

Learning analytics for location-based serious games

Victor M. Perez-Colado, Dan Cristian Rotaru, Manuel Freire, Ivan Martinez-Ortiz, Baltasar Fernandez-Manjon

Dept. of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid, Facultad de Informática
C/ Profesor Jose Garcia Santesmases, 9 28040 Madrid, Spain
{victormp, drotaru}@ucm.es {manuel.freire, imartinez, balta}@fdi.ucm.es

Abstract — Pervasive gaming has experimented a huge commercial growth with location-based game successes such as *Pokémon GO* or *Ingress*. The serious game industry has an opportunity to take advantage of location-based mechanics to better connect games with the real world, creating more authentic immersive learning environments. Games such as historical tours, story-based exploration, laboratories, or flora explorations can greatly benefit from location-based mechanics. Location-based games usually include an augmented map to provide game context, which combines both traditional game-dependent elements such as avatars, and location-based elements such as areas or points of interest. For players, interacting with location-based elements may involve entering or exiting specific areas; or reaching a certain location and looking in a specific direction. To include standards-based learning analytics for location-based serious games (SGs), we have added support for player movement and location-based interactions to the pre-existing *xAPI* serious game profile. We have validated this approach through a case-study example that guided players through different sports-related facilities within a large outdoor area. This work has been carried out and it is available as part of the analytics infrastructure used in EU H2020 RAGE and BEACONING serious game projects.

Keywords—serious games; location-based games; geolocation; learning analytics; *xAPI*

I. INTRODUCTION

Technology can allow learning to take place in more authentic learning environments, outside of books and classrooms, leveraging the capabilities of current smartphones and tablets. The appeal of location-based games is exemplified by *Pokémon Go* [1], a free-to-play smartphone game that combines fun and outdoor physical activity through exploration, that has achieved both commercial [2] and social success. This approach may even have benefits for previously identified issues such as the so-called “physical activity crisis” [3] and those users with severe social withdrawal [4]. Location-based games have also been proved to support effective learning [5]; and are often combined with augmented reality (AR), which has also shown a positive impact on learning [6].

Developing a standardized Learning Analytics (LA) infrastructure that relates in-game user activities with the learning objectives is the first step towards a cost-effective method of validating location-based SGs. To connect and exchange data between location-based games and such a LA infrastructure requires a standardized method for communicating player interaction data. This paper proposes and describes a method that comprises the specificities of

location-based games, as an extension of the *Experience API* (*xAPI*) standard to support location-based SGs. We have tested the proposed solution in a small case-study experiment, where players had to visit and discover the main outdoor sports facilities at the Complutense University of Madrid. This small experiment provided insights on how to use LA for discovering how players explored and learnt about the different sport facilities. We concluded that the proposed *xAPI* extension is a simple but effective, standards-based and extensible means for coping with the LA data exchange between location-based games and a pre-existing *xAPI* standard-based Learning Analytics infrastructure.

A. Pervasive gaming

The generalization of smartphones and tablets has promoted software that can increasingly adapt to the context of everyday life. Applications that adapt to your location, the time, or your environment are some of the possibilities. Further examples that include sport tracking, location visiting, activity registration, or



Fig. 1 Pervasive games *PokémonGO* (left) and *Ingress* (right). Both games share avatar (red call-outs) and located game-elements (yellow call-outs). Those elements are only interactive when the player is inside of their influence area. In addition, device orientation (blue call-outs) is always present in the game.

live weather. Furthermore, applications can use this context information not only to adapt their content, but also as a primary method of interaction. For example, in navigation-based apps, entertainment recommendations or videogames, the primary input is often contextual. Applications that adapt to context or use it as a primary input are known as pervasive or ubiquitous, as they may be used anytime and anywhere, and they adapt to their context of use.

The gaming industry has been greatly influenced by the opportunities of pervasiveness. Since the first location-aware smartphones, games involving the use of location information have captured the interest of both developers and players. Examples include “Zombies, Run!”, “Ingress” or “Pokémon GO” (Fig 1). Generalizing, these videogames use player location as their main source of interactivity. Based on player positioning, events are triggered when players approach certain locations, enter or leave areas, or even look in specific directions. For instance, the game “Zombies, Run!” provides instructions on how to escape (imaginary) zombies based only on location, and audio indications allow the game to be played even without looking at the screen. However, in most location-aware games an augmented map is used to represent the game situation. The map includes real world features among with virtual elements from the game. *Pokémon GO* is a clear example of the usage of an augmented map (Fig. 1), as the player will be placed in the center of the map, and the Pokémon (collectable pet monsters) will appear in the map when the avatar approaches their game-determined locations. To make this kind of interactions possible, location-based elements have an influence range that reacts to the player location; only being displayed once the player enters the area.

In addition, as mentioned earlier, the sources of context used to create pervasive applications can also include time or even friends’ activity or messages from social networks. This can allow the ubiquitous experience to be further expanded. Pervasive games usually involve social interactions, from simple cases, such as as trading information or valuable elements, to more organized, such as group based events where the players should combine their forces to beat difficult challenges.

Pervasive experiences are also highly related with augmented reality (AR), of which they are a super-set. AR offers a complementary way for understanding the environment and enhance it in real-time as the player focuses on elements of interest. Where AR normally depends just on real-time images from the real world, it can be extended to other pervasive sources as location and orientation, and in such cases the AR experience becomes more authentic as it can behave differently depending on players location. For example, a generic image target for information can be placed in relevant in-game locations, but depending on the player’s location, it will show different information about the surroundings.

Commercially, pervasive gaming has been constantly growing as smartphones evolved. The success of mentioned games such as *Ingress* or *Pokémon GO* have settled up a solid foundation for investment and researching. Furthermore, we expect a huge impulse in pervasive gaming growth, as Warner Bros. have recently announced partnership with *Pokémon GO*

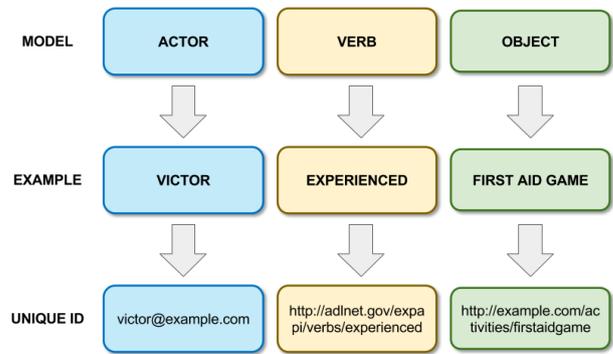


Fig. 2 Experience API data model specification with an example.

creator *Niantic* towards the creation of a pervasive game for the Harry Potter series [7].

For education, pervasive games offer a more authentic and situated learning, as the experiences occur in the real world and allows for learning to emerge from deduction and experimentation. The game can act as a guide to knowledge, and a verification and extender of the content learnt, including some of the benefits of gamification applied to education [8]. Also, it may require students to cooperate (as in real working environments) to complete the in-game challenges. For instance, in a game based on science environmental study, players could take different roles such as meteorologist, geologist or biologist. As the game progresses they would be required to cooperate with the other roles to mix their specialized knowledge towards a common solution.

1) Location-based mechanics

Pervasive games can leverage many different types of interactivity. In this paper we focus on the benefits of location-based games, leaving other approaches (time, social, etc.) as future work. This subsection explores the general mechanics for location-based games identifying low level concepts and atomic interactions.

In this genre of games, player use their position to interact. Because of this, the target elements are areas, and interactions are based on movement or orientation. As the work of T. Heintz [9] summarizes, the interactions happen against two types of elements: regions of interest (*RoI* or areas) and points of interest (*PoI*). The work explores the interactions based on position and orientation. The interaction happens when the player changes his position or orientation over time. Because of this, interactions referring to movement include entering or exiting, and staying inside or outside of an specific area. On the other hand, referring to orientation, the interactions happen when players, which are assumed to hold the device in front of them (so that the player can see what is in front of the device starts), start, stop, or keep looking at a particular location.

Finally, location-based games usually require reaching certain locations. For this purpose, navigation mechanics are usually included, such as reaching a set of points or even passing challenges at each point.

As mentioned, pervasive games can also mix social elements. Interactions with players can be treated as a special case of the previous described interactions (i.e. looking to a player or entering the influence area of a player). Other interactions such as forming groups or sharing game elements are general for any pervasive game, but it's a game-design matter if these mechanics will be location-limited.

Throughout this first section we have introduced the main concepts of pervasive methodologies and location-based gameplay mechanics and their application to SGs. In the next section we briefly describe *Experience API*, which can be extended to address the current lack of a standard for data collection in location-based SGs. Section III proposes such an extension, and in Section IV we analyze the results of applying the proposed solution in an outdoor experiment. Finally, we present a summary with conclusions and future work.

II. EXPERIENCE API

Experience API (xAPI), previously commonly referred to as *Tin Can API*, is a set of specifications for exchanging learning information, specifically to track learning experiences. It is continuously evolving, with the latest revision as of this writing being 1.0.3 (October 2017). *xAPI* is also commonly regarded as the evolution of *SCORM* even if the *xAPI* does not replace *SCORM*. *xAPI* focusses on user tracking in the education experience and not in the packaging or organization of the educational content itself [10] (actually *xAPI* will be used in many cases to track user interaction with *SCORM* content).

Within the *xAPI* specification, data is structured in statements composed of triplets of *noun*, *verb*, and *object*, as depicted in Fig 2. Actor, verb and object are the three main parts of a *statement*, which we will use as synonymous with *trace*, since statements trace the path of a player through a game. First, the *noun* represents an actor defined as anything with the capability to perform actions. Normally, the actor is the player; but any element involved in the application, such as an AI or other players, could potentially be used as an actor. Second, the *object* represents the target of the statement or trace, the element in which the action is being performed. For example, the object could be an in-game element, an area or scenario, or even real elements. Third and last, the *verb* represents the action that is performed. Depending on the *noun* and the *object* there is a set of verbs that can be used to track the different interactions between them. For instance, between an in-game item and the player, the verbs “grab”, “use”, or “inspect” would certainly make sense.

Optionally, an *xAPI* statement might include a *result* field, which represents the consequences of the corresponding action. The result field is used to convey scores, progress, or a set of extensions. Among the possible set of extensions, some are used to complement the information in the statement. For example, *result* fields can contain timestamps, changes to in-game variables, or secondary objects involved in the interaction.

In general, SGs have a set of specific needs regarding data to be tracked in order to gain useful insights of player behavior. Tracking the set of choices that players take while playing a serious game, or the level of completion of different parts of the game (or the game itself) is not easily achievable with the default *xAPI* specification. These serious-game specific requirements have been successfully fulfilled in the *xAPI* serious-games profile, proposed as an extension of *xAPI* and implemented by the Complutense University of Madrid for the H2020 RAGE project in collaboration with the Advanced Distributed Learning (ADL) initiative [11]. The next subsection explores this profile

A. *xAPI* serious games profile

The *xAPI* schema is especially relevant in SGs, where assessment has traditionally been performed by treating the game as a black-box, which once configured, simply outputs a set of grades. However, to play a serious game, players must perform a long set of interactions before reaching the end. In terms of evaluation, reducing this to a single score discards the potential wealth of information that can be gained by analyzing how the goal was reached. The additional information is also useful to assist the game life-cycle, and can guide improvements to the game layer. For instance, unexpected player behavior may be found, at the educational layer, as the player might not be acquiring the knowledge properly due to the way the information is presented. During the assessment process, the player (and the teacher) can benefit from warnings and alerts triggered by in-game actions. Such real-time analytics can help both understand and correct problems long before the game is over.

The serious-games *xAPI* profile (*xAPI SG*)¹ includes most, if not all, interactions that players can perform inside a serious game. For example, some actions include game-element interactions such as characters or items, scenario changes such as game level competitions, choices that the player may choose when alternatives such as questions are presented, game-variable changes relevant for analysis; and tasks that the player must fulfill in order to complete the game. However, in environments where the real-world context surrounding the player is relevant, such as in location-based games, *xAPI SG* offers no clear way to report environmental information such as location or orientation. Hence, for location-based SGs next subsection introduces the new location-based SG profile, covering new interactions and extensions including physical contextual information will be required.

III. XAPI LOCATION-BASED SERIOUS GAMES PROFILE

SGs that use location-based information require new elements in statements, currently absent from the pre-existing SG *xAPI* profile, in order to gather all the information and events produced during such gameplays. With these new elements it will be possible to analyze the player's interactions towards an understanding of the player behavior and even an assessment process. For instance, if a location-based serious game requires the player to navigate towards a specific location, but the player

¹ <http://www.adlnet.org/a-serious-games-profile-for-xapi>

never reaches this location and fails the game, having player positions included in traces will help both game developers and the teachers to identify and correct the cause, either by helping students or by changing the game. In addition, the more location-based context we introduce in each trace, the more potentially-valuable information will be available, via analytics, to improve how pervasive games are applied. Nevertheless, not only the contextual information is important, but also the actions that are derived from the location-based interaction with real world elements such as areas, buildings, parks or points of interest.

Location-based interactions are based on the player's physical actions performed towards these locations, including approaching or looking at them. This is especially relevant in location-based SGs, as one of the benefits of this games is the possibility to make the player deduce information from the real-world. The most cost-effective way to guarantee this is happening is to trace if the player has performed this interaction accordingly, such as looking at a monument, building or plant.

A. Extensions

Location-based extensions offer the possibility to include new contextual information, in particular the (i) location, (ii) orientation and (iii) navigation context of players, in any *xAPI* trace. These extensions are included in most of the cases when a location-based action is performed, and so will be available for use in later analysis to understand how players behave during their playthroughs. Furthermore, by including location-based extensions into the *xAPI SG* statements these traces can be promoted into a location-based analysis, covering the analysis of multiple mechanics in location-based games. For instance, using an in-game item, such as a sensor, is a SG statement, but in a location-based environment, it will result in different outputs depending on player's location. Hence by enriching the *xAPI SG* statement including the location extension makes possible to identify which information the player has obtained.

The *location* extension describes the global position of the player, expressed by latitude and longitude, in the "*GeoPoint*"² format. By including this extension in the traces, we can distinguish between the locations where interaction occurs (i.e. if the player opens the help menu, is it because he is lost?). This extension is especially relevant for location-based interactions as explained later in this paper.

The *orientation* extension holds the compass direction towards which the player's device is pointing, which corresponds to the player's heading in the common scenario where players look at the screen of their devices while holding them in front of their faces. It is expressed with a number in degrees that can have a decimal part (where 0° represents magnetic north, 90° points towards the east, and so on). The orientation extension adds a second layer of context which is especially important in pervasive games based on deduction, where players heading off in the wrong direction can easily explain errors in answering related questions. In addition, in

traces where the action implies looking, knowing the direction being looked at is especially relevant.

Finally, the third extension, the *guide* describes the navigational context being used to guide the player in the current moment. The guide extension is only used when a player is following navigational advice. The format is a list of directions and/or coordinates to follow. The guide extension is useful to detect when the player is misbehaving in terms of following the navigator's indications, or even perform late analysis debugging.

B. Actors and Objects

Describing location-based interactions requires defining new actors and objects for *xAPI* statements. The current proposal is the result of analyzing *GML* [12] and *GeoJSON* [13], and the location-related formats used in APIs of services such as *OpenStreetMap* [14] and *Google Maps* [15]. In these standards and APIs we have found several levels of area descriptions, such as terrain-like features, construction type, or even lower levels such as utilities. While any of location-based interactions could be used within traces, we will limit ourselves to a small subset of specific high-level elements: (i) buildings, for constructions; (ii) urban areas for any place that is not inside of a building but in a city; (iii) green zones for parks, forested areas, seaside areas and nature in general; and (iv) water for natural or artificial water-covered areas. This set of 4 high-level descriptors is enough for common location-based zone actions, and can be refined with additional optional descriptors when needed.

However, not all location-based actions are zone-based. In particular, a very important concept is that of "Point of Interest" (*PoI*), which can represent any important location relevant to the game, such as a monument, a preserved tree, or just the place where an interaction should take place. *PoIs* directly represent important locations that are therefore more relevant to analysis: looking at a *PoI* has an implied meaning for the system.

Since this is generic, sometimes the element is difficult to categorize or the categorization is not truly relevant for the specific application. In this case the *place* type can be used as a generic type to represent any location or area. This type is particularly important for cases where the players add locations to the game collaboratively, because it avoids forcing them to perform more fine-grained classification and therefore simplifies this type of collaboration.

While the previously discussed types provide a simple and compact coverage of location-based context, other types might be added in the future to provide increased descriptive accuracy regarding functionality, quality or shape. Social interactions during location-based gameplays will use the *xAPI SG player* type as both actor or target, and might include extensions with information about the user (i.e. the username).

Finally, in addition to the previous location types, location-based interactions can also include the usage of directions. The

direction type is intended for use in traces that indicate the intention of the player according to directions.

C. Verbs

In this section we describe verbs for location-based interactions based on the previously presented extensions and objects. In terms of interaction, new traces using these verbs allow actions such as moving, entering and exiting, looking, and following directions to be described in a standard way for pervasive games.

1) Moving

The basic behavior in a location-based environment is movement of actors. Whenever any actor changes its location a trace should be emitted indicating the new actor position, using the *moved* verb, specifying the coordinates by including the *location* extension. Since the movement is a reflexive action (what is moved is the actor), *where* it finalizes (when the trace is sent) is used as *target*. As mentioned in previous subsection, several kinds of categories could be used to identify this target, but we recommend using any of the previously-described area types. Optionally, the *orientation* extension could be added too, increasing the precision of the movement, and making it possible to make better predictions for real-time analysis.

2) Entering and exiting

When moving, there is the chance of crossing the limit of any of the relevant game locations. This limit can be either the boundaries of the region, or the influence area of a point of interest. For both cases, when the player crosses the limit there are two possible verbs that identify the player’s final position with respect to it: if an actor that was out of the limit and then goes inside has *entered*, while one that goes from inside to outside has *exited*. In both cases, the object of the trace is the element that is referred to, which should be of any place type. As for extensions, the location is enough to determine the point where the limits have been crossed.

In-game, entering and exiting limits is one of the main location-based mechanics as it can derivate into different game events. For instance, entering an enemy limit could start a battle with the enemy. In cases where crossing the limits changes the game-state, *xAPI SGs* extensions are used to indicate the consequences of this crossing, such as changes in variables [16].

There are some special considerations to append to enter/exit traces. First, places might overlap in the real world. When this is the case, actors can enter several places in a row without leaving any of them. This implies that the analysis cannot expect an order in entering/exiting unless specific conditions, regarding to real world place overlapping, are set in the analysis. The second consideration is that the player could enter or leave limits without explicitly crossing them. This might happen when opening the game or application after entering; or when the device location cannot be determined, due to weak signal, interference, or location being turned off momentarily within the device. While the trace could eventually be sent later (for example once location information is again available), trace consumers must be aware of the possibility of missing location-related events.

Finally, there is the case of games that can only be launched within particular areas, for example within a classroom when first enabled by instructors. In this particular scenario, while the entered trace would be implicit, exited traces would likely result highly relevant for the analytics system.

3) Looking

When playing a location-based game, orientation is very important in the interaction with the real world. This allow to know if players have points of interest (*PoIs*) in their line of sight (*LoS*). For this purpose, the *looked* verb is used to trace when an actor looks towards a relevant element (normally, towards its center to within a specified angular degree of tolerance). In this case, the *orientation* extension must be included to verify that the actor is indeed looking at the object, and, since the direction is subtle to the position, the *location* extension must also be present.

Additional care must be taken when dealing with *PoI* occlusion. For example, within a painting gallery, paintings can only be viewed from certain directions, as they will otherwise be occluded by intervening walls and the fact that the canvas is only painted on one side. To detect this situation, a combination of being inside the viewing area of the *PoI* and looking at the *PoI* itself is required.

4) Following directions

When playing a pervasive game, it is very frequent to require the player to go to specific locations. To reach the location, a set of steps might have to be followed to help the player reach the destination. This is represented with the verb *followed* using the type *direction* as the target of the trace. When navigation steps are being followed, the *guide* extension, described in the extensions section, is used to indicate the current programmed navigation towards the destination, helping to discover misbehaviors.

Action	Noun	Verb	Target	Extensions
Moving	Player	Move	Places (*)	Location
Entering place	Player	Enter	Places (*)	Location
Exiting place	Player	Exit	Places (*)	Location
Looking at place	Player	Look	Places (*), especially POIs	Location Orientation
Following navigation indications	Player	Follow	Direction	Location Guide

Table 1. Summary table of all the combinations for traces including verbs and extensions. In blue, traces related to location interactions, in yellow, traces related to navigation. (*) Places represents all the possible types described in B except direction.

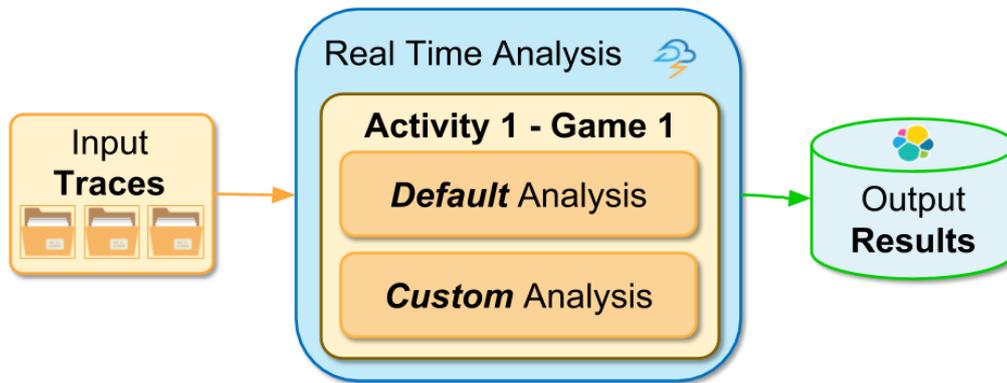


Fig 3. High level overview of the input traces going through the analysis system (default and custom) to produce output results.

D. Summary Table

Table 1 summarizes all the verb and trace combinations described in this section.

IV. ANALYSIS AND VALUE EXTRACTION

Displaying location based visualizations in real time requires to extract value from the data. Traces formatted following the *xAPI* location-based profile are sent to a backend server to be analyzed. There are two main parts interacting with each other in a *Learning Analytics (LA)* infrastructure: (1) client-side, where data is generated and (2) server-side, where data is collected, analyzed and stored. Initially, traces are generated in the mobile devices (client-side) on which the game is being played. During play, traces are sent over the network to a backend service that performs multiple actions on the traces (server-side). The different actions performed by the server-side are to collect the data, transform the data to facilitate later analysis, and then store the results waiting for the data to be retrieved and displayed. Furthermore, the results may be retrieved by an additional service and displayed in forms of an analytics dashboard composed of multiple visualizations.

A. Learning Analytics Infrastructure

LA infrastructures typically need to support types of users. We propose the use of, at least, the roles of developer, teacher and student. Each of these roles has specific requirements that increase the overall complexity of the system [17]. Furthermore, the insights that analytics can uncover are strongly limited by the existence of game-specific analyses and visualizations. We propose two different types of analysis, one generic, and therefore game-independent, and another customized to specific games (see Fig. 3.). In generic analyses, the data is collected and transformed in a game-independent manner. Due to the lack of game-specific input, the results of this analysis require an additional layer of interpretation by the user of the system (student, teacher, developer) when displayed. This is because automatic inference for each specific game is only possible when game-specific analysis models are provided to the system. Analysis models to create custom analyses are provided by the

teachers and game developers to allow assessment and to help the system to obtain deeper and more meaningful game-specific results.

Our system provides a default analysis that automatically obtains game-independent results from *xAPI*-formatted traces, without any additional game-specific input. Finally, as an extension of the infrastructure, custom analyses are supported via the possibility of (1) defining additional game-specific inputs in form of assessment by the teacher and (2) creating additional analysis that infer conclusions from these assessments.

B. Visualizations

One of the endpoints of the value extraction process is the creation of visualizations. For clarification, visualizations are graphical representations of the analyzed data to help the viewer on its interpretation. Thus, these visualizations can have very different formats, from two-dimensional graphics representing scores, to progress bars representing time spent per mission. This diversity of visualizations is a result of the generic purposes of *xAPI* specification. Hence, as a new *xAPI* profile is specified, new visualizations are possible representing this new source of information.

There are two approaches to creating visualizations. On the one hand, given the self-explanatory scheme of *xAPI*, it is possible to establish a series of graphs that represent (or compare) the natural meaning of the traces in a visually interpretable way. These graphs are known as *default* graphs, and can be based on one or more parameters of the analysis results (i.e. showing the player's positions over time). Normally, this type of graphs require an analysis by the interested party to extract a value from them, for example, limiting the cases of the input variables. However, once the educational model of a game is known, it is possible to create visualizations with the objective of representing metrics with educational value. These types of visualizations are known as *custom* visualizations and require the specialized and specific knowledge of the game learning model. For example, even if the player throws different traces of position, only those that occur during a treasure hunt mission are relevant to understanding whether the player has understood the hint that leads to the treasure. In general, we again advocate for

providing simple *defaults* while providing support for *custom* visualizations.

Given the previous description, next subsection explains the new possible location-based *default* and *custom* visualizations.

1) Location-Based visualizations

New visualizations are required to leverage the proposed *xAPI* location-based profile. In particular, data from the location extension from the proposed profile is most naturally displayed on a map. Map-based visualizations represent the location of the students or their behavior. Using the map as a base canvas with real-world information as background, the information can be represented using different colors, areas or graphics. Depending on the time dimension, different visualizations can be created.

When the time is not used, heat maps are recommended to represent the overall amount of activity in each location by changing the color of the active region to warmer colors. Warm colored zones will represent more activity, while colder ones represent sporadic activity. Using heat maps we provide two different *default* visualizations. The first *default* heat-map visualization represents the activity of a student in each location. In this implementation, the heat map visualization uses *OpenStreetMaps* for the back-end graphics and navigation controls, as shown in Fig. 4. The second *default* heat map visualization represents activity of location-based actions (enter, leave and look). For example, by filtering *enter* traces it will be possible to identify main entry points of an area and special of

locations. In this case, *default* visualizations represent the player gameplay route and can be filtered by time intervals.

Besides of map-based visualizations, visualizations of metrics derived from the additional location-based data sent to the server can also be created. For example, developers and instructors may wish to know when players arrive at a certain outdoor area or find a specific item in the map. Furthermore, using the additional verbs proposed in the *xAPI* location-based profile we can generate and display visualizations that describe the behavior of the students in considerable detail. For example, a game could ask its players to explore different type of points of interest within an outdoor area in a specific order. By carefully analyzing the *default* generic map visualizations and filtering by both player and time, teachers could then reconstruct each individual player's performance, annotating the order in which each point of interest was discovered.

In both visualization types, to infer game-specific conclusions about the activity (or metrics) and display it, it is necessary to create *custom* visualizations by providing additional input to the visualization. Together with a *custom* analysis, a *custom* visualization could easily extract this same information and display the results in a much more user-friendly, albeit game-specific visualization; which would however make no sense for games with entirely different goals and mechanics.

C. Assessment

The second endpoint of the analysis is assessment. Assessment complements the visualizations offering a qualitative point of view of students' learning, for example identifying when something is right or wrong. In SGs, assessment can be used as an effective tool to aid the teacher's conclusions about the students' learning [18]. This is also the case in a *Learning Analytics* infrastructure, especially if the assessment process can be automatically inferred by the system when a given set of game-specific inputs are provided by the teachers.

Game-specific information can be provided through multiple inputs to the *Learning Analytics* infrastructure. To begin, a rule-based engine is recommended to be running on top of the default infrastructure that allows teachers to automatically infer conclusions based on different specific parameters of the game. The game-specific parameters must be quantifiable to facilitate the use of the system. Examples include measuring the time spent to achieve a goal, time inside an outdoor area, or the time until a sequence of goals have been successfully achieved.

For the *xAPI* location-based profile itself it is possible to measure different *time* metrics such as the *time* spent in each area and the total completion time. Also, location-based *actions* can be assessed based on the targeted object. However, significant variables sent as game-specific input can be used to assess the students' learning. The variables must be explicitly included by the game developers as extensions to *xAPI* traces, and will generally be *string*, *numeric* or *Boolean* in nature. The assessment rules can then compare the value of the variables with their expected values to give further feedback to the teacher. Assessing on location-based actions is very relevant, as they have an extra meaning of positioning intention. For

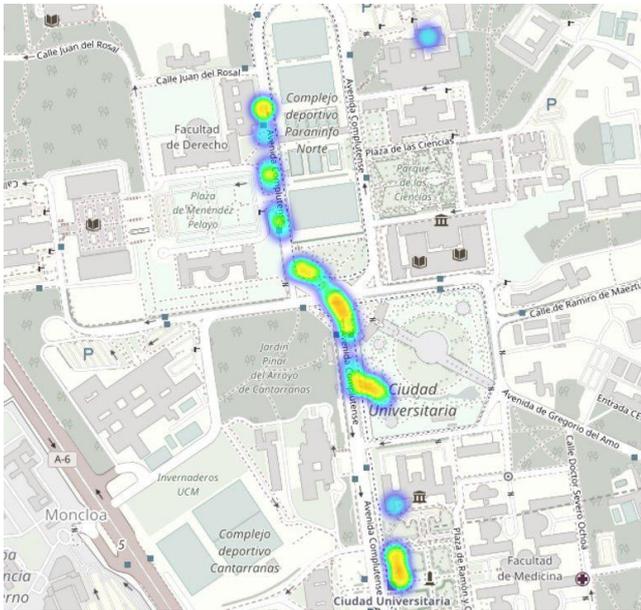


Fig 4. Heat-map of player locations from an experiment where a group of students had to explore the sports-related facilities of the Complutense University of Madrid.

When the time is used, the different locations can be interpreted as a movement with direction. Hence, route based maps can be created using lines connecting the different traced

instance, it is possible to detect when the player is exiting the game-area and notify the teacher accordingly.

Customized analysis algorithms that automatically produce high-level results can also be used to assess the students. The results generally require game-specific information, and will therefore be of little applicability to other games; but high-level results can often be displayed in simple customized visualizations automatically set up by the developer, such as bar-graphs. In more complex cases, developer must create both the customized analysis and the associated visualizations. This process does not require the intervention of the teacher and can result in higher-value insights because of the additional game-related data available to the customized algorithm that is not present in the default analysis.

Assessment is especially relevant for location-based data, which can combine locations with additional time-related information to assess the behavior over time of the students. This is also relevant to detect misconceptions and allow the use of targeted feedback to improve the learning experience [19]. For instance, if specific students spend too much time in an area where the game design did not expect such lingering, this can be easily uncovered using Heat maps in combination with behavior over time; and can allow teachers to reach out to the students and help them continue with the activity, among other possible feedback.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have described a data model designed for reporting location-related aspects of in-game player interactions in pervasive experiences to a learning analytics infrastructure. The underlying data model is proposed as an extension of a pre-existing *xAPI* (*Experience API*). Its use can cover from SGs to gamified applications, and the model can be further extended to match specific needs. We also describe the consequences and benefits of including the proposed profile in a Learning Analytics infrastructure for learning experiences, allowing their users to gain insights based on the location and behavior of students over time in outdoor areas. Additional layers of customization may provide higher-level insights as well as automatically extract conclusions based on students' learning. This is especially relevant for assessment, which is only possible once teacher-defined assessment rules have been specified.

The *xAPI* location-based profile is designed for pervasive experiences; and therefore, relies to an extent on constant internet connection where traces are sent and received correctly and in a timely manner. If the connection fails or is not present, traces should be stored locally in each device and sent to the server once it is successfully reestablished.

One of the limitations of the proposed solution, and learning analytics in general, is the fact that teachers are responsible of extracting meaning from the insights if no additional layer of customization is provided to link inputs to actual learning.

The proposed *xAPI* location-based profile has been tested in terms of functionality in a small experiment where students had to explore different sports-related facilities in an outdoor area at the Complutense University of Madrid. The experiment methodology included both pre- and post- surveys regarding

knowledge of the subject matter (sports facilities in an academic campus); and gathering of all game-generated traces as sent by player devices during the experiment. To gather these traces, the experiment used a modified version of the analytics infrastructure developed for the H2020 RAGE project; after being tested successfully, the modified infrastructure is available as open-source to any interested parties. The main goal of the experiment was to test the capacities of the standard in terms of functionality, while the game-specific assessment layer received considerably less attention. In this sense, future experiments to validate the assessment possibilities for this standard have yet to be performed. We did, however, obtain insights on the new visualizations and assessment opportunities discussed in this paper.

Experiments to further validate the functionality, usability and assessment capabilities in different contexts are currently being planned as part of the H2020 BEACONING. We wish to highlight an experiment that is, as of this writing, being developed by the *GeoMotion* team, designed to work in a web-based environment (JavaScript framework) that will be carried out in several European cities. Additional experiments are being planned by the e-UCM team to test the capabilities of the location-based model integration in the *uAdventure* platform. In this case, the implementation of the location-based tracing is built on top of the Unity environment (C# framework). We expect to obtain exhaustive feedback on the proposed solution, which will allow us to further iterate and adjust the data model, the default visualizations, and the customized assessment capabilities. Future papers will report on the educational value of the model as the assessment capabilities are validated.

We conclude that the proposed solution provides a simple location-based assessment model that can aid and support other methods, and can be used to complement, but not replace, traditional assessment methods. The potential for assessment is especially high if game-specific information is related with the educational design and used within appropriate custom analyses to gain insight and better understand student behavior.

ACKNOWLEDGMENT

This research has been partially financed by the Regional Government of Madrid [eMadrid S2013/ICE-2715], by the Ministry of Education [TIN2013-46149-C2-1-R, TIN2017-89238-R] and by the European Commission [RAGE H2020-ICT-2014-1-644187, BEACONING H2020ICT-2015-687676, Erasmus+ IMPRESS 2017-1-NL01-KA203-035259].

REFERENCES

- [1] A. de Souza e Silva, "Pokémon Go as an HRG: Mobility, sociability, and surveillance in hybrid spaces," *Mob. Media Commun.*, vol. 5, no. 1, pp. 20–23, 2017.
- [2] F. Mäyrä, "Pokémon GO: Entering the Ludic Society," *Mob. Media Commun.*, vol. 5, no. 1, pp. 47–50, 2017.
- [3] A. G. LeBlanc and J. P. Chaput, "Pokémon Go: A game changer for the physical inactivity crisis?," *Preventive Medicine*, vol. 101, pp. 235–237, 2017.
- [4] M. Tateno, N. Skokauskas, T. A. Kato, A. R. Teo, and A. P. S. Guerrero, "New game software (Pokémon Go) may help youth with severe social

- withdrawal, hikikomori,” *Psychiatry Research*, vol. 246, pp. 848–849, 2016.
- [5] C. M. Chen and Y. N. Tsai, “Interactive location-based game for supporting effective English learning,” in *Proceedings - 2009 International Conference on Environmental Science and Information Application Technology, ESAT 2009*, 2009, vol. 3, pp. 523–526.
- [6] M. Dunleavy and C. Dede, “Augmented reality teaching and learning,” ... *Res. Educ. Commun. ...*, pp. 1–34, 2014.
- [7] Pottermore.com, “New mobile game Harry Potter: Wizards Unite coming soon,” *Pottermore.com*, 2017.
- [8] K. Conley and C. Donaldson, “Gamification: The measurement of benefits,” in *Gamification in Education and Business*, 2015, pp. 673–688.
- [9] T. Heinz, “Location-based Game Design Pattern Exploration Through Agent-Based Simulation,” 2017.
- [10] K. C. Lim, “Using xAPI and Learning Analytics in Education,” *Elearning Forum Asia*, no. June 2015, pp. 13–15, 2016.
- [11] Á. Serrano-Laguna, I. Martínez-Ortiz, J. Haag, D. Regan, A. Johnson, and B. Fernández-Manjón, “Applying standards to systematize learning analytics in serious games,” *Comput. Stand. Interfaces*, vol. 50, pp. 116–123, 2017.
- [12] C. Portele, S. J. D. Cox, P. Daisey, R. Lake, and A. Whiteside, “OpenGIS® Geography Markup Language (GML) Encoding Standard,” *OGC Implement. Specif.*, vol. 07–036, p. viii + 426, 2007.
- [13] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub, “The GeoJSON Format,” Aug. 2016.
- [14] OpenStreetMaps, “OpenStreetMaps Places,” 2017. [Online]. Available: <http://wiki.openstreetmap.org/wiki/Places>.
- [15] Google, “Google Places API,” 2017. [Online]. Available: <https://developers.google.com/places/>.
- [16] I. J. Perez-colado, V. M. Perez-colado, M. Freire-moran, I. Martinez-ortiz, and B. Fernandez-manjon, “Integrating learning analytics into a game authoring tool.”
- [17] R. Ferguson, “Learning analytics: drivers, developments and challenges,” *Int. J. Technol. Enhanc. Learn.*, vol. 4, no. 5/6, p. 304, 2012.
- [18] F. Bellotti, B. Kapralos, K. Lee, P. Moreno-Ger, and R. Berta, “Assessment in and of Serious Games: An Overview,” *Adv. Human-Computer Interact.*, vol. 2013, pp. 1–11, 2013.
- [19] Á. Fidalgo-Blanco, M. L. Sein-Echaluce, F. J. García-Peñalvo, and M. Á. Conde, “Using Learning Analytics to improve teamwork assessment,” *Comput. Human Behav.*, vol. 47, pp. 149–156, 2015.