Computer Science and Information Systems 11(1):195–213 DOI: 10.2298/CSIS130214003G

Behaviour-based identification of student communities in Virtual Worlds

Antonio Gonzalez-Pardo[1], Angeles Rosa[2], and David Camacho[1]

[1] Departamento de Ingenieria Informatica.
Escuela Politecnica Superior.
Universidad Autonoma de Madrid.
C/Francisco Tomas y Valiente 11, 28049 Madrid, Spain.
{antonio.gonzalez,david.camacho}@uam.es
[2] Universidad de Málaga.
Málaga, Spain.
angeles.rosaalvarez@alu.uma.es

Abstract. Virtual Worlds (VW) have gained popularity in the last years in domains like training or education mainly due to their highly immersive and interactive 3D characteristics. In these platforms, the user (represented by an avatar) can move and interact in an artificial world with a high degree of freedom. They can talk, chat, build and design objects, program and compile their own developed programs, or move (flying, teleporting, walking or running) to different parts of the world. Although these environments provide an interesting working place for students and educators, VW platforms (such as OpenCobalt or OpenSim amongst others) rarely provide mechanisms to facilitate the automatic (or semi-automatic) behaviour analysis of users interactions. Using a VW platform called VirtUAM, the information extracted from different experiments are used to analyse and define students communities based on their behaviour. To define the individual student behaviour, different characteristics are extracted from the system, such as the avatar position (in form of GPS coordinates) and the set of actions (interactions) performed by students within the VW. Later this information is used to automatically detect behavioural patterns. This paper shows how this information can be used to group students in different communities based on their behaviour. Experimental results show how community identification can be successfully perform using K-Means algorithm and *Normalized Compression Distance*. Resulting communities contains users working in near places or with similar behaviours inside the virtual world.

Keywords: Community finding, Behavioural patterns,Clustering, Virtual Worlds.

1. Introduction

The Metaverse concept [53] can be described as a collective on-line shared space, created joining some virtually enhanced physical reality with a physically persistent virtual space [12]. It has been defined as a digital or electronic representation of the real world where people can interact freely using the metaphor of their real lives in a non-limited world by physics, age or other real world characteristics. In the last years, some new software applications have been developed such as Virtual Worlds (VWs) or augmented reality.

Virtual Worlds provide a 3D environment that can be used as a fictional virtual world where people with different interests and skills can interact, share or cooperate in a wide

range of activities [51]. Users can interact in the VW with other users or with the elements (objects) contained in the world through avatars that represent themselves. Characteristics such as the simple use, collaborative facilities or the attractiveness of the 3D features which provide a new and highly immersive sensation in the user, have made of VWs an interesting scenario to use it in different areas. In our case specifically, VWs provide a good environment to develop educational, training, and collaborative tasks [7, 8] using new techniques and tuning our previous results.

Using an innovative educational platform, named VirtUAM (from Virtual Worlds at Universidad Autónoma de Madrid) [8], some new techniques are integrated to find students communities based-on their behaviour. The goal of this work is to test whether students behaviour characteristics (extracted from the VW using the monitoring tools) can be used to classify students in different behavioural communities, extending the work presented in [4, 19].

This paper is structured as follows: Section 2 shows a brief description of the different virtual world platforms that have been applied to educational domains. Section 3 describes the software architecture of the platform used in this work, *VirtUAM*, and the description of the environment is shown in Section 4. Data extracted from avatar interactions, and how these data are analysed, are described in Section 5. Section 6 shows the obtained experimental results and finally, conclusions appear in Section 7.

2. Related Work

Education in virtual environments has gained popularity in the recent years. This is due to virtual environments (such as Social Networks or Virtual Worlds) are the most popular technologies amongt students. Although several researchers are focused on this area [17, 26], the main problem of SNs is that students consider them a ”social glue” [40, 27] rather than a formal teaching tool [50, 36]. Despite the fact that the findings on the educational potentials of popular SNSs are still limited, many researchers express caution about invading a social networking space that students feel clearly is theirs in order to utilize this space for teaching purposes [3, 40, 50, 57].

Virtual Worlds technologies have been traditionally applied in different domains from economy [51] to massively multiplayer online games [12, 28] which is nowadays the most popular application. VWs and videogames have become, also, very popular in learning processes [5, 18, 42] and this popularity has carried out many educational institutions (universities and high schools) to recognise the great potential of educational VWs [1, 15]. These environments have improved educative techniques [7, 8, 49] allowing teachers and students to use innovative learning strategies such as practical training, team work, discussions, field practices, simulations, and visualizations of concepts. One of the advantages of VWs is that these environments can reproduce situations that are not reproducible in the classroom or in the face-to-face teaching.

Currently, there are an important number of available VWs platforms that can be used to design and implement virtual spaces, some of the most popular are: Active Worlds[1],

[1] http://www.activeworlds.com/

Second Life[2] (SL), OpenCobalt[3] and OpenSim[4]. Second Life has been applied to different learning and educational processes [25, 35, 55, 59, 29]. For example, this platform has been used by psychology instructors as a meeting space with students to create labs, buildings and objects that can be used to learn psychology contents and skills [2]. Cunha [16] uses Second Life as an environment for collaborative learning and generation of new educational contents, and De Lucia et al. [37] uses Second Life to create a collaborative learning environment where objects support the synchronous role-based collaborative activities required by the jigsaw learning technique. In the specific context of teaching technical subjects, Second Life has been used with medical and health educators to explore its pedagogical potential [9, 30]. Another approach in [10] analyses how multiple remote participants can engage in 3D geometry within a virtual environment.

There are many researchers focused on virtual worlds which have applied this technology to education domains. Park et al. [44] compare instructor-led versus simulation-based environments for engineering students, and measure two variables: achievement and interest. They conclude that both environments produce similar results concerning these variables. Slator et al. [52] present a virtual world without teachers (named ProgrammingLand MOOseum). In this work, computer science students explore rooms populated with interactive objects that facilitate the learning experience. Nelson and Ketelhut [43] use an Individualized Guidance System (IGS) for students, in a virtual world (named River City) with no real teachers. The IGS prompts students with questions and hints and collects data about simple student activities such as clicking on pictures, or reading charts.

Even though all the research previously described, rarely present mechanisms to extract information from students behaviour, to improve the learning process. The most common approach is to extract the avatar information in order to generate charts that describe the use of the platform. In this work, the results of a statistical module developed in VirtUAM architecture, are presented. Although the data processing is performed off-line, the final goal of this research is to incorporate an on-line evaluation method that improve the students learning process in real time, i.e. while the students are working in the VW.

3. VirtUAM Platform Description

VirtUAM is the acronym from Virtual World at Universidad Autónoma de Madrid and it is the platform used to develop the statistical module described in this work. VirtUAM is built on top of the Open Simulator (OpenSim) framework developed by IBM. The reasons to select OpenSim framework can be briefly summarized as follows:

- OpenSim is an open source-software, which provides to administrators the possibility to modify and adapt the platform to their needs.
- The platform can be accessed only by registered users. Usually, VW platforms such as ActiveWorlds or SL, are public virtual places but OpenSim is not a public virtual place where administrators can configure the VW to restrict the access to limited

[2] http://secondlife.com/
[3] http://www.opencobalt.org/
[4] http://www.opensimulator.org/

users. This means that other external non-authorized users would not be able to visit the VW-learning environment and possibly interfering with the tasks developed in the VW.

– The virtual space that can be built by users, as well as, the number of objects created in this platform, are both unlimited. Users can build their own virtual space with an almost unlimited number of objects to interact with them. Although, there are some problems related to the performance of the system when the number of objects and programs running in the VW grows faster. It can be managed using the grid-based configuration of the platform.

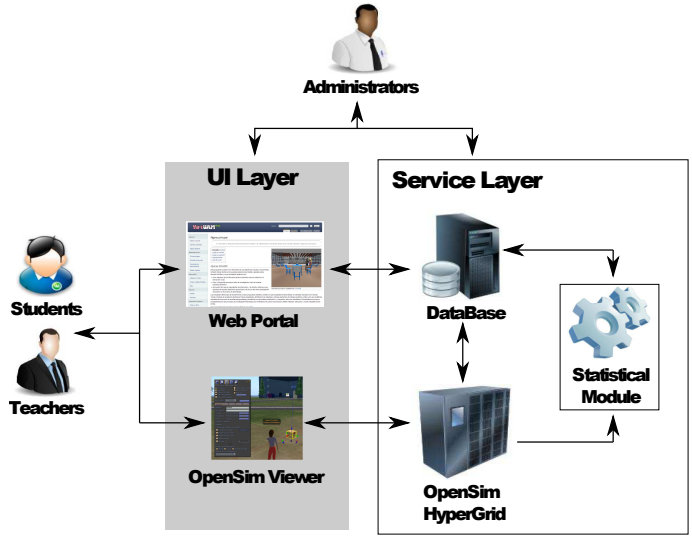


Fig. 1. Software architecture of the VirtUAM platform.

The software architecture of the VirtUAM platform is shown in Figure 1 and it is composed by the following modules:

1. A grid of computers hosting the Virtual spaces (islands, buildings or any other virtual spaces that could be requested from users) that allows the execution and management of all the VWs generated. In these virtual spaces, lectures, laboratory activities, or working activities, can be placed, and the educational objects created by teachers and students are stored. If the number of objects within the VW increases, then several computers are needed in order to avoid performance problems.
2. A Web portal to provide users access to public information and data about the courses (technical guides, construction and programming tutorials, etc). There are three different types of user roles in this web portal: administrator, teachers and students. Teacher and administrator users can use the web portal for administration tasks. For instance, teachers can create courses, include new students into their courses,

and obtain statistics that describe the student behaviour and course performance. Furthermore, administrator users can manage teacher accounts, analyse the system performance and get access to the logs stored in the database system. On the other hand, students can change their own information and they can access to the documentation related to courses.

3. A back-end service built over different Databases (DDBB) servers which contain all the required data: technical and user guides, user information (groups, teachers and students profiles), data mining information (such as logs, chat conversations, tracking movements, documents and objects developed by users, or student and educator interactions in the VW). This large amount of information can be used by both, teachers and administrators, to perform data analysis of student behaviour.
4. A VW client, called *viewer*. This program is needed to access to the VW. Some of the most popular viewers are *RealXtent*, *HippoViewer* or *Imprudence*.
5. The designed statistical module that receives the data from avatar interactions in the VW. Once these data are adequately mapped, they are processed using data mining techniques to retrieve behavioural patterns from them.

4. Virtual World Environment

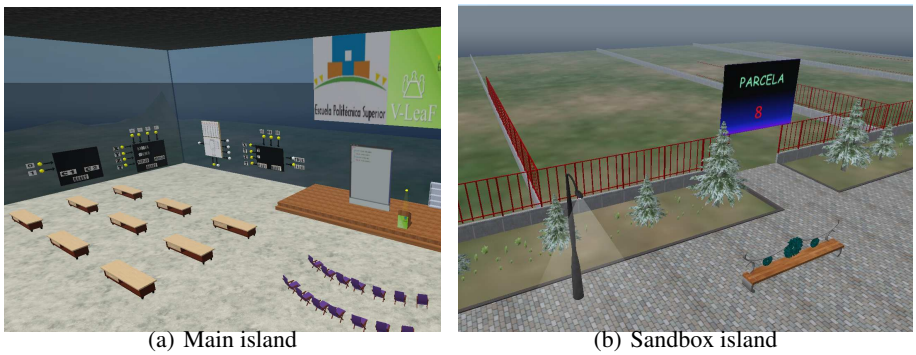
Once the VW platform has been described, it is needed to understand the environment where the experiments take place. This section provides a snapshot of the initial state of the virtual world, i.e. the different objects created in the VW and the different tasks that students are asked to do.

In this work, VW is composed by only two islands, i.e. two VW regions. The first one (called *Main island*) is used as an example island containing different objects (see Figure 2(a)). Any object can be analysed taking into account its **3D details**, that is the visual aspect of the object, and its **programmed behaviour** using LSL scripts, LSL stands from *Linden Scripting Language*. This island is used to guide students by showing them different examples.

The second island, called *Sandbox island*, has been designed to place the different experiments. This island contains several parcels and the students must select one of those parcels to build there whatever they want. Figure 2(b) shows a parcel contained in this island. There are three different types of structures that the students could select to build:

1. A composition focused mainly in the graphical design. This composition is built with many objects and the 3D aspect is very detailed. Most of the effort carried out is dedicated to the graphical aspect so little programmed functionality is expected.
2. A composition with simple graphical aspect but with a lot of programmed functionality. This case is the opposite to the previous one, in this composition the graphical aspect is not really important because all the effort is applied to program the functionality with script files.
3. A mixed between previous cases, where the composition is not very realistic but it has some programmed functionality.

Students decide what they are going to build and which is the proportion between 3D details and programmed behaviour. With this goal, different student profiles are expected to appear in the VW. For example, the behaviour of those students focused



(a) Main island (b) Sandbox island

Fig. 2. Examples of the two different islands used in this work. The left figure shows the *Main island* that contains several objects and it is used as an example island. The right figure shows the *Sandbox island* where students must work in different parcels.

on programming should differ from the behaviour of those students building a realistic composition.

5. Extracted Avatar Information and Behaviour Detection

As it was described in Section 3, one of the modules of VirtUAM is in charge of inserting in a database the avatar characteristics and interactions. In this section a description about the extracted data and how it has been analysed is given.

5.1. Avatar position

The first characteristic analysed in this work is the **Avatar position** in form of GPS coordinates. This information is inserted each 5 seconds, only if the current position is different to the last registered position for this avatar. This check avoids the insertion of duplicated information in the database. The avatar position can be used to define communities composed by those students working in near places. The algorithm used to create avatar communities based on the avatar position is *K-means* [38, 39, 31].

K-means is a popular and well known partitional clustering algorithm. It is a straightforward clustering guided method (usually by a heuristic or directly by a human) to classify data in a predefined number of clusters. Given a fixed number of clusters, K-means tries to find a division of the dataset based on a set of common features given by distances or metrics that are used to determine what elements belong to each cluster.

K-means algorithm depends on a parameter called k. This parameter fixes the number of clusters in which data are going to be grouped. There are different ways to specify the value for this parameter:

1. **Giving a range of values**. This approach tries to group data using different values of k between a *min* and *max* value that must be specified.

2. **Giving a value determined by the problem**. In this case, the modelled problem determines the fixed number of clusters.
3. **Determining the value using statistical measures**. Some other works use the Bayesian information criterion [24, 46] over data generated by Gaussian distributions. Other works assume that data are generated by a Poisson distribution [22] and other papers work with Monte Carlo techniques [21]. In all these cases the statistical approach used generates the best value for k.
4. **Giving the same value as the number of classes**. This option is used with classification techniques and it is very similar to option 2. In this case, the user wants to classify data into a given number of classes. Therefore the number of clusters is the same as the number of classes, because each class is represented by one cluster.
5. **Specifying the value using data visualization**. This technique uses the data visualization to determine the number of clusters. This option can be considered as *Supervised* clustering. The disadvantage of using this approach is that sometimes it is not possible to use this method because there is so many information that the visual identification of the number of clusters is very difficult.

In this work, the algorithm described in [47] has been implemented and used. Authors use a function to evaluate the goodness of a given value of k, this function ($f(K)$) takes into account the distance between each value to its corresponding centroid penalizing those clusters composed by only one data. In order to define the value of k, authors launch the *k-means* several times changing the value of k, from 1 to N clusters, where N represents the number of elements that compose the dataset. For each k-value, the value $f(k)$ is computed. Finally, the k-value with lowest $f(k)$ is considered as "the best" value for k.

Although K-means algorithm has been used in many different domains, and it has been improved several times [32, 34, 60] or even combining k-means with other algorithms such as Genetic algorithms [41, 48], in this work the classical algorithm is used. Applying the K-means algorithm to avatar position, the communities composed by students working in near places will appear. This information is useful because in VWs students are free to move around the world and thus students would go away from the teacher and not pay attention to the given task.

5.2. Avatar behaviour

The second characteristic is the **Avatar behaviour**, defined by those actions that students can perform in the VW. The set of actions taken into account are the following:

- **Move Running (MR)**. This action is registered when the avatar is running.
- **Move Flying (MF)**. This action represents the avatar while he/she is flying.
- **Move Teleporting (MP)**. Another action is teleporting. Using the teleport avatars can move immediately to any place in the VW.
- **Public Chat (XA)**. With this action any public chat conversations are represented.
- **Private Chat (XO)**. Students can chat with each other through a private channel, in this case the action is registered as XO.
- **Build Compiling (BC)**. While students are compiling or writing some functionality on scripts, this action is registered.

– **Build Touching (BT)**. This action represents the moment when the avatar is building physical objects.

While students are working in the VW, there is a procedure that stores in the Data Base (DB) the different actions performed. The result is a string composed by a sequence of actions (*MR,MF,MP,XA,XO,BC,BT*) that represents in a sequential order what was doing the avatar in each moment. Figure 3 shows both, the action string and the related actions done by the avatar. The resulting file containing these actions is called *behavioural file*.

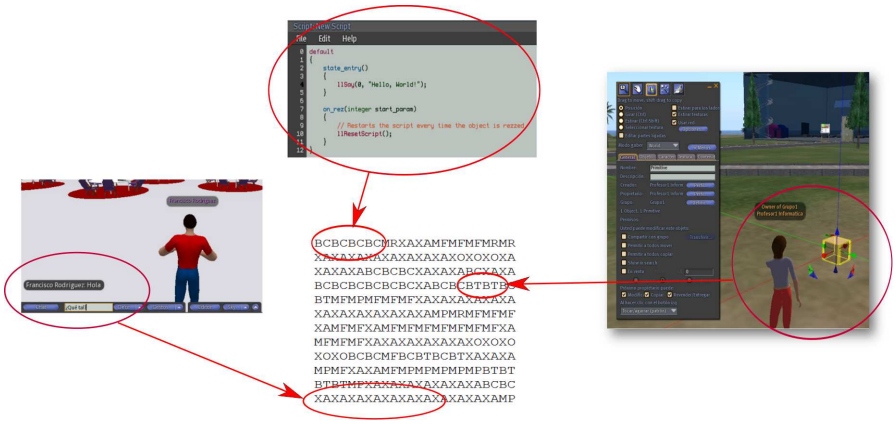


Fig. 3. Example of different actions that avatars can perform in the VW and how this actions are represented in the behavioural file.

Once actions for all avatars have been stored, these behavioural files can be compared. In order to classify the avatar behaviour a metric is needed, but this metric should takes into account that our behavioural model is based on a text representation. For this reason, the *Normalized Compression Distance (NCD)* has been selected. This metric provides a measure of the similarity between two objects, x and y, using compressors. The definition is as follows:

$$NCD(x,y) = \frac{\max\{C(xy) - C(x), C(yx) - C(y)\}}{\max\{C(x), C(y)\}}, \tag{1}$$

where C is a compression algorithm, $C(x)$ is the size of the C-compressed version of x, and $C(xy)$ is the compressed size of the concatenation of x and y. NCD generates a non-negative number $0 \leq NCD(x,y) \leq 1$. Distances near 0 indicate similarity between objects, while distances near 1 reveal dissimilarity.

The NCD is just one of the many similarity distances that use compression algorithms. Other distances [6, 33, 61] are small variations and can be easily reduced to it, as it is possible to prove that this distance is as good as any other that can be computed by a

universal Turing machine [56]. However, this metric has been successfully used in a wide number of domains based on text mining and classification [20, 23, 54].

In this paper, CompLearn Toolkit [14] is used to compute and visualize the resulting NCD-based clustering described in [13]. This clustering algorithm comprises two phases. First, the NCD matrix is calculated using a compression algorithm. In this work the compression algorithm used is LZMAX, which is a Lempel-Ziv-Markov chain algorithm [45]. Second, the NCD matrix is used as input to the clustering phase and a dendrogram is generated as output. A dendrogram is an undirected binary tree diagram, frequently used for hierarchical clustering, that illustrates the arrangement of the clusters produced by a clustering algorithm. Several dendrograms can be seen in Figure 8(a) and Figure 9.

6. Experimental results

To evaluate the performance of our behavioural classification model, two different experiments have been carried out. The first one is composed by 5 students whose task is to work in groups and build a set of objects in two parcels. All these students have worked in the VW before, so they have a deep knowledge about the platform functionality. The second experiment is carried out in an open scenario, where 30 students work during a month (7 or 8 students per week). These students work individually but they are free to visit his/her friends' parcels to watch what they are building.

In both experiments, students positions are analysed using *K-means algorithm* while *Normalized Compression Distance* is used to identify behavioural communities, with the main goal of discovering communities or avatar groups based on their behaviours.

6.1. K-Means analysis

In this subsection the results obtained with *K-means* algorithm using avatars position are shown. This analysis is applied to both the control group and the open scenario.

Control Group

As it was previously described, the control group is composed by 5 students with wide knowledge in VirtUAM. The main task for these students was to build some objects and program some functionality using LSL language in two different parcels. Three different working groups were created, one of them was composed by 2 students. Another group was composed by other 2 students, and the remaining student simply walked around the VW without cooperating with any of the described groups. For that reason the expected *k-value* was 2 or 3, depending on how fine our approach works to identify the student communities. This control group working under a controlled experiment tries to show empirically whether our approach is correctly working or not.

The results of applying *K-means* algorithm are shown in Figure 4. Right-figure shows the $f(k)$ value associated to this experiments, the mean value is obtained when $k = 3$ and the left-figure shows the resulting clustering using 3 different clusters.

Note that Figure 4 and the right images of Figure 5(a) and Figure 5(b) contain more data than the number of students involved in the experiments. This is produced because each point does not represent an avatar but it represents only a position in a specific

moment. In this work, k-means algorithm uses the data of a Snapshot, i.e. k-means works with all data generated in time intervals of 40 seconds. For this reason avatars can contribute with more than one pair of data $< Xposition, Yposition >$.

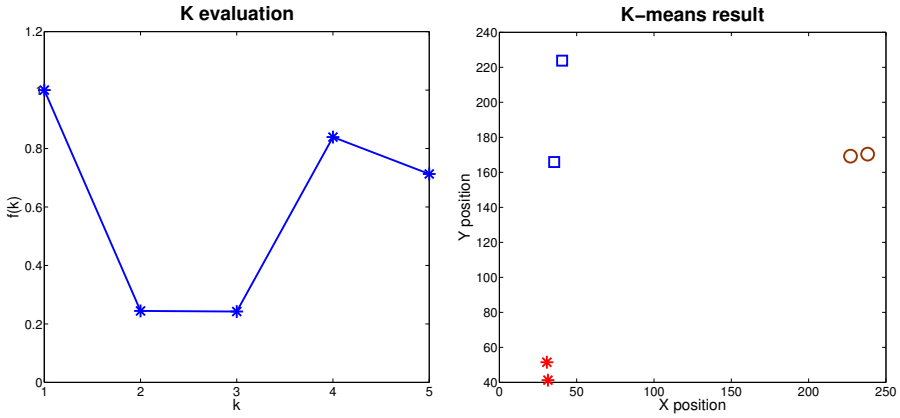


Fig. 4. Result of applying K-means with the control group data. Left figure shows the evaluation of the parameter k and right figure shows the result of the algorithm with the mean k-value obtained on the left figure.

As Figure 4 shows, the K-means algorithm positional avatar information is able to identify three different communities of students.

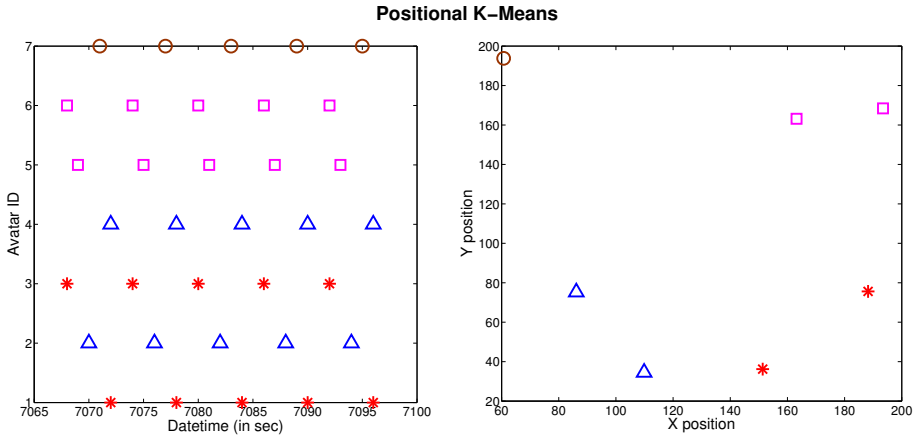
Open scenario

Once k-means algorithm has been tested with the control group data, its application using data extracted from the open scenario is analysed. This open scenario is composed by 7 non-skilled users working in the VW during a whole week (4 hours per day).

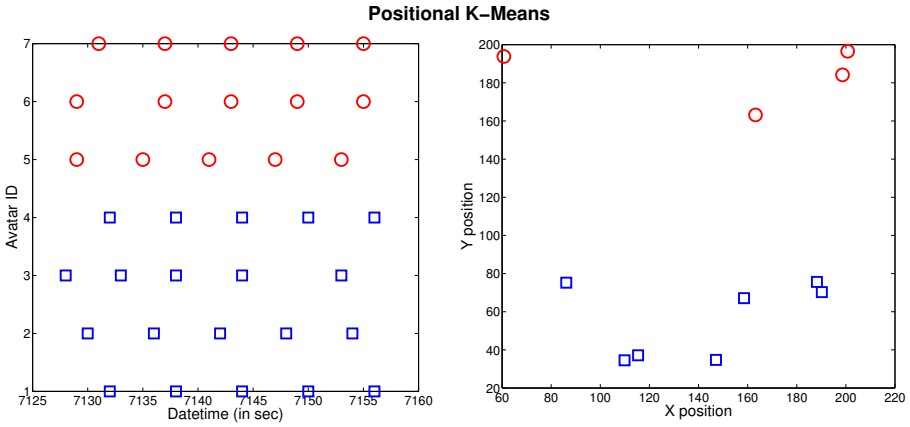
In this case, a new problem appears. It is possible that k-means algorithm does not have enough data to perform the clustering task in a specific moment. For this reason, a time interval containing all data between two moments is used to guarantee enough data to k-means algorithm. Experimentally, this time interval (called *Snapshot*) was fixed to 40 seconds.

The results obtained by applying k-means over this data are shown in Figure 5. As can be seen in this figure, avatars can be easily group in clusters depending on their positions.

Analysing the results of applying *k-means* algorithm over consecutive snapshots some interesting features can be observed. For example, while in Figure 5(a) students do not move a lot and this characteristic makes the *k-means* algorithm cluster students in three clusters, the movement of students in Figure 5(b) makes the algorithm identify two different groups. Therefore, from one snapshot to the following one, a cluster has disappeared. This means that some students have virtually move to other communities of students to cooperate with them.



(a) Snapshot n



(b) Snapshot $n+1$

Fig. 5. Results of applying K-means algorithm to two consecutive snapshot extracted from the open scenario where 7 students work in the Virtual World during a week.

Also Figure 6 shows the evolution of different clusters in three consecutive snapshots. Initially there are 3 different clusters (or communities), the first one is composed by $Avatar7$, the second cluster is composed by $Avatar1$, $Avatar5$ and $Avatar6$, while $Avatar2$, $Avatar3$ and $Avatar4$ compose the third cluster. In the next snapshot, the third cluster is split in other two clusters, one of them composed by $Avatar4$, and other clusters composed by $Avatar2$ and $Avatar3$. Finally, the cluster just created (composed by $Avatar2$ and $Avatar3$) is merged with the initial cluster composed by $Avatar1$, $Avatar5$ and $Avatar6$. Figure 7 shows the results shown in Figure 6 using a more graphical representation. In Figure 7 dotted arrows represent the avatar movement between snapshots.

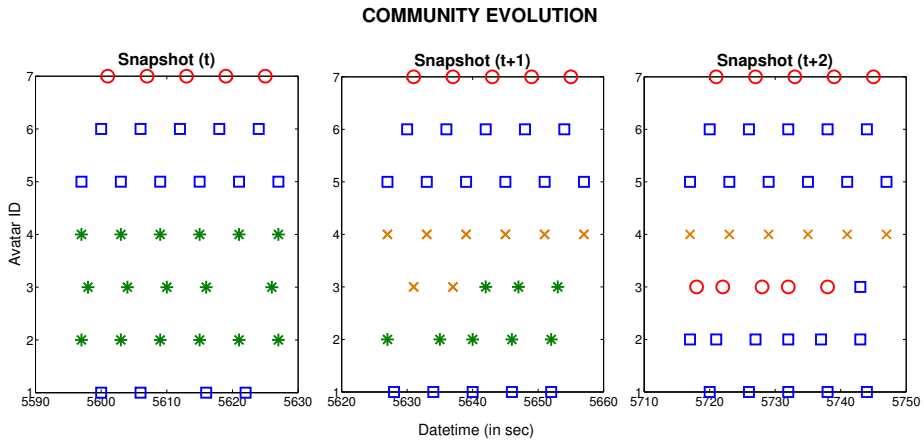


Fig. 6. Evolution of different clusters in three consecutive time snapshots (Detailed version).

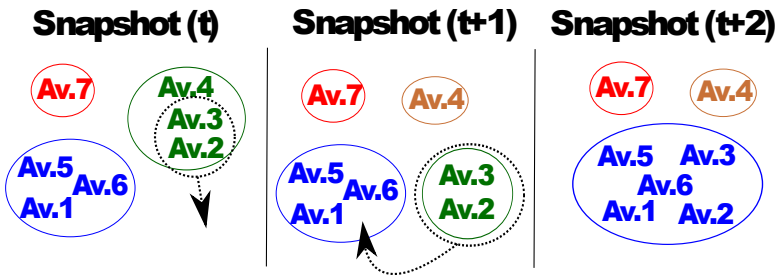


Fig. 7. Evolution of different clusters in three consecutive snapshots (Graphical version).

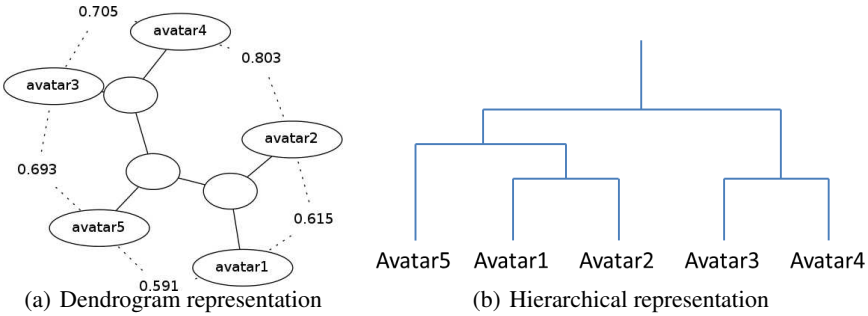
It is important to highlight the movement of $Avatar3$ in Figure 6. Initially, this avatar belonged to a cluster composed by $Avatar2$, $Avatar3$ and $Avatar4$. Then this cluster is split into two clusters, one of them composed by $Avatar3$ and $Avatar4$ (named C_1) and the other cluster (named C_2), by $Avatar2$. But in 10 seconds $Avatar3$ leaves this new cluster (C_1), to be part of the cluster named C_2. But in the next snapshot, $Avatar3$ belongs to the cluster composed by $Avatar7$ and finally, $Avatar3$ is included in a big cluster composed by $Avatar1$, $Avatar2$, $Avatar5$, $Avatar6$ and himself. This detailed analysis shows how our approach is able to detect easily the modification and evolution of avatar communities in short periods of time only based on the avatar positions.

6.2. NCD analysis

The next step is the behaviour analysis using the *Normalized Compression Distance* (NCD) described in section 5.2. The goal of these experiments is to group together those students that show similar behaviours.

Control Group

The resulting dendrogram that contains the avatar behaviour clustering for the Control group experiment is shown in Figure 8(a), nodes represent avatars while the numerical values between nodes represent the distance value between the nodes. A dendrogram is a hierarchical clustering, Figure 8(b) shows the same results as 8(a) but using a classical representation.



(a) Dendrogram representation (b) Hierarchical representation

Fig. 8. Results of applying *Normalized Compression Distance* with the avatar behaviour extracted from the Control Group experiment.

As can be seen in Figure 8(a), dendrogram allows to define different communities taking into account students behaviour where $Avatar1$ and $Avatar2$ performed similar actions, as $Avatar3$ and $Avatar4$ did.

Open scenario

Analysing the different files that contain the control group behaviour, a new problem related to the size of the resulting files appears. The larger the student session, the bigger resulting files. This unfortunately difficult the NCD analysis because it needs huge computational resources. For this reason, some new representation schemas are needed to simplify and reduce the size of these files.

Four different representations was studied and analysed:

1. **Representation #1**. This representation is the one showed in Figure 3, where the actions performed are stored in chronological order.
2. **Representation #2**. In this representation, actions are simplified with a number indicating the number of consecutive repetitions of this action. For example, ”*2MF3BT1XA*” written using representation #1 means *MFMFBTBTBTXA* (i.e. 2 times *Move Flying*; 3, *Build Touching* and 1 *Chat All*).This representation reduce the size of the behavioural files, as other compression techniques used to reduce the size of the file deleting repeated information.
3. **Representation #3**. In this type of representation, the number following each action represents the total number of times that this action has been reproduced. With this

representation, "*2MF3BT1XA*" does not means the same as representation #2, but it means that avatar has moved flying (MF) 2 times (in the whole session). In this case the previous example could be "*MFBTMFXABTBT*", or "*MFBTXABTMFBT*" or "*MFBTBTMFBTXA*. This representation keeps the appearance order, this means that two behavioural files could not start with the same sequence of actions, because one avatar can start the session building new objects (MF) while other avatar start going for a walk inside the VW.

4. **Representation #4**. Finally, this representation is the same as representation #3 but it keeps the same order all the time. This means that behavioural files have the following structure:

$$\alpha\mathbf{XA}\beta\mathbf{MF}\gamma\mathbf{MR}\delta\mathbf{BT}\epsilon\mathbf{BC}\rho\mathbf{XO}\eta\mathbf{MP} \tag{2}$$

where $\alpha, \beta, \gamma, \delta, \epsilon, \rho, \eta$ are the number of times that the actions are repeated. Using this representation the size of the behavioural files are strongly reduced but this representation lacks of temporal representation.

Finally, the results of the NCD applied to the behavioural files with these representations are shown in Figure 9. This figure shows the result of applying *NCD* over the behavioural files using the described representations. Representation #1 is show in subfigure *a*, whereas representation #2 is shown in subfigure *b*. And subfigures *c* and *d* show representations #3 and #4 respectively.

There is a trade-off between the size of the behavioural files and the accuracy of the NCD. On one hand, representation #1 is the more detailed representation (and thus it provides the more precise clustering results) but the resulting behavioural files can be extremely large complicating the *NCD* clustering task. On the other hand, representation #4 is the most compacted representation but *NCD* results are not very precise. This is shown in Figure 9, where avatar with similar behaviours (i.e. the communities composed by $Avatar1$ and $Avatar4$, and $Teacher1$ and $Teacher2$) are very close using representation #1 (Subfigure *a*) but if the representation is less precise, these avatars appear more separated in the corresponding dendrogram. This fact is shown in Subfigure *d* where $Avatar1$ and $Teacher1$ are located far away from their corresponding partners $Avatar4$ and $Teacher2$ respectively.

Also the loss of accuracy is reflected in the NCD-values shown between nodes in the dendograms. Representation #1 (Subfigure *a*) is the most accurate representation because it contains all the information in chronological order, i.e. it is the most complete representation. This fact is reflected with high NCD-values that goes from 0.75 to 0.95. The less accurate is the representation, the lower NCD-values are obtained. In this case, representation #4 is the representation with less information and this is shown with NCD-values between 0.55 and 0.7. It seems that files using representation #4 are similar, and this is a reasonable because all files contains the structure.

7. Conclusions

Virtual Worlds (VW) have gained popularity in recent years in domains like training or education mainly due to their highly immersive and interactive 3D characteristics. This popularity has been reflected in the increment of the number of research works applied to VWs.

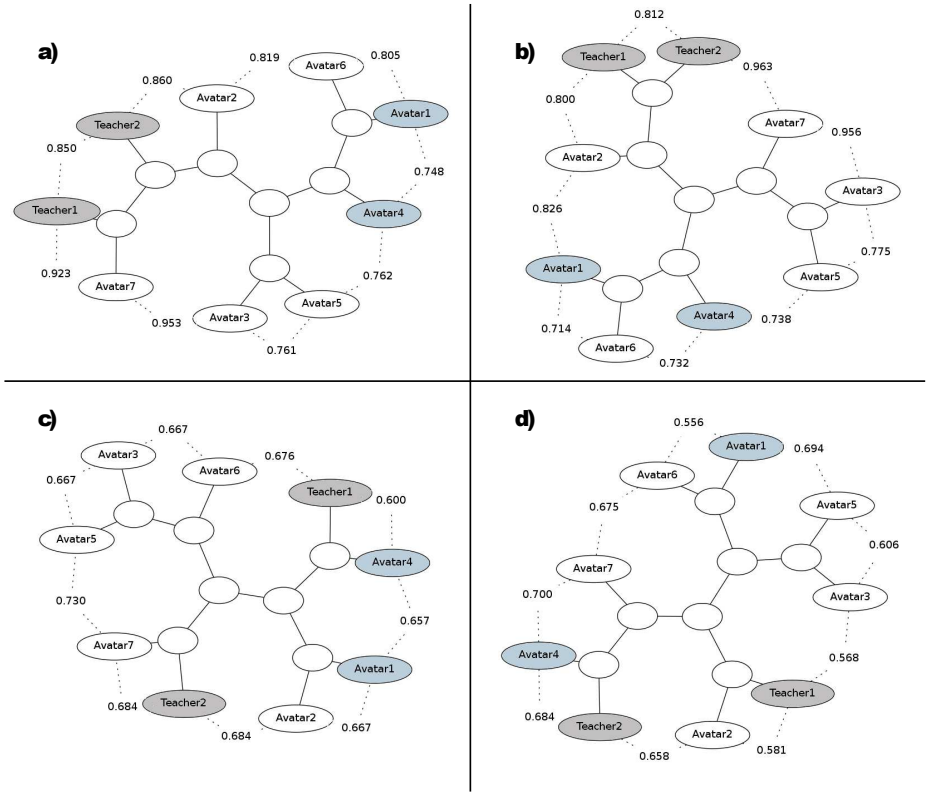


Fig. 9. This figure shows the results of applying *NCD* over the behavioural files using the described representations. Representation #1 is show in subfigure a, representation #2 is shown in subfigure b. And subfigures c and d show representations #3 and #4 respectively. Two identified communities ($Teacher1$ and $Teacher2$, and also $Avatar1$ and $Avatar4$) has been highlighted.

Despite the different VW platforms (such as OpenCobalt or OpenSim amongst others), almost none of them provide mechanisms to facilitate the automatic (or semi-automatic) behaviour analysis of users interactions. For this reason, this work presents the results of a developed statistical module for a VW platform called VirtUAM.

In order to analyse the avatar behaviour, two main characteristics are taken into account, the avatar position and their actions in the VW. The former is the position of the avatar in Global Positioning System (GPS), the latter is a text file containing the different actions performed by the avatar in the VW. This work uses K-means algorithm and the *Normalized Compression Distance* (*NCD*) to perform two different types of clustering. As K-means algorithm needs the definition of a parameter, called k, this work computes this value using the algorithm described in [47].

Experimental results, shown in Section 6, reveal that both algorithms can be applied to clustering techniques. K-means algorithm can be used to identify avatar communities based on the avatar positions. Also this algorithm detects the avatar movements, and the

different changes performed to the community compositions, i.e. members leaving and joining different communities. Therefore, it is possible to detect community evolution by analysing the different resulting graphs. As a future work, it would be interesting the inclusion of different algorithms that perform community evolution tasks automatically (such as CommTracker [11, 58]).

Therefore, a next step in this research could be the integration of both techniques into a common and reliable method to automatically detect and track the evolution of those communities based on the characteristics shown in this work.

Acknowledgments. This work has been funded by the Spanish Ministry of Science and Innovation under the project ABANT (TIN2010-19872/TSI).

References

1. Aldrich, C.: Learning online with games, simulations and virtual worlds. Strategies for online instruction. John Wiley (2009)
2. Baker, S.C., Wentz, R.K., Woods, M.M.: Using virtual worlds in education: Second life as an educational tool. Teaching of Psychology 36(1), 59–64 (2009), http://dx.doi.org/10.1080/00986280802529079
3. Baran, B.: Facebook as a formal instructional environment. British Journal of Educational Technology 41(6) (2010)
4. Bello-Orgaz, G., R-Moreno, M.D., Camacho, D., Barrero, D.F.: Clustering avatars behaviours from virtual worlds interactions. In: Proceedings of the 4th International Workshop on Web Intelligence and Communities. pp. 1–7. ACM (2012), http://doi.acm.org/10.1145/2189736.2189743
5. Bellotti, F., Berta, R., De Gloria, A., Zappi, V.: Exploring gaming mechanisms to enhance knowledge acquisition in virtual worlds. In: Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts. pp. 77–84. DIMEA '08, ACM (2008), http://doi.acm.org/10.1145/1413634.1413653
6. Benedetto, D., Caglioti, E., Loreto, V.: Language Trees and Zipping. Physical Review Letters 88(4), 48702 (2002)
7. Berns, A., Gonzalez-Pardo, A., Camacho, D.: Designing videogames for foreign language learning. In: 4th International Conference ICT for Language Learning, Florence (Italy) (2011)
8. Berns, A., Gonzalez-Pardo, A., Camacho, D.: Game-like language learning in 3-d virtual environments. Computers and Education 60(1), 210 – 220 (2013)
9. Boulos, M.N.K., Hetherington, L., Wheeler, S.: Second life: an overview of the potential of 3-d virtual worlds in medical and health education. Health Information & Libraries Journal 24, 233–245 (2007)
10. Bourke, P.: Chaos and graphics: Evaluating second life for the collaborative exploration of 3d fractals. Comput. Graph. 33, 113–117 (2009), http://dx.doi.org/10.1016/j.cag.2008.08.004
11. Cajias, R., Gonzlez-Pardo, A., Camacho, D.: A swarm simulation platform for agent-based social simulations. In: 5th International Symposium on Intelligent Distributed Computing (IDC 2011). vol. 382, pp. 265 – 270. Springer Berlin / Heidelberg (2011)
12. Castronova, E.: Synthetic Worlds : The Business and Culture of Online Games. University Of Chicago Press (2005)
13. Cilibrasi, R., Vitanyi, P.: Clustering by compression. IEEE Transactions on Information Theory 51(4), 1523–1545 (2005)
14. Cilibrasi, R., Cruz, A.L., de Rooij, S., Keijzer, M.: CompLearn Toolkit. [Online] Available: http://www.complearn.org/

15. Consortium, T.N.M.: The horizon report. Tech. rep. (2007)
16. Cunha, M., Raposo, A.B., Fuks, H.: Educational technology for collaborative virtual environments. In: Proceedings of the 12th International Conference on CSCW in Design, CSCWD. pp. 716–720 (2008)
17. Edwards, W.K.: Putting computing in context: An infrastructure to support extensible context-enhanced collaborative applications. ACM Transactions on Computer-Human Interaction (TOCHI 12 (2005)
18. de Freitas, S.: Learning in immersive worlds: A review of game-based learning. Tech. rep., JISC e-Learning Programme (2006)
19. Gonzalez-Pardo, A., de Borja Rodriguez, F., Pulido, E., Camacho, D.: Using virtual worlds for behaviour clustering-based analysis. In: ACM Workshop on Surreal Media and Virtual Cloning. pp. 9 – 14. ACM (2010)
20. Gonzalez-Pardo, A., Granados, A., Camacho, D., de Borja Rodriguez, F.: Influence of music representation on compression-based clustering. In: IEEE Congress on Evolutionary Computation (CEC). pp. 2988 – 2995. IEEE (2010)
21. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods: Part i. SIGMOD Record 31(2), 40–45 (2002)
22. Hardy, A.: On the number of clusters. Computational Statistics & Data Analysis 23(1), 83–96 (1996)
23. Helmer, S.: Measuring the structural similarity of semistructured documents using entropy. In: Proceedings of the 33rd international conference on Very large data bases. pp. 1022–1032. VLDB '07, VLDB Endowment (2007)
24. Ishioka, T.: Extended k-means with an efficient estimation of the number of clusters. In: Intelligent Data Engineering and Automated Learning - IDEAL 2000, Data Mining, Financial Engineering, and Intelligent Agents, Second International Conference, Shatin, N.T. Hong Kong, China, December 13-15, 2000, Proceedings. pp. 17–22. Springer (2000)
25. Jarmon, L., Traphagan, T., Mayrath, M., Trivedi, A.: Virtual world teaching, experiential learning, and assessment: An interdisciplinary communication course in second life. Computers & Education 53(1), 169–182 (2009)
26. Jung, J.J.: Social grid platform for collaborative online learning on blogosphere: A case study of elearning@bloggrid. Expert Syst. Appl. 36(2), 2177–2186 (2009)
27. Jung, J.J.: Ubiquitous Conference Management System for Mobile Recommendation Services Based on Mobilizing Social Networks: a Case Study of u-Conference. Expert Syst. Appl. 38(10), 12786–12790 (2011)
28. Jung, J.J.: Semantic Optimization of Query Transformation in a large-scale peer-to-peer network. Neurocomputing 88, 36–41 (2012)
29. Jung, J.J.: ContextGrid: A Contextual Mashup-based Collaborative Browsing System. Information Systems Frontiers 14(4), 953–961 (2012)
30. Jung, J.J.: Evolutionary Approach for Semantic-based Query Sampling in Large-scale Information Sources. Information Sciences 182(1), 30–39 (2012)
31. Jung, J.J.: Cross-lingual Query Expansion in Multilingual Folksonomies: a Case Study on Flickr. Knowledge-Based Systems 42, 60–67 (2013)
32. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Anal. Mach. Intell. 24(7), 881–892 (2002)
33. Kraskov, A., Stoegbauer, H., Andrzejak, R., Grassberger, P.: Hierarchical clustering using mutual information. Europhysics Letters 70(2), 278–284 (2005)
34. Lai, J.Z.C., Liaw, Y.C.: Improvement of the k-means clustering filtering algorithm. Pattern Recogn. 41(12), 3677–3681 (2008)
35. Lamb, A., Johnson., L.: The potential, the pitfalls, and the promise of multiuser virtual environments: Getting a second life. Teacher Librarian 36(4), 68–72 (2009)

36. Lipka, S.: For professors, 'friending' can be fraught. (cover story). Chronicle of Higher Education 54(15) (2007)
37. Lucia, A.D., Francese, R., Passero, I., Tortora, G.: Supporting jigsaw-based collaborative learning in second life. In: Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies. pp. 806–808. ICALT '08, IEEE Computer Society (2008)
38. MacKay, D.: Information Theory, Inference and Learning Algorithms. Cambridge University Press (2003)
39. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. pp. 281–297 (1967)
40. Madge, C., Meek, J., Wellens, J., Hooley, T.: Facebook, social integration and informal learning at university: it is more for socialising and talking to friends about work than for actually doing work?. Learning Media And Technology 34(2), 141–155 (2009)
41. Min, W., Siqing, Y.: Improved k-means clustering based on genetic algorithm. In: Computer Application and System Modeling (ICCASM), 2010 International Conference on. vol. 6, pp. V6–636 –V6–639 (2010)
42. Nardi, B.A., Ly, S., Harris, J.: Learning conversations in world of warcraft. Hawaii International Conference on System Sciences 0 (2007)
43. Nelson, B.C., Ketelhut, D.J.: Exploring embedded guidance and self-efficacy in educational multiuser virtual environments. In: Proceedings of the 8th iternational conference on Computer supported collaborative learning. pp. 548–550. CSCL'07, International Society of the Learning Sciences (2007)
44. Park, Y.B., Lee, Y., Lee, J., Kang, J., W.B.: Effects of 3d-simulation-based instruction on students' achievement and interests in a manufacturing engineering class. International Journal of Engineering Education 24, 843–849 (2008)
45. Pavlov, I.: LZMAX. [Online] Available: http://www.7-zip.org/sdk.html
46. Pelleg, D., Moore, A.: X-means: Extending k-means with efficient estimation of the number of clusters. In: In Proceedings of the 17th International Conf. on Machine Learning. pp. 727–734. Morgan Kaufmann (2000)
47. Pham, D.T., Dimov, S.S., Nguyen, C.D.: Selection of k in k -means clustering. Proceedings of the I MECH E Part C Journal of Mechanical Engineering Science 219(1), 103–119 (2005)
48. Prabha, K., Saranya, R.: Refinement of k-means clustering using genetic algorithm. Journal of Computer Applications (JCA) 4(2), 40 – 44 (2011)
49. Ritzema, T., Harris, B.: The use of second life for distance education. Journal of Computing Sciences in Colleges 23(6), 110 – 116 (2008)
50. Selwyn, N., Grant, L.: Researching the realities of social software use an introduction. Learning, Media and Technology 34, 79 – 86 (2009)
51. Sinrod, E.J.: Virtual world litigation for real (2007), http://news.cnet.com/Virtual-world-litigation-for-real/2010-1047_3-6190583.html
52. Slator, B., Hill, C., Del Val, D.: Teaching computer science with virtual worlds. IEEE Transactions on Education 47(2), 269–275 (2004)
53. Stephenson, N.: Snow Crash. Bantam Books (1992)
54. Telles, G., Minghim, R., Paulovich, F.: Normalized compression distance for visual analysis of document collections. Computers & Graphics 31(3), 327–337 (2007)
55. Trotter, A.: Educators get a 'second life'. Education Week 27(42), 1–17 (2008)
56. Turing, A.: On computable numbers, with an application to the entscheidungsproblem. Proceedings of the London Mathematical Society 2(42), 230–265 (1936)
57. Usluel, Y.K., Mazman, S.G.: Adoption of web 2.0 tools in distance education. Procedia Social and Behavioral Sciences 1(1), 818–823 (2009)
58. Wang, Y., Wu, B., Pei, X.: Commtracker: A core-based algorithm of tracking community evolution. In: Proceedings of the 4th international conference on Advanced Data Mining and Applications. pp. 229–240. ADMA '08, Springer-Verlag, Berlin, Heidelberg (2008)

59. Waters, J.: A second life for educators. Journal - Water Pollution Control Federation 36(1), 29 (2009)
60. Wu, J., Yu, W.: Optimization and improvement based on k-means cluster algorithm. In: Knowledge Acquisition and Modeling, 2009. KAM '09. Second International Symposium on. vol. 3, pp. 335 –339 (2009)
61. Zhang, X., Hao, Y., Zhu, X., Li, M.: Information distance from a question to an answer. In: KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 874–883. ACM, New York, NY, USA (2007)

Antonio González Pardo is a teaching assistant in Universidad Autónoma de Madrid. He holds a BSc in Computer Science from Universidad Carlos III de Madrid (2009) and a MSc in Computer Science from Universidad Autónoma de Madrid (2011). Nowadays, he is a Computer Science PhD candidate at Escuela Politécnica Superior (UAM). Currently he is involved with AIDA interest research group at EPS-UAM, his main research interests are related to computational intelligence (genetic algorithms, PSO, SWARM intelligence, etc), simulation and multi-agent systems. The application domains for his research are Constraint Satisfaction Problems (CSP), Video-games and Virtual Worlds.

Angeles Rosa is a last-year Computer Science student in Universidad de Málaga (UMA). She collaborates with AIDA research group at EPS-UAM in different projects related to her main research interests like Artificial Intelligence and Data Mining.

David Camacho is currently working as Associate Professor in the Computer Science Department at Universidad Autónoma de Madrid (Spain). He received a Ph.D. in Computer Science (2001) from Universidad Carlos III de Madrid, and a B.S. in Physics (1994) from Universidad Complutense de Madrid. He has published over 100 journal, books, and conference papers. His research interests include Data Mining (Clustering), Evolutionary Computation (GA & GP), Multi-?Agent Systems and Computational Intelligence (Swarm computing), Automated Planning and Machine Learning.

Received: February 14, 2013; Accepted: November 13, 2013.