not enhance interoperability of video games.

Even though there has been some research on using ontologies in video games, hyperlinking of games together is not a widely explored area. The existing solutions are more ad-hoc for certain situations and require significant effort from game developers to connect games of different companies together.

3 Video game ontology

Our main contribution is the creation of Video Game Ontology[11] (VGO), an ontology which aims to address the issue of interoperability among video games. The ontology has been developed by carrying out several activities defined in the NeOn methdology [13]. In this section we focus on reporting ontology requirements creation and how these requirements were specified. Afterwards, we present the ontology conceptualization and discuss its main concepts. Finally, we present the ontology evaluation in terms of terminology, competency question answering, and common pitfall detection, as is required by the ontology creation methodology.

3.1 Ontology creation and specification requirements

In order to create a proper ontology for capturing knowledge of video games, we started by creating a set of competency questions that could be answered with the use of the Video Game Ontology. The competency questions were based on brainstorming sessions as suggested by Uschold & Gruninger [15]. For the brainstorming session we invited 8 video game developers to imagine imagine new ways to connect games together and how such scenarios could be described. The brainstorming lead into 83 different example scenarios [9] that were further processed. Based on the game developer scenario descriptions, we formed questions that would cover a range of different scenarios on how games could be connected and what kind of information developers would like to know about their games. Even though the developer interviews and brainstorming did not bring any unexpected insights, the goal of the process was to remove researcher bias and base the ontology development on the requirements and needs of the developers, not on the views of the researchers.

[10] http://musicontology.com/

[11] http://purl.org/net/VideoGameOntology

Table 1 The most commonly occurring terms in competency questions

Term	# Occurrences	Ontology concept
player	46	vgo:Player
Game	40	vgo:Game
Item	18	vgo:Item
Achievement	8	vgo:Achievement
In-app purchase	8	vgo:GameOffering
Friend	5	vgo:isFriendWithPlayer
Genre	4	vgo:Genre
Level	4	vgo:PlayingArea

The purpose for these questions was to work as a requirement for the ontology creation. The ontology should be able to answer to all of the questions. The competency questions also served as a starting point for forming the main concepts of the Video Game Ontology. In total, we created 68 different competency questions.[12] The goal was to capture relevant knowledge about a certain game (e.g., which items are available or which achievement exists in the game), as well as about a gameplay (i.e., how players have acted in a game). In addition, we wanted to be able to gather statistical data and to create a general solution for sharing and querying game-related information.

We analyzed the competency questions to find out which are the most often appearing terms in the questions. As the model needs to answer to the competency questions, the most common terms were then used as the initial building blocks in forming the ontology. The listing of most commonly occurring terms in the competency questions can be seen in Table 1. The table also shows how the final ontology concepts map to the original competency question terms.

The competency questions were then divided into different categories based on their overall topic. Table 2 shows all the four competency question categories and example questions that describe a typical question in that category. The categories were named the following: 1) *Game Metadata* to describe general information about the games, e.g. who is the creator of this game. 2) *Gameplay Knowledge* describes how a game is being played with questions like what is the most played map in the game. 3) *Event Knowledge* contains questions that describe event-based knowledge, such as how many players use an item after gaining it. Finally, 4) *Business Knowledge* contains questions related to money, e.g. how many players make in-app purchases in a specific game.

3.2 Video game ontology conceptualization

After defining the requirements for the ontology, we created a conceptual model to capture the required knowledge and to answer to the competency questions. The conceptual model was created as a result of an iterative process, by assessing it against the competency questions. After several iterative rounds, the model was able to answer to all the requirements. The resulting Video Game Ontology of the modeling process is shown in Fig. 1 (with prefix vgo to refer to its main concepts). The figure gives an overview of the concepts and their relationships.

[12] Competency questions are available at: goo.gl/lxz8yU

Table 2 Categories of the created competency questions with a typical example question

Category	Example competency question
Game metadata	What games are similar to this game?
Gameplay knowledge	What is the most played map in the game?
Event knowledge	After gaining an item, how many players use it?
Business knowledge	How many players who have spend money in the game, spend money again?

The most important concepts in the ontology are related to the video game itself (vgo:Game) and to the player of the game (vgo:Player). To enable interoperability among video games, the ontology needs to model contents of the game, so that the related information can be exchanged between different games. All player characters (vgo:Character), items of the games (vgo:Item) and even the achievements (vgo:Achievement) are modeled to enable exchange of information about them.

In addition to game contents, the events (vgo:InstantaneousEvent) taking place in games play a major role in the Video Game Ontology. The events are used to track everything that has happened while playing the game, such as gaining achievements (vgo: Achievement) and consuming items (vgo:Item).

To also cater to the business needs of the game companies, we extended the ontology to contain concepts from widely adopted vocabularies, such foaf[13] and Good Relations.[14] The goal of these extensions was to relate data from other sources and to model data from other resources, such as defining who are friends of a player or to enhance business possibilities by linking purchasable products with Good Relations concepts. All the used extensions are shown in Table 3. Thus, the ontology provides also business possibilities on top of enabling interoperability.

In the core of the ontology are the game itself and the player. The vgo:Game describes the video game in question. Game is further linked with descriptive concepts, such as belonging to a specific genre, that it was created by a person or organization and what kind of leaderboards, playing areas (e.g. levels) and items it might have.

The other vital part of the video game ontology is the vgo:Player. In the scope of the ontology, a player refers to a person playing a video game. As one of the goals of the ontology is to capture the behavior of the player in a game, this concept plays an important role. All events taking place in a game happen either to the player (e.g. a player gains an achievement or makes an in-app purchase) or to the character the player is controlling (e.g. character dies, character gains a level).

3.2.1 Game metadata

In order to answer the competency questions, the ontology needs to capture the metadata about the game developers, player characters and players themselves, such as where are they living, who are they playing with and who is the creator of a game.

For multiplayer purposes, we want to record if the player has spent time online, or in other words and who are his friends. Digital gaming platforms, such as Steam[15] and Xbox Live,[16]

[13] http://www.foaf-project.org/

[14] http://purl.org/goodrelations/

[15] http://store.steampowered.com/

[16] http://www.xbox.com/en-US/live

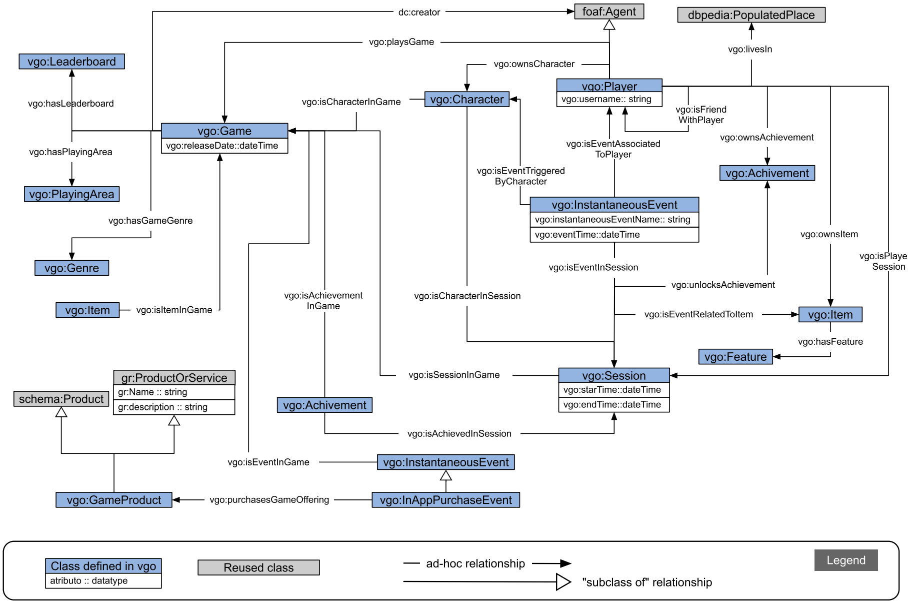


Fig. 1 Video game ontology overview

often provide possibility to befriend with a player in order to provide shortcuts to play together with them. For the purposes of modeling the friendship between players, we defined the vgo: isFriendWithPlayer relationship (subproperty of foaf:knows).

Answering to other questions about player locations, we used dbpedia:PopulatedPlace to describe the player location. This enables to tell more about the geographical location of players and to understand, for example, where the biggest player audiences are located.

Finally, to gather some knowledge of the actual developer of the game, we used dc:creator to define who (a person or an organization) is the creator of a certain game. This information can be used for example to suggest new games from the same creator to the player or for the developer to search for high level information of all their games (e.g. how many players from Germany in total are playing my games?).

Table 3 Prefix listing of references ontologies

Prefix	Name	URI
vgo	Video Game Ontology	http://purl.org/net/VideoGameOntology
gr	Good Relations	http://purl.org/goodrelations/
terms	Dublin Core Terms	http://dublincore.org/documents/dcmi-terms/
foaf	Friend of a Friend	http://www.foaf-project.org/
dbpedia	Dbpedia	http://wiki.dbpedia.org/
schema	Schema.org	https://schema.org/

3.2.2 Gameplay knowledge

A highly important part in understanding how games are played is to gather gameplay data of the players. In order to tie events of a game to a meaningful continuum, we decided to tie all player gameplay knowledge to vgo:Session. A session in the scope of the ontology means the time span between starting a game to the moment of player shutting down the game. For example, starting a game of Mario, playing two levels and then turning of the console would count as one session.

A vgo:Player participates in a session, where different events take place, items are used and achievements are gained. Each session has also a start and end time, allowing to see how long each session has lasted. The main advantage of collecting session data in the described way is to enable analysis of how players behave, what happens in sessions and to even profile different player types according to certain events taking place in a session.

3.2.3 Event knowledge

Another core concept is the vgo:InstantaneousEvent. This concept describes events that can happen, either to a player or to a game character. For example, this can be a player gaining an achievement, a character being killed by an enemy or making an in-app purchase in a game. The Video Game Ontology contains four subclasses, vgo:PlayerEvent, vgo:GameEvent, vgo:InAppPurchaseEvent and vgo:MenuEvent. The basic event hierarchy is shown in Fig. 2. These concepts are used to describe an event that has taken place during a gameplay in more detail. The subclass concepts also separate the events from one one another, enabling to tell whether the event in question was an in-app purchase, a game, a menu or a player event. In addition, we added two subclasses vgo:GainEvent and vgo:LoseEvent to deal with temporal issues, such as defining when a player has received and item and when the player has consumed or sold it.

3.2.4 Business knowledge

An important part of video games is the logic of how to earn money, e.g. the business logic. To model marketing activities, such as making in-app purchases inside a game or purchasing other

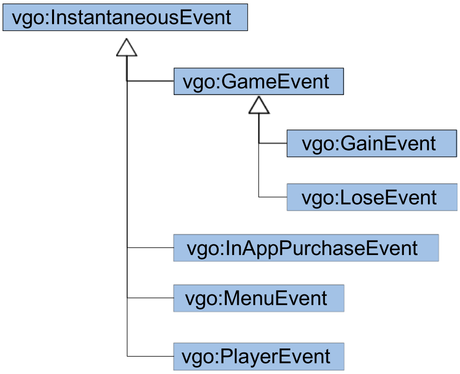
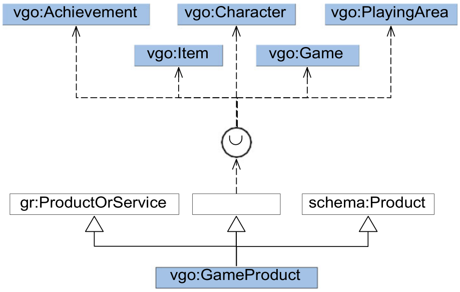


Fig. 2 Hierarchy of Events in the VGO

Fig. 3 Game products modelling



downloadable content for a video game, we extended some of the concepts with the Good Relations[17] and Schema.org[18] vocabularies.

As seen in Figs. 1 and 3, we chose to use the concepts of gr:ProductOrService and schema:Product to describe concepts that are available for purchase in video games. The decision to use both Good Relations and Schema was made to integrate digital products with Google and to give a wider support for modeling digital products.

In the Video Game Ontology, an item, a game, an achievement, a playing area or even a character can be bought. Every purchasable concept in the ontology can be modeled as a subclass of vgo:GameProduct, as is shown in Fig. 3.

3.3 Ontology evaluation overview

In this section we describe the evaluation of the Video Game Ontology by examining how the ontology concepts match with the terminology in the competency questions and whether the ontology answers to the previously specified competency questions or not. The completeness of the ontology is evaluated against the previously formed competence questions. Also, we briefly describe the results of the analysis of the Video Game Ontology with OOPS![19] (Ontology Pitfall Scanner!) [10] and Vapour[20] tools.

3.3.1 Terminology

The previously presented ontology concepts need to be validated against the amount of occurrences in the competency questions. Table 1 shows the amount a certain term appeared in the competency questions. The terms have been gathered by counting the words or phrases that match the same concept, for example by mapping weapon, consumable item and drinkable potion to be under the same term item. All the main concepts of the VGO appear most of the times in the list of terms, thus suggesting that the core concepts have been defined properly.

[17] purl.org/goodrelations/
[18] https://schema.org
[19] http://www.igi-global.com/article/oops-ontology-pitfall-scanner/116450
[20] http://linkeddata.uriburner.com:8000/vapour

3.3.2 Answering competency questions

Taking a look at the concepts presented in the Video Game Ontology, we need to reflect their answering power to the core competency questions. Our original listing of core competency questions has 68 different questions. We discuss some illustrative examples of the competency questions to ensure that they have been answered with the created ontology. The rest of the competency questions can be found online.[21]

For example, one of the competency questions was related to understanding how many players have made in-app purchases in a certain game. To answer this question, we can ask to count all the players that have in-app purchase events in a certain game. To take the use-case a step further, a developer could ask for all in-app purchases in all of their games and even sort the amount of player purchases by their geographical location. This question allows to give statistical information about the players of a certain game, which is quite important for the game developers to understand how their games are being played. To answer the question, we can ask for all in-app purchases of players in the game. Then, we can extend the query to ask where the players live, to count all the locations together and to see which place has most appearances. An example SPARQL query of this is shown in Listing 1.

Another competency question that is related to how players act in games is "What is the first action done by a particular player after an event?" (e.g., after they die in a boss battle or when they see an ingame- advertisement). For example, we can ask for the events that have happened after a player has scored a goal in a game of Pong. With the help of vgo:Session, we can see all the events that have happened in one gameplay session and then retrieve all events that have happened after the scoring goal event (for simplicity, we assume the URI of the event to be known). An example of performing this query is shown in Listing 2.

One important thing to notice in the use of the Video Game Ontology is the general applicability of the queries. When the games are modeled according to the same concepts, the same queries can be used with different games, just by changing the name of the game. For example, the query for asking in-app purchase events work in all games that have in-app purchases in them.

3.3.3 Evaluating the ontology with OOPS and vapour

In order to detect common mistakes done by developers when implementing ontologies we have used the online ontology evaluation system OOPS!.[22] Currently, the only pitfall detected is related to the lack of connectivity between the classes schema:Product and gr:ProductOrService with the rest of the ontology. However, the detected pitfall is not considered a problem in this particular case, as such classes are specified through the class vgo:GameProduct which is actually linked to the rest of the model. The other errors found by the tool were corrected accordingly.

Finally, to validate that the content negotiation mechanisms for publishing the ontology were correctly implemented, we used a tool called Vapour,[23] which provides

[21] Shortened link to competency questions: goo.gl/lxz8yU

[22] http://www.igi-global.com/article/oops-ontology-pitfall-scanner/116450

[23] http://linkeddata.uriburner.com:8000/vapour

the means to trace the responses of the server when requesting different content to the same URI.

4 Examples and use cases of video game ontology

In this section, we show three different usage examples of the Video Game Ontology (VGO) and how the ontology enables interoperability between them. In order to demonstrate the viability of the VGO, we tried to contact video game companies that were involved in our developer interviews. However, due to time and resource limitations of the companies involved, we had to limit the ontology evaluation to smaller, open-source video games. For demonstration purposes, we created three different video games that take advantage of the VGO in a different manner. The example games are modified from open source games in order to concentrate on the ontology implementation. The games use a previously created research platform Gamecloud[24] which can be used to automatically model the ontologies, requiring only a little developer effort. All of the games are HTML5 based and can be played straight from an Internet browser.[25]

The three example games are presented here in the order of the modeling complexity. The first game, Asteroids, tests the possibility to model game data. The second game, Coil, measures an amount of different player actions, thus testing gathering of session data and seeing how querying the dataset works. The third game, Browser Quest, takes full advantage of the ontology, modeling game items, achievements and events (except in-app purchases, as our goal is not to make profit with the test games) and even allowing events from the previous two games to have an effect in the game of Browser Quest.

Finally, in the end of this chapter, we demonstrate the use of VGO to enable interoperability between the example games. We link the modeled data of the example games to demonstrate how gameplay data can be used to connect multiple video games together.

4.1 Game 1 - Asteroids

The first test game is a classic game called Asteroids.[26] The goal of the game is to destroy and dodge asteroids floating around the player. When an asteroid is hit by a bullet, it divides into smaller asteroids that continue floating in the game, until they cannot be divided into smaller asteroids anymore. For the purposes of testing basic core parts of the game, we extended the Video Game Ontology to capture the game-specific knowledge.[27] The modeled concepts contain basic definition for the game, being an arcade game that has the events for asteroid being hit by a bullet, a ship being hit by an asteroid, player starting a new game and player hitting game over.

[24] http://www2.it.lut.fi/GRIP/about

[25] The games are also available for testing at http://www.gamecloudgames.com

[26] Asteroids game is available at http://www.gamecloudgames.com/HTML5-Asteroids

[27] The Asteroids ontology is available at http://w3id.org/games/spec/asteroids

Listing 1: Example SPARQL Query for asking where the players with most in-app purchases live

```
SELECT count(?event) AS ?purchase ?name ?location where {
 ?event a vgo:InAppPurchaseEvent.
 ?event vgo:isEventAssociatedToPlayer ?player.
 ?event vgo:isEventInGame <Target Game URI>
 ?player vgo:livesIn ?location.
 ?player foaf:name ?name.
}
```

Listing 2: Example SPARQL Query for asking what is the next player action after scoring a goal

```
SELECT ?laterEvents where {
?event a <http://www.example.com/PongScoreEvent>.
?event vgo:isEventAssociatedToPlayer <http://www.example.com/Player1>.
<particularEventURI> vgo:isEventInSession ?session.
<particularEventURI> vgo:eventTime ?evTime.
?session vgo:isSessionInGame <http://www.example.com/AtariPong/>.

?laterEvents a vgo:InstantaneousEvent.
?laterEvents vgo:isEventInSession ?session.
?laterEvents vgo:eventRime ?evTime2.

 Filter(?evTime2>?evTime).
}
```

As an example play-through of starting a new game, destroying one asteroid and then getting killed by an asteroid would be defined according to the ontology in the manner as shown in Listing 3 (the events are listed in the order of appearance).

These few examples allow us to model the game according to the ontology and to define the events that can happen in the game. All this collected information allows us to track important events happening in the game and to see for example, how many games have been started during a single day. The Asteroids example presents that the ontology is good enough to describe a simple arcade game that produces only limited amount of information. In addition, with only the simple ontology modeling, it is possible to ask questions that would not be easily possible previously. An example of such query would be to “show me the results of all the games a player has won”.

4.2 Game 2 - Coil

Coil[28] is another browser-based arcade game where the goal is to draw circles around blue balls (enemies), that appear to the screen, by moving finger on a touchscreen or by a mouse. Each ball is only visible for a short moment, before it turns into a yellow ball and eventually explodes. The player has only a limited amount of energy, which is decreased when a blue ball explodes before it was enclosed, eventually leading into a game over scenario. In addition, from time to time red balls (bombs) appear on the playing field and encircling them causes the player to lose energy. The longer the game progresses, the more balls appear to the playing field, until player cannot draw circles around the balls fast enough.

[28] The game is available at http://www.gamecloudgames.com/Coil

Listing 3: Example of short gameplay of Asteroids as modelled according to Video Game Ontology

```
ex:PlayerStartsNewGame123 a asteroids:Asteroids_Start_New_Game ;
    vgo:eventTime "2015-09-20T20:00:00"^xsd:DateTime ;
    vgo:isEventAssociatedTo ex:Player1 .

ex:AsteroidHitByBulletEvent567 a asteroids:AsteroidHitByBulletEvent ;
    vgo:eventTime "2015-09-20T20:00:50"^xsd:DateTime ;
    vgo:isEventAssociatedTo ex:Player1 .

ex:PlayerHitByAsteroidEvent444 a asteroids:PlayerHitByAsteroidEvent ;
    vgo:eventTime "2015-09-20T20:01:30"^xsd:DateTime ;
    vgo:isEventAssociatedTo ex:Player1 .
```

The purpose of modeling this game was to take a deeper look into the data gathering and analysis of the gameplay. Therefore, we modeled all the events in the game[29] (e.g. enemy ball explodes, bomb ball explodes, enemy ball enclosed, bomb ball enclosed). The game is rather quick paced and even a short game can produce dozens of ontology triples (instances of the ontology). Thus we also tested the tracking of player gameplay by linking each event to a certain vgo:Session and starting a new game caused a new session to be recorded.

The modeled data can be used to track how play sessions usually turn out. In our example, it would be possible to ask how many players have played in sessions where they have enclosed at least 100 blue balls or how many players have returned to play the game again after five sessions. In addition, we can use the ontology to see how many players have returned to play the game after 1, 5 or 10 days of first try. Such statistics are quite commonly used in analyzing games and these statistics can be gathered with the Video Game Ontology for later consumption.

4.3 Game 3 - Browser quest

The third example game, Browser Quest[30] is a multiplayer adventure game originally developed by Mozilla Foundation to demonstrate the possibilities of HTML5 technology. The goal of the game is to adventure around the world, exploring dungeons, fighting enemies and collecting better weapons and armors for the player character. The game allows multiple players to adventure in the same world simultaneously.

We modeled the game data using the Video Game Ontology to demonstrate the possibility to track player inventory and to demonstrate that the ontology can be used to model more complex games than the previously presented arcade games. We use the ontology to represent and capture player achievements, owned items, player actions and killed enemies[31] (using the prefix bquest). The actions of players can thus be carried over to other games.

To test how well player inventory can be followed, we implemented each item in the game and also the equivalent game events that describe losing and gaining the item in question. This way we were able to define the temporality of owning an item. For example, gaining and losing an axe is modelled as shown in Listing 4.

[29] The Coil ontology is available at http://w3id.org/games/spec/coil

[30] The game is available at http://www.gamecloudgames.com/BrowserQuest

[31] The Browser Quest ontology is available at http://w3id.org/games/spec/browserquest

Modeling the items through gaining and losing events, we can track the items possessed by a player and whether someone has gained or lost them. Checking if a player has a certain item, we can ask for the latest gain event and lose events and if the last event is a vgo:GainEvent, the player still has the item. In situations when player can have many consumable items (e.g. health potions), we can subtract the total amount of vgo:LoseEvents from the vgo:GainEvents. The result of the subtraction is the amount of items the player possesses. In addition to checking the items of a player, we can model how players use those items, for example to kill enemies, by linking a bquest:KillEvent with a weapon used by the player. This way we can follow how players use the items, which are most favored ones and so on. This allows developers to gain even more fine-grained knowledge of how players act in the game and whether changes to game mechanics change player behavior over time.

Listing 4: A VGO Example of modelling gaining and losing an axe in Browser Quest

```
ex:Axe1 a bquest:Axe .
ex:GainAxe1 a vgo:GameEvent ;
    vgo:isEventRelatedToItem ex:Axe1 .
ex:LoseAxe1 a vgo:GameEvent ;
        vgo:isEventRelatedToItem ex:Axe1 .
```

4.4 Interoperability of test games

The previously presented three case example games show that it is possible to model and query information about separate games and how they have been played. However, the goal of the ontology is to enable interoperability between separate games. Thus, we take a step further by connecting all the previously mentioned games together by taking advantage of the already modeled data.

One of the main advantages of modeling the video game content and player behavior data is to enable transfer of knowledge between separate games. Connecting video games together with the Video Game Ontology enables creation of player experiences that are are difficult to implement otherwise. Even though some game companies have created ad-hoc solutions for connecting their own games together (such as CCP Games with Eve Online and Dust 514), the ontology enables to connect separate video games without prior discussions or need to implement complex data exchange protocols.

For the purposes of demonstrating how games can be connected together with the Video Game Ontology, we implemented a blue armor (the "Armor of Arcade Awesomeness") that is only available to players that have gained at least five achievements in any arcade- type of games (i.e. in our first test games, Asteroids and Coil). Without semantic knowledge of other games, Browser Quest would need to know all possible games and their achievements to grant the bonus to a player. However, with the modeled data, the game can use the ontology to ask "How many achievements this player has gained in arcade-type games?". Even if new arcade games start to publish modeled gameplay data, no changes are needed for Browser Quest to take into account the achievements of the video games added afterwards.

In addition, to demonstrate the possibility to connect games together by "advertising" another game inside the currently played one, we changed behavior of an agent character in the starting town of Browser Quest. The replies of the character were modified to contain an advertisement link to Asteroids game. If the player clicks the link, she is awarded with an extra life in Asteroids (by using the ontology to tell that the player in question was involved in a "clicked asteroids link event”. Every time a player starts a new game of Asteroids, the game checks if the player has the event in question. It does not matter from where this event fired, as the only matter is to see that the player has the event. With the addition of only two queries, we were able to connect the three games together and to demonstrate that the ontology supports interoperability of separate game products. Data from a game modeled according to the Video Game Ontology can be used later to have an effect in another game, without needing to change anything in the first one.

Also, the Video Game Ontology enables querying similar information about all the games with just a single query. For example, the same query can be used to ask what achievements in the game a player has gained, only by changing the name of the game in question. An example of such query is shown in Listing 5.

Listing 5: A VGO Example of a single query that can be used to display all achievements that a player has gained in a certain game

```
SELECT ?achievement
{
        ?achievement vgo:isAchievementInGame <Game> .
        <Player> vgo:ownsAchievement ?achievement .
}
```

5 Conclusion and future work

Linking multiple games in a general manner is a challenging task. Some ad-hoc examples have been created to achieve interoperability between two games or via an intermediary service such as Xbox Live or Steam, but no general solution for interoperability exists. On one hand, the problem with ad-hoc solutions is that they do not scale to a large variety of games and require a lot of developer effort on negotiating the formats of data exchange. On the other hand, intermediary platforms control what can be done between the games and what data can be exchanged. This restricts the freedom of developers, as for example, extending existing standards is not possible without the agreement of the intermediary platform’s owner. The proposed Video Game Ontology (VGO) is a research thrust towards creating a generalized standard for video game interoperability. The VGO ontology enables developers to annotate video game information, to exchange the information between separate games and to gather statistical knowledge of players’ gameplay behavior.

We have demonstrated the usage of the ontology with three different video game examples. In addition, we have implemented an example of interoperability among the games, in which a player is rewarded with a special item in one game, if he gains five different achievements in any arcade-type video games. Furthermore, we have shown an example on how to integrate an advertisement of one games into the gameplay of another. When player interacts with the link, a reward is automatically granted to the player in the game where the link leads to. All this is enabled through the implemented Video Game Ontology.

The ontology may also be used to query and to profile the kind of games people are playing and how they are playing the games. The Video Game Ontology enables capturing of large amounts of different data and using the modeled information to make complex queries against it. Also, the ontology model enables creating single queries that can be used to retrieve similar information from different games, thus making it easier for developers to analyze gameplay data.

By publishing the Video Game Ontology as open- source and presenting the creation process and the example use cases, we invite both the research community and the industry to take part in validation of the ontology. Standards and best practices are often of- ten results of real-world usage and experience. For Human-Computer Interaction the exchange of video game content and knowledge of player behavior data in a semantically modeled form can open doors for totally new experiences. Carrying players' decisions are from one game to another has potential for totally new immersive gameplay.

The possibility of exchanging gameplay knowledge between games brings both new challenges and possibilities for the developers. An important future direction is to evaluate how connecting multiple games affects the design of video games. On one hand the influences coming outside from other games should not break the core mechanics of the game in question. On the other hand, connecting multiple games together could bring immersion of continuum between video games and serve as a motivating aspect for hard core players or as introductory tool to invite new players to try out different game products.

For future research, we aim to analyze larger datasets of commercial games, determining how the ontology can be applied to commercial products. The ontology may be used in a commercial product to analyze player-related information, for example, to understand what kind of games a certain player group plays or whether they are fans of already existing franchises. This knowledge may then be used to tailor more intelligent in-game advertisement for the players. Another research path is to study how players react to actions that are carried over from one game to another. This may enhance the player experience and deepen the immersion of the game. Also, the VGO may bring e-commerce benefits, as the digital products can be modeled with the ontology, for example enhancing search engine visibility of purchasable in-game products.

Acknowledgements This paper has been supported by the FPU grants (FPU2012/04084 and AP2010-1393) of the Spanish Ministry of Education, Culture and Sport. In addition, the research has been supported by TEKES, the Finnish Funding Agency for Innovation.

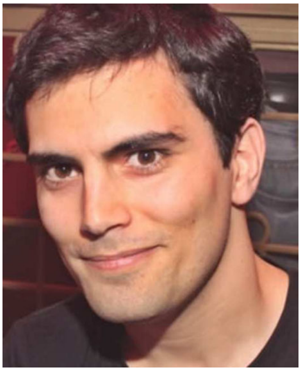
References

1. Blackett T, Boad RW (1999) Co-branding: the science of alliance. Palgrave Macmillan
2. Carter M, Gibbs M (2013) ESports in EVE Online: Skullduggery, fair play and acceptability in an unbounded competition. In: Proceedings of the 8th International Conference on the Foundations of Digital Games, pp 47–54
3. Chan JT, Yuen WY (2008) Digital game ontology: Semantic web approach on enhancing game studies. In: 9th International Conference on Computer-Aided Industrial Design and Conceptual Design
4. Entertainment Software Association (2013) Essential facts about the computer and video game industry: 2013 sales, demographic and usage data. Entertainment Software Association
5. Gulati R, Garino J (1999) Get the right mix of bricks & clicks. Harv Bus Rev 78(3):107–114
6. Juul J (2012) A casual revolution: reinventing video games and their players. The MIT Press
7. Lisboa MTCAG, Corruble RV, Santos ALM, Freitas F. Helping developers to look deeper inside game sessions
8. Newzoo. Global report: US and Chinae take half of $113bn games market in 2018, 2015. available at http://newzoo.com/insights/articles/us-and-china-take-half-of-113bn-games-market-in-2018/
9. Parkkila J, Hynninen T, Ikonen J, Porras J, Radulovic F (2015) Towards interoperability in video games. In: Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter, CHI- taly, pages 26–29, New York, NY, USA, 2015. ACM
10. Poveda-Villalón M, Gómez-Pérez A, Suárez-Figueroa MC (2014) Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems (IJSWIS) 10(2):7–34
11. Prata W, de Moraes A, Quaresma M (2012) User's demography and expectation regarding search, purchase and evaluation in mobile application store. Work 41(Suppl 1): 1124–1131
12. Sotamaa O, Karppi T (2010) Games as services-final report TRIM Research Reports 2, O. Sotamaa and T. Karppi

13. Suárez-Figueroa MC, Gómez-Pérez A, Fernández-López M (2015) The NeOn Methodology framework: A scenario-based methodology for ontology development. Appl Ontol, (Preprint), 1–39
14. Tang T, Newton GD, Wang X (2007) Does synergy work? an examination of cross-promotion effects. Int J Media Manage 9(4):127–134
15. Uschold M, Gruninger M (1996) Ontologies: principles, methods and applications. Knowl Eng Rev 11(02):93–136
16. Van Buskirk RE, Wright DL (2010) Virtual worlds seamless object drop integration. US Patent App. 12/756, 505
17. Washburn J, Till B, Priluck R (2000) Co-branding: brand equity and trial effects. J Consum Mark 17(7):591–604
18. Zagal J, Bruckman A (2008) The game ontology project: supporting learning while contributing authentically to game studies. In: Proceedings of the International Conference of the Learning Sciences ICLS 2008
19. Zagal JP, Bruckman A (2008) The game ontology project: Supporting learning while contributing authentically to game studies. In: Proceedings of the 8th international conference on International conference for the learning sciences-Volume 2, pages 499–506. International Society of the Learning Sciences

Janne Parkkila is a PhD student at Innovation and Software at Lappeenranta University of Technology. His focus is in the field of game interoperability and semantic data modeling.

Filip Radulovic is a PhD student in Artificial Intelligence within the Ontology Engineering Group at the Computer Science Faculty at Universidad Politécnica de Madrid (UPM), and an FPU scholarship holder. He is a graduate engineer in Information Systems from the University of Belgrade, and he holds a Master degree in Artificial Intelligence Research from the UPM. His research activities are mainly focused in quality modelling, decision support systems, semantic web and ontology engineering.



Daniel Garijo is a postdoc student in the Ontology Engineering Group at the Artificial Intelligence Department of the Computer Science Faculty of Universidad Politécnica de Madrid. His research activities focus on e-Science and the Semantic web, specifically on how to increase the understandability of scientific workflows using provenance, metadata, intermediate results and Linked Data.

María Poveda-Villalón María Poveda-Villalón is a Ph.D student at the Artificial Intelligence Department of the Computer Science Faculty of Universidad Politécnica de Madrid, in the Ontology Engineering Group. Her research activities focus on Ontological Engineering, Knowledge Representation and the Semantic Web. Previously she finished her studies as an engineer in Computer Science (2009) by Universidad Politécnica de Madrid, and then she moved to study the Artificial Intelligence Research Master finished in 2010 in the same university. She has collaborated during a four-month research stay in 2013 with Mondeca (París, France), during a three-month stay in 2012 with the Free University of Berlin and with the University of Liverpool in a three-month stay in 2011.

Jouni Ikonen is an associate professor at Innovation and Software at Lappeenranta University of Technology. His research focus is in Online collaboration, brokering data for value added services, sharing data between computer games, educational solutions and open access networks.

Jari Porras is a full professor at Innovation and Software at Lappeenranta University of Technology. His interests include wireless networks, ad-hoc networking, peer-to-peer computing, and aspects of Grid computing as well as distributed computing and distributed environments.

Asunción Gómez-Pérez is Full Professor at UPM (2007), Director of the Artificial Intelligence department (2008), Director of the Ontology Engineering Group (1995, 8th group in the UPM ranking), Director of the Master (2010) and Ph.D Program (2013) on Artificial Intelligence, Director of the Co-founder of the COM joint institute between Santander Bank and UPM (2012). PhD in Computer Science (1993) and Master on Business Administration (1992). Before joining UPM, she was visiting (1994-1995) as a postdoc the Knowledge Systems Laboratory at Stanford University. She has supervised 18 Ph.D thesis, she has coordinated 4 EU projects SEALS, SemSorGrid4Env and Ontogrid and now she is coordinating LIDER. She has participated 21 EU projects (FP5, FP6 and FP7) as main researchers, and in more than 40 national projects funded by Spanish research agencies and companies. Her main research interests are ontologies, semantic technologies, linked data and the semantic Web. She has published more than 150 papers and two books on Ontological Engineering. Her works on Ontological Engineering about Methontology and the NeON methodology are world-wide known. She has been co-director of the summer school on Ontological Engineering and the Semantic Web since 2003 up to 2011. She acts as reviewers in journals and conferences related with semantic technologies.