
VIDEO GAMES ONTOLOGY

IA301

Alexandre Dazat

`alexandre.dazat@telecom-sudparis.eu`

Julien Denize

`julien.denize@telecom-sudparis.eu`

A Basics

A.1 Ontology name

Video-games ontology, v1.0

A.2 Ontology owner

Julien DENIZE, Alexandre DAZAT

A.3 Ontology license

Creative Commons Attribution 3.0 (CC BY 3.0)

A.4 Ontology URL

<https://github.com/julindenize/Video-Games-Ontology/blob/master/ontology.owl>

A.5 Ontology repository

<https://github.com/julindenize/Video-Games-Ontology>

A.6 Methodological framework

No specific methodological framework has been used. Properties and classes selection were made to fit the context and requirements.

B. Motivation

B.1 Need

Single ontology resource that enables users to look up for video games in databases with fine-grained granularity regarding the platforms, people involved in development, famous players and genres.

B.2 Competition

The closest ontology to what we did can be found at the following URL :

<https://bartoc.org/en/node/18344>

Besides, it mainly aims at encapsulating knowledge on events that happen in video games and information about players, whereas ours is made in order to store general information, as well as in depth features including genre or production team of games in order to retrieve it from large market databases.

B.3 Target audience

The Video Game Ontology aim to be used by video game digital distribution service platforms such as Steam or Amazon to provide quality modeling in games representation so that consumers can provide the most accurate queries and get to the product they are looking for or discover interesting ones based on the various functional attributes available.

C. Scope, requirements, development community

C.1 Scope and coverage

The ontology focuses on representing relevant features for consumers and users to be able to retrieve video games they are looking for. This goal requires to provide at least some key features : Genre, platform, PEGI.

Indeed, one might also be searching for some games produced by their favourite video games company, or more specifically some well-known developer or sound designer.

Besides, it is also likely that the rise of streaming platforms and e-sport paves the way for new consuming habits of people looking for games based on which games their favourite professional players or streamers broadcasts.

In this context, we extended the scope of the ontology by adding the following classes : Company, Job, Human (subclasses in section [...])

The overall scope has been chosen in order to answer the type of questions one can find in section G.1 - Testing

C.2 Development community

Alexandre DAZAT and Julien DENIZE

C.3 Communication

<https://github.com/juliendenize/Video-Games-Ontology>

D. Knowledge acquisition

D.1 Knowledge acquisition methodology

Personal general knowledge acquired through 15+ years of gaming and structured into an ontology through the coursework given in IA301.

D.2 Source knowledge location

We collected data from Wikipedia and existing video games recommendation platforms (Steam, Amazon etc ..)

D.3 Content selection

In order to provide good quality representation the Video Games Ontology has to gather enough well-known games, production team leaders and games of diverse genre. The goal is to provide at least the means for retrieving the most commonly searched games, the more the database grows in individuals and the more consumers will find what they are looking for.

E. Ontology content

E.1 Knowledge Representation language

OWL version 2, EL profile.

E.2 Development environment

Protégé 5.5.0

E.3 Ontology metrics

Ontology metrics		Ontology metrics	
Metrics			
Axiom	826	FunctionalObjectProperty	3
Logical axiom count	496	InverseFunctionalObjectProperty	1
Declaration axioms count	187	TransitiveObjectProperty	0
Class count	61	SymmetricObjectProperty	0
Object property count	16	AsymmetricObjectProperty	0
Data property count	9	ReflexiveObjectProperty	0
Individual count	104	IrreflexiveObjectProperty	15
Annotation Property count	2	ObjectPropertyDomain	15
		ObjectPropertyRange	15
		SubPropertyChainOf	0
Class axioms		Data property axioms	
SubClassOf	65	SubDataPropertyOf	8
EquivalentClasses	13	EquivalentDataProperties	0
DisjointClasses	9	DisjointDataProperties	0
On count	0	FunctionalDataProperty	0
Hidden GCI Count	11	DataPropertyDomain	8
		DataPropertyRange	7
Object property axioms		Individual axioms	
SubObjectPropertyOf	15	ClassAssertion	104
EquivalentObjectProperties	0	ObjectPropertyAssertion	158
InverseObjectProperties	5	DataPropertyAssertion	55
DisjointObjectProperties	0		
		NegativeObjectPropertyAssertion	0
		NegativeDataPropertyAssertion	0
		SameIndividual	0
		DifferentIndividuals	0
Annotation axioms			
		AnnotationAssertion	143
		AnnotationPropertyDomain	0
		AnnotationPropertyRangeOf	0

E.4 Incorporation of other ontology's

The ontology was made from scratch with no other ontology incorporated.

E.6 Identifier generation policy

Spaces are replaced by underscores in individual names. Properties and classes are named according to CamelCase norm.

E.7 Entity metadata policy

Each class and property has a description.

E.8 Upper ontology

No upper ontology has been used, the ontology was made from scratch

E.9 Ontology relationships

We could have reused the Organization and product relationships from Schemas.org but the scope of the ontology made it hard to derive and adapt almost all properties for the use case. Instead we built our own model to fit the video games area.

E.10 Axiom patterns

No specific axiom pattern has been chosen for this ontology.

F. Managing Change

F.1 Sustainability plan

The ontology will not be actively maintained.

F.2 Entity deprecation strategy

There is no particular management for deprecated classes.

F.3 Versioning policy

The Video Games ontology versioning is managed through the github repository.

Object property	Description
isEmployedBy	An employee is employed by a company
hasPlayed	A player played a video game
hasDeveloped	An employ develops video games
hasEmployed	A company employ employees
hasFounded	A company has been founded by an employee
hasGender	A person has a gender
hasGenre	A video game can have multiple genres
hasJob	An employee has one or several jobs
hasPEGI	A video game has a PEGI
hasPlatform	A video game has a platform
hasProduced	A company has produced a video game
isDevelopedBy	A video game is developed by an Employee
isFoundedBy	A company is founded by an employee (inverse of hasFounded)
isPlayedBy	A video game is played by players (inverse of hasPlayed)
isProducedBy	A video game is produced by a company (inverse of hasProduced)

Object Property	Func	InvFunc	Trans	Asym	Refl	Irrefl	Domain	Range	Inverse
isEmployedBy	x					x	Employee	Company	hasEmployed
hasPlayed						x	Player	VideoGame	isPlayedBy
hasDeveloped						x	Employee	VideoGame	isDevelopedBy
hasEmployed		x				x	Company	Employee	isEmployedBy
hasFounded						x	Employee	Company	isFoundedBy
hasGender	x					x	Human	Gender	
hasGenre						x	VideoGame	Genre	
hasJob						x	Employee	Job	
hasPEGI	x					x	VideoGame	PEGI	
hasPlatform						x	VideoGame	Platform	
hasProduced						x	Company	VideoGame	isProducedBy
isDevelopedBy						x	VideoGame	Employee	hasDeveloped
isFoundedBy						x	Company	Employee	hasFounded
isPlayedBy						x	VideoGame	Player	hasPlayed
isProducedBy	x					x	VideoGame	Company	hasProduced

Data Property	Domain	Range
hasAge	Person	positiveInteger
hasEmployees	Company	positiveInteger
hasSurname	Person	string
hasName	Person	string
isFoundedYearIn	VideoGame	positiveInteger
hasReleasedYear	Thing	positiveInteger
hasGameReleasedYear	Human	positiveInteger
hasPlatformReleasedYear	Platform	positiveInteger

G. Quality Assurance

G.1 Testing

The ontology has to provide the means for querying games through fine-grained features including the production team involved, company owning, genre, platform, age restrictions. We tested three requests bellow before and after the reasoner in protege :

Query (class expression)

PCVideoGame and isProducedBy some LargeCompany and hasGenre some Multiplayer

Query results

Instances (6 of 6)

- ◆ Call_of_Duty
- ◆ Fornite
- ◆ League_of_Legends
- ◆ Minecraft
- ◆ Rocket_League
- ◆ Warcraft_III

Figure 1: query of multiplayer video games made by large companies

Query (class expression)

VideoGame and isPlayedBy some (Player and hasSurname value "Faker")

Query results

Instances (2 of 2)

- ◆ League_of_Legends
- ◆ Warcraft_III

Figure 2: query of video games played by professional player "Faker"

Query (class expression)

VideoGame and hasGenre some Multiplayer and (isDevelopedBy some (Employee and hasName value "Markus_Persson"))

Query results

Instances (2 of 2)

- ◆ Age_Of_Wonders_III
- ◆ Minecraft

Figure 3: query of multiplayer video games made by famous developer "Markus Persson"

G.2 Evaluation

The video games ontology has to provide the means for querying the games through the most relevant features at the time and given level of granularity it conveys we expect it to meet its initial goal.

G.3 Value of use

See queries in section testing G.1 Testing.

G.4 Institutional endorsement

No particular institutional endorsement.

G.5 Evidence of use

Same type of ontology may be used by digital video games retail contenders such as Steam or Amazon.

Appendix

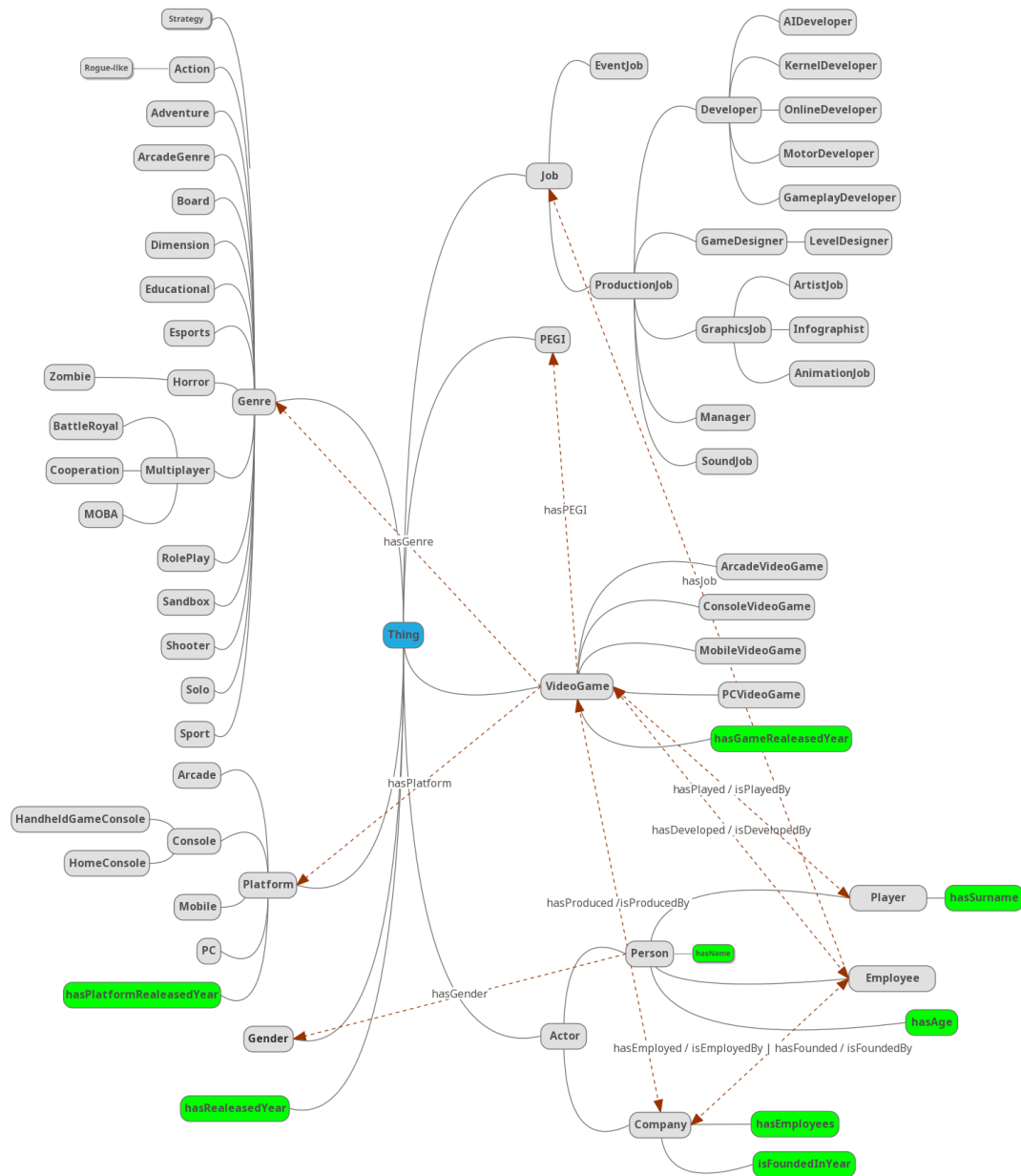


Figure 4: Ontology map