A Webshop for Digital Assets in Virtual Worlds Supported by a 3D Object Representation.

Michael Erwin Steurer
Institute for Information Systems and Computer Media
Graz University of Technology, Austria
michael.steurer@iicm.tugraz.at

***Abstract*—Virtual worlds became very popular over the past years and users spend more and more time in it. One of its main characteristics is the creation of content and a working economy that allows users to buy, sell, and even trade these objects. Existing multi-vendor platforms for virtual items lack of user friendliness for customers and merchants. In this paper we present the design of a multi-vendor webshop for virtual items enhanced by a 3D representation within the virtual world. Merchants can directly add items from their avatar's inventory to the webshop and promote it with additional meta-information. Customers can choose items from the webshop and additionally get a 3D preview prior to buying them. With the provided solution merchants do not need in-world stores to offer their objects because they are directly added to the webshop from their inventory. Customers can find products at a centralized webshop to search, compare and rate objects but use existing in-world mechanisms to buy them. The presented design has been implemented as a prototype to prove the concept.**

***Keywords*-3D Virtual Worlds, Selling Platform, OpenSimulator, Second Life, Open Metaverse**

I. INTRODUCTION

Virtual Worlds as depicted in Figure 1 are computer generated three dimensional environments and where first mentioned in Neil Stevenson's novel "Snow Crash" in 1992 [1]. The human representation within this environment (*i.e.*, residents of the virtual world), are referred to as avatars. In contrast to Massively Multiplayer Online Games virtual worlds are not quest oriented but focus on creativity, user interaction and are open-ended [2], [3]. According to Sivans 3D3C properties a 3D world simulates the real world, allows avatars to create social networks, create content, and trade with objects [4]. The virtual world tries to imitate the real world but avatars have additional capabilities like flying or teleporting to far distanced places. They can move around in this virtual world freely and interact with other avatars by using text chat, voice chat and limited forms of gestures. Users can create objects within the virtual world that base on primitive shapes like boxes or cylinders. Objects can either embed other objects or simple scripts to do autonomous tasks (*e.g.*, send embedded items to the avatar touching the object). Avatars store objects they own in their inventory. It is organized by folders and can not be accessed by other avatars. Users can "rezz" (in virtual world's jargon)

Figure 1. A typical scene from the virtual world of Second Life.

objects from their avatar's inventory which means that they make the objects appear in the virtual world. A requirement for building and creating objects is the permission of the landowner. [5]

An important factor for economy in a virtual world is a virtual currency that can be exchanged to real currency [6]. Virtual money like the Linden Dollar (L$) for Second Life or the Open-Metaverse-Cent (OM¢) for OpenSimulator based virtual worlds is directly bound to real currency. Therefore, avatars can buy, sell and even trade with virtual objects for real money [7].

Todays most successful virtual worlds are Second Life (www.secondlife.com) with 800.000 average monthly repeated logins, OpenSimulator (www.opensimulator.org) based virtual worlds with over 13.000 active users in January 2011 and Habbo Hotel (www.habbo.com) with 18 million unique logins per month [8], [9], [10], [11].

Most virtual worlds are designed as server-client architecture and users have to download a special viewer software to connect the servers. To access a virtual world users register and provide username and password. After logging in with these credentials the viewer software renders and displays the virtual world on the user's computer and provides an interface for interaction with mouse and keyboard.

The virtual world of Second Life is maintained by Linden Labs but due to it's closed source servers developers can not add any functionality. Basing on the communication protocol between the world of Second Life and the viewer

 ISBN: 978-1-61208-139-7

software a project named OpenSimulator project was founded in 2007 by Darren Guard. It is an open source 3D application server with an extensible and modular architecture. Every application server represents a particular area of virtual land, also referred to as *region*, that form a virtual world if connected to each other. Due to this network structure a virtual world it is also referred to as *grid*. Due to the modular design and the open source of the application server developers can implement new modules and extend it's capabilities. The framework presented in this paper can be applied to other open source virtual worlds but we focus on OpenSimulator based virtual worlds.

The remainder of the paper is organized as follows. In Section II we give a overview of existing selling platforms that are either web based or completely in-world. Section III describes the usage of the system and lists the necessary requirements. Basing on this Section IV points out the design and the prototypical implementation of the webshop. Finally, Section V concludes the paper and gives ideas for further work.

II. Existing Selling Platforms

In open ended virtual environments with a high degree of freedom and user interaction vanity has a large influence on the behavior of users [12]. They want to be entertained when spending time in-world and a lack of usability or complex usage can prevent users from buying and selling objects [12]. A presentation by the founders of Second Life states that 75% of all Second Life users were buyers and 25% of them were sellers[13]. Due to this it's even more important to bring the ease of use to both parties. In 2010 Second Life had on average 485.000 monthly economy participants. This number are residents with at least one activity on their virtual money account. The sales volume of their webshop was approximately 2.6 Million Euro in the fourth quarter of 2010 [9].

Merchants can offer their virtual items either in in-world stores or in webshops that are connected to the virtual world.

A. In-World Stores

In order to sell items merchants have to rent a salesroom to present their products. See Figure 2 for an in-world store that offers flowers in the OpenSimulator based virtual world of *German Grid*. Depending on the location of the salesroom the costs for renting can vary. The prices depend on the avatar traffic at the particular location and supply and demand of stores. Merchants with a small portfolio of products hesitate to invest in a such a shop because of the little income if compared to the effort to decorate and promote these items. Although items can be modified in a limited way through object-embedded scripts in-world stores lack of automated mechanisms for modification. Objects are

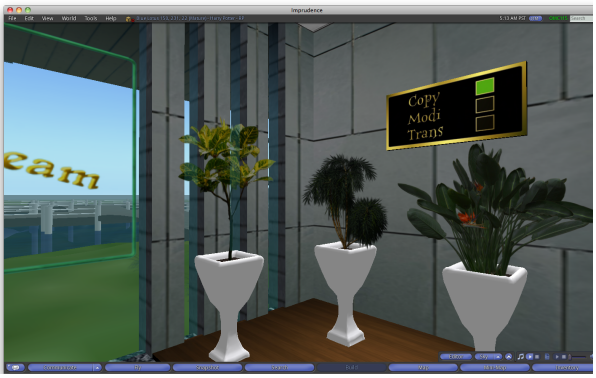

Figure 2. In in-world stores all objects are already present in the virtual world and can be bought by customers by touching the object.

created statically in the store and a users have to delete them manually. Customer buy object in in-word stores by clicking on the desired item and use grid internal mechanisms for payment and consignment. All bought objects are immediately delivered to the avatar's inventory.

Shops in virtual worlds are quite similar to shops in the real world. Avatars have to visit them to see and finally buy items. There is no automatic recommendation system but customers are attracted by other items in the shop.
Searing for items is important for a positive shopping experience but in OpenSimulator based virtual worlds there are no mechanisms to search for sellable objects or items [14], [3]. Interested users can only walk or fly through the virtual world and find objects and items accidentally. The combination of a missing search function and a locally limited market make the comparison of similar objects nearly impossible. [12] and [15] focus on these in-world stores and present studies about the buying behavior of users within a virtual world.

In-world stores are more similar to real world shops and users immediately see which objects they buy. Unfortunately, these stores also have drawbacks for customers and merchants.

B. External Webshops

There are several webshops for items in virtual worlds. SL Marketplace (http://marketplace.secondlife.com) for Second Life, and GridBay (http://www.gridbay.de) and Avatar Marketplace (http://avatarmarketplace.com) for Second Life and OpenSimulator based virtual worlds. As depicted in Figure 3 merchants put the items $m_{1..n}$ into an in-world object (also referred to as "magic box") that acts as a vault and all items in it are indexed by the webshop, $index(C_M, m_{1..n})$. Merchants immediately see their added items on the webpage and can add a more detailed textual description, some photos and extra keywords for the search engine.

 ISBN: 978-1-61208-139-7

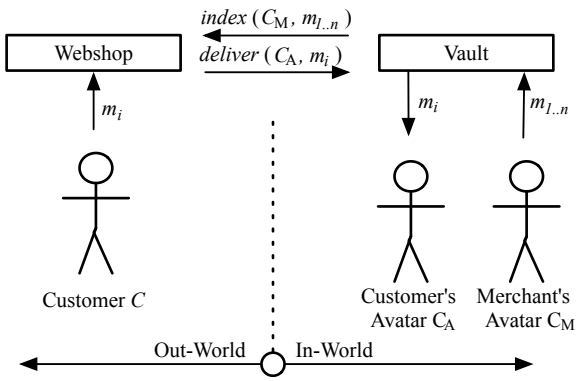

Figure 3. In existing webshop Merchants put items $m_{1..n}$ into an in-world vault which are automatically added to the webshop. A customer buys and object m_i from the webshop and the request to the vault initializes the consignment.

Customers put the desired items m_i from the webshop into a virtual trolley and do the checkout. All payments are made on the webshop's side in order to deliver the objects. As all items remain in-world the webshop sends a request with identifiers for avatar and bought objects to the vault to initialize the consignment of the objects, *deliver*(C_A, m_i) [16]. This requires the webshop to maintain a separate payment system but on the other hand it can serve different virtual worlds with different currencies.
External webshops benefit from well known techniques like browsing through items, filtering, or even comparing them but only provide the meta-information to their customers because the actual object remains in the virtual world. A common technique to motivate customers for further purchases are recommendation system that base on other users buying behavior.

Webshops are well known by users and they gain more information if compared to the plain 3D object. Webshops can provide recommendation systems for similar objects and customer reviews from other customers.

III. Functionality and Requirements

The basic idea of this paper is to describe a webshop enhanced by the benefits of an in-world store.

A. Usage and general description

Merchants have to register their avatar with the webshop and connect an in-world avatar to the account. After logging into the webshop they are provided with a list of objects retrieved from their avatar's inventory and can select the objects to be added to the offered items in the webshop. Due to security and privacy reasons merchants can only access a predefined subfolder in their inventory and are only allowed to add self-created objects to the webshop. To promote objects merchants can add additional product meta-information for the customers. The selling price is taken from the in-world properties of the object and can not be changed.

To gather data about the buying behavior and to improve the shopping experience customers are also required to register with the service and connect an avatar to this account in order to buy items. Then they can browse the webshop to find items upon the given meta-information or keywords. In existing webshops customers put their items in a virtual trolley and checkout to complete the purchase. In the proposed solution all items added to the trolley are automatically rezzed from the merchants inventory to the virtual world. The avatar that is connected to the actual account gets a notification with the accurate location of the rezzed objects and teleport to this location to inspect the created object. From this point the object behaves just like an object from an in-world store and customers can use the same mechanisms to buy the object. The purchase is initialized in-world and the entire payment and consignment mechanism is done in-world. Figure 4 gives an overview of the functionality of the webshop.

B. Requirements

The functionality of such a selling platform can be tracked down to the following requirements:

- Ease of use for merchants. Merchants are able to access their inventory from the webshop to select items to be listed on the selling platform and do not need a salesroom or vault in the virtual world. They can promote their objects by adding keywords, a better description, and pictures.
- 3D representation. Customers get a 3D preview of the objects within the virtual world prior to buying them with in-world mechanisms.
- Meta-information. Besides the three dimensional representation customers can retrieve more information about objects. This includes a detailed textual description, pictures, and information about the creator.
- Payment system and consignment. All purchases are done in-world. There is no external payment and consignment system but existing in-world mechanisms to transfer the money and send objects to the customers are used instead.
- Multivendor search and recommendations. Customers can search the entire webshop upon the item's names, their description and the set keywords. For every item the customer is provided with with a list of similar and related objects in the webshop.
- Security and trust. Merchants can define the permissions of the objects to be rezzed as a users's preview. Further, all objects created for customers have a certain time to life. If users do not buy the items they are

 ISBN: 978-1-61208-139-7

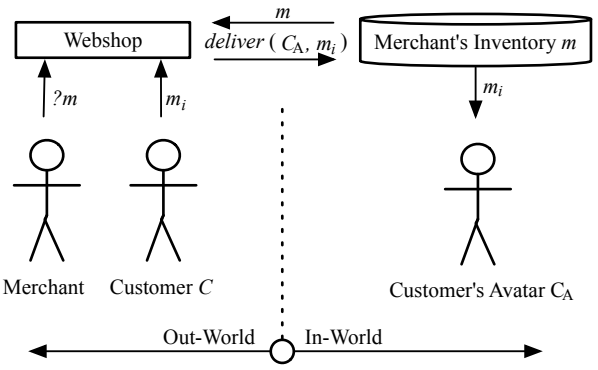

Figure 4. Merchants can request their avatar's inventory items m through the webshop ($?m$). A customer buys an object m_i and the webshop directly accesses the merchant's inventory to consign the actual object, $deliver(C_A, m_i)$.

periodically purged from the virtual world. Instead of deleting these objects they are return to the merchants inventory to prevent from data loss.

To demonstrate all these features we have implemented a prototype selling platform.

IV. Implementation

Users can interact with the virtual world using the client viewer. It detects user actions and sends the according requests to the application server (*e.g.*, rezz objects from the avatar's inventory in the virtual world and delete them). The design of the proposed solution has the same client-server architecture but the graphical client viewer is replaced by the webshop. Instead it bypasses the standard login procedure with the avatars standard log-in credentials and directly connects to the 3D application server. Hence, the webshop can send requests to the application server upon a user's request. As depicted in Figure 5 the webshop provides a web interface for users and sends these requests to the application server.

A. OpenSimulator Module

OpenSimulator based virtual worlds are a network of connected application servers and developers can extend a simulators functionality by adding modules. To provide this additional functionality to the entire virtual world every application server needs the module. It can detect requests sent by the webshop, extract the passed parameters and process them.
The developed module has an XML-RPC interface but does not support encryption. Hence, all requests sent to the module are not protected against wiretapping and forgery. For a prove of concept the unencrypted communication is sufficient but the productive use requires a more secure protocol as described in [2]. The module processes the XML-RPC requests and responds data in JSON format. In the prototypical implementation there are request to get a user's inventory, rezz items from the inventory to the virtual world and also delete these objects from the virtual world. In the following we describe it's basic functionality:

1) Access Inventory: Avatars as well as items in the avatar's inventory are identified by a unique identifier. Users can specify wether these objects are sellable, the price for the customers, and the permissions for the next owner. An XML-RPC request to get the list of an avatar's inventory contains the identifier of the avatar and replies with the data in JSON format. This response includes the folder structure of the entire inventory, unique identifiers, names, description, and the permissions of the objects. Due to privacy reasons merchants can specify a subfolder for the request to limit the response.

After fetching the inventory merchants can specify which items to be listed on the webshop. Hence, all items in the webshop are linked to virtual objects in the inventory of an avatar. If a merchant deletes one of the listed items in the avatar's inventory but not in the webshop the link is broken. To prevent from this scenario and to keep the items in the webshop up to date there is an additional request to the avatar's inventory. It contains identifiers for the avatar and a certain item in the inventory to check if it still exists. This request is executed periodically to purge all zombie records in the webshop.

2) Create and Delete Objects: Existing client viewer allow users to rezz items from their avatar's inventory only if the region owner allows it and the introduced framework does not bypass this system. After retrieving the list of inventory items users can request the 3D application server to make an object from an avatar's inventory appear in the virtual world. Besides the identifiers for the avatar and the particular item this request also requires information about the location of the new object. This location can be chosen freely but requires the user to have sufficient privileges to create items. The object appears in-world as if it was directly rezzed from the merchant's inventory by using the client viewer.

De-rezzing an object requires the identifier of the actual item and the creator of the object. In order to prevent from data loss we do not delete the object directly but move it automatically into a folder of the merchant's inventory.

3) Notify avatars: To inform avatars about rezzed objects we can send message to them. The parameters passed with the request are an identifier to specify the avatar and the actual message.

 ISBN: 978-1-61208-139-7

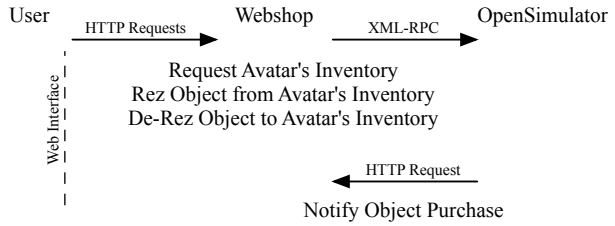

Figure 5. Users access the webshop by web interface and can request their inventory, rezz objects from the inventory and also de-rezz these items. After a successful purchase the simulator sends a notification to the webshop.

B. Webshop

It contains the database and the control logic of the entire architecture. Customers can access the webshop through web-interface and the webshop can access the OpenSimulator module through XML-RPCs. In order to use the service, both customers and merchants have to register with the webshop and link their avatars to the actual account. The registration requests the user to enter a username, a password and the name of the related in-world avatar. The webshop creates a secret confirmation PIN and sends it as private message to the specified avatar within the virtual world. If the name of the avatar is correct and the user has the credentials to access the avatar's inbox it can complete the registration and enter the secret PIN in the webshop's registration form. This mechanism proves the existence of the avatar and confirms the link between webshop account and in-world avatar [17]. Both account types are identical and so merchants can buy objects and vice versa.

Users have to log in and can either access their inventory to add items to the webshop or browse the existing catalogue. After fetching the inventory the merchant is provided with information of the avatar's inventory (*i.e.*, name, description, permissions), and can add these items to the webshop. To promote them users can add meta-information to the selected items that is stored locally in the webshop's database but directly linked to the inventory items. Meta-information can contain a more detailed description, keywords for classification, and some pictures for the customers to get a first glance of the offered items.

We have implemented a simple search function that bases on the keywords of the listed objects. Hence, customers can search the webshop for desired items, get a list of matching items choose objects from the list upon the name, detailed description and pictorial information.

There is no checkout as in existing webshops but a customer can initialize to rezz the object in-world instead. To do so, the webshop sends a request to the OpenSimulator module with identifiers for merchant and actual item. No matter whether the merchant's avatar is logged in or not the item appears at a certain position in-world. To inform the customer about this location an instant message is sent to it's avatar with the detailed information. Now, the customer logs into the virtual world to access the message and locate the rezzed object. The customer can either buy the objects in-world with in-world purchase mechanisms or continue shopping. If the customer decides to continue shopping and select a new object to be rezzed in-world the webshop requests the simulator to remove the old object and replace it with the new one. Otherwise the rezzed object will be automatically purged after a certain period of time again by webshop request. The OpenSimulator module can not only detect XML-RPC from the gateway but also trigger on object purchases. As a customer decides to buy an in-world object the module detects this event and sends a HTTP request to the webshop. With this method the webshop detects all purchases and obtains additional information for customers (*e.g.*, recommendation systems).

V. Results and Conclusion

Existing solutions that list virtual 3D objects in a 2D webshop are complicated to use. The proposed idea has the following benefits if compared to existing solutions:

- Registration Process. The registration process proves the link between the webshop account and the in-world avatar. This is required in order to access an avatar's inventory to add items to the webshop and to inform customers about rezzed objects.
- Access Inventory. Instead of "magic boxes" users can easily access their inventory from the webshop to offer items. They can add additional meta-data to give customers a more detailed description of the items.
- Purchasing Objects. Customers can employ all features of a webshop to search and compare items. The use of keywords and knowledge about previous payments can be used as a recommendation system.
- Consignment and Payment. The webshop just provides meta-information about the objects and the actual object remains inside the virtual world. Within the webshop users can request the selected object to be rezzed in-world to use in-world mechanisms for purchasing objects. This implies that the webshop does not need a separate payment module.

The complete framework of the merchant system has been implemented as a prototype without any security measures to prevent a users privacy. Obviously, this system can not be used for productive use but is a simple proof of concept.

For future work we will add the necessary functionality for security and test the shop with a limited number of merchants.

 ISBN: 978-1-61208-139-7

REFERENCES

[1] N. Stephenson, *Snow Crash.* Bantam Books, 1992.

[2] F. Kappe and M. Steurer, "The open metaverse currency (omc) - a micropayment framework for open 3d virtual worlds," in *EC-Web*. ACM, 2010.

[3] M. Q. Tran, "Understanding the influence of 3d virtual worlds on perceptions of 2d e-commerce websites," in *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems - EICS '10*, 2010.

[4] Y. Sivan, "3d3c real virtual worlds defined: The immense potential of merging 3d, community, creation, and commerce," *Journal of Virtual Worlds Research*, vol. 1, no. 1, 2008.

[5] M. Rymaszewski, W. J. Au, M. Wallace, C. Winters, C. Ondrejka, B. Batstone-Cunningham, and P. Rosedale, *Second Life: The Official Guide*. Alameda, CA, USA: SYBEX Inc., 2006.

[6] H. Yamaguchi, "An analysis of virtual currencies in online games," http://ssrn.com/abstract=544422, 2004, [accessed on 19-April-2011].

[7] P. Crowther and R. Cox, "Building content in second life issues facing content creators and residents," in *Computer-Mediated Social Networking*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009.

[8] Sulake, "Habbo hotel – where else?" http://www.sulake.com/habbo/, 2011, [accessed on 19-April-2011].

[9] Linden Lab, "The second life economy in q4 2010," http://blogs.secondlife.com/community/features/blog/2011/01/26/the-second-life-economy-in-q4-2010, 2010, [accessed on 19-April-2011].

[10] M. Korolov, "June 2010 opensim grid statistics," http://www.hypergridbusiness.com/statistics/june-opensim-grid-statistics/, 2010, [accessed on 19-April-2011].

[11] M. Korolov, "December 2010 opensim grid statistics," http://www.hypergridbusiness.com/statistics/december-opensim-region-counts/, 2010, [accessed on 19-April-2011].

[12] Y. Guo and S. Barnes, "Virtual item purchase behavior in virtual worlds: an exploratory investigation," *Electronic Commerce Research*, pp. 77–96, 2009.

[13] S. Papagiannidis, M. Bourlakis, and F. Li, "Making real money in virtual worlds: MMORPGs and emerging business opportunities, challenges and ethical implications in metaverses," *Technological Forecasting and Social Change*, pp. 610–622+, 2008.

[14] OpenSimulator, "Opensimsearch," http://opensimulator.org/wiki/OpenSimSearch, 2011, [accessed on 19-April-2011].

[15] P. R. Messinger, E. Stroulia, K. Lyons, M. Bone, R. Niu, K. Smirnov, and S. Perelgut, "Virtual worlds past, present, and future: New directions in social computing," *Decision Support Systems*, vol. 47, no. 3, 2009.

[16] M. E. Steurer and F. Kappe, "A micropayment enabled webshop for digital assets in virtual worlds," in *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*, ser. MindTrek '10. ACM, 2010.

[17] C. Trattner, M. E. Steurer, and F. Kappe, "Socializing virtual worlds with facebook: a prototypical implementation of an expansion pack to communicate between facebook and opensimulator based virtual worlds," in *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*, ser. MindTrek '10. ACM, 2010.

 ISBN: 978-1-61208-139-7