EXTENDING THE PERVASIVE GAME ONTOLOGY THROUGH A CASE STUDY

Trygve Pløhn\textsuperscript{1}, John Krogstie\textsuperscript{2} and Hong Guo\textsuperscript{2}
\textsuperscript{1} Faculty of Business, Social and Environmental Sciences, Nord-Trøndelag University College
\textsuperscript{2} Department of Computer and Information Science, Norwegian University of Science and Technology

trygve.plohn@hint.no, krogstie@idi.ntnu.no, guohong@idi.ntnu.no

Abstract: Pervasive gaming is a gaming genre where games are not confined in the virtual world, but extend into the real world and players’ everyday life. Pervasive games have proven to be effective within social relationship building, advertising and education. When designing Pervasive Games, one would have benefited greatly by being able to model main aspects of the game before it is realized, however, within the domain of Pervasive Gaming, a well-defined, suitable and generally accepted conceptual modelling language does not exist. One approach to over-come this problem is PerGO, a proposed ontology to structure and produce domain specific outputs within the domain of pervasive gaming. This paper aims to evaluate and extend the ontology within the field of pervasive gaming by mapping an existing pervasive game to the categories defined in PerGO. As a result we suggest two new perspectives, Pervasive Storytelling and Real Game World, to be included in the ontology.

1. INTRODUCTION

Pervasive gaming is a research area that is becoming increasingly popular. The terms pervasive game and pervasive gaming are widely used on a lot of different types of games, toys and experiences (Magerkurth et al. 2005). Currently, Pervasive gaming is a gaming genre where the game is not confined to the virtual domain of the computer, but the gaming experience is extended out into the real world - be it on city streets, in the remote wilderness, or a living room (Benford et al. 2005). The players must interact with the environment and with real objects to achieve certain goals (game objectives and missions). In contrast to traditional computer games, which take place in limited and well-defined settings, pervasive games erase the boundaries between spatial, temporal, and social expansion. Pervasive games are staged in reality and their main attractiveness is generated by using reality as a resource in the game (Waern et al. 2009).

Pervasive games are - within the scientific community - used as a framework to learn more about the use of technology, game concepts and educational games, just to name a few applications. Since this is a new research area, much research is also done just to understand the concept of pervasive gaming itself. Researchers have approached the genre from different perspectives which have resulted in multiple definitions of the term pervasive game (Nieuwdorp 2007). This paper follows a technologically independent definition:

* A pervasive game is a game that is pervasive relative to the player’s everyday life (Pløhn 2014).

A pervasive game is therefore required to be both spatially and temporally pervasive relative to the player’s everyday life for the whole duration of the game.

Conceptual modelling is - among other things – used to understand and analyse phenomena in a domain by providing a description of the phenomena at some level of abstraction expressed in a semi-formal or formal diagrammatical language (Krogstie 2012). Within most areas where Pervasive Games are being used, one could benefit greatly by being able to *model* the main aspects of the game before it
is realized, assuming that there is a well-defined, suitable and generally accepted conceptual modelling language available. However, when it comes to the domain of Pervasive Gaming, this is not yet the case. The reason for this is mainly because pervasive gaming is a new field of study, but also that it is not well defined. Extending the ontology and populating the domain of pervasive gaming may also help to define what makes a pervasive game. The research goal to be pursued in this paper is to develop a framework for the representation of important concepts in pervasive games.

The remainder of this paper is organized as follows. In Section 2, we will present pervasive games and modelling of (pervasive) games in more detail. There are many different requirements for what constitutes a good model, and in Section 3 we introduce PerGO, an ontology which can be used to structure the domain analysis work for pervasive game development with a model based approach (Guo et al. 2014). It is indicated what might be needed to be represented in a model of a pervasive game to enable high semantic quality (Krogstie 2012). Then we will further evaluate and extend PerGO by mapping an existing pervasive game to the categories defined in PerGO. The existing Pervasive Learning Game “Nuclear Mayhem” ((Pløhn 2014), (Pløhn et al. 2014)) act as a case study and is described in Section 4. The mapping of concepts in the case to PerGO is found in Section 5 where we also describe in more detail the case study research methodology. In Section 6 we conclude the paper, pointing to further work to be done in this area.

2. MODELLING OF PERVERSIVE GAMES

Models help us to understand a complex problem and its possible solutions. Models can act as documentation artefacts and can be used more directly for development of the running software (France and Rumpe 2007). In Model Driven Software Development (MDSD) models have become the primary products of software development rather than computer programs (Selic 2003). The motivation of MDSD is to move working focus from programming to solution modelling (Stahl et al. 2006). This is achieved by two important mechanisms: providing abstractions that are closer to the problem domain and generating programs from their corresponding models. Using concepts that are closer to the problem domain makes models less sensitive to the chosen technology of their solutions and to the evolutionary change brought by the technology (Selic 2003). MDSD should be domain specific (Kelly and Tolvanen 2008). This requires domain specific languages to raise the level of abstraction. That is why Domain Specific Modelling (DSM) using Domain Specific Languages (DSL) is regarded as an alternative terminology to MDSD in many settings.

As DSMs has been applied in many domains and successfully achieved expected benefits, researchers have tried to apply it in other domains such as the computer games domain (Albright et al. 2008), (Behrens 2010), (Andre et al. 2011), (Hernandez and Ortega 2010), (Moreno-Ger et al. 2007), (Walter and Masuch 2011). Computer games seem to be desirable for this approach due to its complexity and traditions. Computer games can be extremely complex due to complicated architecture and the variety of needed domain knowledge. Applying DSM can potentially alleviate the overall complexity by separating and hiding the domain complexity in a DSL. Further, it is quite common that game software utilizes a generic game engine which executes specific level descriptions that are produced in a level editor. DSM proposes a similar approach: utilizing generic code patterns which execute a specific data (model) that is produced in a DSL editor, but in a more structured way. DSM applies for games that do not have a fully functional game engine or level editors as well. In (Zhu et al. 2015), the authors present a conceptual framework for game architectures – Game Worlds Graph (GWG) – for classifying existing architectures, communication and sense-making of game architectures, and exploring and discovering future game architectures. In their work they identify 40 game architectures.

Modelling techniques are used for game design in general, and also for games in specific usage areas such as game-based learning (Tang and Hanneghan 2011). In particular, techniques within the area of MDSD have been applied and specialized for game design (Dormans 2012; Maier and Volk 2008), often by developing domain specific modelling languages (Furtado and Santos 2006; Maier and Volk 2008; Tang and Hanneghan 2011).
One of the 40 game architecture types relates to pervasive games. Among all the articles describing various efforts towards MDSD for games, it is difficult to find comprehensive descriptions about how their domain analysis was carried out, or where the abstractions in their DSLs came from. The work in (Furtado et al. 2010; Furtado and Santos 2006) may be the only that illustrates the detailed domain analysis process and the result in a structured way. PerGO is a recent comprehensive attempt to address this where core dimensions for the game development were considered and analysed. The ontology is at a domain modelling level (Maria-Cruz and Valiente 2010). By using the ontology to structure and accelerate the production of main domain analysis outputs (concepts, commonality and variability), it is expected that better DSLs can be produced in an efficient way.

3. PERVERSIVE GAME ONTOLOGY (PERGO)

PerGO is designed to support a structured and efficient domain analysis. PerGO consists of a domain vocabulary and a domain analysis procedure. The domain vocabulary defines a number of commonly used domain-specific concepts (as well as the relations among them) and will be our focus in this paper. Two benefits are expected when using PerGO: 1) The domain analysis is accelerated by using the pre-defined concepts and relations, directly or with minor modification; and 2) The domain analysis is regulated and structured by following the ordered steps.

In the domain vocabulary, more than 100 concepts are defined. These concepts are organized in six perspectives which focus on different aspects of game software such as Gameplay (challenges and actions), Artificial Intelligence (AI), Virtual Game World, World Elements, Control, and Presentation. There are two levels of concepts within PerGO: high-level concepts which are common to all computer games and low-level concepts which are specific to pervasive games.

![Figure 1 - Core parts of PerGO (Guo et al. 2014)](image)

There are numerous concepts within one domain. PerGO focuses on identifying those that are promising to work as DSL concepts (primarily for the support of automatic code generation instead of only conceptual modelling). Three criteria were used to select or make (low-level) concepts in PerGO: 1) the concepts should have an appropriate complexity. Simple attributes expressed by one variable of primitive type are thought to be of improper complexity and are excluded; 2) the concepts should be at a proper abstraction level. Those concepts that appear in the discussion of game design or level design are chosen, if it is possible to implement them as a class or some other encapsulated data structure. This is to achieve balance between the expressiveness and the flexibility of the DSL. High level concepts which are difficult to implement or low level concepts which do not appear in the common discussion.
are thought to be of an improper abstraction level; 3) the concepts should be constructive. In other words, the concepts describe some nouns or some rule.

A UML class diagram (Booch 2005) was used to formalize the domain vocabulary of PerGO. This is primarily because PerGO will be further used to construct DSL meta-models which are often based on UML class diagram too. By doing so, the transition between domain analysis and a DSL definition can be smooth. In addition, because UML is well known to many software developers, the threshold of using it can be lower. Comparatively, other ontology languages like OWL (McGuinness and Van Harmelen 2004) may not be so widely known to software developers. What is more, UML class diagram contains all the standard hierarchical relations (Krogstie 2012) (classification, aggregation, generalization and association) that we need to describe relationships among concepts. In Figure 1, an overview of the core part concepts and the relationships among them are shown in a UML class diagram based on (Hong et al. 2014). A brief introduction of these concepts is presented below:

- The Gameplay perspective includes all the concepts describing challenges and actions from players’ perspective. While [Challenge] describes what the players need to overcome, [Action] is about what the players are allowed to do to overcome the challenges (Rouse III 2010).
- The Artificial Intelligence (AI) perspective includes concepts which describe rules from the game elements’ perspective, regarding how game elements react to players, or how game elements evolve when no interaction happening. An abstract concept [AI] is defined in the core part standing for such kind of logic.
- While Gameplay and AI depict the dynamic characteristics of a game, GameWorldElement focuses on the static ones. The perspective contains all the concepts standing for game elements like [Character], and [Group].
- To make the game interactive, Control focuses on how various physical inputs control the game status, by invoking actions or changing variables. The concept [Controller] is defined in the core part referring to the logic which accepts physical inputs and controls part of the game world (which implements the interface [Controllable]).
- The Presentation perspective focuses on how the game utilizes various output devices to make the game status explicit, and influence the physical world. Similar to the control perspective, two concepts [Presenter] and [Presentable] are defined in the core part corresponding to the presentation logic and the objects that can be presented.
- The CtrlPresentation perspective is defined due to the complexity of control and presentation within the pervasive game domain. Some artefacts exist which control and/or present the game world in various ways. While Controller and Presenter are usually connected with one and only one I/O device, [CtrlPre] is more flexible and can be connected with several different devices (speaker, display, force feedback gamepad, or physical UI devices). CtrlPre can be connected to both Controllable and Presentable, but usually these two interfaces are implemented by one game element. This makes CtrlPre differ with a combination of several Controller and Presenter.

A four-step domain analysis procedure is proposed in PerGO based on the domain vocabulary. The procedure is used to produce three major outputs of domain analysis: a domain vocabulary, commonality space, and variability space. As indicated by (Mernik et al. 2005), DSL and some other DSM artefacts can be developed according to the information gathered in the domain analysis phase.

PerGO was chosen as a basis to be further developed through the case-study of the pervasive game Nuclear Mayhem on the basis that it is to our knowledge the only method available that is specifically designed to be applied on pervasive games.

4. THE CASE OF THE PERVERSIVE GAME NUCLEAR MAYHEM

Nuclear Mayhem (NM) is a pervasive game developed to support learning. The game is designed to support university studies in Multimedia and Web-game technology at a University College (Pløhn 2013). The gameplay in NM and the course syllabus is strongly related and to play the game well, the
players must learn and understand the syllabus of the course. The game starts when the course starts and ends when the course is completed hence both the game and the course last for nine weeks. In this nine week period, game related activity can happen anywhere at any time of the day for the whole duration of the game – thereby making the game pervasive. Nuclear Mayhem is a game that is designed for game based learning, but the learning aspects of NM is not in focus in this evaluation and case study of the game. How to map and model the learning aspects of pervasive games that is used in game based learning is extremely challenging and beyond the scope of this paper. NM was chosen as the game for our case study because of its uniqueness when it comes to the narrative aspect (Pløhn et al. 2014) of the game and how the game incorporated the real world in the game.

The narrative aspect of pervasive gaming is not a part of the current version of PerGO. Storytelling or narrative is commonly used in both digital entertainment and serious games in order to motivate players by first engaging and then keeping them engaged. Work by Lazzaro (Lazzaro 2004), Fullerton (Fullerton et al. 2004) and Yee (Yee 2005) all identified storytelling elements as key motivations for players in entertainment games. Storytelling, for similar reasons, is also regarded as a key element for the design of serious games (Pløhn et al. 2014).

In games that take place over a long period of time, i.e. days or weeks, an important success criterion is to provide features that 1) support in-game awareness and 2) increases the pervasiveness of the game (Pløhn et al. 2014). The main design feature in NM to support and enhance these two important properties is the game story itself. This was achieved by following the design criteria described in the Dynamic Pervasive Storytelling approach (Pløhn et al. 2014). This approach makes the game story a major feature of the game and a much more important part of the game than what is usual. In NM, this means that the game story is integrated in every part of the game and not just a presentation layer at the “top of the game” which is the more normal design approach to storytelling in games. The game story in NM is an integrated part of the game play, the challenges and the motivation for why the player has to perform certain tasks and actions during the game. The game story aims to become a part of the players’ everyday life to remind them that they are participating in an ongoing game.

Figure 2 - Using real life events to create the game story and the game plot

The game story in NM is illustrated in Figure 2 above. The game story is built up and supported by real life events that happened before the game started (the white triangles) and by real life events that happen during the game (the black triangles). All the events that are included in the game are motivated by the game story.

In the model, time passes from left to right and the game starts on the left side of the model indicated by the symbol Game start and ends when the player reaches the end of the game indicated with the symbol End of the game on the right side of the model. Each Quest/Riddle/Assignment arrow symbolizes a quest consisting of different puzzles to be solved and/or tasks that must be performed. All quest leads to an academic challenge (the white squares labelled 1, 2 .. n) that must be solved/conducted to reveal a code which will then be registered in the game client symbolized by $C_1$, $C_2$ .. $C_n$. The fact that there are more than one Quest/Riddle/Assignment arrow means that there are several different paths towards the goal, but all the players have to solve the same academic challenge.
no matter how they get to this point in the game (the academic challenge is related to the subject taught in the course).

The narrative strategy undertaken in NM is to create an overall game story (the main storyline) based on previous real life events that form the complete game story, and if nothing happens in the real world that can be related to the story or offer opportunities to explore different aspects of the game topic, this will be the complete game story. If a relevant real life event occurs, the main storyline can be put aside temporarily in order to allow for the development of an emerging story line (based on timely real-life events) until the story branches back again to the main overall storyline as shown in the Figure 3 below. The role of the Game Master is thus both opportunistic and creative in the sense that it is up to the Game Master to scrutinize media for relevant news reports and assess the opportunities these could provide with respect to enriching the pervasive game experience.

Figure 3 – The unfolding of the game story as relevant real life events occurs (Plöhn, et al, 2014, p.465).

To increase the pervasiveness the game penetrates the player’s everyday life in many different ways. The game makes use of social media (Facebook), the cityscape in the town of the University College, SMS, actors, Internet and real life events to provide game related clues and tasks that the players have to discover and accomplish. Major parts of the game were not implemented as software, but took place in the real world making use of real life objects such as interactions with real people (“actors” playing a role in the game which the players had to identify and interact with), real life events that occurred while the game was going on was used in the game (the game story, create challenges, motivate actions) and even real life events that was not originally a part of the game were perceived by some players as game related and by that became a part of the game.

Both the narrative and the real world aspect of NM have the potential to extend PerGO and we will therefore focus on these two aspects when we perform the mapping of NM to PerGO in Section 5.

5.  MAPPING THE PERVERSIVE GAME NUCLEAR MAYHEM TO PERGO

In this section, we examine the pervasive game example Nuclear Mayhem (NM), presented in the previous section, in the light of PerGO which earlier have been evaluated on a different case study (Guo et al. 2014). We use single-case design for the case study because of the uniqueness of NM. Since the narrative and real world design approach of NM are the boundaries of our case study, and we have not be able to identify any other pervasive game that uses the same narrative and real world approach as NM, a multi-case design study is not feasible. When analysing NM, we follow the four-step domain analysis proposed in PerGO (Guo et al. 2014).

A large part of the game and gameplay in NM was not implemented as software or mobile technology (as in many other types of Pervasive Games). The game took place in the real world where the players had to relate to the real world, understand and interpret the game story and locate and interact with
**actors** (people having an active role in the game). The game included events within the same time frame that may or may not be relevant for the game - such as news events featured in the evening news on TV. These types of real life events are an essential and important part of NM (Pløhn et al. 2014), but can never be implemented in the game as software.

The current version of PerGO focuses on identifying those concepts within the domain that than can support automatic code generation (software development), hence it may be argued that PerGO should not be used to model and represent those parts of a game that cannot be implemented as software. However, the non-technological part of games such as Nuclear Mayhem is as argued above more important for the game than the technical part of the game. Both the technical and non-technical part of these types of games needs to be planned and designed as one game. We will therefore argue that during the initial planning and design of this type of game it will be an advantage that every part of the game, both the technological and non-technical parts, are modelled in the same modelling language although logically separated in different sub-models. This is similar to what we find in e.g. enterprise modelling supporting sense-making and communication and manual deployment just as much as automatic deployment and system development (Krogstie 2012). It is therefore important that also the non-technical parts of pervasive games can be modelled using PerGO. This leads us to proposing two new perspectives to PerGO, the Pervasive Storytelling Perspective and the Real Game World Perspective (Table 1). The Real Game World perspective will have many of the same concepts as the Virtual Game World perspective such as Game, Players, and Group, since it is one and the same game in “both worlds”, with the same players organised in the same groups, however Real Game World consist of everything related to the game that happens in the real world that are not represented in the Virtual Game World.

Below we will describe some relevant parts of the game Nuclear Mayhem in more detail in order to populate the Domain Analysis Table. Due to limited space only a few game scenarios are presented here.

Scenario 1 (Real Game World, Pervasive Storytelling): The overall game scenario: The NM scenario features a plot line set in the context of the escalating international diplomatic conflict related to the Iranian nuclear programme. A timely current affair topic at the time of running the game was a growing international scepticism on the legitimacy of the Iranian nuclear programme (RT 2009). In this context, computing students play the role of cyber-saboteurs in the wake of the Stuxnet virus attack on Iranian nuclear facilities in 2010 (CNN 2010).

This scenario (building the main story line – the overall game scenario) involves media analysis (AnalyseMedia – see Table 1) to find and decide which events to use as building blocks to create the game story (what is the story about?). This involves adaptions of the chosen events so they are suitable to be used in the story (to support and enhance the pervasiveness and awareness properties of the game, the events should be chosen according to the criteria described in the Dynamic Pervasive Storytelling approach (Pløhn et al. 2014)) (AdaptRLEvent – see Table 1). This will be the story as experienced from the players view if no relevant real life events occur during the game allowing the main storyline to be put aside temporarily in order to allow the development of an arising story line based on those events (CreateREStory – see Table 1). This process has to be completed during the initial planning and design of the game (the main storyline must exist before the game can start, since the overall game story and game scenario is important tools for the Game Master).

Scenario 2 (Real Game World): One of the professors at the university played a role as a secret agent. The players had to, by secretly taking pictures of the faculty staff, identify which one of the professors was the secret agent and when they had a positive identification, they had to consult him in an undisturbed place in order to tell him a secret code (a strange sentence). If there were other people present, the professor would pretend that he did not understand anything of the “strange” sentence the player said to him, the same if the sentence was not 100% correct. Only when the professor was consulted in his office with no other people present except the player and himself, and the correct code word (sentence) was said, the professor would reveal the secret information the player need to know to be able to advance in the game.
This scenario involves the concept TakePictureCha from the Gameplay perspective already included in PerGO but it also involves concepts that are not yet a part of PerGO. The gameplay in this challenge do not make use of any type of technology at all (except for the part that is needed to identify the professor as the secret agent (HumanActor – see Table 2) - take picture of a person, submit the picture, analyse the picture, confirm identity from picture etc.). The professor is a real human person with a name, he has a life that he lives as normal in the duration of the game, there is a time frame where he is available for the players (he is not available when he is not at work, when he is lecturing etc.). There is a location where the player has to talk to him (his office), and the exchange of information (InteractWithHumanActor – see Table 2) is oral – the interface is “talking and listening”. Based on the actions of the player the professor will either pretend to know nothing (a real world “error message”) or reveal the information the player needs (a real world confirmation message).

Scenario 3 (Pervasive Storytelling): During the game an Iranian diplomat at the Iranian embassy in Brussels fled to Norway where he sought asylum (BBC 2010). This was a major news-event and was featured in Norwegian newspapers, online newspapers, and in newscasts on national TV and radio. As a part of the overall game scenario, Iranian agents were sent from Iran to find and eliminate the saboteurs and as the game unfolds, plot by plot, the Iranian agents are gradually getting closer to the players. The players are – in the duration of the game – informed about the Iranian agents whereabouts (they have left Iran, they are now somewhere in Europe, then have come to Scandinavia, they have just arrived Oslo). The news about the Iranian diplomat fleeing to Norway to seek asylum was in the game turned into a story about an Iranian diplomat that was a spy for the group that secretly was helping the players to sabotage the Iranian nuclear programme (AdaptRLEvent – see Table 1). This diplomat was the one that had provided the information about the Iranian agents’ whereabouts, but he had now been compromised (revealed as a spy) and had to flee to save his life. As a result of this one had lost track of the whereabouts of the Iranian agents, one suspects that they now had come all the way up to Steinkjer (the location of the players), and the players – because of this – had to perform a challenge to make sure they were not followed (a rebus challenge leading the players to different physical locations in the city of Steinkjer). This storyline and the challenge was revealed for the players at the same period of time as the news story about the Iranian diplomat fleeing to Norway seeking asylum was featured in all the major national news media in the players normal everyday life (blending and removing the border between the real life and the game).

This scenario involves monitoring the news for current events, analysing each event according to predefined demands, accept or reject the event as relevant to the game and if accepted the event must be adapted to the overall game scenario (AnalyseMedia, TypeOfRLEvent, MediaCredibility, AdaptRLEvent – see Table 1) and (if possible) a challenge based on the event must be designed and incorporated in the game play (CreateREStory – see Table 1). The adapted event (storyline) and challenge then must be made available for the players in a manner that is suited for the game.

The concept, commonality and variability details we identified when analysing Nuclear Mayhem according to the domain analysis procedure described in PerGO are listed in Table 1 and Table 2 bellow.

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Concept</th>
<th>Commonality Details</th>
<th>Variability Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pervasive Storytelling</td>
<td>AnalyseMedia</td>
<td>Identify real life events from media that has the potential of being adapted to the game story (both before and during the game)</td>
<td>RLEvent (Title, Pictures, Video, Text, Sound) RelevantRLEvent (yes/no) MediaType (newspaper, TV, Radio, Internet)</td>
</tr>
<tr>
<td></td>
<td>TypeOfRLEvent</td>
<td>What type of event is this?</td>
<td>CurrentNews PastNews Documentary FakeNews</td>
</tr>
<tr>
<td></td>
<td>MediaCredibility</td>
<td>Is the media publishing the event trustworthy?</td>
<td>levelOfCredibility (strong, normal, weak)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Relates to</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>AdaptRLEvent</td>
<td>Adapt the chosen RLEvent to the game story</td>
<td>Title, Pictures, Video, Text, Sound</td>
<td></td>
</tr>
<tr>
<td>CreateMainStory</td>
<td>Design the overall game story that is to be used if none relevant RLEvent happens during the game</td>
<td>Game, Player, Challenge, NewsEvent and RelevantNewsEvent</td>
<td></td>
</tr>
<tr>
<td>PublishMainStory</td>
<td>Involve the players in the story/make the player aware of the story as it unfolds (email, humanActor, SMS, video, sound)</td>
<td>Game, Player and Challenge</td>
<td></td>
</tr>
<tr>
<td>CreateREStory</td>
<td>GetNewsEvent (if relevant) AdaptNewsEvent (adapt the event to the main story)</td>
<td>AnalyseMedia, RelevantNewsEvent, EventdataType (text, pictures, video or combinations of these)</td>
<td></td>
</tr>
<tr>
<td>CreateREStoryCha</td>
<td>Create a challenge based on a REStory (scenario 3)</td>
<td>Game, Player</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2 - Domain Analysis Table – Real Game World - Nuclear Mayhem**

<table>
<thead>
<tr>
<th>Perspective</th>
<th>Concept</th>
<th>Commonality Details</th>
<th>Variability Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Game World</td>
<td>Game</td>
<td>Name, story, intro, timepast, timeleft, winner</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group</td>
<td>ID, gamepoint</td>
<td>Related to Player</td>
</tr>
<tr>
<td></td>
<td>PlayerProxy</td>
<td>Gamepoint</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HumanActor</td>
<td>Name, Location, RoleInGame, Availability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>InteractWithHuman-Actor</td>
<td>TalkTo, CallOnPhone, SendEmailTo, SendSMS, ListenTo</td>
<td>code word, secret, location, availability</td>
</tr>
<tr>
<td></td>
<td>IdentifyHumanActor</td>
<td>Identify if a person is a part of the game or not</td>
<td>name, identity, roleInGame</td>
</tr>
<tr>
<td></td>
<td>LocateHumanActor</td>
<td>Find and consult a given person that has a role in the game</td>
<td>name, appearance, location, availability</td>
</tr>
<tr>
<td></td>
<td>AvoidHumanActor</td>
<td>Avoid contact with a given person that has a role in the game</td>
<td>name, appearance, location, availability</td>
</tr>
<tr>
<td></td>
<td>RealLifeObject</td>
<td>Any real life object that is used one way or another in the game</td>
<td>Id of object, Type of object, Role of object, Location of object</td>
</tr>
<tr>
<td></td>
<td>LocateRLObject</td>
<td>Find a specific object used in the game such as a book or a tombstone in a cemetery.</td>
<td>ID of object, location (GPS data), availability</td>
</tr>
<tr>
<td></td>
<td>UseRLObject</td>
<td>Use the real life object to solve a task (challenge)</td>
<td>Description on the correct use of the given object</td>
</tr>
<tr>
<td></td>
<td>RLLocation</td>
<td>Location to an object that is used in the game (an object or a physical place)</td>
<td>ObjectID, GPS data</td>
</tr>
</tbody>
</table>
Figure 4 - Core parts of PerGo, extended version from (Guo et al, 2014, p.652)

Compared to the six perspectives in the original PerGo meta-model (Guo et al. 2014), two additional perspectives have been identified in the analysis of Nuclear Mayhem (Table 1), and Figure 4 shows the extended meta-model of PerGo with the RealGameWorld and PervasiveStoryTelling perspectives included (indicated in yellow). RealGameWorld, being related to Map and WorldElement, and PervasiveStoryTelling, where events might be related to setting up new challenges. Since PerGo originally only focuses on modelling the parts of the game that can be implemented as software, a perspective that extends the model beyond the software aspect of the game to include real world events, such as proposed in this paper, causes some apparent overlap of concepts between perspectives. For example is RealGameWorld. LocateHumanActor very similar to Challenge.FindPerson and RealGameWorld.Group is conceptual similar to VirGameWorld.Group. The reason for this is to be able to clearly differentiate between implementable and non-implementable (in a technical sense) aspects. A potential problem with this may be that the concepts are not placed properly in the model and should be moved to a higher level or a new type of perspective. The two games so far used in case studies to construct the model are very different types of games (even if both are labelled as pervasive games). The first game (Guo et al 2014) is more like an augmented reality game while Nuclear Mayhem is a relatively new type of game where many parts of the game are not controlled by software or gadgets, but the interaction between people and events that unfolds (predicted and unpredicted) in the real world during the game. Case studies of other types of pervasive games may be helpful in addressing the problems of overlap between concepts.

6. SUMMARY AND CONCLUSION

In this paper we have stated that the research area of pervasive gaming would benefit if there were tools available to construct good quality conceptual models to be used to analyse, understand or evaluate pervasive games and game concept without having to develop fully functional games. We have pointed out that this has yet be done since the not all areas within the domain of pervasive gaming is well understood and the construction of a good quality conceptual model requires in-depth knowledge of the domain. This task is made more difficult by the fact that researchers have approached the genre from different perspectives leading to multiple definitions of the term pervasive game (Nieuwendorp 2007). To achieve a better understanding of the domain, we have investigated the Pervasive Game Ontology (PerGO) as a candidate for a basis of an ontology for the area, to be able to
judge the domain appropriateness of a modelling approach in the field, by looking at its applicability in a case study using the pervasive learning game Nuclear Mayhem. The domain analysis helped us identify two new perspectives, *Real Game World* and *Pervasive Storytelling* that we suggest as an extension to PerGO. The domain analysis also helped us identify major design decisions in a structured and efficient way. We conclude that PerGO domain analysis procedure is a suitable approach for structuring the domain analysis process and to populate the pervasive game domain. However, we also revealed some overlap of concepts between the perspectives and this has to be addressed to develop and refine PerGO further in future versions. Even if both games are labelled as *pervasive games*, Nuclear Mayhem is clearly a very different type of game than the game used in the first case study of PerGO (Guo et al 2014) and these differences were useful in relation to identifying new perspectives and concepts. Given the wide definition of *pervasive game* there is a need to use a wide spectre of very different types of games in future case studies of PerGO to refine, restructure and consolidate the pervasive game domain further which is the main idea for further work on the approach.

REFERENCES


Behrens, H. 2010. MDSD for the iPhone: developing a domain-specific language and IDE tooling to produce real world applications for mobile devices Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion. ACM, Reno/Tahoe, Nevada, USA, 123-128.


CNN. 2010. Iran’s main nuclear powerplant Bushehr targeted by the Stuxnet computer worm. CNN.


Furtado, A.W.B., A.L.M. Santos. 2006. Using domain-specific modeling towards computer games development industrialization The 6th OOPSLA Workshop on Domain-Specific Modeling (DSM06), University of Jyväskylä, Finland.


RT. 2009. No trigger to pull? Iran rejects nuclear bomb component test RT, YouTube.


