

Learning Diary: Human Computer Interaction in Virtual Worlds and Mobile Commerce

Topic: Second Life / Virtual Worlds

Tobias Glocker

University of Vaasa

Tobias.Glocker@uwasa.fi

Student Number: p87915

June 27, 2011

1 INTRODUCTION

Nowadays Virtual Worlds are becoming more and more popular since they enable technologies for the business models and practices in the future. The virtual world, Second Life, gives the opportunity of sharing ideas in the information economy. Furthermore Second Live plays an essential role in the futures education. We will be able to participate in lectures at Universities all around the world. Besides the advantages, displaying a virtual world requires many graphical computations and thus a powerful hardware on the server and client side is needed. In addition an Internet connection with a high bandwidth is essential to achieve a smooth movement in the virtual world.

2 SECOND LIFE

In Second Life users participate in a virtual world as avatars. Avatars have the capability to fly and to teleport. A lot of attention was paid to Second Life due to its embrace of open source software that give users the possibility to design their own environments and virtual goods. Some users have become professional entrepreneurs in Second Life and sell virtual goods, services, furnitures and legal advices. Users can buy goods and services with Linden dollars ($L\$250 = 1$ US dollar) or with real world currency. Second Life has become also popular in the public sector. Sweden, for example, "has created a virtual embassy in Second life". Some schools and the Harvard University offer courses in Second Life. Kim, Lyons & Cunningham (2008) categorize "immersive business innovations into the following three classes: immersive prototyping, immersive event simulation and immersive commerce". Immersive prototyping can be useful for getting new ideas or for trying out new ways to achieve improvements. For example Starwood hotels have developed a Second Life version of their hotels in order to see what spaces and furnitures are attractive for their guests to equip their hotels in real life with furnitures, people liked in Second Life. Immersive event simulation is helpful for simulating real-worlds events in Second Life. Emergency response worker could be educated in Second Life by simulating cases of national crises. The immersive commerce "provides opportunities for companies and individuals to engage in commerce services in Second Life". As already mentioned virtual goods can be exchanged for Linden dollars. The purpose of the immersive commerce is to "increase commerce activity in real life". (Kim et al. 2008.)

3 EVALUATING IMPLICATIONS OF VIRTUAL WORLDS ON SERVER ARCHITECTURE USING SECOND LIVE

To reach a high graphical refresh rate when walking through the virtual world, a powerful hardware on the client and on the server side is needed. According to Makineni, Tickoo, Terell, Young & Newell (2009) "Virtual worlds can be broadly characterized into two types". One type is Massively Multiplayer Online Games (MMOG) such as World of Warcrafts, Club Penguin and so on. The other type of virtual worlds have no focus on playing games but "more on interaction, socialization and commerce". In comparison to the first mentioned type where the "simulated world is mostly built by game developers", this type allows the users to build their

own virtual world according to their wishes. To this type belongs Second Live, Olive etc. One of the most famous virtual world is Second Life developed from Linden Lab. To simulate this virtual world more than 2000 multi-core servers are necessary. More than 75000 users are active in Second Life at any time. Second Life requires a well structured server architecture to handle all the users. A Second Life server architecture is illustrated in **Figure 1** and consists of a simulator, map server, asset server, space server and a data server.

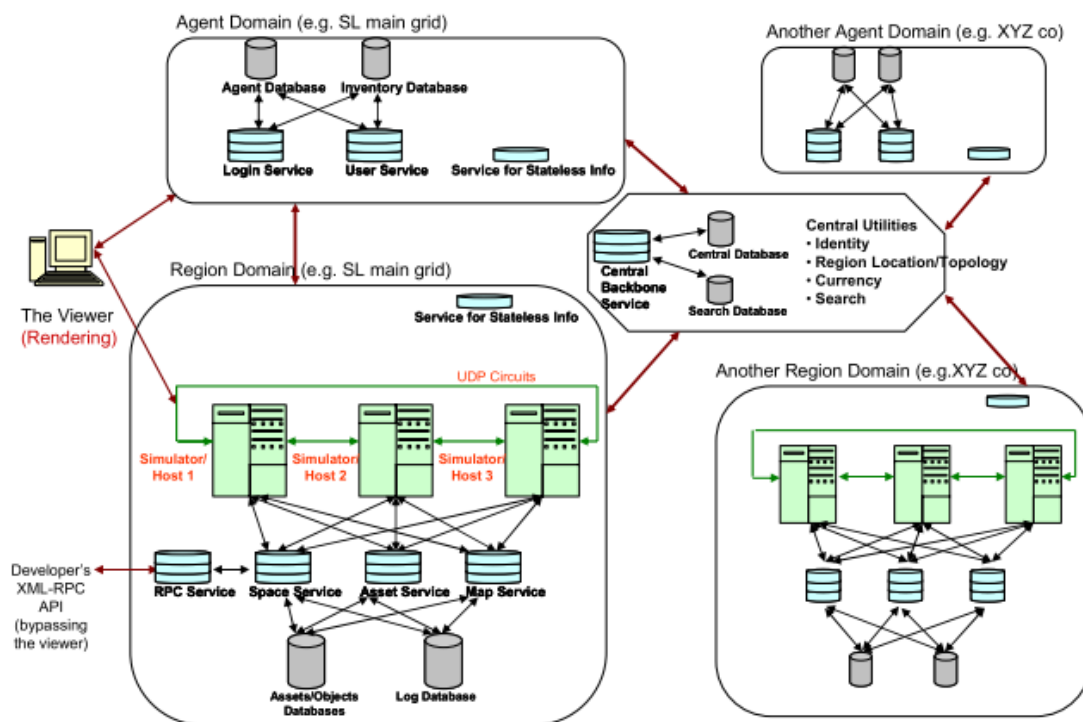


Figure 1. Server Architecture in Second Life. (Makineni et al. 2009.)

The simulator is mainly responsible for simulating the virtual world. "This service (process) is responsible for storing object state, terrain information and performing visibility calculations to send to each connected viewer". It is the so called workhorse of the Second Life server. In addition the simulator process has also the duty to talk to other services and to the database server whenever needed. The map server contains the information about "the entire world being simulated". As soon as the user clicks the map button, the map server sends information about active regions and the number of users that are active in this region. An asset server handles the asset information (each object in Second Life is treated as an asset). To this duty

belongs also the management of each avatar’s inventory like clothes, shoes, built objects and so on. The space server keeps track of ”which regions are where on the grid” and it maintains the neighbor information as well. A data server takes care of the connections to all active databases like log, inventory, central etc.

To evaluate the performance of Second Life, Makineni et al. (2009) built a network containing a Second Life Server and several client machines with a Nehalem (Core i7) processor. The test was done with two different processor types on the server side. In a Clovertown processor based platform there are two 4MB L2 caches that are shared by two cores. Nehalem processor based platforms support Hyper-threading, meaning that ”each core has two logical processors (hardware threads) that share the core resources”. Moreover each core in a Nehalem processor has its own 256KB L2 cache. An 8MB L3 cache is shared among all 4 cores. Makineni et al. (2009) focus ”on 3 very important aspects of the simulator (sim) process, which are physics processing, script processing and avatar handling”. **Table 1** gives a rough overview on what the 3 aspects of the simulation are focused. (Makineni et al. 2009.)

Table 1. A rough overview on what the 3 aspects of the simulation are focused.

Aspect	Focus
Physics	”measuring performance of physics engine and interactions between it and the sim process”
Scripting	”measuring scripting engine’s performance inside Second Life sim process”
Avatars	”sim’s ability to process avatars and their movement around the region”

An interesting result shown in the paper written by Makineni et al. (2009) is the performance on a Nehalem processor versus a Clovertown processor. For the comparison of the performance Makineni et al. (2009) loaded 4 sims on each system. No Hyper-threading was used. **Figure 2** illustrates the Nehalem processor efficiency over a Clovertown processor. A time per frame metric for 5 different scenarios shown on the x-axis is used to compute the efficiency. As it can be seen from the graph, a Nehalem processor based system is 40 to 50% more efficient, except for 1 sim, 5000 cubes scenario. The reason is that a Nehalem processor has a much better cache management that leads to the advantage that the misses per instruction and the

cycles per instruction are much lower than on a Clovertown processor. This result clarifies, what an important role the hardware architecture plays.

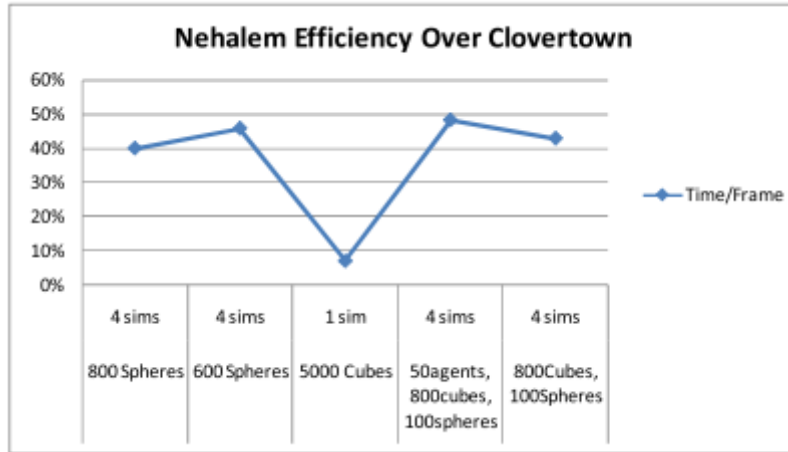


Figure 2. Nehalem benefit for Second Life - Time/Frame. (Makineni et al. 2009.)

Another remarkable result in the paper of Makineni et al. (2009) is the achieved improvement by using Hyper-threading. In the previous Second Life simulation process Hyper-threading was not used, meaning that maximum 4 sims per CPU were possible. When Hyper-threading is enabled, two hardware threads in each core run. Both hardware threads share the same resources (execution ports, cache, etc.) of the core. **Figure 3** shows the results of the Second Life simulation process "where each sim is handling 50 avatars moving randomly around the region, 800 cubes rotating and changing color and 100 spinning spheres".

	4 sims	8 sims
	4 cores	8 threads
CPU	87%	100%
Cycles/100Inst	149	219
L2 Req/10KI	239	366
L2 Miss/10KI	105	164
LLC Miss/10KI	43	59
Fr/Sec	71	43

Figure 3. Simulation without and with Hyper-threading - 50 Agents, 800 Rotating Cubes, 100 Spinning Spheres. (Makineni et al. 2009.)

It can be seen that running 8 sims slow down each sim significantly but there is still one advantage that can be reached with Hyper-threading. The comparison of the frame rate shows that 8 sims deliver 43 frames/second/sim while 4 sims deliver 71 frames/second/sim. In total 8 sims generate 344 frames and 4 sims generate 284 frames. Hyper-threading offers 20% more frames/second but it is not very beneficial when sims are heavily loaded.

4 BUILDING VIRTUAL WORLDS WITH THE VIRTUAL REALITY MODELING LANGUAGE (VRML)

The Virtual Reality Modeling Language is often used when "virtual worlds incorporating 3D shapes, light sources, fog animation, and even sound effects" need to be built. A virtual world can be "described by one or more VRML world files with a .wrl filename extension". For executing world files a VRML browser is needed that interprets the world file and draws the virtual world "into a rectangular region".

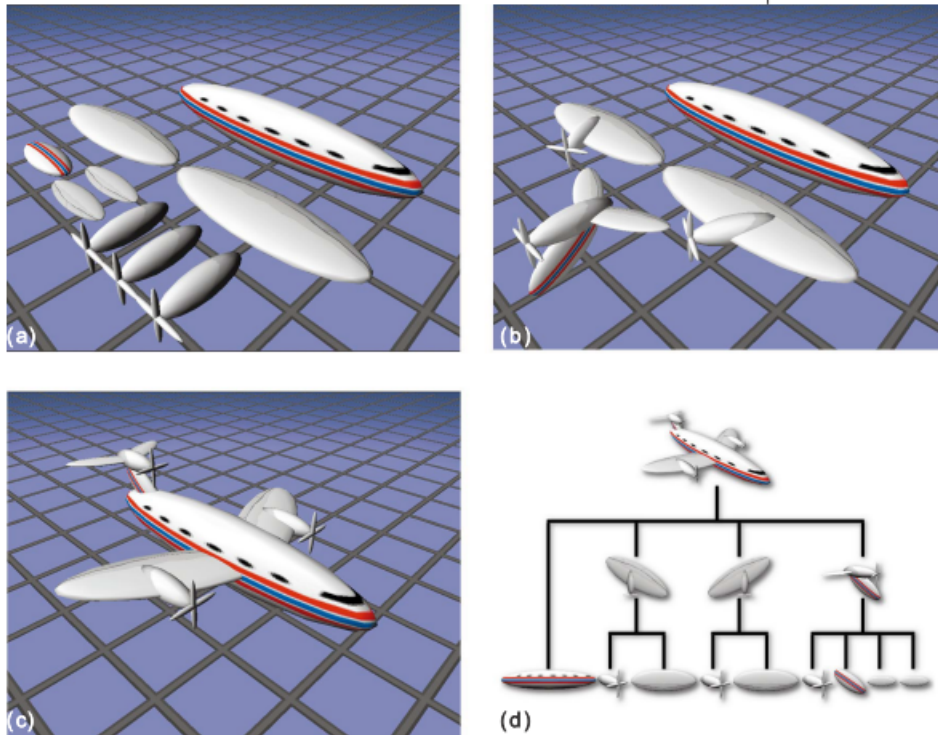


Figure 4. Building an airplane using a scene graph. (Nadeau 1999.)

”Scene data within a VRML file are organized into a scene graph, a hierarchy of groups and shapes arranged like a family tree”. The shapes, lights, sounds, and so on are called the children in the scene graph. **Figure 4** illustrates how an airplane can be built by using a scene graph. **Figure 4a** illustrates the separate parts of the airplane. **Figure 4b** shows the assembly for each wing and its engine as well as for the tail and engine. In **Figure 4c** all parts are assembled together. **Figure 4d** presents the scene graph diagram of this hierarchical grouping.

```
#VRML V2.0 utf8
# Build a 3D cylinder
Shape {
  appearance Appearance {
    material Material {
      diffuseColor 0.0 0.5 1.0
    }
  }
  geometry Cylinder {
    height 2.0
    radius 1.5
  }
}
```

Figure 5. VRML example code.

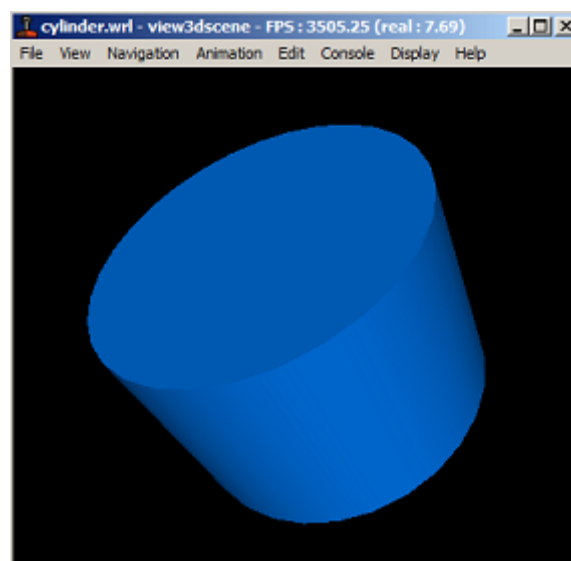


Figure 6. Building a blue cylinder with VRML.

The parents and children within a VRML graph are VRML nodes - "building blocks for creating a VRML world". "VRML nodes construct 3D shapes, position the user's viewpoint, aim light sources, define animation paths, group shapes together, add fog to the environment, position sounds, and so forth". As it can be seen from the VRML example presented in **Figure 5**, the VRML syntax is relatively easy. It requires 13 lines of code to describe a 3D cylinder shape with a specific height and radius. In this example there are 4 VRML nodes - Shape, Appearance, Material and Cylinder. "The Cylinder node has height and radius fields; the Material node has a diffuseColor field, and so on". **Figure 6** shows the graphical representation of the VRML example file. (Nadeau 1999.)

5 CONCLUSION

Virtual worlds are growing faster and faster due to the amazing user experience. The more users join the better needs to be the server architecture of the virtual world. At the moment, 75000 users are active in Second Life at any time but in the future there will be more users since the users capabilities are going to be enhanced. Also the education will enrich the virtual world. People can participate in lectures all around the world. Furthermore the commerce will play an essential role. In the virtual world people will be able to design the goods that they need in the real world. The virtual world can simulate real accidents in order to educate emergency workers. According to my opinion the virtual world will bring us so many advantages that we can't imagine to live without it.

6 BIBLIOGRAPHY

- Kim, Henry M., Kelly Lyons & Mary Ann Cunningham (2008). Towards a Framework for Evaluating Immersive Business Models: Evaluating Service Innovations in Second Life. 1530-1605/08 ©2008 IEEE.
- Makineni, Srihari, Omesh Tickoo, Aaron Terrell, Jessica Young & Donald Newell (2009). Evaluating Implications of Virtual Worlds on Server Architecture using Second Life. 978-1-4244-4921-7/09/ ©2009 IEEE.
- Nadeau, David R. (1999). Building Virtual Worlds with VRML. 0272-1716/99/ ©1999 IEEE.