CSE 591: Adaptive Web
Object Oriented Tutorial using Open User model in adaptive augmented
Group Report - 2018

Abhinethri Tumkur Umesh	Jithin Perumpral John	Pallavi Duggal	Chetan Gupta	Sukanya Panda
1213187714	1213175858	1212940207	1213185179	1213317363
atumkuru@asu.edu	jperumpr@asu.edu	pduggal@asu.edu	cgupta6@asu.edu	spanda7@asu.edu

ABSTRACT

The data on the World Wide Web has been developing exponentially and users are required to perform explicit searches with the end goal to discover data that is most significant to their own needs. This turns into a more concerning issue with regards to a learning community that expects users to take in proactively without adequate intervention from the instructor or developer. Adaptive Support in primary fields has turned out to be in excess of a need. Our project depicts the diverse techniques in which adaptive support can be given to users and to exhibit this, we have developed a unity 3D game with a hidden explicit feedback framework dependent on cooperative filtering. The development progress is part of 5 different areas: the Introduction, the plan, the Implementation, Evaluation, and future work.

Keywords: Adaptive feedback, Adaptive support, Unity3D, Open user model, Learn java, accelerometer

1. INTRODUCTION

The objective of the project is to build an augmented reality game application utilizing unity3D. A game engine, for example, Unity 3D for instance, enables you to see the project running progressively, in a "Scene" window. The user would then be able to add objects to the scene, for example, cubes, spheres, canvas (and so forth.). In any case, every one of the objects that are added to the scene, are static. The game enables users to play a ball game and learn from feedback, see statistics and dynamic visualizations regarding their execution. The user gets versatile feedback about his/her performance. We endeavored to implement open user model. Open User Model - A user model is the accumulation and classification of individual information related with an explicit user. The key segments of our model are transparency, controllability, and visualization. In this project, we're planning an Open User Model utilizing Augmented Reality, by considering user interests and preferences, to assist a user with learning the Object Oriented Concepts. This report will manage you through every single part of the functionalities and the engineering of our project. Our goal is to gather every one of the pins by rolling a ball. Each pin is related to an Object Oriented Concept. The user's activities will be logged and shown utilizing the Open User Model. A working model had been executed.

2. MOTIVATION

Late headways in advancements have given affordances to improving the experience of learning and giving adaptive help to users to viably expand the convenience of nature and encourage a higher level of learning. Today there is the huge importance of building a far-reaching user model in giving the "adaptation impact" and consolidating components of personalization into the learning model. The essential goal behind any adaptive framework is to gather data from different sources dependent on the user's communication with the framework. User modeling can be extensively arranged into three layers in particular:

- Nature of the data being modeled
- Representation and structure of the data
- Maintenance of various types of models

The data required for user profiling is gathered utilizing the strategies of explicit and implicit feedback. Information gathering utilizing explicit feedback includes an extra weight the users while a feedback framework that gathers information unequivocally requires a lot of help from the framework. There is a considerable measure of advancements of different frameworks for recognizing fascinating applications wherein they have built user profiles dependent on explicit feedback recorded by evaluations given by the user.

Ontological user profiling is another fascinating method utilized for building a user profile that focuses more on learning the conduct and the deductions that can be made

dependent on the data are definitely known to the framework. User models advance dependent on the learning achieved utilizing the data gathered from feedback frameworks. Adaptive introduction of the data gathered assumes a vital job in numerous fields, for example, in the education field, in this day and age. The strategies associated with handling the modeled data and rendering it to the user enables the user to assess the adequacy of the application. Another intriguing component that an adaptive instructive programming should fuse is the open social understudy modeling. Helping user draw in with the framework in making his user model gives a more profound comprehension of the user and a user-driven way to deal with building up the user model.

3. DATA COLLECTION AND ANALYSIS

3.1. Data Source

We collected our data from the accelerometer and gyroscope built into the device. Apart from this, we also did a user study wherein we collected data for the user using a device which can collect accelerometer and, gyroscope data for the user holding and playing in the device. After collecting the data, the next task was to label the data we had to manually label the data using the video captured while collecting the data.

After extracting features from the raw data fast and slow players, we append both the dataset matrix. We then apply PCA on it and thereafter multiple outputs of the PCA(obtained top 5 eigenvectors with the above feature set matrix to obtain a projection of the dataset into these eigenvectors. We also label the top half rows as slow and later half as fast after the two matrices are appended, assuming n as the total number of rows(actions) in the matrix. This matrix finally contains all our raw data collected during the experiment.

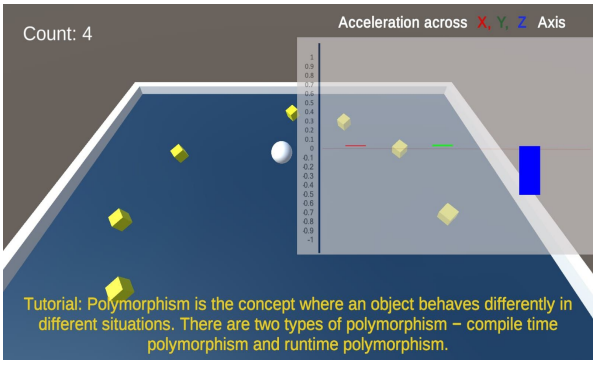


Fig1: User's speed data collection

After finding out the dominant features among the raw data containing x,y,z values for accelerometer and gyroscope, we were able to identify and differentiate between slow and fast players. For the slow player, we decided to retain the same speed and for fast players who are unable to collect the yellow cubes, we have decided to reduce the speed by half. This was one of the adaptation we did in the game.

3.2. Normalization

Since the accelerometer data is very noisy, we decided to add a few filters before collecting the raw data. After the filters (threshold accelerations) are applied, we normalize the data and remove the outliers from the raw signals by ignoring the extrema values of the accelerometer and gyroscope sensors.

Normalization is basically a scaling technique or can be said to be a mapping technique or a pre-processing stage. Here, it is possible to find the new range from an existing one range. This technique is used popularly in predicting the future values for a dataset. This technique is used to maintain variation in the data and increase the variance of the dataset.

4. METHODOLOGY

The goal of this project is to teach a user the Object Oriented Concepts using Adaptive Augmented technology. For helping the user to learn, we implement an Open User model.

The entire application has been divided into following subsystems for the purpose of reusability and maintenance.

Application subsystem: This section holds the application code written in C# and all the dependent libraries

User Interface subsystem: Unity 3D has Scenes using which user can login to access the application and use accelerometer and gyroscope sensors to control the ball.

Presentation Layer: The objects included are 2D objects i.e. textbox, button etc. and 3D i.e. sphere, cube, pins etc.

Tracking system: update() function of MonoBehaviour class has the code for tracking part which logs all the information required for user model.

Context: SQlite database stores the context information and other info required for developing the user model.

World model subsystem: Information about real world is collected using normal as well as AR camera which is part of unity.

In the game, the user can control the ball which could be rolled. There are 12 pins placed around the ball. Each pin has an Object Oriented concept embedded inside. When the ball collides with a pin, that pin is collected, and the associated Java concept with the pin is displayed to the user. The user could learn the concept. The goal of the user is to collect all the 12 pins.

Open User Model is implemented to help the user to achieve his goal of collecting all the Object oriented concepts, and help him learn better. Using this model, the system is made adaptive to the user. We collect multiple implicit and explicit feedbacks of the user, and design the open user model and then make it adaptive to the environment.

The explicit feedback collected are User Id and login information of the user, and also we give an option to the user to switch between ball or cube for the game. By default ball is selected and if the user feels like he's not performing well, he could choose cube which slides slower than the roll of a ball. The implicit feedback collected are the speed of the ball, accelerometer values, orientation of the device etc.

Open user model is implemented using live feedback of the speed, accelerometer values and the progress of the user. This is discussed further in the next topic.

The user is taught Object oriented concepts explicitly by displaying the Object oriented concepts in the screen and also implicitly. The implicit learning is of inheritance. When the user selects a cube instead of a ball, this could be considered as inheritance. The user class is inherited to ball or cube based on the use case. Additionally, reducing the speed of the ball could be considered as method overloading. The default value of the speed of the ball is overwritten by passing a value half of the original value.

5. IMPLEMENTATION

In this project, we have implemented the Open User Model for teaching a user on the Java concepts using Adaptive Augmented technology. We're collecting different implicit feedbacks from the user. This includes the speed of the ball for a user, accelerometer values, position of the ball, and average count on the previous games. These data are collected to make the game adaptive to the user and also help him to collect all the Java tutorial pins assigned to the user.

The goal is to collect all the concepts embedded in the pin object, and by collecting these information we help the user to achieve this. For this purpose, we've implemented the Open User model to provide a status on the progress and performance of the user and also to complete the goal. The key components of an Open User model are Transparency, Controllability and Visualization. In our project, we've implemented these key concepts.

5.1. Transparency

The collection of related issues of addressing the learner's right of access to data about themselves, supporting control over their learning through greater control over their learner model and increasing trust in a system by making transparent the information used for adaptation, where all of these relate to broader principles of management of personal information[2]. Transparency helps the user to identify what data is collected and hence improving the trust of the user in the system.

In this project, we ensure transparency by displaying the average speed of the ball across X-axis, Y-axis and Z-axis collected as a live visualization along with the game. This helps the user to identify whether he's still in control of the game, or whether it's uncontrollable for him.

Additionally, the game is made adaptive by reducing the speed of the ball to 50% if the user can't control the game for multiple attempts. In this case, we display that information to the user, and making it more transparent.

5.2. Controllability

A user has control on what information to be shared and what information should not be shared in the model. This is called Controllability. This is important as there could be many information collected about the user and how he interacts with the model. The user should have

profound control on these data collected and should be able to identify the key things to be shared.

In our game, since we're not implementing the social aspect of the open user model, all the interactions of a user are shown to the current user and no information about another user would be displayed here.

In addition to that, the user would have control on selecting a cube if he's not performing well in the game. The user is given an option to choose for cube, which moves slower compared to the ball, and he's given the control to choose. The user has the control to share this information with other users or not.

Below is the sample snippet of the code to control the speed of the ball:

```
Vector3 movement = new Vector3(Input.acceleration.x, 0f, Input.acceleration.y);

rb.AddForce(movement * speed2 * Time.deltaTime);
```

5.3. Visualization

The third key feature of the Open User model is Visualization. It helps a user to identify the progress and how well he's performing. It also acts as a feedback on how he could improve the interaction. From a mere concept of visualizing the data collected, this key concept could be extended as a powerful tool to provide a holistic view on how a user could interact better with the space he's using.

There are two main streams of work on open student models. One stream focuses on visualizing the model to support students' self-reflection and planning; the other one encourages students to participate in the modeling process, such as engaging students through negotiation or collaboration on the construction of the model. Representations of the student model vary from displaying high-level summaries (such as skill meters) to complex concept maps or Bayesian networks. A range of benefits of opening the student models to the learners have been reported, such as increasing the learner's awareness of the developing knowledge, difficulties and the learning process, and students' engagement, motivation, and knowledge reflection explore interactive open learner modeling by engaging learners in negotiating with the system during the modeling process investigated active open learner models in order to motivate learners [1]. In our project, since we're focussing mainly on the self reflection and planning part of the visualization rather than the social adaptiveness of the project.

In this project, we show 5 key visualizations:-

1. Object Oriented Concept Collected: There are a total of 12 pins in the game and in each pin there's a Java concept embedded in the pin. When one pin is collected, we show what's the concept associated with the pin in the bottom of the screen as below:

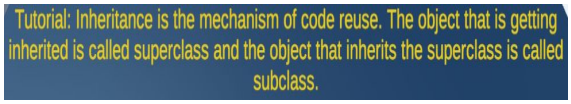


Fig 2: Sample Object Oriented Concept

User could read and learn this tutorial and then collect the subsequent pins.

2. Count: Count is displayed in the top of the screen and it just shows the progress on the number of pins collected.
3. Live Visualization on the Speed across X, Y, Z:

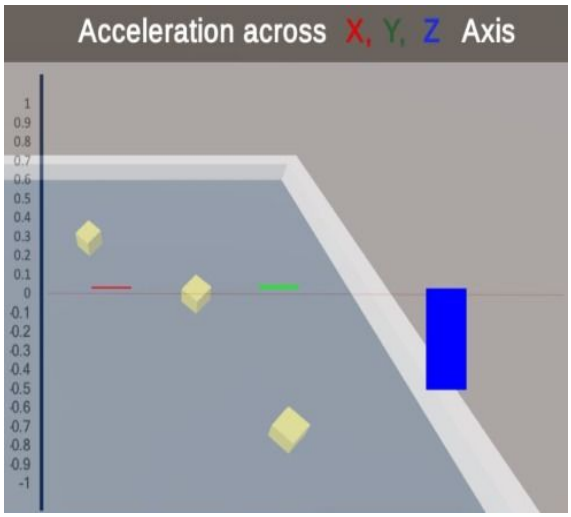


Fig 3: Live Visualization of speed of the ball

The data visualization is represented with the help of both line graph and bar graph displaying the average speed, hit count and live acceleration respectively. This is one of the important visualization used in the project. This shows the average speed of the user along X-axis, Y-axis and Z-axis. This engages the user and also alerts him if he's overspeeding across any axis. This is a live feedback which the user could take advantage of to improve his performance. As the area of the board is limited across X-axis and Z-axis could cautiously monitor this and align the game accordingly. They could also see which speed

increases spontaneously so they could understand which orientation of the mobile device is causing the issue. Additionally, if there pin to be collected are in a corner, using this adaptive feedback he could carefully roll the ball and collect the pin and read the object oriented concept embedded.

This feedback could also be used for future work to analyze a pattern using any regression technique to identify the best set of orientation and values where the user could perform the best, aligns the mobile device accordingly.

4. Average Speed in the Game:

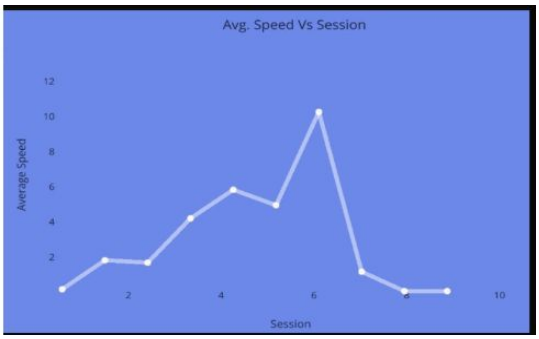


Fig 4: Average speed for last 10 games

From the values collected in speed across X,Y, Z axis throughout the game, we calculate the average speed of the ball in the game. Once the game ends, we display this information. The graph is plotted for the average speed of the ball for the previous 10 attempts.

This visualization is really important as it shows a comprehensive view on the user's speed with the system. We make the speed adaptive based on this statistics. If the speed peaked continuously for 5 consecutive attempts, it indicates that the user does not have good control on the ball. Then we reduce the speed of the ball is made 50% so he could play the game better and complete the goal of collecting the object oriented concepts.

A peak will be observed if the ball speed away to some boundary and cross them. The field of the game is placed in AR projection. This indicates that the orientation of the mobile device is not aligned with the plane of projection. Using this information, we reduce the speed of the ball so he could adjust and play the game better.

5. Average Hit Count

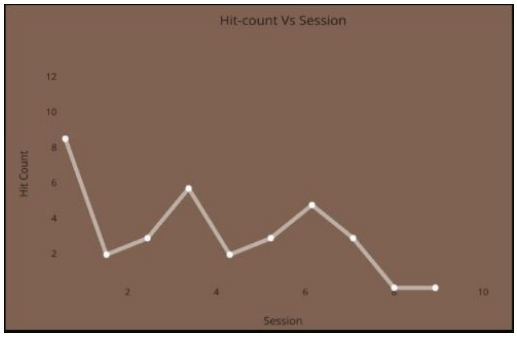


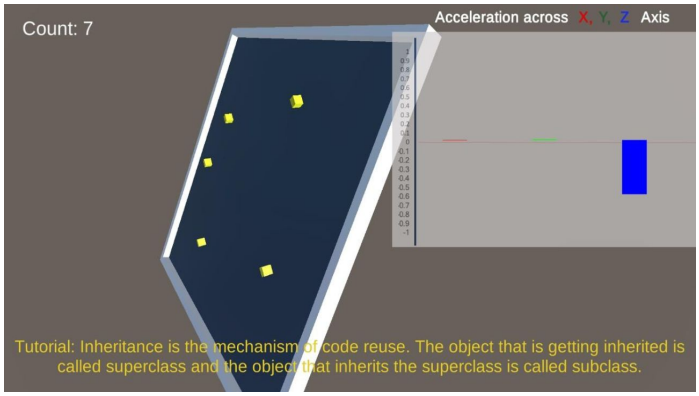
Fig 5: Average concepts learnt in last 10 games

6. This visualization shows the user's progress on the game. There are maximum 12 sprites representing the 12 recent hit count/ avg speed. The game also shows the pins collected in the last 10 games. This is a good indicator of the performance status of user.

   It's really important for future extension of the game as it clearly shows the user's status and combining this visualization with the live feedback or average speed, the system could learn the best orientation or speed to get the best results.

5.4. Augmented Reality

The image on which the game is projected by keeping the target image in the Vuforia database.



6. TECHNOLOGY USED

Software: Unity, Vuforia

Database: SQLite

Scripts: C#

Platform: Android (apk file)

Open User model: Visualization

7. OBSERVATION

We have done two adaptations in our system, the first one is related to the speed of the player and second is related to object change from sphere to the cube.

Using this system we implemented an open user model and are trying to give motivation to the user to learn Java programming concepts using an Augmented reality game which can be played using any general android phone. More the user gain yellow cubes, new java concepts are shown to the user, also since the ball keep adjusting its speed and adapt based on user behavior, the user doesn't lose interest in the game.

We can see the graph changing dynamically 5 times in every second. The change is done with the help of update () function which calls itself every 200 milliseconds. This is how the graph updates dynamically. As and when the orientation of the device changes, acceleration across the X, Y, Z axis are updated and same is being displayed in the graph.

There are line graphs showing Average and speed and Average Count of the entire slot of the game.

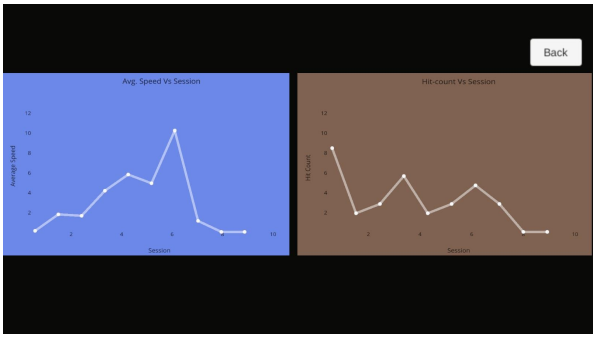


Fig6: Performance and session information

User has the capability of observing his live performance one the right panel of the screen where he can check his current performance. We also show the total number of cubes gained, speed of the player and results compared to his last session (using past history of the user). This provides a continuous engagement to the user and based on our observation users spend on average 19 minutes playing our game.

Because the system has the login capability, the user feel secure and his data is confidential. We encrypt the username, password and feature vectors, hence the system is well designed from the security perspective as well. Since the game is augmented on to an external object it provides an extended field of view to the user, in spite of he playing it on a small mobile screen, due to augmented reality he can experience a large field of view in the game, hence it keeps user engaged.

Major problems with the learning games in a mobile device is limited availability of view to the user. Another common issue is lack of adaptation available specific to the user. We tried to solve both these issues in this system. Using the live streaming data from the sensor, we try to improve user experience by adjusting speed and object specific to that user. Also, we not only provide performance charts to evaluate his past and current performance but also because we are using augmented reality, user don't realize that the game is been played in a small mobile screen.

8. EVALUATION PLAN

The most difficult piece of this project was advancing the game and make it keep running on the Android platform. In spite of the fact that the project concentrated on the advancement of an adaptive application, the game was created in the work area and just ported to the mobile platform in late phases of improvement. This choice was made so that there were no execution limitations that would result in features being removed or not implemented into the game. Nonetheless, there were various misfortunes that severely affected the advancement of the project. For instance, unity3D supports various emulation so that a game can be tested on a platform by running in the simulator. It was a hard time making sense of the correct simulator and approach to troubleshooting the code. Subsequent to encountering various accidents, designs issues and execution, the adaptive version of the game figured out how to accomplish a similar work area level experience just in a small and compact bundle. Our game fulfills every one of the criteria that were taught in class to emerge as a good adaptive application. We give transparency in results. We give intuitiveness and controllability in the application by enabling the users to see visualizations and learnings. At last period of testing, every individual from the group tried the game and gave singular feedback.

9. CONCLUSION

There are many projects to train a user in some concept using Open User model, but our project stand out because of the novelty approach. We've made the Object

oriented concepts training very interactive and interesting using the rollaball game. The game is also projected into AR space so the board for the game is in the projection and user could interact more with the model.

We've also incorporated all the key concepts of Open User model transparency, controllability and Visualization. Transparency is implemented using by displaying the speed information collected from the user. The speed across X, Y, and Z axis are collected and shown the user so he could have a control on the ball. By displaying the data collected from the user, he would have more trust on the model and would interact more. Controllability is implemented by ensuring the user has control on what information to be shared and also by selecting the ball or cube for the user. Visualization is one of the key feature for our project, where we implemented 5 important visualizations. These visualizations help the user to understand his progress and also the current performance in the game. This encourages the user to perform better in the current and next games. Different type of visualizations has been developed after a thorough understanding of their effectiveness and considering the audience of different levels of knowledge.

Using the Open User model and the data collected, we make the game adaptive to the user. If the user could not control the ball, the speed of the ball is reduced to 50%. This is done by analyzing the average speed across last 5 games. Additionally, the user is also given a control to switch to a cube if he still can not control the ball.

Using this game and open user model, the user would be taught 12 concepts of Object Oriented techniques explicitly. In addition to that, we also teach the concepts of inheritance and method overloading using this game.

10. FUTURE WORK

Just like the case for any framework with no restricted degree, our prototype to has some degree for extension and future work in a few zones. As far as the future improvement, planned features incorporate numerous game-levels with various objectives, the reintroduction of the first individual camera, moving ground units and the usage of the object framework for various ball models. A multiplayer can be executed later on to investigate social adaptive highlights of user model also. At the present time, the game was created for Android platform only, it tends to be likewise produced for IOS platform. Regarding the user model, our application can be reached out to consider implicit feedback or background knowledge so it can have earlier information about users past experience and accordingly learnings can be made more customized. The framework can likewise be made to offer some adaptability to users concerning how they might want to play the game by presenting user inclinations. Our application center around showing concepts of object-oriented concepts. It very well may be stretched out to numerous different zones also.

11. ACKNOWLEDGEMENT

We would like to extend our sincerest thanks to Dr. Sharon Hsiao for her direction and support all through the project. We truly value the helpful feedback given to us during the design phase of our application. Her insightful remarks helped us improve our application.

12. REFERENCES

[1]. Hsiao, I. H., Bakalov, F., Brusilovsky, P., & König-Ries, B. (2011). *Open social student modeling: visualizing student models with parallel introspective views. In User Modeling, Adaptation and Personalization (pp. 171-182).* Springer Berlin Heidelberg.

[2]. Bull, S., & Kay, J. (2007). *Student models that invite the learner in: The SMILI:() Open learner modelling framework.International Journal of Artificial Intelligence in Education, 17(2), 89-120.*

[3]. G. Adomavicius and A. Tuzhilin. *Recommender Systems Handbook, chapter Context-Aware Recommender Systems, pages 217–253. Springer US, Boston, MA, 2011.*

[4]. F. M. Harper, F. Xu, H. Kaur, K. Condiff, S. Chang, and L. Terveen. *Putting users in control of their recommendations. In Proceedings of the 9th ACM Conference on Recommender Systems.*

[5]. Bunt, A., Carenini, G., Conati, C.: *Adaptive content presentation for the web. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume*

[6]. Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval. Addison-Wesley (1999).*

[7]. Bull, S., Mabbott, A., & Abu-Issa, A.S. (2007). UMPTEEN: *Named and Anonymous Learner Model Access for Instructors and Peers. To appear in International Journal of Artificial Intelligence in Education.*

[8]. Benyon, D., Kay, J., & Thomas, R.C. (1992). *Building user models of editor usage. In E. Andre, R. Cohen, W.H. Graf, B. Kass, C.L. Paris & W. Wahlster (Eds.) UM 92: Third International Workshop on User Modeling.*

[9]. Gasparetti, F.: *Adaptive Web Search: User Modeling based on Associative Memory and Multi-Agent Focused Crawling. PhD Thesis, University of Roma Tre, 2005.*

[10]. Kelly, D., Teevan, J.: *Implicit feedback for inferring user preference: a bibliography. ACM SIGIR Forum 37(2), 18–28 (2003)*

[11]. Asa MacWilliams, Thomas Reicher, Gudrun Klinker and Bernd Bruegge, *"Design Patterns for Augmented Reality Systems"*

[12] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier and Blair MacIntyre, *"Recent Advances in Augmented Reality" – 2001*