Virtual World Interoperability of Avatar Information

by

Andrew M. Kane

A Capstone Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Information Technology

Supervised by

Dianne Bills
Department of Information Technology
B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York
April 2010

Approved By:

Dianne Bills
IT Graduate Coordinator & Associate Professor, Department of Information Technology
Advisory Board Chair

Christopher Egert
Assistant Professor, Department of Interactive Games & Media
Advisory Board Member

Contents

1 Project Details **3**
- 1.1 Abstract . . . 3
- 1.2 Introduction . . . 3
- 1.3 Project Description . . . 4
- 1.4 Project Overview & Goals . . . 6
  - 1.4.1 Proof of Concept Protocol Architecture . . . 7
  - 1.4.2 Avatar Data Exchange Protocol Documentation . . . 8

2 Review of Current Solutions **8**

3 Methodology Protocol Documentation **10**

4 Protocol Proof of Concept **13**
- 4.1 Data Restriction & Conformity . . . 13
- 4.2 Data Transmission . . . 14

5 Protocol Implementation **15**

6 Conclusion **17**

7 Future Development **19**

8 References **21**

Appendices **23**

A SleekBot Plugin Source Code **23**

B Initial Testing Client Source Code **39**

C Final Demonstration Client Source Code **51**

D Final Demonstration XML Schema **71**

E Business Logic Database Schema **75**
- E.1 Business Logic Database Schema Data Dictionary . . . 76

F Business Logic SQL Script **77**

G Ragnarok Online Database Schema **84**

List of Figures

1 Two virtual worlds that contain varied avatar information 5
2 Sample Implementation XML Document 11
3 Avatar Flow Diagram 12
4 Virtual World Compatibility Matrix Example 12
5 Information Data Flow 15
6 Information Transportation Flow 16
7 Business Logic for Avatar"Class" Transformation 75

List of Tables

1 Summary of Current Solutions 8

Thanks to The RIT Electronic Gaming Society, the RIT Information Technology Department, the RIT Interactive Games & Media Department, Dr. Kevin I. Smith, Jennifer C. Patel, Mom, and Dad

1 Project Details

1.1 Abstract

As users create avatars to represent themselves in virtual worlds, the number of replicated user representations grows, in addition to the amount of independent data associated with that avatar. To reduce the amount of replication and increase a user's familiarity, one should be able to transfer one world's avatar information to a destination world. Additional business logic and data restrictions will also be required to ensure proper data identification and proper usage for the destination world, possibly utilizing the type of worlds involved in the transfer. This technique will allow users to bring their previous experiences and information with them to a new world, creating a more static entity that represents themselves in virtualized worlds.

1.2 Introduction

With the growth of online communities and authenticated systems, users have begun to accumulate a large number of authentication tokens, typically consisting of username and password pairs. Authentication and access control solutions have arisen in an attempt to assist users in countering this plethora of tokens. Services such as the XMPP communication protocol [16], OpenID [20] and others allow one to utilize a single authentication token for multiple systems. The sharing of information across various communities, such as multiplayer games, blogs, forums, and websites has also become very popular, prompting the launch of several social development frameworks such as Facebook Applications [15] and other methodologies of sharing information.

Similarly, as virtual worlds have emerged, the issue of managing authentication tokens has become an important topic. Occasionally, virtual worlds manage additional information, such as user preferences and avatar information, and players in these worlds have started to encounter the same issue of having too many authentication tokens. Contrary to the fast pace of virtual world development, authentication and token management services have begun to appear more slowly, such as NCsoft's "PlayNC™" service [18] and Nexon's "Nexon Passport" service [19]. These services allow games to share a small number of user preferences, avatar information, and authentication tokens, similar to the previously mentioned access control services. However, the sharing of information between virtual worlds has not gone beyond these existing authentication services involving small amounts of user information and preferences or bare statistics. Experiments [24] have been

performed in which extensive amounts of player information and preferences, such as entire player avatars, have been transmitted between virtual worlds. However, non-experimental virtual world access control services currently only offer advantages that are similar to, if not the same as, the standard authentication token access-control services.

Within the current services available that offer grouping of authentication and basic user information, the issue of sharing avatar information has not been addressed. A single sign-on is the ideal solution for users; however converting current infrastructures to such a system would require multiple iterations of change. The changes that would have to be done, and possibly re-evaluated as services expand, would include identifying core avatar information that will be transferable, creating a "middle-man" service to facilitate the conversion, and finally implementing a centralized single sign-on. This project will focus upon the steps within an iteration where multiple virtual worlds are beginning to share avatar information across worlds, to facilitate the sharing of avatars via a middle-man service. A initial prototype will be created to demonstrate the flow of base avatar information from a source virtual world to a destination virtual world, in addition to applying minor business logic.

My goal is to develop a more robust and extensible methodology for communicating and transferring avatar information between virtual worlds. This methodology will go beyond the currently existing authentication services to address the need for interoperability between virtual environments. My focus is the transmission of avatar information and user preferences, as a first initial step towards allowing avatar data to be transmitted across virtual worlds and ideally eliminating the need for numerous authentication tokens.

1.3 Project Description

To facilitate the flow and distribution of avatar data from virtual worlds, a common protocol is required. Utilizing this standard protocol, one would be able to pull distinct attributes (i.e. eye color, hair color) pertaining to a player's avatar, in addition to partial listings of attributes or even full profiles. However, depending on a virtual world's parameters, environment, virtual locale, availability of avatar customizations, and restrictions pertaining to its specifications, some avatar information may need to undergo business logic transformations, due to restrictions and conversions, to ensure that it complies with the destined virtual world's universe of discourse, environment, and business rules.

Numerous virtual worlds allow for the detailed customization of avatar characteristics and physical

attributes which results in ample user-specified data. However, some virtual worlds restrict a user's ability for customization or set preferences due to its setting or environmental preference options & limitations. Therefore when transferring avatars, business logic will be required to ensure conformity with the destination virtual world; so that improper or unnecessary avatar data originating from the source virtual world is either omitted or not requested. As an example, in the virtual world *ToonTown* [22] (pictured in *Figure 1(a)*), users take the form of humanoid animals, such as a mouse and a dog, and venture amongst a cartoon-like landscape oriented for young children. Comparing this to the avatar choices and "battle-ridden" environment of the *EverQuest II* [23] virtual world (pictured in *Figure 1(b)*), we see that conformity would be required to transfer avatars between these two games. With this conformity, certain elements would need to be converted between the two environments, such as the avatar's physical appearance, personal history and achievements, and social statuses. To assist users with sharing avatar information and making various decisions, a protocol to standardize such sharing is needed.

(a) Screenshot of *ToonTown* [8]

(b) Screenshot of *EverQuest II* [1]

Figure 1: Two virtual worlds that contain varied avatar information are *ToonTown* and *EverQuest II*. Both virtual worlds have users playing as their personal customized avatars. However, the environment and art style of each game is very distinct. *ToonTown* is designed to provide a fun environment for younger users, while *EverQuest II* provides an action combat-oriented game targeted towards more mature users.

The creation of such a sharing protocol, as well as an approach for enforcing a dynamic set of attribute transferrals to allow for a varying range of flexible virtual environment rules, will be the main focus of this project. To create such a fluid, straight-forward, and accessible protocol to also support various attribute types and business logic restrictions, where applicable, will be a secondary goal of the project. Additionally, the management of authenticated access will also be core to this project and will be covered in detail with examples where necessary.

1.4 Project Overview & Goals

The overall goal of this project is to provide a prototype that allows players to transport avatar information between select virtual worlds that are compatible to sharing, with the ideal situation being that a user can transport his or her avatar to multiple cooperating virtual worlds. To accomplish the overall goal, numerous sub-goals were created:

- Choose a communication technology to act as a basis for sending the avatar data between the cooperating virtual worlds.
- Develop the tools necessary to facilitate transfer of information.
- Design a core standard format and data layout for transferring avatar information.
- Develop a connection between two virtual worlds using the created tools and layout.

The initial stage of this project is where I conducted research into pre-existing integration implementations, such as OpenID [20] and the interoperability milestone documentation compiled by Linden Labs and IBM [24]. By studying these pre-developed intercommunication examples, pre-established methodologies and experimental ideas were evaluated for implementation as a basis for the project's prototype. Components such as common protocol methodologies, recommended best practices, and previous experiment results were extracted from these resources, providing benefitial recommendations of usage.

This way, similar functionality or portions of pre-existing technologies were evaluated for inclusion in the proposed protocol, reducing the amount of retooling required for users experienced with such technologies and building upon these proven methodologies. Identifying possible technologies that could be utilized will also assist in any migrations that will need to be made for systems that wish to support the proposed protocol, such as online virtual worlds attempting to communicate with other worlds.

Following this review, documentation was drafted detailing the proposed protocol for transferring avatar information. Utilizing previous research and current technological progress, an outline of said protocol was created, providing ample information for dynamic usage and application of the data contained within the protocol. In addition to documentation, a sample case and implementation example are included to inform and possibly assist any readers with possible uses.

A number of components are included in this project. These include a review and critical analysis of currently available solutions, a finalized prototype and documentation for the avatar transference protocol,

and a few examples that demonstrate the transfer of an avatar between virtual worlds. Two proofs of concept are also provided: one that demonstrates the polling of the "source" virtual world for avatar information and a second demonstration that emulates transferring an avatar from one virtual world to another in addition to applying foundation business logic.

The benefit of including these components is that it allows one to understand the current state of technologies that address the issue of sharing avatar information and also demonstrates my solution to the issue by providing a robust description of the formats and technologies utilized. Similarly, the included proof of concepts provide developers with a starting position to begin implementation of such a platform that will allow the transfer of avatars.

1.4.1 Proof of Concept Protocol Architecture

The proofs of concept of the protocol methodology developed establish how the described protocol can be utilized to export and, depending on the desired usage and business decisions, to transfer avatar information between virtual worlds or even to third party systems, such as an avatar preview system or gallery. This demonstrates that not only is the documentation and source code provided sound, but also usable and presentable for a variety of applications.

Examples of the protocol in action are provided as a deliverable. Said examples include the generation of protocol messages by pulling avatar information from virtual worlds and the receiving, reading and utilization of these protocol messages. These demonstrations are written as proofs of concept and allow for the ease of adoption by providing future developers pre-existing examples upon which to base their own systems.

Such demonstrations will also be helpful for individuals wishing to see a variety of situations to which the protocol could apply and assist the developers of technologies for interoperability and user adaptability. In addition to assisting developers, these examples also include Use Cases which will allow others to visualize where and how this protocol may apply to a variety of situations. The first example that is provided in *Appendix B* demonstrates the polling of avatar information, while the second example in *Appendix C* features an avatar being transferred to a secondary virtual world.

1.4.2 Avatar Data Exchange Protocol Documentation

This document provides an overview of the project and the associated protocol, the layout of the protocol's data payload with descriptions of the payload attributes, source code for this protocol, and a brief example of how to utilize the provided payload information. This document is meant to provide an implementation-ready version of the protocol, such as where a developer would utilize such a document to evaluate the adaptation, to understand the amount of data that may be available from or to other virtual worlds, or to discuss the usability of such a mechanism.

2 Review of Current Solutions

Table 1: Summary of Current Solutions

Solution Name	**Open Source?**	**User Grouping**	**Information Available**	**Communication Technology**
Facebook Connect	No	Facebook accounts	User Data (identity, photos, etc.) and listing of friends & groups	HTTP [11]
Nexon Passport	No	Nexon accounts	Unknown (currently, internal usage only)	Unknown
OpenID	Yes	By server and username (*protocol://server/username*)	Standard user information	HTTP
PlayNC™	No	NCSoft accounts	Unknown (currently, internal usage only)	Unknown
Second Life [14]	Yes	Second Life or per-server OpenSim accounts	Some avatar information and avatar status	XML-RPC & HTTP
Steam	No	Steam accounts	User information, games owned/played, and listing of friends & groups	Unknown
XFire	No	XFire accounts	User information, games played, listing of friends & groups, game statistics	Custom "Toucan" Protocol [26]
XMPP	Yes	By server and username (*username@server*)	Various (extendable [17]), but usually standard user information	XML

To assess the state of current solutions to the aforementioned interoperability issue, my first step was to identify some of the more popular technologies available that targeted the issue of bridging information and data. My scope, however, was not limited to only virtual worlds, but included other networked solutions that dealt with the sharing of information and authentication tokens. A brief summary of my findings can be seen in *Table 1* and each reviewed technology will be further explained within this section.

Facebook Connect [15] is used to connect to third party websites, allowing users to utilize their pre-existing Facebook accounts to interact with said websites, and eliminating the need for extra accounts. Having users

link their Facebook accounts with a third party website, however, has both benefits and drawbacks. The primary benefit of this link is quick access to user information and friends, but the drawback is that *too* much information may be available to third party websites, including the location of the user and an entire listing of their contacts. Facebook also uses REST-ful services [11] which allows data to be transported via the HTTP protocol, for ease-of-use.

Used for a handful of online games, such as *Maplestory* and *Mabinogi*, and their forums, the Nexon Passport [19] alleviates the requirement of numerous logins for all of Nexon's websites and range of products. However Nexon's Passport system is closed source and no contact information was available, so I was unable to determine what technologies it utilizes.

As one of the most popular cross-domain authentication and user-management services, OpenID [20] (`http://openid.net`) was one of the first technologies chosen as a possible solution. Using OpenID, one can obtain not only a simple authentication, but also standard user contact information and, by taking advantage of various extensions [12], additional data. This technology uses basic HTTP requests and responses [13], but normally utilizes browser or HTTP redirections between the service requesting authentication and the OpenID service. The flexibility of this technology needs to be taken into consideration, since the user's identity can differ depending on the username chosen and the server on which their user exists; users can create any number of accounts on any number of servers, even start their own OpenID servers!

Akin to Nexon's passport system, NCSoft (`http://www.ncsoft.com`) has developed their PlayNC™[18] system. NCSoft uses this system to authenticate accounts for its various games and for its forum. No external uses of the PlayNC™system have been found and it is also closed source, so no information is available.

Outside of the Second Life virtual world (`http://www.secondlife.org`), Second Life accounts are only used for billing and forum authentication purposes. However, once inside Second Life's virtual world or their open source counterpart OpenSimulator [14], avatar information, such as an inventory listing and location, is available via in-game scripts. Second Life uses HTTP and XML-RPC communication [9], triggered by in-game scripts, allowing for external scripts to be run when user-approved. However, only very limited information is available about user avatars, making it difficult to transfer the majority of an avatar's information set.

As one of the two most popular in-game communication services, Steam (`http://www.steampowered.com`) not only serves as a community portal and content-distribution system, but also as a communication system that allows players to send messages while playing a game. Through this service, one can obtain user

information, listings of friends and the groups to which a user belongs, the games that they own, and the user's statistics concerning said games. However, since Steam is closed source, the technologies that it is based upon are unknown.

Another very popular in-game communication service is XFire (`http://www.xfire.com`). Similar to Steam, XFire provides a communication system for players to send messages while playing a game. Using XFire, one can obtain user information and listings of friends and groups, like Steam. However, XFire provides a much more detailed tally of statistics for a user's games. XFire, too, is closed source but uses a custom protocol named "Toucan" [26].

Finally, XMPP [16] (`http://xmpp.org`) is available primarily as an open source messaging protocol. Similar to OpenID, XMPP allows for user and server flexibility, with each server containing a listing of users and operating independently. Various XMPP extensions are also available, allowing a user to access additional data, such as an avatar picture or the user's current emotions, or provide users with additional features. XMPP is normally utilized for user authentication, the storage of basic user information, and the sending of messages. However, with various extensions [17], XMPP can send and receive other types of information, all through standard, schema-based XML [16].

After researching the various technologies presented above that could be utilized as a base communication method for a prototype, I chose XMPP as the base for the proposed protocol. The decision to base the project upon XMPP was determined by the fact that not only is it an open standard that takes advantage of an easily readable XML-based layout of data, but also due to its extensibility and its pre-existing use for message communication between different domains. This makes XMPP an ideal choice as the proposed methodology since it can be easily extended and already is targeted towards sending messages and metadata about users between domains.

3 Methodology Protocol Documentation

The ability to transfer avatar information between virtual worlds depends upon the degree of similarity that exists between the avatars and environments of the two worlds. In the sample implementation of the proposed protocol, the avatar data structures of two virtual worlds were compared: *World of Warcraft's* players' public avatar XML and the *Ragnarok Online* character attributes created by eAthena. The goal was to discover common data that would facilitate avatar transfer. It was found that the two structures

only have a small amount of similar information, consisting of the avatar's name, skills, gender, class or job, race, last login time, level, and inventory. This information was then constructed into a standard XML format that would be utilized to transmit avatar information between the two worlds. An example of said XML documents can be seen in *Figure 2*, which contains an example XML data set about an avatar; this generic avatar is named "Test Avatar" and is a level 80 male human thief who has various items, such as a "Respirator Mask", and is located in the *Ragnarok Online* virtual world. A visualization of the avatar information flow can be seen in *Figure 3*, which demonstrates a sample of what information may be available about an avatar and how one can select only certain types of information to utilize.

```
1  <?xml version="1.0" ?>
2  <avatar xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation=
       "http://www.andrewmkane.com/projects/capstone/avatar.xsd">
3          <name>Test Avatar</name>
4          <stats defense="144" dexterity="115" intelligence="35" speed="112" spirit="60" strength="180
               "/>
5          <gender>M</gender>
6          <class id="6">Thief</class>
7          <race id="1">Human</race>
8          <lastlogin>2009-09-07T17:48:52</lastlogin>
9          <level>80</level>
10         <inventory>
11                 <item amount="2" era="fantasy" id="867" name="Assassin Dagger" type="Dagger" /
                       >
12                 <item amount="1" era="fantasy" id="4078" name="Goggles" type="Apparel" />
13                 <item amount="1" era="fantasy" id="101" name="Cotton Shirt" type="Cloth" />
14                 <item amount="1" era="fantasy" id="115" name="Slacks" type="Cloth" />
15                 <item amount="1" era="fantasy" id="132" name="Leather Boots" type="Leather" />
16                 <item amount="1" era="fantasy" id="5088" name="Respirator Mask" type="Apparel"
                       />
17         </inventory>
18 </avatar>
```

Figure 2: Sample Implementation XML Document that provides details about an avatar. The avatar is represented in a standardized XML document and restricted to only topical information with a XML Schema. In this instance, this user's avatar is male, named "Test Avatar", has a level of 80, and has a variety of items or accessories

To identify the attributes that can be shared as avatar information, a universe of discourse must be identified. A universe of discourse in this domain is a set of relevant avatars, entities, objects, and other related materials that define the virtual world. The size of this set can be large or small, depending upon how different or how similar the two virtual worlds are. For example, referencing the matrix I created in *Figure 4*, one can create a similar matrix to assist in understanding the likelihood of sharing information between only two virtual worlds. By using this matrix, it becomes apparent that certain types of virtual worlds offer higher levels of compatibility with other types of worlds. Creating a universe of discourse for a game further assists in identifying attributes and information that can be shared between virtual worlds.

This information can then be broken down into categories and attributes, allowing one to create a structure that would allow for data of either world. Following the example in *Figure 4*, transferring avatars between a Racing game and a Sports game will offer more information and a higher chance of interoperability than transferring avatars between a Racing game and a Role-Playing game.

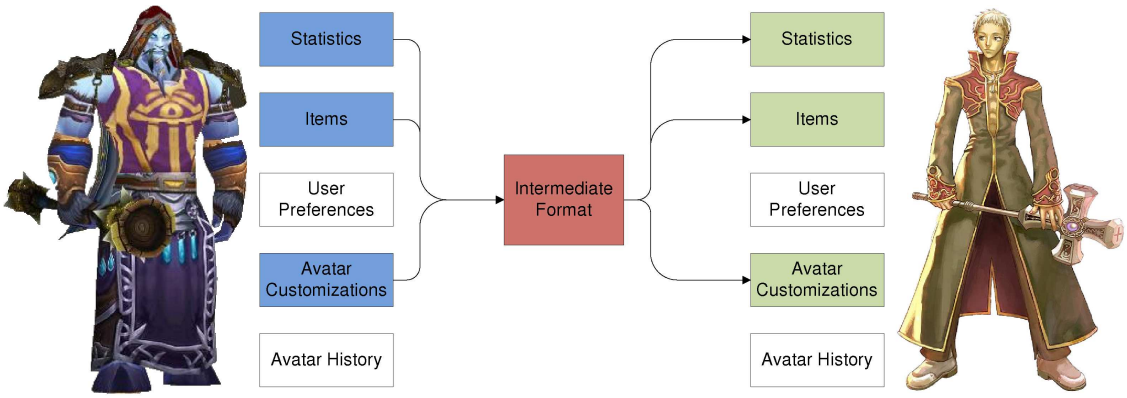


Figure 3: Avatar Flow Diagram demonstrating the flow of information when transferring avatar data. This diagram features a Draenei Shaman on the left, from the virtual world of *World of Warcraft* and a Priest from the virtual world of *Ragnarok Online* on the right. Between these two worlds, the avatars have similar categories of information, each represented within this diagram. "Statistics" represents the various numbers that define the user's avatar, such as their amount of money, character level, physical strength, dexterity, etc. "Items" represents the list of items that they own. "User Preferences" defines the various options that a user can customize, such as their screen resoltuion and audio configuration. Finally, "Avatar Customizations" and "Avatar History" define what makes up the user's avatar: the physical appearance of the avatar and the achievements and history of their avatar. Transferring an avatar between the two worlds requires that the information be placed into an Intermediate Format so that the information can be identified and read correctly.

Once a universe of discourse has been defined, a transport system can be created, utilizing a standard XML transitional document, akin to the one featured in *Figure 2*. A transport system then must be created to allow the identification of the source and destination virtual worlds in addition to parsing and formatting the avatar information to the chosen standard XML document format. Using this methodology, the finalized XML documents, containing the avatar information, can easily be passed between virtual worlds over the chosen transport system, in this case the XMPP messaging system.

Type Name	**Sports**	**Racing**	**Role-Playing**	**Adventure**	**Shooter**
Sports	High	High	Low	Low	Medium
Racing	High	High	Low	Medium	Medium
Role-Playing	Low	Low	High	Medium	Medium
Adventure	Low	Low	Medium	High	Medium
Shooter	Medium	Medium	Medium	Medium	High

Figure 4: Virtual World Compatibility Matrix Example showing which games are more compatible with one another and more likely to provide transferrable avatar information to the destination world.

To further assist with the understanding and conversion of data, additional businses logic can be added as appropriate within a universe of discourse. For example, matching world-centric information to other related information, such as an avatar's profession or species, may be very useful when transferring avatars that have

similar categories of information. In *Appendix F*, a SQL script is provided, demonstrating a link between the *World of Warcraft*, *Ragnarok Online*, and *Second Life* virtual world information known as "classes" which define a user's specific avatar type, such as an avatar can have a class or profession of being a priest or a paladin. As shown in this script, two of the associated virtual worlds have a link between similar classes, while the third world does not have classes. This type of extended busines logic allows for faster analysis and transfer.

4 Protocol Proof of Concept

4.1 Data Restriction & Conformity

Just as the messages are standardized with the XML language syntax, one must also mandate conformity of the messages' structure and content. Conformity will assist with parsing, identification, document structure, and even security measures. To check the conformity of messages being passed, a XML schema is required. Utilizing the universe of discourse and a sample of avatar information from each virtual world, a XML schema can be created that will facilitate and identify information when it is transferred. A XML schema was created for the proof of concept, which can be viewed in *Appendix D*, detailing the message structure for the transfer of avatars between the *World of Warcraft* and *Ragnarok Online* virtual worlds.

Transforming avatar data from one world's requirements to that of another world requires some type of inference engine. For the proof of concept provided, the conformance of avatar data was handled by a XML schema and a set of business logic rules. The XML schema sets down a "base set" of information that the two worlds can share, restricting the data set to only allowed information, while the business logic transforms the provided values between the two worlds. This transferral schema could allow a large amount of information or a minute amount of information depending on how much information can be shared between the two worlds, as the compatibility chart in *Figure 4* suggests. Business logic utilizes the information stored within the XML structures to translate the avatar to the destination world's information schema. This process could be implemented in a number of ways, ranging from a robust inference engine to a set of scripts and database relations, such as the one shown in *Appendix E*. This example also shows how business logic can provide a recommended destination "class" (or job occupation) for an avatar depending on the source world's "class."

4.2 Data Transmission

Since this project uses XMPP as a base protocol for providing the avatar information, a server is required to transmit XMPP messages for the newly developed avatar tranasfer methodology. I chose to use the OpenFire collaboration server software package, due to its open source software license (GNU General Public License v2, `http://www.gnu.org/licenses/gpl-2.0.txt`), its ease of setup, and its support for other open technologies, such as Java and MySQL. In addition, OpenFire is quite scalable [25] and even has an enterprise version, which could be used for administrating virtual worlds with larger user populations.

For testing purposes, such as sending test messages to the scripts supporting the transport mechanisms and assisting with the configuration of said scripts, I used the Pidgin instant-messaging client (`http://www.pidgin.im`). Pidgin is an open source messaging application able to connect with multiple instant messaging systems [2], one being XMPP, and is one of the most popular [21] messaging clients available. Utilizing this application, I was able to send basic messages to the supporting scripts to ensure that the messages were being parsed properly and correct data was returned.

The proof of concept that I created utilizes the aforementioned technologies to demonstrate the extraction of information from a virtual world and conforming it to a standardized XML document. This XML document, then, can be used to create an avatar in a compatible virtual world depending upon the universe of discourse. The flow of this information, in a theoretical production environment, is shown in *Figure 5*.

Figure 5 features three distinct sections: "Debug Communication," "Destination Virtual World," and "Source Virtual World." The "Debug Communication" section is for debugging the communication between virtual worlds. This section is what the prototype utilizes to show the XML document that would normally be parsed by the client virtual world to transform an avatar from another world to its own. The "Source Virtual World" section is the location from which the source avatar information is pulled, through its associated XMPP Server, for transferral into the destination world's XMPP server. Similarly, in the "Destination Virtual World" section, said information is received via the destination world's XMPP server, parsed for values, and utilized to create an avatar that exists within the destination virtual world's settings and which conforms to its business logic.

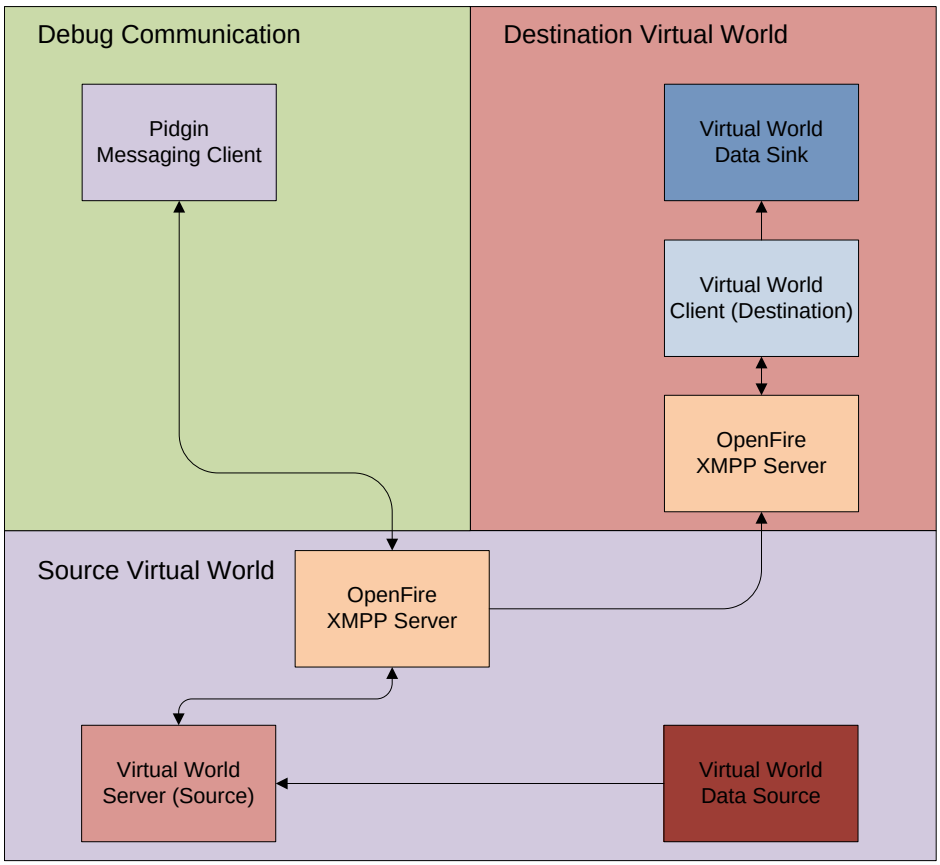


Figure 5: Information Data Flow that shows how data will flow between the different subsystems of the prototype, such as the source & destination virtual world data source & sink, debug communication method, and intermediate XMPP Server.

5 Protocol Implementation

To demonstrate the transfer of an avatar between two virtual worlds, a series of scripts were created to handle the transformation of information. The diagram shown in *Figure 6* specifies the multiple tiers of communication required and provides the proposed protocol's preferred framework for transferring avatar information. My sample implementation utilizes the PHP script shown in *Appendix C*, as a client, taking advantage of the XMPPHP [7] library, and allows the user to interact with and specify which avatar's information is transferred. For the transport portion, I created a Python script, included in *Appendix A*, that utilizes the SleekXMPP [5] library and pulls information from various data sources, depending on the amount, types, and depth of information that one requires for an avatar. In the sample implementation, the World of Warcraft Armory [6] (currently up to the *Wrath of the Lich King* expansion), which provides avatar information in XML format, and an unofficial Ragnarok Online MySQL database, a database schema created for storing Ragnarok Online information by the eAthena project [10] (featured in Appendix G), are

both utilized as data sources with *World of Warcraft* functioning as the source world and *Ragnarok Online* functioning as the target world.

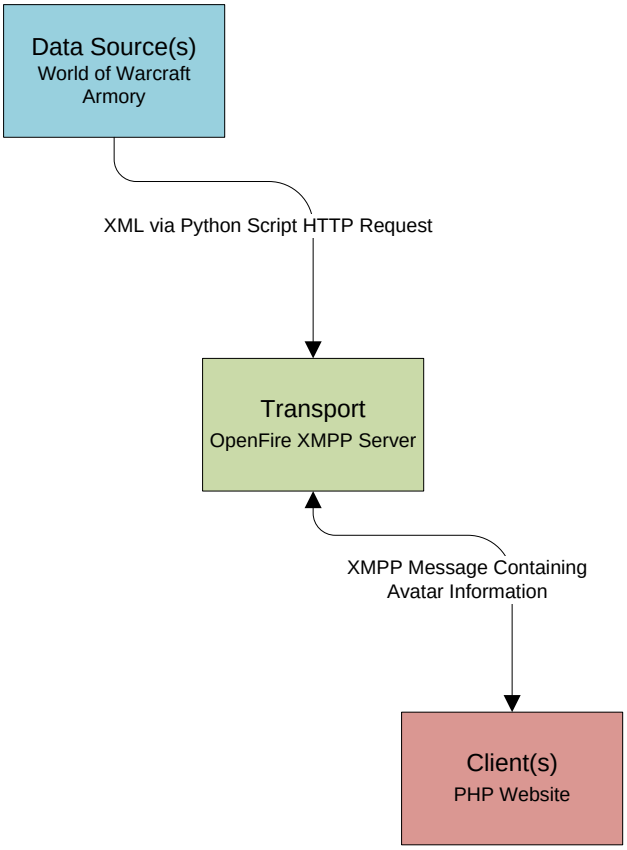


Figure 6: Information Transportation Flow that describes how information flows between a data source, an intermediate transport system, and a client service. In this case, the data source is represented by the *World of Warcraft* Armory, the intermediate transport system is handled by an OpenFire XMPP Server, and the client is fulfilled by a PHP Website.

The client, which is a PHP website in this case, issues a XMPP message containing a command to the transport, currently setup by a set of scripts and the OpenFire XMPP server, which in turn pulls a XML document consisting of the avatar's core attributes from the source virtual world's data source, in this case the *World of Warcraft Armory*. Once this message is received, the client processes the XML document and, based upon the contents, will determine suggested avatar settings and choices in the destination virtual world based upon business logic within the client. After this, the user has the ability to review the settings and change them to what they prefer. This final step, however, is optional depending on the business requirements of the destination virtual world and could even be fully automated.

After utilizing the PHP website client in this example, the user can then enter the destination virtual world and utilize their newly created avatar. In this example, a customized *Ragnarok Online* client is utilized to connect to a custom eAthena server. This step demonstrates that a user can transfer their avatar from a

source virtual world to a target virtual world, have suggestions provided to them via business logic, choose these additional options to further define their avatar in the new world, and finally utilize the newly created avatar.

Within the transport section of the sample implementation, the client's XMPP message is parsed to identify key terms, such as the source virtual world and a specific avatar name. Afterwards, the script pulls data from the source virtual world's data provisions that correlates with the target avatar and transforms it into a standard XML format. This XML is then returned to the originating client, who requested the information, where business logic is applied and a recommended avatar is provided.

To facilitate the transfer of this avatar information, two functions were written in the transport section script for my sample implementation: the "avatar" message and the "avatar_xml" message. The "avatar" message was created initially as a debugging and testing message to support plaintext listing of basic avatar information and was utilized only in the initial testing script (source code available in Appendix B). After ensuring that the debug function was working properly and returning the correct information, the "avatar_xml" function was created. This function correctly formats the basic transferable avatar information into a common XML format for the universe of discourse that was chosen for the sample implementation. Utilizing the "avatar_xml" functions, avatar information is transported via XMPP messages easily and with a common structure.

This three-tier methodology, composed of the client; transport; and server, allows a user to utilize a single client to request data via this transport mechanism from multiple servers. In addition, it also allows a single point of business intelligence that enforces a consistent filtration and parsing system across all incoming avatar information so that only correct and proper avatar information is provided. Virtual worlds can then utilize this by providing players with transfer suggestions based on prior game experiences, bonuses, or additional content based on previous games purchased or achievements gained. Finally, users will be able to utilize this to adapt much more quickly to new virtual worlds, by being able to use an avatar that resembles ones that they have used in other virtual worlds.

6 Conclusion

In this project, I have demonstrated that it is possible to move a user's avatar information from one virtual world to a secondary virtual world. Additionally, I have shown that business logic is also possible,

allowing a system to accept avatar information and perform transformation to ensure its conformance for the destination world. This proves that although being able to transfer avatars is currently nonexistant amongst popular virtual worlds, it is possible and can be done! I can only hope that with future developments, virtual world interoperability will be evaluated as a part of any virtual world creation project.

Thankfully, during the development of this project, only minimal problems were encountered. The first issue that hindered my progress was the lack of documentation in the SleekXMPP library, which was utilized to communicate between the PHP client script and the XMPP transport system. Minimal information was provided on the functions that the library provided; however numerous sample plugins were included with the source code of the project. These samples proved to be invaluable in understanding the usage of the library and assisted the development of the script that is utilized for the transformation and transmission of avatar information.

An additional problem was obtaining avatar information from various source world data sources. I was able to obtain data from the *World of Warcraft Armory* (currently up to the *Wrath of the Lich King* expansion) in addition to my custom *Ragnarok Online* database. However, I was unable to obtain much information from *Second Life*, which I intended on using as a third virtual world for transferring avatars to. It seems that *Second Life* only allows public access to certain information, as users are freely able to convert real money into virtual money within their game. This makes information about avatars very important to users, since their avatars are tied to their actual funds, and thus, most information is unavailable to the public.

The *World of Warcraft Armory* introduced an additional complication as a source world in that it detected browser user agents for XML provision and stylizing. It seems that the service analyzes the user agent that a browser or stream provides in order to determine the information that is to be returned. Since I was pulling the information via a Python script, I had to emulate a "proper" user agent, such as a browser that can display XLST-transformed XML, in order to obtain the avatar information in XML format. In this case, I mimicked the Mozilla Firefox browser's user agent which allowed me to gain access to the avatar information XML.

7 Future Development

Future development and changes to this protocol would not be surprising. As virtual worlds evolve, require more information, and allow for additional features, transfer protocols like this will, likewise, change and evolve. I foresee new transitional document formats being accepted, additional functionality being added, and many more avatar information types being added to protocols.

As more avatar information is added to the process of transferring avatars, there is a possibility that a large amount of data will be transmitted between the virtual worlds, creating payloads that are very large and take a considerable amount of time to transfer and parse. In this circumstance, the avatar's information may need to be split into multiple parts and transmitted in separate messages. This way, the information payload is reduced for each message and less segmentation and parsing time is required for the transmission of information. Additionally, new commands could be created within a protocol to support the transfer of segmented avatar information, such as sending only the currently applicable information to the destination virtual world on an as-needed basis.

Furthermore, as the number of avatar XML messages increases in addition to the amount of XML parsing that is required of said messages, the amount of time that is required to parse each message will increase. With the added possibility of additional avatar information being required, the time that larger XML documents take to parse could become an issue. If this occurs, faster alternatives would have to be found instead of XML, such as JSON [3] and Protocol Buffers [4] which provide faster parsing, but less flexibility than XML.

I also forsee consumer applications being created as protocols like this gain support. As more virtual worlds support the transfer and collection of data, centralized systems could begin to emerge. Websites and communities will log and track user achievements and their progressions through various virtual worlds, such as many of the virtual worlds' own websites. This pooling of information will also link users with avatars between worlds, which before could only be handled by proprietary systems or numerous interexchanging methodologies. Utilizing these methods, various services and communities will arise where multiple virtual worlds' activities will be logged for users of that community.

Inferring what avatar information a virtual world can provide is extremely difficult. Creating a matrix, such as that featured in *Figure 4* assists with predicting how compatible two worlds will be with one another, but is far from perfect. As this type of protocol continues to evolve and grow over time, the creation of an inference engine would be an ideal component. Predicting and perhaps even discovering exactly what data

a world can provide would be an invaluable addition to this protocol type.

In the future, I believe that this type of protocol will continue to thrive and evolve with the various other uses that developers see that it may be usable for. Additions to intermediate XML documents, concerning the world's universe of discourse, and changes between common intermediate document types may occur. However the basis of the protocol will remain the same: a simple and easily-parsed intermediate document that conforms to corresponding universes of discourse for similar virtual worlds. This simple ideal will allow this protocol to adapt and be utilized for our current generation of virtual worlds and other virtual worlds to come.

8 References

[1] Everquest ii sentinel's fate. http://everquest2.station.sony.com/media/screenshots. Accessed on February 7, 2010.

[2] *About Pidgin, the universal chat client*. http://pidgin.im/about/. Accessed on July 26, 2009.

[3] *JSON*. http://www.json.org/. Accessed on September 14, 2009.

[4] *Protocol Buffers*. http://code.google.com/apis/protocolbuffers/. Accessed on September 14, 2009.

[5] *sleekxmpp*. http://code.google.com/p/sleekxmpp/. Accessed on September 02, 2009.

[6] *The World of Warcraft Armory*. http://www.wowarmory.com. Accessed on September 02, 2009.

[7] *xmpphp*. http://code.google.com/p/xmpphp/. Accessed on September 02, 2009.

[8] Footballtoon photos. http://www.toontowncentral.com/gallery/showphoto.php?photo=13557, July 2008. Accessed on February 7, 2010.

[9] *Category:LSL Communications*. http://wiki.secondlife.com/wiki/Category:LSL_Communications, October 2008. Accessed on July 20, 2009.

[10] Main page - eathena wiki. www.eathena.ws/wiki/index.php/Main_Page, June 2009. Accessed on November 5, 2009.

[11] *API - Facebook Developer Wiki*. http://wiki.developers.facebook.com/index.php/API, July 2009. Accessed on July 19, 2009.

[12] *Final: OpenID Attribute Exchange 1.0 - Final*. http://openid.net/specs/openid-attribute-exchange-1_0.html, 2009. Accessed on July 20, 2009.

[13] *Final: OpenID Authentication 2.0 - Final*. http://openid.net/specs/openid-authentication-2_0.html, 2009. Accessed on July 20, 2009.

[14] *Main Page - OpenSim*. http://opensimulator.org/wiki/Main_Page, July 2009. Accessed on July 19, 2009.

[15] Facebook, Inc. *Get Started*. http://developers.facebook.com/get_started.php?tab=principles, 2009. Accessed on April 12, 2009.

[16] XMPP Standards Foundation. *About XMPP*. http://xmpp.org/about/, 2009. Accessed on April 12, 2009.

[17] XMPP Standards Foundation. *XMPP Extensions*. http://xmpp.org/extensions/, 2009. Accessed on July 20, 2009.

[18] NCsoft Corporation. *Welcome to PlayNC*. http://www.plaync.com/us/about/, 2009. Accessed on April 12, 2009.

[19] Nexon Corporation. *New To Nexon?* http://www.nexon.net/Support/NewToNexon.aspx, 2009. Accessed on April 12, 2009.

[20] OpenID Foundation. *What is OpenID?* http://openid.net/what/. Accessed on April 12, 2009.

[21] *Adam Pash. Five Best Instant Messengers*. http://lifehacker.com/375391/five-best-instant-messengers, April 2008. Accessed on October 29, 2009.

[22] *Disney*. What is toontown? http://play.toontown.com/about.php. Accessed on November 9, 2009.

[23] *Sony Online Entertainment*. Character races. http://everquest2.station.sony.com/allraces.vm. Accessed on November 9, 2009.

[24] Steven Tomasco. *Linden Lab and IBM Achieve Major Virtual World Interoperability Milestone*. July 2008. Accessed on April 12, 2009.

[25] Steve Traut. *Openfire Scalability Enhancements*. 3 2007. Accessed on July 26, 2009.

[26] Xfire, Inc. *Game SDK*. http://www.xfire.com/cms/xf_game_sdk/, 2009. Accessed on July 19, 2009; had to explore source-code for protocol information.

Appendices

A SleekBot Plugin Source Code

```
1  # Import the required libraries that we need
2  import string
3  import time
4  import xml.etree.ElementTree as etree
5  import urllib2
6  import sys
7  import MySQLdb
8
9  # Define our class
10 class avatar(object):
11         # Define the class constructor function
12         def __init__(self, bot, config):
13                 try:
14                         # Set the basic SleekBot Information
15                         self.bot = bot
16                         self.config = config
17                         # What's our plugin about?
18                         self.about = "Allows servers to obtain avatar information from other servers."
19                         # Specify the /avatar command
20                         self.bot.addIMCommand('avatar', self.process_avatar)
21                         self.bot.addMUCCommand('avatar', self.process_avatar)
22                         self.bot.addHelp('avatar', 'Avatar Information', "Information about an avatar in
                               plaintext", 'avatar')
23                         # Specify the /avatar_xml command
24                         self.bot.addIMCommand('avatar_xml', self.process_avatar)
25                         self.bot.addMUCCommand('avatar_xml', self.process_avatar)
26                         self.bot.addHelp('avatar_xml', 'Avatar Information', "Information about an avatar
                               in XML", 'avatar_xml')
```



```
27              except:
28                      print "Error adding the plugin to SleekBot"
29                      sys.exit(1)
30
31      # Define our process_avatar function
32      def process_avatar(self, command, args, msg):
33              # Initialize the avatar info object to false first
34              avatar_info = False
35
36              # Split the argument string into a listing of parameters
37              params = args.split();
38              # Check if we have any parameters, if not, simply return an error-like message
39              if len(params) <= 0:
40                      return "Usage: /" + command + " <avatar name> <service>"
41
42              # Set the service name and avatar name
43              service = params[len(params) - 1]
44              service_param = ""
45              if service.find(":") != -1:
46                      serv_params = params[len(params) - 1].partition(":")
47                      service = serv_params[0]
48                      service_param = serv_params[2]
49              params.pop()
50              avatar_name = string.join(params, "")
51
52              print "Param is %s, Name is %s, and Service is %s\n" % (service_param, avatar_name,
                    service)
53
54              # Grab data from the specified service
55              # In this case, we specified World of Warcraft, so snag the information
56              # from Warcraft's "WoW Armory", which publishes XML representations of player
                    avatars
```



```
57          if service.lower() == "wow" or service.lower() == "warcraft":
58
59              # Grab the XML contents
60              try:
61                  if service_param != None:
62                      wow_url = "http://www.wowarmory.com/character-sheet.xml?r
                            =%s&n=%s" % (service_param, avatar_name)
63                  else:
64                      wow_url = "http://www.wowarmory.com/character-sheet.xml?r
                            =Deathwing&n=%s" % avatar_name
65                  opener = urllib2.build_opener()
66                  opener.addheaders = [('user-agent','Mozilla/5.0 (X11; U; Linux x86_64;
                        en-US; rv:1.9.1.1) Gecko/20090715 Firefox/3.5.1'),]
67                  # Parse the contents of the avatar page as XML to create a DOM object
68                  wow_xml = etree.XML(opener.open(wow_url).read())
69              except:
70                  return "Error: Unable to obtain Warcraft XML information"
71
72              # Create the avatar_instance object
73              try:
74                  avatar_info = avatar_instance(wow_xml.find(".//characterInfo/character")
                        .get("name"))
75              except:
76                  return "Error: Unable to create Avatar Information object"
77
78              # Set the stats of the avatar from the XML DOM
79              try:
80                  stats_node = wow_xml.find(".//characterInfo/characterTab/baseStats")
81                  avatar_info.setStats(
82                      int(stats_node.find(".//strength").get("base")),
83                      int(stats_node.find(".//agility")[0].get('base')),
84                      int(stats_node.find(".//stamina")[0].get('base')),
```



```
85                                  int(stats_node.find(".//intellect")[0].get('base')),
86                                  int(stats_node.find(".//spirit")[0].get('base')),
87                                  int(stats_node.find(".//armor")[0].get('base'))
88                          )
89                  except:
90                          avatar_info.setStats(None, None, None, None, None, None)
91
92                  # Set the gender of the avatar from the XML DOM
93                  try:
94                          avatar_info.setGender(string.lower(wow_xml.find(".//characterInfo/
                                character").get("gender")))
95                  except:
96                          avatar_info.setGender(None)
97
98                  # Set the class of the avatar from the XML DOM
99                  try:
100                         avatar_info.setClass(wow_xml.find(".//characterInfo/character").get("
                                class"), int(wow_xml.find(".//characterInfo/character").get("classId"
                                )))
101                 except:
102                         avatar_info.setClass(None)
103
104                 # Set the race of the avatar from the XML DOM
105                 try:
106                         avatar_info.setRace(wow_xml.find(".//characterInfo/character").get("race
                                "), int(wow_xml.find(".//characterInfo/character").get("raceId")))
107                 except:
108                         avatar_info.setRace(None)
109
110                 # Set the guild of the avatar from the XML DOM
111                 try:
```

```
112                 avatar_info.setGuild(wow_xml.find(".//characterInfo/character").get("
                        guildName"), int(wow_xml.find(".//characterInfo/character").get("
                        guildId")))
113             except:
114                 avatar_info.setGuild(None, None)
115
116             # Set the avatar's last login date from the XML DOM
117             try:
118                 curtime = time.strptime(str(wow_xml.find(".//characterInfo/character").
                        get("lastModified")), "%B %d, %Y")
119                 avatar_info.setLastLogin(time.strftime("%Y-%m-%dT%H:%M:%S",
                        curtime))
120             except:
121                 avatar_info.setLastLogin(None)
122
123             # Set the avatar's level based on data from the XML DOM
124             avatar_info.setLevel(int(wow_xml.find(".//characterInfo/character").get("level")))
125
126             # Set the level of the avatar from the XML DOM
127             # however, no color information is available from the provided XML
128             avatar_info.setColors(None, None, None)
129
130             # Set the avatar's amount of money from the XML DOM
131             # however, no monitary information is available from the provided XML
132             avatar_info.setMoney(None)
133
134             # Set the avatar's inventory from -various new- XML DOMs (oh boy)
135             inventory_items = wow_xml.findall(".//characterInfo/characterTab/items/item")
136             for item in inventory_items:
137                 wow_item_url = "http://www.wowarmory.com/item-tooltip.xml?i=%d"
                        % int(item.get("id"))
138                 wow_item_xml = etree.XML(opener.open(wow_item_url).read())
```



```
139                             try:
140                                     wow_item_name = wow_item_xml.find(".//itemTooltips/
                                            itemTooltip/name").text
141                             except:
142                                     wow_item_name = None
143                             try:
144                                     wow_item_id = wow_item_xml.find(".//itemTooltips/itemTooltip/
                                            id").text
145                             except:
146                                     wow_item_id = None
147                             try:
148                                     wow_item_subclass = wow_item_xml.find(".//itemTooltips/
                                            itemTooltip/equipData/subclassName").text
149                             except:
150                                     wow_item_subclass = None
151                             avatar_info.addInventory(
152                                     wow_item_name,
153                                     wow_item_id,
154                                     wow_item_subclass,
155                                     "fantasy"
156                             )
157
158                     # In this case, we specified Ragnarok, so we need to connect
159                     # to the MySQL server that powers the Ragnarok server
160                     # and pull information from that!
161                     elif service.lower() == "ro" or service.lower() == "ragnarok":
162                             # Open the MySQL connection
163                             conn = MySQLdb.connect( host = "localhost",
164                                                     user = "ragnarok",
165                                                     passwd = "capstone",
166                                                     db = "ragnarok")
167
```

```
168                 # Create a cursor to query the database
169                 cursor = conn.cursor(MySQLdb.cursors.DictCursor)
170
171                 # Query the database for avatar information
172                 cursor.execute("SELECT `char`.char_id, login.account_id, sex, `char`.name AS
                        character_name, class, class_name, base_level, job_level, zeny, str, agi, vit, `int
                        `, dex, luk, guild.guild_id, hair, hair_color, clothes_color, guild.name AS
                        guild_name, lastlogin, item_id, item_name FROM (login INNER JOIN (((`
                        char` INNER JOIN classes ON `char`.class = classes.class_id) LEFT JOIN
                        guild ON `char`.guild_id = guild.guild_id) LEFT JOIN (SELECT char_id,
                        item_id, item_name, amount FROM (inventory INNER JOIN (SELECT id as
                         item_id, name_english AS item_name, type FROM item_db UNION SELECT
                         id as item_id, name_english AS item_name, type FROM item_db2) item_dbs
                        ON item_dbs.item_id = inventory.nameid) WHERE equip > 0)
                        item_inventory ON item_inventory.char_id = `char`.char_id) on login.
                        account_id = `char`.account_id) WHERE lower(`char`.name) = lower('%s');
                        " % avatar_name)
173
174                 # Grab the query results
175                 rows = cursor.fetchall()
176
177                 # Start parsing the results
178                 for row in rows:
179                     if avatar_info:
180                         avatar_info.addInventory(row["item_name"], int(row["item_id"]))
181                     else:
182                         avatar_info = avatar_instance(row["character_name"], int(row["
                                char_id"]))
183                         avatar_info.setStats(int(row["str"]), int(row["agi"]), int(row["dex"
                                ]), int(row["int"]), int(row["luk"]), int(row["vit"]))
184                         avatar_info.setGender(row["sex"])
185                         avatar_info.setClass(row["class_name"], int(row["class"]))
```

```
186                                     if row["guild_id"] and row["guild_id"] != "NULL":
187                                             avatar_info.setGuild(row["guild_name"], int(row["guild_id
                                                    "]))
188                                     if row["lastlogin"] and row["lastlogin"] != "NULL":
189                                             curtime = time.strptime(str(row["lastlogin"]), "%Y-%m
                                                    -%d %H:%M:%S")
190                                             avatar_info.setLastLogin(time.strftime("%Y-%m-%dT%
                                                    H:%M:%S", curtime))
191                                     avatar_info.setLevel(int(row["base_level"]), int(row["job_level"]))
192                                     avatar_info.setColors(None, int(row["hair_color"]), int(row["
                                            clothes_color"]))
193                                     avatar_info.setMoney(int(row["zeny"]))
194                                     if row["item_name"] != None and row["item_id"] != None:
195                                             avatar_info.addInventory(row["item_name"], int(row["
                                                    item_id"]))
196
197                             # Close the cursor
198                             cursor.close()
199
200                             # Close the MySQL connection
201                             conn.close()
202
203                     else:
204                             return "Invalid Service; please provide a valid service"
205
206                     # Check what command we ran to see how we should return the data
207                     if command.lower() == "avatar":
208                             return avatar_info.export()
209                     elif command.lower() == "avatar_xml":
210                             return avatar_info.exportXML()
211
212 # The instance of a single avatar
```

```
213 class avatar_instance:
214         # Constructor, takes in the avatar's id and name (or similar identifiers)
215         def __init__(self, name, id = None):
216                 self.name = name
217                 self.id = id
218                 self.inventory = [];
219                 # Set defaults
220                 self.strength = None
221                 self.speed = None
222                 self.dexterity = None
223                 self.intelligence = None
224                 self.spirit = None
225                 self.defense = None
226                 self.gender = None
227                 self.class_name = None
228                 self.class_id = None
229                 self.race_name = None
230                 self.race_id = None
231                 self.guild_name = None
232                 self.guild_id = None
233                 self.lastLogin = None
234                 self.main_level = None
235                 self.sub_level = None
236                 self.skin = None
237                 self.hair = None
238                 self.clothes = None
239                 self.money = None
240         # Function that sets the avatar's attributes
241         def setStats(self, strength, speed, dexterity, intelligence, spirit, defense):
242                 self.strength = strength
243                 self.speed = speed
244                 self.dexterity = dexterity
```



```
245             self.intelligence = intelligence
246             self.spirit = spirit
247             self.defense = defense
248         # Function that sets the avatar's gender
249         def setGender(self, gender):
250             if gender.lower() == "male" or gender.lower() == "m":
251                 self.gender = "m"
252             elif gender.lower() == "female" or gender.lower() == "f":
253                 self.gender = "f"
254             else:
255                 self.gender = None
256         # Function that sets the avatar's class
257         def setClass(self, class_name, class_id = None):
258             self.class_name = class_name
259             self.class_id = class_id
260         # Function that sets the avatar's race
261         def setRace(self, race_name, race_id = None):
262             self.race_name = race_name
263             self.race_id = race_id
264         # Function that sets the avatar's guild or clan
265         def setGuild(self, guild_name, guild_id = None):
266             self.guild_name = guild_name
267             self.guild_id = guild_id
268         # Sets the avatar's last login
269         def setLastLogin(self, lastLogin):
270             self.lastLogin = lastLogin
271         # Sets the avatar's level(s)
272         def setLevel(self, main_level, sub_level = None):
273             self.main_level = main_level
274             self.sub_level = sub_level
275         # Sets the avatar's various colors
276         def setColors(self, skin = None, hair = None, clothes = None):
```



```
277             self.skin = skin
278             self.hair = hair
279             self.clothes = clothes
280         # Sets the avatar's amount of money
281         def setMoney(self, money):
282             self.money = money
283         # Add something to an avatar's inventory
284         def addInventory(self, name, id = None, type = None, era = None, amount = 1):
285             self.inventory.append({
286                 "id" : int(id),
287                 "name" : name,
288                 "type" : type,
289                 "era" : era,
290                 "amount" : int(amount)
291             })
292         # Exports the avatar in plaintext
293         def export(self):
294             output = "name: %s" % self.name
295             if self.id and self.id != None:
296                 output += "\nid: %d" % self.id
297             if self.strength and self.strength != None:
298                 output += "\nstrength: %d" % self.strength
299             if self.speed and self.speed != None:
300                 output += "\nspeed: %d" % self.speed
301             if self.dexterity and self.dexterity != None:
302                 output += "\ndexterity: %d" % self.dexterity
303             if self.intelligence and self.intelligence != None:
304                 output += "\nintelligence: %d" % self.intelligence
305             if self.spirit and self.spirit != None:
306                 output += "\nspirit: %d" % self.spirit
307             if self.defense and self.defense != None:
308                 output += "\ndefense: %d" % self.defense
```



```
309             if self.gender and self.gender != None:
310                     output += "\ngender: %s" % self.gender
311             if self.class_name and self.class_name != None:
312                     output += "\nclass: %s" % self.class_name
313             if self.class_id and self.class_id != None:
314                     output += "\nclass_id: %d" % self.class_id
315             if self.race_name and self.race_name != None:
316                     output += "\nrace: %s" % self.race_name
317             if self.race_id and self.race_id != None:
318                     output += "\nrace_id: %d" % self.race_id
319             if self.guild_name and self.guild_name != None:
320                     output += "\nguild: %s" % self.guild_name
321             if self.guild_id and self.guild_id != None:
322                     output += "\nguild_id: %d" % self.guild_id
323             if self.lastLogin and self.lastLogin != None:
324                     output += "\nlast_login: %s" % self.lastLogin
325             if self.main_level and self.main_level != None:
326                     output += "\nlevel: %s" % self.main_level
327             if self.sub_level and self.sub_level != None:
328                     output += "\nsub_level: %s" % self.sub_level
329             if self.skin and self.skin != None:
330                     output += "\nskin_color: %s" % self.skin
331             if self.hair and self.hair != None:
332                     output += "\nhair_color: %s" % self.hair
333             if self.clothes and self.clothes != None:
334                     output += "\nclothes_color: %s" % self.clothes
335             if self.money and self.money != None:
336                     output += "\nmoney: %d" % self.money
337             if self.inventory and self.inventory != None:
338                     for item in self.inventory:
339                             if item["name"] != None:
340                                     output += "\ninventory_item: %s" % item["name"]
```

```
341                                     if item["id"] != None:
342                                             output += "\t%d" % item["id"]
343                                     else:
344                                             output += "\t0"
345                                     if item["type"] != None:
346                                             output += "\t%s" % item["type"]
347                                     else:
348                                             output += "\tnone"
349                                     if item["era"] != None:
350                                             output += "\t%s" % item["era"]
351                                     else:
352                                             output += "\tnone"
353                                     if item["amount"] != None:
354                                             output += "\t%d" % item["amount"]
355                                     else:
356                                             output += "\t0"
357             return output;
358     # Exports the avatar in XML
359     def exportXML(self):
360             root = etree.Element("avatar")
361             root.set("xmlns:xsi", "http://www.w3.org/2001/XMLSchema-instance")
362             root.set("xsi:noNamespaceSchemaLocation", "http://www.andrewmkane.com/projects/
                    capstone/avatar.xsd")
363             node = etree.SubElement(root, "name")
364             node.text = self.name
365             if self.id != None:
366                     node = etree.SubElement(root, "id")
367                     node.text = str(self.id)
368             if self.strength != None or self.speed != None or self.dexterity != None or self.intelligence
                    != None or self.spirit != None or self.defense != None:
369                     node = etree.SubElement(root, "stats")
370                     if self.strength != None:
```

```
371                         node.set("strength", str(self.strength))
372                 if self.speed != None:
373                         node.set("speed", str(self.speed))
374                 if self.dexterity != None:
375                         node.set("dexterity", str(self.speed))
376                 if self.intelligence != None:
377                         node.set("intelligence", str(self.intelligence))
378                 if self.spirit != None:
379                         node.set("spirit", str(self.spirit))
380                 if self.defense != None:
381                         node.set("defense", str(self.defense))
382         if self.gender != None:
383                 node = etree.SubElement(root, "gender")
384                 node.text = self.gender.upper()
385         if self.class_name != None:
386                 node = etree.SubElement(root, "class")
387                 if self.class_id != None:
388                         node.set("id", str(self.class_id))
389                 node.text = self.class_name
390         if self.race_name != None:
391                 node = etree.SubElement(root, "race")
392                 if self.race_id != None:
393                         node.set("id", str(self.race_id))
394                 node.text = self.race_name
395         if self.guild_name != None:
396                 node = etree.SubElement(root, "guild")
397                 if self.guild_id != None:
398                         node.set("id", str(self.guild_id))
399                 node.text = self.guild_name
400         if self.lastLogin != None:
401                 node = etree.SubElement(root, "lastlogin")
402                 node.text = str(self.lastLogin)
```



```
403             if self.main_level != None:
404                     node = etree.SubElement(root, "level")
405                     node.text = str(self.main_level)
406             if self.sub_level != None:
407                     node = etree.SubElement(root, "sub_level")
408                     node.text = str(self.sub_level)
409             if self.skin != None or self.hair != None or self.clothes != None:
410                     node = etree.SubElement(root, "color")
411                     if self.skin != None:
412                             node.set("skin", str(self.skin))
413                     if self.hair != None:
414                             node.set("hair", str(self.hair))
415                     if self.clothes != None:
416                             node.set("clothes", str(self.clothes))
417             if self.money != None:
418                     node = etree.SubElement(root, "money")
419                     node.text = str(self.money)
420             if self.inventory != None and len(self.inventory) > 0:
421                     node = etree.SubElement(root, "inventory")
422                     for item in self.inventory:
423                             item_node = etree.SubElement(node, "item")
424                             if item["name"] != None:
425                                     item_node.set("name", str(item["name"]))
426                             if item["id"] != None:
427                                     item_node.set("id", str(item["id"]))
428                             if item["type"] != None:
429                                     item_node.set("type", str(item["type"]))
430                             if item["era"] != None:
431                                     item_node.set("era", str(item["era"]))
432                             if item["amount"] != None:
433                                     item_node.set("amount", str(item["amount"]))
434             #output.appendChild(root)
```

```
435             #return output.toprettyxml()
436             #return output.toxml("utf-8")
437             return "" + etree.tostring(root, "utf-8")
```

B Initial Testing Client Source Code

```
1  <?php
2          if(!isset($_POST["type"]) || $_POST["type"] != "xml") {
3  ?>
4  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
5          "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
6  <html xmlns="http://www.w3.org/1999/xhtml">
7  <head>
8          <title>Capstone Demonstration #1</title>
9          <script type="text/javascript" src="./jquery-1.3.2.min.js"></script>
10         <script type="text/javascript">
11                 $(function() {
12                         if($("p#service_p select").val() == "wow")
13                                 $("p#server_p:hidden").show();
14                         $("p#service_p select").change(function(){
15                                 if($(this).val() == "wow")
16                                         $("p#server_p:hidden").fadeIn();
17                                 else
18                                         $("p#server_p:visible").fadeOut();
19                         });
20                         return;
21                 });
22         </script>
23         <style type="text/css">
24                 p#server_p {
25                         display: none;
26                 }
27         </style>
28 </head>
29 <body>
30 <?php
31         } else {
```



```
32              header ("content-type: text/xml");
33      }
34
35      if(isset($_POST["type"]) && isset($_POST["name"]) && isset($_POST["service"])) {
36              // Require the inclusion of the XMP PHP library
37              require_once("XMPPHP/XMPP.php");
38
39              try {
40                      // Connect to the XMPP server
41                      $conn = new XMPPHP_XMPP("qrro2.servegame.com", 5222, "client", "garden",
                            "AvatarDemo", "qrro2.servegame.com");
42                      $conn->connect();
43
44                      // While are still connected...
45                      while(!$conn->isDisconnected()) {
46                              // Process various things unil the events "message" or "session_start"
47                              $events = $conn->processUntil(array(
48                                      "message", "session_start"
49                              ));
50                              // Utilizing the data provided...
51                              foreach($events as $event) {
52                                      // Switch on the event name
53                                      switch($event[0]) {
54                                              // If it's a message...
55                                              case "message":
56                                                      // Let's spit out the body! That's the content of
                                                            the message!
57                                                      // Depending on the output, we may have to add
                                                            additional PHP headers or HTML
58                                                      if($_POST["type"] == "xml") {
59                                                      } else {
60                                                              echo "<pre>";
```

```
61                                  }
62                                  echo $event[1]["body"];
63                                  if($_POST["type"] != "xml")
64                                          echo "</pre>";
65                                  // Then disconnect, since we have what we need
66                                  $conn->disconnect();
67                                  break;
68                          case "session_start":
69                                  // Set that we are present and alive!
70                                  $conn->presence("Pulling Information");
71                                  $service = htmlspecialchars(stripslashes(
                                        $_POST["service"]));
72                                  // If a server was specified, append that to the
                                        service
73                                  if(isset($_POST["server"]) && strlen($_POST["
                                        server"]) > 0)
74                                          $service .= ":" . htmlspecialchars(
                                                stripslashes($_POST["server"]));
75                                  // Send a message to the bot depending on the
                                        type provided
76                                  if($_POST["type"] == "xml")
77                                          $conn->message("bot@qrro2.servegame.
                                                com", "/avatar_xml " .
                                                htmlspecialchars(stripslashes(
                                                $_POST["name"])) . " ". $service);
78                                  else
79                                          $conn->message("bot@qrro2.servegame.
                                                com", "/avatar " .
                                                htmlspecialchars(stripslashes(
                                                $_POST["name"])) . " " . $service);
80                                  break;
81                          }
```

```
82                              }
83                      }
84              } catch(XMPPHP_Exception $e) {
85                      print $e;
86              }
87      } else {
88  ?>
89      <form action="index.php" method="POST">
90              <p><label for="type">Display Type</label> <select id="type" name="type">
91                      <option value="plain">PlainText</option>
92                      <option value="xml">XML</option>
93              </select></p>
94              <p><label for="name">Avatar Name</label> <input type="text" value="" id
                    ="name" name="name" /></p>
95              <p id="service_p"><label for="service">Service</label> <select id="service"
                    name="service">
96                      <option value="ro">Ragnarok Online</option>
97                      <option value="wow">World of Warcraft</option>
98              </select></p>
99              <p id="server_p"><label for="server">Server</label> 
100                     <select id="server" name="server">
101                             <option value="Aegwynn">Aegwynn</option>
102                             <option value="Aerie Peak">Aerie Peak</option>
103                             <option value="Agamaggan">Agamaggan</option>
104                             <option value="Aggramar">Aggramar</option>
105                             <option value="Akama">Akama</option>
106                             <option value="Alexstrasza">Alexstrasza</option>
107                             <option value="Alleria">Alleria</option>
108                             <option value="Altar of Storms">Altar of Storms</option>
109                             <option value="Alterac Mountains">Alterac Mountains</option>
110                             <option value="Aman'Thul">Aman'Thul</option>
111                             <option value="Andorhal">Andorhal</option>
```



```
112                 <option value="Anetheron">Anetheron</option>
113                 <option value="Antonidas">Antonidas</option>
114                 <option value="Anub'arak">Anub'arak</option>
115                 <option value="Anvilmar">Anvilmar</option>
116                 <option value="Arathor">Arathor</option>
117                 <option value="Archimonde">Archimonde</option>
118                 <option value="Area 52">Area 52</option>
119                 <option value="Argent Dawn">Argent Dawn</option>
120                 <option value="Arthas">Arthas</option>
121                 <option value="Arygos">Arygos</option>
122                 <option value="Auchindoun">Auchindoun</option>
123                 <option value="Azgalor">Azgalor</option>
124                 <option value="Azjol-Nerub">Azjol-Nerub</option>
125                 <option value="Azshara">Azshara</option>
126                 <option value="Azuremyst">Azuremyst</option>
127                 <option value="Baelgun">Baelgun</option>
128                 <option value="Balnazzar">Balnazzar</option>
129                 <option value="Barthilas">Barthilas</option>
130                 <option value="Black Dragonflight">Black Dragonflight</option>
131                 <option value="Blackhand">Blackhand</option>
132                 <option value="Blackrock">Blackrock</option>
133                 <option value="Blackwater Raiders">Blackwater Raiders</option>
134                 <option value="Blackwing Lair">Blackwing Lair</option>
135                 <option value="Blade's Edge">Blade's Edge</option>
136                 <option value="Bladefist">Bladefist</option>
137                 <option value="Bleeding Hol">Bleeding Hol</option>
138                 <option value="Blood Furnace">Blood Furnace</option>
139                 <option value="Bloodhoof">Bloodhoof</option>
140                 <option value="Bloodscalp">Bloodscalp</option>
141                 <option value="Bonechewer">Bonechewer</option>
142                 <option value="Borean Tundra">Borean Tundra</option>
143                 <option value="Boulderfist">Boulderfist</option>
```

```
144                 <option value="Bronzebeard">Bronzebeard</option>
145                 <option value="Burning Blade">Burning Blade</option>
146                 <option value="Burning Legion">Burning Legion</option>
147                 <option value="Caelestrasz">Caelestrasz</option>
148                 <option value="Cairne">Cairne</option>
149                 <option value="Cenarion Circle">Cenarion Circle</option>
150                 <option value="Cenarius">Cenarius</option>
151                 <option value="Cho'gall">Cho'gall</option>
152                 <option value="Chromaggus">Chromaggus</option>
153                 <option value="Coilfang">Coilfang</option>
154                 <option value="Crushridge">Crushridge</option>
155                 <option value="Daggerspine">Daggerspine</option>
156                 <option value="Dalaran">Dalaran</option>
157                 <option value="Dalvengyr">Dalvengyr</option>
158                 <option value="Dark Iron">Dark Iron</option>
159                 <option value="Darkspear">Darkspear</option>
160                 <option value="Darrowmere">Darrowmere</option>
161                 <option value="Dath'Remar">Dath'Remar</option>
162                 <option value="Dawnbringer">Dawnbringer</option>
163                 <option value="Deathwing" selected="selected">Deathwing</option>
164                 <option value="Demon Soul">Demon Soul</option>
165                 <option value="Dentarg">Dentarg</option>
166                 <option value="Destromath">Destromath</option>
167                 <option value="Dethecus">Dethecus</option>
168                 <option value="Detheroc">Detheroc</option>
169                 <option value="Doomhammer">Doomhammer</option>
170                 <option value="Draenor">Draenor</option>
171                 <option value="Dragonblight">Dragonblight</option>
172                 <option value="Dragonmaw">Dragonmaw</option>
173                 <option value="Drak'Tharon">Drak'Tharon</option>
174                 <option value="Drak'thul">Drak'thul</option>
175                 <option value="Draka">Draka</option>
```

```
176                 <option value="Drakkari">Drakkari</option>
177                 <option value="Dreadmaul">Dreadmaul</option>
178                 <option value="Drenden">Drenden</option>
179                 <option value="Dunemaul">Dunemaul</option>
180                 <option value="Durotan">Durotan</option>
181                 <option value="Duskwood">Duskwood</option>
182                 <option value="Earthen Ring">Earthen Ring</option>
183                 <option value="Echo Isles">Echo Isles</option>
184                 <option value="Eitrigg">Eitrigg</option>
185                 <option value="Eldre'Thalas">Eldre'Thalas</option>
186                 <option value="Elune">Elune</option>
187                 <option value="Emerald Dream">Emerald Dream</option>
188                 <option value="Eonar">Eonar</option>
189                 <option value="Eredar">Eredar</option>
190                 <option value="Executus">Executus</option>
191                 <option value="Exodar">Exodar</option>
192                 <option value="Farstriders">Farstriders</option>
193                 <option value="Feathermoon">Feathermoon</option>
194                 <option value="Fenris">Fenris</option>
195                 <option value="Firetree">Firetree</option>
196                 <option value="Fizzcrank">Fizzcrank</option>
197                 <option value="Frostmane">Frostmane</option>
198                 <option value="Frostmourne">Frostmourne</option>
199                 <option value="Frostwolf">Frostwolf</option>
200                 <option value="Galakrond">Galakrond</option>
201                 <option value="Garithos">Garithos</option>
202                 <option value="Garona">Garona</option>
203                 <option value="Garrosh">Garrosh</option>
204                 <option value="Ghostlands">Ghostlands</option>
205                 <option value="Gilneas">Gilneas</option>
206                 <option value="Gnomeregan">Gnomeregan</option>
207                 <option value="Gorefiend">Gorefiend</option>
```

```
208                 <option value="Gorgonnash">Gorgonnash</option>
209                 <option value="Greymane">Greymane</option>
210                 <option value="Grizzly Hills">Grizzly Hills</option>
211                 <option value="Gul'dan">Gul'dan</option>
212                 <option value="Gundrak">Gundrak</option>
213                 <option value="Gurubashi">Gurubashi</option>
214                 <option value="Hakkar">Hakkar</option>
215                 <option value="Haomarush">Haomarush</option>
216                 <option value="Hellscream">Hellscream</option>
217                 <option value="Hydraxis">Hydraxis</option>
218                 <option value="Hyjal">Hyjal</option>
219                 <option value="Icecrown">Icecrown</option>
220                 <option value="Illidan">Illidan</option>
221                 <option value="Jaedenar">Jaedenar</option>
222                 <option value="Jubei'Thos">Jubei'Thos</option>
223                 <option value="Kael'thas">Kael'thas</option>
224                 <option value="Kalecgos">Kalecgos</option>
225                 <option value="Kargath">Kargath</option>
226                 <option value="Kel'Thuzad">Kel'Thuzad</option>
227                 <option value="Khadgar">Khadgar</option>
228                 <option value="Khaz Modan">Khaz Modan</option>
229                 <option value="Khaz'goroth">Khaz'goroth</option>
230                 <option value="Kil'jaeden">Kil'jaeden</option>
231                 <option value="Kilrogg">Kilrogg</option>
232                 <option value="Kirin Tor">Kirin Tor</option>
233                 <option value="Korgath">Korgath</option>
234                 <option value="Korialstrasz">Korialstrasz</option>
235                 <option value="Kul Tiras">Kul Tiras</option>
236                 <option value="Laughing Skull">Laughing Skull</option>
237                 <option value="Lethon">Lethon</option>
238                 <option value="Lightbringer">Lightbringer</option>
239                 <option value="Lightning's Blade">Lightning's Blade</option>
```



```
240                 <option value="Lightninghoof">Lightninghoof</option>
241                 <option value="Llane">Llane</option>
242                 <option value="Lothar">Lothar</option>
243                 <option value="Madoran">Madoran</option>
244                 <option value="Maelstrom">Maelstrom</option>
245                 <option value="Magtheridon">Magtheridon</option>
246                 <option value="Maiev">Maiev</option>
247                 <option value="Mal'Ganis">Mal'Ganis</option>
248                 <option value="Malfurion">Malfurion</option>
249                 <option value="Malorne">Malorne</option>
250                 <option value="Malygos">Malygos</option>
251                 <option value="Mannoroth">Mannoroth</option>
252                 <option value="Medivh">Medivh</option>
253                 <option value="Misha">Misha</option>
254                 <option value="Mok'Nathal">Mok'Nathal</option>
255                 <option value="Moon Guard">Moon Guard</option>
256                 <option value="Moonrunner">Moonrunner</option>
257                 <option value="Mug'thol">Mug'thol</option>
258                 <option value="Muradin">Muradin</option>
259                 <option value="Nagrand">Nagrand</option>
260                 <option value="Nathrezim">Nathrezim</option>
261                 <option value="Nazgrel">Nazgrel</option>
262                 <option value="Nazjatar">Nazjatar</option>
263                 <option value="Ner'zhul">Ner'zhul</option>
264                 <option value="Nesingwary">Nesingwary</option>
265                 <option value="Nordrassil">Nordrassil</option>
266                 <option value="Norgannon">Norgannon</option>
267                 <option value="Onyxia">Onyxia</option>
268                 <option value="Perenolde">Perenolde</option>
269                 <option value="Proudmoore">Proudmoore</option>
270                 <option value="Quel'dorei">Quel'dorei</option>
271                 <option value="Quel'Thalas">Quel'Thalas</option>
```



```
272                 <option value="Ragnaros">Ragnaros</option>
273                 <option value="Ravencrest">Ravencrest</option>
274                 <option value="Ravenholdt">Ravenholdt</option>
275                 <option value="Rexxar">Rexxar</option>
276                 <option value="Rivendare">Rivendare</option>
277                 <option value="Runetotem">Runetotem</option>
278                 <option value="Sargeras">Sargeras</option>
279                 <option value="Saurfang">Saurfang</option>
280                 <option value="Scarlet Crusade">Scarlet Crusade</option>
281                 <option value="Scilla">Scilla</option>
282                 <option value="Sen'jin">Sen'jin</option>
283                 <option value="Sentinels">Sentinels</option>
284                 <option value="Shadow Council">Shadow Council</option>
285                 <option value="Shadowmoon">Shadowmoon</option>
286                 <option value="Shadowsong">Shadowsong</option>
287                 <option value="Shandris">Shandris</option>
288                 <option value="Shattered Halls">Shattered Halls</option>
289                 <option value="Shattered Hand">Shattered Hand</option>
290                 <option value="Shu'halo">Shu'halo</option>
291                 <option value="Silver Hand">Silver Hand</option>
292                 <option value="Silvermoon">Silvermoon</option>
293                 <option value="Sisters of Elune">Sisters of Elune</option>
294                 <option value="Skullcrusher">Skullcrusher</option>
295                 <option value="Skywall">Skywall</option>
296                 <option value="Smolderthorn">Smolderthorn</option>
297                 <option value="Spinebreaker">Spinebreaker</option>
298                 <option value="Spirestone">Spirestone</option>
299                 <option value="Staghelm">Staghelm</option>
300                 <option value="Steamwheedle Cartel">Steamwheedle Cartel</option>
301                 <option value="Stonemaul">Stonemaul</option>
302                 <option value="Stormrage">Stormrage</option>
303                 <option value="Stormreaver">Stormreaver</option>
```



```
304                 <option value="Stormscale">Stormscale</option>
305                 <option value="Suramar">Suramar</option>
306                 <option value="Tanaris">Tanaris</option>
307                 <option value="Terenas">Terenas</option>
308                 <option value="Terokkar">Terokkar</option>
309                 <option value="Thaurissan">Thaurissan</option>
310                 <option value="The Forgotten Coast">The Forgotten Coast</option>
311                 <option value="The Scryers">The Scryers</option>
312                 <option value="The Underbog">The Underbog</option>
313                 <option value="The Venture Co">The Venture Co</option>
314                 <option value="Thorium Brotherhood">Thorium Brotherhood</option
                        >
315                 <option value="Thrall">Thrall</option>
316                 <option value="Thunderhorn">Thunderhorn</option>
317                 <option value="Thunderlord">Thunderlord</option>
318                 <option value="Tichondrius">Tichondrius</option>
319                 <option value="Tortheldrin">Tortheldrin</option>
320                 <option value="Trollbane">Trollbane</option>
321                 <option value="Turalyon">Turalyon</option>
322                 <option value="Twisting Nether">Twisting Nether</option>
323                 <option value="Uldaman">Uldaman</option>
324                 <option value="Uldum">Uldum</option>
325                 <option value="Undermine">Undermine</option>
326                 <option value="Ursin">Ursin</option>
327                 <option value="Uther">Uther</option>
328                 <option value="Vashj">Vashj</option>
329                 <option value="Vek'nilash">Vek'nilash</option>
330                 <option value="Velen">Velen</option>
331                 <option value="Warsong">Warsong</option>
332                 <option value="Whisperwind">Whisperwind</option>
333                 <option value="Wildhammer">Wildhammer</option>
334                 <option value="Windrunner">Windrunner</option>
```



```
335                                 <option value="Winterhoof">Winterhoof</option>
336                                 <option value="Wyrmrest Accord">Wyrmrest Accord</option>
337                                 <option value="Ysera">Ysera</option>
338                                 <option value="Ysondre">Ysondre</option>
339                                 <option value="Zangarmarsh">Zangarmarsh</option>
340                                 <option value="Zul'jin">Zul'jin</option>
341                                 <option value="Zuluhed">Zuluhed</option>
342                         </select>
343                 </p>
344                 <p><input type="submit" value="Obtain Information" /></p>
345         </form>
346 <?php
347         }
348
349         if(!isset($_POST["type"]) || $_POST["type"] != "xml") {
350 ?>
351 </body>
352 </html>
353 <?php
354         }
355 ?>
```

C Final Demonstration Client Source Code

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/
       DTD/xhtml11.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4          <title>Capstone Demonstraton #2</title>
5          <style type="text/css">
6                  div#new_avatar_preview {
7                          width: 37px;
8                          height: 100px;
9                  }
10         </style>
11         <script type="text/javascript" src="http://www.google.com/jsapi"></script>
12         <script type="text/javascript">
13                 // Load jQuery
14                 google.load("jquery", "1.3.2");
15
16                 // Generate the avatar based on selected attributes
17                 function generateAvatar() {
18                         var gender = $("select#new_avatar_gender").val().toLowerCase();
19                         var avatar_class = $("select#new_avatar_class").val();
20                         var hairstyle = $("select#new_avatar_hairstyle").val();
21                         var haircolor = $("select#new_avatar_haircolor").val();
22                         var clothcolor = $("select#new_avatar_clothcolor").val();
23
24                         var image_url = "";
25
26                         if(gender == 'm')
27                                 image_url += "./images/1/";
28                         else if(gender == 'f')
29                                 image_url += "./images/0/";
30                         else {
```



```
31                          return;
32                  }
33
34                  if(avatar_class > 0)
35                          image_url += avatar_class + "/";
36                  else {
37                          return;
38                  }
39
40                  if(hairstyle > 0)
41                          image_url += hairstyle + "/";
42                  else {
43                          return;
44                  }
45
46                  if(haircolor > 0)
47                          image_url += haircolor + "/";
48                  else {
49                          return;
50                  }
51
52                  if(clothcolor >= 0)
53                          image_url += clothcolor + ".png"
54                  else {
55                          return;
56                  }
57
58                  $("div#new_avatar_preview").css("background", "url('" + image_url + "')");
59                  $("div#new_avatar_preview").css("background-position", "-81px -49px");
60
61                  return;
62          }
```

```
63          </script>
64  </head>
65  <body>
66          <?php
67          if(!isset($_POST["create_ro_avatar"])) {
68                  if(isset($_POST["transfer"])) {
69                          require_once("XMPPHP/XMPP.php");
70
71                          try {
72                                  // Connect to the XMPP Server
73                                  $conn = new XMPPHP_XMPP("qrro2.servegame.com", 5222, "client", "
                                        garden", "AvatarDemo2", "qrro2.servegame.com");
74                                  $conn->connect();
75
76                                  // While we are still connected...
77                                  while(!$conn->isDisconnected()) {
78                                          // Process various things until the events "message" or "
                                                session_start"
79                                          $events = $conn->processUntil(array(
80                                                  "message", "session_start"
81                                          ));
82                                          // Utilizing the data provided...
83                                          foreach($events as $event) {
84                                                  // Switch on the event name
85                                                  switch($event[0]) {
86                                                          case "message":
87                                                                  // Let's split out the body! That's the
                                                                        content of the message!
88                                                                  $avatar_xml = $event[1]["body"];
89                                                                  $conn->disconnect();
90                                                                  break;
91                                                          case "session_start":
```

```
92                                                  // Set that we are present and alive!
93                                                  $conn->presence("Pulling Information!"
                                                        );
94                                                  $conn->message("bot@qrro2.servegame.
                                                        com", "/avatar_xml " .
                                                        htmlspecialchars(stripslashes(
                                                        $_POST["wow_name"])) . " wow:" .
                                                        htmlspecialchars(stripslashes(
                                                        $_POST["wow_server"])));
95                                                  break;
96                                              }
97                                          }
98                                      }
99                      } catch(XMPPHP_Exception $e) {
100                         print $e;
101                     }
102             }
103     ?>
104     <?php
105             if(!isset($_POST["transfer"])) {
106     ?>
107             <div id="source_div">
108                     <h1>Source Avatar</h1>
109                     <form action="./index2.php" method="post">
110                             <p><label for="wow_name">Name</label> <input type="text"
                                    id="wow_name" name="wow_name" /></p>
111                             <p><label for="wow_server">Server</label> 
112                             <select id="wow_server" name="wow_server">
113                                     <option value="Aegwynn">Aegwynn</option>
114                                     <option value="Aerie Peak">Aerie Peak</option>
115                                     <option value="Agamaggan">Agamaggan</option>
116                                     <option value="Aggramar">Aggramar</option>
```



```
117                 <option value="Akama">Akama</option>
118                 <option value="Alexstrasza">Alexstrasza</option>
119                 <option value="Alleria">Alleria</option>
120                 <option value="Altar of Storms">Altar of Storms</
                        option>
121                 <option value="Alterac Mountains">Alterac Mountains
                        </option>
122                 <option value="Aman'Thul">Aman'Thul</option>
123                 <option value="Andorhal">Andorhal</option>
124                 <option value="Anetheron">Anetheron</option>
125                 <option value="Antonidas">Antonidas</option>
126                 <option value="Anub'arak">Anub'arak</option>
127                 <option value="Anvilmar">Anvilmar</option>
128                 <option value="Arathor">Arathor</option>
129                 <option value="Archimonde">Archimonde</option>
130                 <option value="Area 52">Area 52</option>
131                 <option value="Argent Dawn">Argent Dawn</option>
132                 <option value="Arthas">Arthas</option>
133                 <option value="Arygos">Arygos</option>
134                 <option value="Auchindoun">Auchindoun</option>
135                 <option value="Azgalor">Azgalor</option>
136                 <option value="Azjol-Nerub">Azjol-Nerub</option>
137                 <option value="Azshara">Azshara</option>
138                 <option value="Azuremyst">Azuremyst</option>
139                 <option value="Baelgun">Baelgun</option>
140                 <option value="Balnazzar">Balnazzar</option>
141                 <option value="Barthilas">Barthilas</option>
142                 <option value="Black Dragonflight">Black Dragonflight
                        </option>
143                 <option value="Blackhand">Blackhand</option>
144                 <option value="Blackrock">Blackrock</option>
```

```
145                 <option value="Blackwater Raiders">Blackwater
                        Raiders</option>
146                 <option value="Blackwing Lair">Blackwing Lair</
                        option>
147                 <option value="Blade's Edge">Blade's Edge</option>
148                 <option value="Bladefist">Bladefist</option>
149                 <option value="Bleeding Hol">Bleeding Hol</option>
150                 <option value="Blood Furnace">Blood Furnace</
                        option>
151                 <option value="Bloodhoof">Bloodhoof</option>
152                 <option value="Bloodscalp">Bloodscalp</option>
153                 <option value="Bonechewer">Bonechewer</option>
154                 <option value="Borean Tundra">Borean Tundra</
                        option>
155                 <option value="Boulderfist">Boulderfist</option>
156                 <option value="Bronzebeard">Bronzebeard</option>
157                 <option value="Burning Blade">Burning Blade</
                        option>
158                 <option value="Burning Legion">Burning Legion</
                        option>
159                 <option value="Caelestrasz">Caelestrasz</option>
160                 <option value="Cairne">Cairne</option>
161                 <option value="Cenarion Circle">Cenarion Circle</
                        option>
162                 <option value="Cenarius">Cenarius</option>
163                 <option value="Cho'gall">Cho'gall</option>
164                 <option value="Chromaggus">Chromaggus</option>
165                 <option value="Coilfang">Coilfang</option>
166                 <option value="Crushridge">Crushridge</option>
167                 <option value="Daggerspine">Daggerspine</option>
168                 <option value="Dalaran">Dalaran</option>
169                 <option value="Dalvengyr">Dalvengyr</option>
```

```
170                 <option value="Dark Iron">Dark Iron</option>
171                 <option value="Darkspear">Darkspear</option>
172                 <option value="Darrowmere">Darrowmere</option>
173                 <option value="Dath'Remar">Dath'Remar</option>
174                 <option value="Dawnbringer">Dawnbringer</option>
175                 <option value="Deathwing" selected="selected">
                        Deathwing</option>
176                 <option value="Demon Soul">Demon Soul</option>
177                 <option value="Dentarg">Dentarg</option>
178                 <option value="Destromath">Destromath</option>
179                 <option value="Dethecus">Dethecus</option>
180                 <option value="Detheroc">Detheroc</option>
181                 <option value="Doomhammer">Doomhammer</option
                        >
182                 <option value="Draenor">Draenor</option>
183                 <option value="Dragonblight">Dragonblight</option>
184                 <option value="Dragonmaw">Dragonmaw</option>
185                 <option value="Drak'Tharon">Drak'Tharon</option>
186                 <option value="Drak'thul">Drak'thul</option>
187                 <option value="Draka">Draka</option>
188                 <option value="Drakkari">Drakkari</option>
189                 <option value="Dreadmaul">Dreadmaul</option>
190                 <option value="Drenden">Drenden</option>
191                 <option value="Dunemaul">Dunemaul</option>
192                 <option value="Durotan">Durotan</option>
193                 <option value="Duskwood">Duskwood</option>
194                 <option value="Earthen Ring">Earthen Ring</option>
195                 <option value="Echo Isles">Echo Isles</option>
196                 <option value="Eitrigg">Eitrigg</option>
197                 <option value="Eldre'Thalas">Eldre'Thalas</option>
198                 <option value="Elune">Elune</option>
```

```
199                 <option value="Emerald Dream">Emerald Dream</
                        option>
200                 <option value="Eonar">Eonar</option>
201                 <option value="Eredar">Eredar</option>
202                 <option value="Executus">Executus</option>
203                 <option value="Exodar">Exodar</option>
204                 <option value="Farstriders">Farstriders</option>
205                 <option value="Feathermoon">Feathermoon</option>
206                 <option value="Fenris">Fenris</option>
207                 <option value="Firetree">Firetree</option>
208                 <option value="Fizzcrank">Fizzcrank</option>
209                 <option value="Frostmane">Frostmane</option>
210                 <option value="Frostmourne">Frostmourne</option>
211                 <option value="Frostwolf">Frostwolf</option>
212                 <option value="Galakrond">Galakrond</option>
213                 <option value="Garithos">Garithos</option>
214                 <option value="Garona">Garona</option>
215                 <option value="Garrosh">Garrosh</option>
216                 <option value="Ghostlands">Ghostlands</option>
217                 <option value="Gilneas">Gilneas</option>
218                 <option value="Gnomeregan">Gnomeregan</option>
219                 <option value="Gorefiend">Gorefiend</option>
220                 <option value="Gorgonnash">Gorgonnash</option>
221                 <option value="Greymane">Greymane</option>
222                 <option value="Grizzly Hills">Grizzly Hills</option>
223                 <option value="Gul'dan">Gul'dan</option>
224                 <option value="Gundrak">Gundrak</option>
225                 <option value="Gurubashi">Gurubashi</option>
226                 <option value="Hakkar">Hakkar</option>
227                 <option value="Haomarush">Haomarush</option>
228                 <option value="Hellscream">Hellscream</option>
229                 <option value="Hydraxis">Hydraxis</option>
```

```
230                 <option value="Hyjal">Hyjal</option>
231                 <option value="Icecrown">Icecrown</option>
232                 <option value="Illidan">Illidan</option>
233                 <option value="Jaedenar">Jaedenar</option>
234                 <option value="Jubei'Thos">Jubei'Thos</option>
235                 <option value="Kael'thas">Kael'thas</option>
236                 <option value="Kalecgos">Kalecgos</option>
237                 <option value="Kargath">Kargath</option>
238                 <option value="Kel'Thuzad">Kel'Thuzad</option>
239                 <option value="Khadgar">Khadgar</option>
240                 <option value="Khaz Modan">Khaz Modan</option>
241                 <option value="Khaz'goroth">Khaz'goroth</option>
242                 <option value="Kil'jaeden">Kil'jaeden</option>
243                 <option value="Kilrogg">Kilrogg</option>
244                 <option value="Kirin Tor">Kirin Tor</option>
245                 <option value="Korgath">Korgath</option>
246                 <option value="Korialstrasz">Korialstrasz</option>
247                 <option value="Kul Tiras">Kul Tiras</option>
248                 <option value="Laughing Skull">Laughing Skull</
                        option>
249                 <option value="Lethon">Lethon</option>
250                 <option value="Lightbringer">Lightbringer</option>
251                 <option value="Lightning's Blade">Lightning's Blade</
                        option>
252                 <option value="Lightninghoof">Lightninghoof</option
                        >
253                 <option value="Llane">Llane</option>
254                 <option value="Lothar">Lothar</option>
255                 <option value="Madoran">Madoran</option>
256                 <option value="Maelstrom">Maelstrom</option>
257                 <option value="Magtheridon">Magtheridon</option>
258                 <option value="Maiev">Maiev</option>
```

```
259                 <option value="Mal'Ganis">Mal'Ganis</option>
260                 <option value="Malfurion">Malfurion</option>
261                 <option value="Malorne">Malorne</option>
262                 <option value="Malygos">Malygos</option>
263                 <option value="Mannoroth">Mannoroth</option>
264                 <option value="Medivh">Medivh</option>
265                 <option value="Misha">Misha</option>
266                 <option value="Mok'Nathal">Mok'Nathal</option>
267                 <option value="Moon Guard">Moon Guard</option>
268                 <option value="Moonrunner">Moonrunner</option>
269                 <option value="Mug'thol">Mug'thol</option>
270                 <option value="Muradin">Muradin</option>
271                 <option value="Nagrand">Nagrand</option>
272                 <option value="Nathrezim">Nathrezim</option>
273                 <option value="Nazgrel">Nazgrel</option>
274                 <option value="Nazjatar">Nazjatar</option>
275                 <option value="Ner'zhul">Ner'zhul</option>
276                 <option value="Nesingwary">Nesingwary</option>
277                 <option value="Nordrassil">Nordrassil</option>
278                 <option value="Norgannon">Norgannon</option>
279                 <option value="Onyxia">Onyxia</option>
280                 <option value="Perenolde">Perenolde</option>
281                 <option value="Proudmoore">Proudmoore</option>
282                 <option value="Quel'dorei">Quel'dorei</option>
283                 <option value="Quel'Thalas">Quel'Thalas</option>
284                 <option value="Ragnaros">Ragnaros</option>
285                 <option value="Ravencrest">Ravencrest</option>
286                 <option value="Ravenholdt">Ravenholdt</option>
287                 <option value="Rexxar">Rexxar</option>
288                 <option value="Rivendare">Rivendare</option>
289                 <option value="Runetotem">Runetotem</option>
290                 <option value="Sargeras">Sargeras</option>
```

```
291                 <option value="Saurfang">Saurfang</option>
292                 <option value="Scarlet Crusade">Scarlet Crusade</
                        option>
293                 <option value="Scilla">Scilla</option>
294                 <option value="Sen'jin">Sen'jin</option>
295                 <option value="Sentinels">Sentinels</option>
296                 <option value="Shadow Council">Shadow Council</
                        option>
297                 <option value="Shadowmoon">Shadowmoon</option>
298                 <option value="Shadowsong">Shadowsong</option>
299                 <option value="Shandris">Shandris</option>
300                 <option value="Shattered Halls">Shattered Halls</
                        option>
301                 <option value="Shattered Hand">Shattered Hand</
                        option>
302                 <option value="Shu'halo">Shu'halo</option>
303                 <option value="Silver Hand">Silver Hand</option>
304                 <option value="Silvermoon">Silvermoon</option>
305                 <option value="Sisters of Elune">Sisters of Elune</
                        option>
306                 <option value="Skullcrusher">Skullcrusher</option>
307                 <option value="Skywall">Skywall</option>
308                 <option value="Smolderthorn">Smolderthorn</option
                        >
309                 <option value="Spinebreaker">Spinebreaker</option>
310                 <option value="Spirestone">Spirestone</option>
311                 <option value="Staghelm">Staghelm</option>
312                 <option value="Steamwheedle Cartel">Steamwheedle
                        Cartel</option>
313                 <option value="Stonemaul">Stonemaul</option>
314                 <option value="Stormrage">Stormrage</option>
315                 <option value="Stormreaver">Stormreaver</option>
```

```
316                 <option value="Stormscale">Stormscale</option>
317                 <option value="Suramar">Suramar</option>
318                 <option value="Tanaris">Tanaris</option>
319                 <option value="Terenas">Terenas</option>
320                 <option value="Terokkar">Terokkar</option>
321                 <option value="Thaurissan">Thaurissan</option>
322                 <option value="The Forgotten Coast">The Forgotten
                        Coast</option>
323                 <option value="The Scryers">The Scryers</option>
324                 <option value="The Underbog">The Underbog</
                        option>
325                 <option value="The Venture Co">The Venture Co</
                        option>
326                 <option value="Thorium Brotherhood">Thorium
                        Brotherhood</option>
327                 <option value="Thrall">Thrall</option>
328                 <option value="Thunderhorn">Thunderhorn</option>
329                 <option value="Thunderlord">Thunderlord</option>
330                 <option value="Tichondrius">Tichondrius</option>
331                 <option value="Tortheldrin">Tortheldrin</option>
332                 <option value="Trollbane">Trollbane</option>
333                 <option value="Turalyon">Turalyon</option>
334                 <option value="Twisting Nether">Twisting Nether</
                        option>
335                 <option value="Uldaman">Uldaman</option>
336                 <option value="Uldum">Uldum</option>
337                 <option value="Undermine">Undermine</option>
338                 <option value="Ursin">Ursin</option>
339                 <option value="Uther">Uther</option>
340                 <option value="Vashj">Vashj</option>
341                 <option value="Vek'nilash">Vek'nilash</option>
342                 <option value="Velen">Velen</option>
```

```
343                                         <option value="Warsong">Warsong</option>
344                                         <option value="Whisperwind">Whisperwind</option>
345                                         <option value="Wildhammer">Wildhammer</option>
346                                         <option value="Windrunner">Windrunner</option>
347                                         <option value="Winterhoof">Winterhoof</option>
348                                         <option value="Wyrmrest Accord">Wyrmrest Accord
                                                </option>
349                                         <option value="Ysera">Ysera</option>
350                                         <option value="Ysondre">Ysondre</option>
351                                         <option value="Zangarmarsh">Zangarmarsh</option>
352                                         <option value="Zul'jin">Zul'jin</option>
353                                         <option value="Zuluhed">Zuluhed</option>
354                                     </select>
355                                 </p>
356                                 <p><input type="submit" value="Transfer" name="transfer" /></p>
357                             </form>
358                     </div>
359             <?php
360                     }
361
362
363                     if(isset($_POST["transfer"])) {
364             ?>
365             <div id="dest_div">
366                     <h1>Destination Avatar</h1>
367 <?php
368             // If we have Avatar XML data...
369             if(isset($avatar_xml)) {
370                     // Parse the data!
371                     $avatar_dom = new DOMDocument();
372                     $avatar_dom->loadXML($avatar_xml);
373
```

```
374             // Parse out the name, if possible
375             $node_list = $avatar_dom->getElementsByTagName("name");
376             foreach($node_list as $node)
377                     $avatar_name = $node->nodeValue;
378
379             // Parse out the gender, if possible
380             $node_list = $avatar_dom->getElementsByTagName("gender");
381             foreach($node_list as $node)
382                     $avatar_gender = $node->nodeValue;
383
384             // Parse out the class name, if posible
385             $node_list = $avatar_dom->getElementsByTagName("class");
386             foreach($node_list as $node)
387                     $avatar_class = $node->nodeValue;
388
389             // Use "business logic" to determine what class they should use in the destination game
390             // Based upon the class they used in the source game
391             if(isset($avatar_class)) {
392                     switch(strtolower($avatar_class)) {
393                             case 'death knight':
394                                     $recommended_class = "swordsman";
395                                     break;
396                             case 'druid':
397                                     $recommended_class = "merchant";
398                                     break;
399                             case 'hunter':
400                                     $recommended_class = "archer";
401                                     break;
402                             case 'mage':
403                                     $recommended_class = "magician";
404                                     break;
405                             case 'paladin':
```

```
406                                     $recommended_class = ”swordsman”;
407                                     break;
408                             case 'priest':
409                                     $recommended_class = ”acolyte”;
410                                     break;
411                             case 'rogue':
412                                     $recommended_class = ”thief”;
413                                     break;
414                             case 'shaman':
415                                     $recommended_class = ”acolyte”;
416                                     break;
417                             case 'warlock':
418                                     $recommended_class = ”acolyte”;
419                                     break;
420                             case 'warrior':
421                                     $recommended_class = ”swordsman”;
422                                     break;
423                     }
424             }
425     }
426 ?>
427             <form action=”./index2.php” method=”post”>
428                     <p><label for=”new_avatar_name”>Avatar Name</label> <input type
                            =”text” id=”new_avatar_name” name=”new_avatar_name” value=”<?=((
                            isset($avatar_name)) ? $avatar_name : ””)?>” /></p>
429                     <p><label for=”new_avatar_gender”>Avatar Gender</label> 
430                             <select id=”new_avatar_gender” name=”new_avatar_gender” onchange=”
                                    generateAvatar();”>
431                                     <option value=”S”>Select a Gender</option>
432                                     <option value=”M”<?=((isset($avatar_gender) && strtolower(
                                            $avatar_gender) == ”m”) ? ” selected=\”selected\”” : ””)
                                            ?>>Male</option>
```



```
433                         <option value="F"<?=((isset($avatar_gender) && strtolower(
                                $avatar_gender) == "f") ? " selected=\"selected\"" : "")
                                ?>>Female</option>
434                     </select>
435                 </p>
436                 <p><label for="new_avatar_class">Avatar Class</label> 
437                     <select id="new_avatar_class" name="new_avatar_class" onchange="
                            generateAvatar();">
438                         <option value="0">Select a Class</option>
439                         <option value="1"<?=((isset($recommended_class) &&
                                $recommended_class == "swordsman") ? " selected=\"
                                selected\"" : "")?>>Swordsman</option>
440                         <option value="2"<?=((isset($recommended_class) &&
                                $recommended_class == "magician") ? " selected=\"selected
                                \"" : "")?>>Magician</option>
441                         <option value="3"<?=((isset($recommended_class) &&
                                $recommended_class == "archer") ? " selected=\"selected
                                \"" : "")?>>Archer</option>
442                         <option value="4"<?=((isset($recommended_class) &&
                                $recommended_class == "acolyte") ? " selected=\"selected
                                \"" : "")?>>Acolyte</option>
443                         <option value="5"<?=((isset($recommended_class) &&
                                $recommended_class == "merchant") ? " selected=\"
                                selected\"" : "")?>>Merchant</option>
444                         <option value="6"<?=((isset($recommended_class) &&
                                $recommended_class == "thief") ? " selected=\"selected\""
                                : "")?>>Thief</option>
445                     </select>
446                 </p>
447                 <p><label for="new_avatar_hairstyle">Avatar Hair Style</label> 
448                     <select id="new_avatar_hairstyle" name="new_avatar_hairstyle"
                            onchange="generateAvatar();">
```



```
449                         <option value="0">Select a Hair Style</option>
450                         <option value="1">Hair Style #1</option>
451                         <option value="2">Hair Style #2</option>
452                         <option value="3">Hair Style #3</option>
453                         <option value="4">Hair Style #4</option>
454                         <option value="5">Hair Style #5</option>
455                         <option value="6">Hair Style #6</option>
456                         <option value="7">Hair Style #7</option>
457                         <option value="8">Hair Style #8</option>
458                         <option value="9">Hair Style #9</option>
459                         <option value="10">Hair Style #10</option>
460                         <option value="11">Hair Style #11</option>
461                         <option value="12">Hair Style #12</option>
462                         <option value="13">Hair Style #13</option>
463                         <option value="14">Hair Style #14</option>
464                         <option value="15">Hair Style #15</option>
465                         <option value="16">Hair Style #16</option>
466                         <option value="17">Hair Style #17</option>
467                         <option value="18">Hair Style #18</option>
468                         <option value="19">Hair Style #19</option>
469                         <option value="20">Hair Style #20</option>
470                         <option value="21">Hair Style #21</option>
471                         <option value="22">Hair Style #22</option>
472                         <option value="23">Hair Style #23</option>
473                         <option value="24">Hair Style #24</option>
474                         <option value="25">Hair Style #25</option>
475                     </select>
476                 </p>
477                 <p><label for="new_avatar_haircolor">Avatar Hair Color</label> 
478                     <select id="new_avatar_haircolor" name="new_avatar_haircolor"
                        onchange="generateAvatar();">
479                         <option value="0">Select a Hair Color</option>
```

```
480                         <option value="1">Blonde</option>
481                         <option value="2">Purple</option>
482                         <option value="3">Light Brown</option>
483                         <option value="4">Green</option>
484                         <option value="5">Blue</option>
485                         <option value="6">White</option>
486                         <option value="7">Dark Brown</option>
487                         <option value="8">Red</option>
488                         <option value="9">Black</option>
489                         <option value="10">Gray</option>
490                     </select>
491                 </p>
492                 <p><label for="new_avatar_clothcolor">Avatar Clothing Color</label> 
493                     <select id="new_avatar_clothcolor" name="new_avatar_clothcolor"
                            onchange="generateAvatar();">
494                         <option value="0">Standard Coloration</option>
495                         <option value="1">Red</option>
496                         <option value="2">Blue</option>
497                         <option value="3">Green</option>
498                         <option value="4">Brown</option>
499                         <option value="5">Alternate Dark Gray</option>
500                         <option value="6">Alternate Light Gray</option>
501                         <option value="7">Alternate Brown</option>
502                     </select>
503                 </p>
504                 <div id="new_avatar_preview"></div>
505                 <p><label for="new_avatar_username">Username</label> 
506                     <input type="text" name="new_avatar_username" id="
                            new_avatar_username" />
507                 </p>
508                 <p><label for="new_avatar_password">Password</label> 
```

```
509                                 <input type="password" name="new_avatar_password" id="
                                        new_avatar_password" />
510                             </p>
511                             <p><input type="submit" value="Create Avatar" name="create_ro_avatar" id
                                    ="create_ro_avatar" /></p>
512                     </form>
513             </div>
514 <?php
515
516             }
517 } else {
518             // Step 3 - create the ragnarok avatar
519             if(isset($_POST["new_avatar_name"]) && strlen($_POST["new_avatar_name"]) > 0 &&
520             isset($_POST["new_avatar_gender"]) && $_POST["new_avatar_gender"] != "S" &&
521             isset($_POST["new_avatar_class"]) && $_POST["new_avatar_class"] > 0 &&
522             isset($_POST["new_avatar_hairstyle"]) && $_POST["new_avatar_hairstyle"] > 0 &&
523             isset($_POST["new_avatar_haircolor"]) && $_POST["new_avatar_haircolor"] > 0 &&
524             isset($_POST["new_avatar_clothcolor"]) && $_POST["new_avatar_clothcolor"] >= 0) {
525
526                     $link = new mysqli("localhost", "ragnarok", "capstone", "ragnarok");
527                     $result = $link->query("SELECT TRUE FROM `char` WHERE name = '" . $link->
                            real_escape_string($_POST["new_avatar_name"]) . "'");
528                     if($result->num_rows > 0) {
529 ?>
530                             <p>It seems the avatar '<?=$_POST["new_avatar_name"]?>' already exists!</p
                                    >
531 <?php
532                     } else {
533
534                     if(strtolower($_POST["new_avatar_gender"]) == "m")
535                             $image_url = "./images/1/";
536                     else
```



```
537                     $image_url = "./images/0/";
538                 $image_url .= $_POST["new_avatar_class"] . "/" . $_POST["new_avatar_hairstyle"] . "/" .
                        $_POST["new_avatar_haircolor"] . "/" . $_POST["new_avatar_clothcolor"] . ".png";
539
540                 $account_id = 0;
541                 $result = $link->query("SELECT account_id FROM `login` WHERE userid = '" . $link
                        ->real_escape_string($_POST["new_avatar_username"]) . "' AND user_pass = MD5('
                        " . $link->real_escape_string($_POST["new_avatar_password"]) . "')");
542                 while($row = $result->fetch_object()) {
543                     $account_id = $row->account_id;
544                 }
545
546                 if($account_id <= 0) {
547 ?>
548                     <p>It seems that your username and password were incorrect!</p>
549 <?php
550                 } else {
551                     $link->query("UPDATE login SET sex = '" . $link->real_escape_string(
                            $_POST["new_avatar_gender"]) . "' WHERE account_id = '" . $link->
                            real_escape_string($account_id) . "'");
552                     $result = $link->query("SELECT COUNT(char_id) AS \"num\" FROM `char`
                            WHERE account_id = " . $link->real_escape_string($account_id));
553                     $char_num = 1;
554                     while($obj = $result->fetch_object())
555                         $char_num = $obj->num;
556                     $result->close();
557                     $link->query("INSERT INTO `char` (account_id, char_num, name, class, str, agi
                            , vit, `int`, dex, luk, hair, hair_color, clothes_color) VALUES (" . $account_id
                            . ", " . $char_num . ", '" . $link->real_escape_string($_POST["
                            new_avatar_name"]) . "', " . $_POST["new_avatar_class"] . ", 5, 5, 5, 5, 5, 5, "
                            . $_POST["new_avatar_hairstyle"] . ", " . $_POST["new_avatar_haircolor"] .
                            ", " . $_POST["new_avatar_clothcolor"] . ")");
```



```
?>
        <div id="avatar">
                <p>Your new avatar has been created:</p>
                <h2><?=$_POST["new_avatar_name"]?></h2>
                <p><img id="avatar_image" alt="<?=$_POST["new_avatar_name"]?>" width="200"
                    height="200" src="<?=$image_url?>" /></p>
        </div>
<?php
                }
                }
                $link->close();
        }
}
?>
</body>
</html>
```

D Final Demonstration XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:element name="avatar">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="name" maxOccurs="1" minOccurs="1">
                                        <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                        <xs:whiteSpace value="collapse"/>
                                                </xs:restriction>
                                        </xs:simpleType>
                                </xs:element>
```



```
13                  <xs:element name="id" maxOccurs="1" minOccurs="0" type="
                        xs:integer" />
14                  <xs:element maxOccurs="1" minOccurs="0" name="stats">
15                      <xs:complexType>
16                          <xs:attribute name="defense" type="xs:integer"/>
17                          <xs:attribute name="dexterity" type="xs:integer"/>
18                          <xs:attribute name="intelligence" type="xs:integer"/>
19                          <xs:attribute name="speed" type="xs:integer"/>
20                          <xs:attribute name="spirit" type="xs:integer"/>
21                          <xs:attribute name="strength" type="xs:integer"/>
22                      </xs:complexType>
23                  </xs:element>
24                  <xs:element name="gender" maxOccurs="1" minOccurs="0">
25                      <xs:simpleType>
26                          <xs:restriction base="xs:string">
27                              <xs:enumeration value="M"/>
28                              <xs:enumeration value="F"/>
29                          </xs:restriction>
30                      </xs:simpleType>
31                  </xs:element>
32                  <xs:element name="class" maxOccurs="1" minOccurs="0">
33                      <xs:complexType>
34                          <xs:simpleContent>
35                              <xs:extension base="xs:string">
36                                  <xs:attribute name="id" type="
                                        xs:integer"/>
37                              </xs:extension>
38                          </xs:simpleContent>
39                      </xs:complexType>
40                  </xs:element>
41                  <xs:element name="race" maxOccurs="1" minOccurs="0">
42                      <xs:complexType>
```

```
43                          <xs:simpleContent>
44                              <xs:extension base="xs:string">
45                                  <xs:attribute name="id" type="
                                        xs:integer"/>
46                              </xs:extension>
47                          </xs:simpleContent>
48                      </xs:complexType>
49                  </xs:element>
50                  <xs:element name="lastlogin" type="xs:dateTime" maxOccurs="1"
                        minOccurs="0"/>
51                  <xs:element name="level" type="xs:integer" minOccurs="0"/>
52                  <xs:element name="sub_level" type="xs:integer" minOccurs="0"/>
53                  <xs:element name="color" maxOccurs="1" minOccurs="0">
54                      <xs:complexType>
55                          <xs:attribute name="clothes" type="xs:string"/>
56                          <xs:attribute name="hair" type="xs:string"/>
57                          <xs:attribute name="skin" type="xs:string"/>
58                      </xs:complexType>
59                  </xs:element>
60                  <xs:element name="money" maxOccurs="1" minOccurs="0" type="
                        xs:decimal" />
61                  <xs:element name="inventory" maxOccurs="unbounded" minOccurs="0
                        ">
62                      <xs:complexType>
63                          <xs:sequence>
64                              <xs:element name="item" minOccurs="1"
                                    maxOccurs="unbounded">
65                                  <xs:complexType>
66                                      <xs:attribute default="1"
                                            name="amount" type="
                                            xs:integer" use="optional"/
                                            >
```



```
67                                                          <xs:attribute name="era" type
                                                                ="xs:string"/>
68                                                          <xs:attribute name="id" type=
                                                                "xs:integer"/>
69                                                          <xs:attribute name="name"
                                                                type="xs:string"/>
70                                                          <xs:attribute name="type" type
                                                                ="xs:string"/>
71                                                      </xs:complexType>
72                                                  </xs:element>
73                                              </xs:sequence>
74                                          </xs:complexType>
75                                      </xs:element>
76                                  </xs:sequence>
77                              </xs:complexType>
78                      </xs:element>
79  </xs:schema>
```


E Business Logic Database Schema

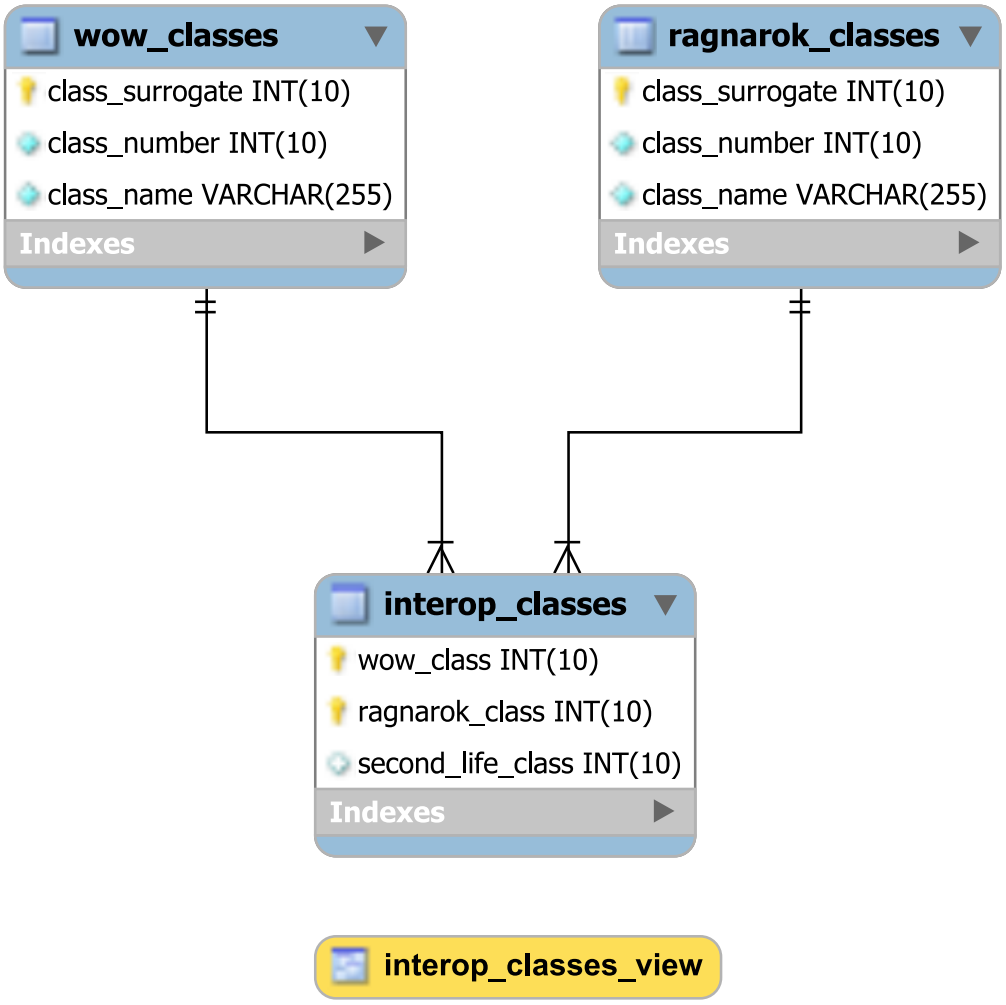


Figure 7: Business Logic for Avatar "Class" Transformation

E.1 Business Logic Database Schema Data Dictionary

Data Object Name	Description
class_surrogate	Surrogate auto-incrementing number that is the primary key of the wow_classes or ragnarok_classes table.
class_number	The number representing the class/profession in the *World of Warcraft* or *Ragnarok Online* virtual world. Does not necessarily have to match the class_surrogate primary key.
wow_classes	Table that will contain a listing of *World of Warcraft* classes/professions.
ragnarok_class	Table that will contain a listing of *Ragnarok Online* classes/professions.
interop_classes	Table that links together related or suggested classes between virtual worlds. Contains world-centric business logic.
wow_class	Foreign key to the primary class_surrogate of the wow_classes table. Refers to a *World of Warcraft* class.
ragnarok_class	Foreign key to the primary class_surrogate of the ragnarok_classes table. Refers to a *Ragnarok Online* class.
second_life_class	Numeric representation of a class/profession from *Second Life*. However, *Second Life* currently does not implement classes or professions, so this column is only kept for possible future uses and is NULL for all instances currently.
interop_classes_view	A view that represents a join between the interop_classes table, the wow_classes table, and the ragnarok_classes table to discover what classes from *World of Warcraft* are suggested or related to classes in *Ragnarok Online*.

F Business Logic SQL Script

```
1  CREATE TABLE ragnarok_classes (
2          class_surrogate INT UNSIGNED NOT NULL AUTO_INCREMENT,
3          class_number INT UNSIGNED NOT NULL,
4          class_name VARCHAR(255) NOT NULL,
5          PRIMARY KEY(class_surrogate)
6  ) ENGINE=InnoDB;
7
8  CREATE TABLE wow_classes (
9          class_surrogate INT UNSIGNED NOT NULL AUTO_INCREMENT,
10         class_number INT UNSIGNED NOT NULL,
11         class_name VARCHAR(255) NOT NULL,
12         PRIMARY KEY(class_surrogate)
13 ) ENGINE=InnoDB;
14
15 CREATE TABLE interop_classes (
16          wow_class INT UNSIGNED,
17          ragnarok_class INT UNSIGNED,
18          second_life_class INT UNSIGNED,
19          PRIMARY KEY(wow_class, ragnarok_class),
20          FOREIGN KEY (wow_class) REFERENCES wow_classes(class_surrogate),
21          FOREIGN KEY (ragnarok_class) REFERENCES ragnarok_classes(class_surrogate)
22 ) ENGINE=InnoDB;
23
24 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Novice',0);
25 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Swordman',1);
26 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Mage',2);
27 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Archer',3);
28 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Acolyte',4);
29 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Merchant',5);
30 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Thief',6);
31 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Knight',7);
```



```
32 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Priest',8);
33 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Wizard',9);
34 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Blacksmith',10);
35 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Hunter',11);
36 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Assassin',12);
37 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Knight',13);
38 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Crusader',14);
39 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Monk',15);
40 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Sage',16);
41 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Rogue',17);
42 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Alchem',18);
43 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Alchemist',18);
44 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Bard',19);
45 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Dancer',20);
46 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Crusader',21);
47 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Wedding',22);
48 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('SuperNovice',23);
49 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Gunslinger',24);
50 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Ninja',25);
51 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Xmas',26);
52 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Novice High',4001);
53 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Swordman High',4002);
54 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Mage High',4003);
55 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Archer High',4004);
56 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Acolyte High',4005);
57 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Merchant High',4006);
58 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Thief High',4007);
59 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Lord Knight',4008);
60 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('High Priest',4009);
61 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('High Wizard',4010);
62 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Whitesmith',4011);
63 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Sniper',4012);
```



```
64 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Assassin Cross',4013);
65 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Lord Knight',4014);
66 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Paladin',4015);
67 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Champion',4016);
68 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Professor',4017);
69 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Stalker',4018);
70 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Creator',4019);
71 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Clown',4020);
72 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Gypsy',4021);
73 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Paladin',4022);
74 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby',4023);
75 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Swordman',4024);
76 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Mage',4025);
77 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Archer',4026);
78 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Acolyte',4027);
79 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Merchant',4028);
80 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Thief',4029);
81 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Knight',4030);
82 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Priest',4031);
83 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Wizard',4032);
84 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Blacksmith',4033);
85 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Hunter',4034);
86 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Assassin',4035);
87 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Knight',4036);
88 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Crusader',4037);
89 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Monk',4038);
90 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Sage',4039);
91 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Rogue',4040);
92 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Alchem',4041);
93 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Alchemist',4041);
94 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Bard',4042);
95 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Dancer',4043);
```



```
96  INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Baby Crusader',4044);
97  INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Super Baby',4045);
98  INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Taekwon',4046);
99  INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Star Gladiator',4047);
100 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Star Gladiator',4048);
101 INSERT INTO ragnarok_classes(class_name, class_number) VALUES ('Soul Linker',4049);
102
103 INSERT INTO wow_classes(class_name, class_number) VALUES ('Warrior', 1);
104 INSERT INTO wow_classes(class_name, class_number) VALUES ('Paladin', 2);
105 INSERT INTO wow_classes(class_name, class_number) VALUES ('Hunter', 3);
106 INSERT INTO wow_classes(class_name, class_number) VALUES ('Rogue', 4);
107 INSERT INTO wow_classes(class_name, class_number) VALUES ('Priest', 5);
108 INSERT INTO wow_classes(class_name, class_number) VALUES ('Death Knight', 6);
109 INSERT INTO wow_classes(class_name, class_number) VALUES ('Shaman', 7);
110 INSERT INTO wow_classes(class_name, class_number) VALUES ('Mage', 8);
111 INSERT INTO wow_classes(class_name, class_number) VALUES ('Warlock', 9);
112 INSERT INTO wow_classes(class_name, class_number) VALUES ('Druid', 11);
113
114 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 2);
115 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 8);
116 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 14);
117 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 15);
118 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 23);
119 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 30);
120 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 36);
121 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 42);
122 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 43);
123 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 50);
124 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 52);
125 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 58);
126 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 64);
127 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 65);
```



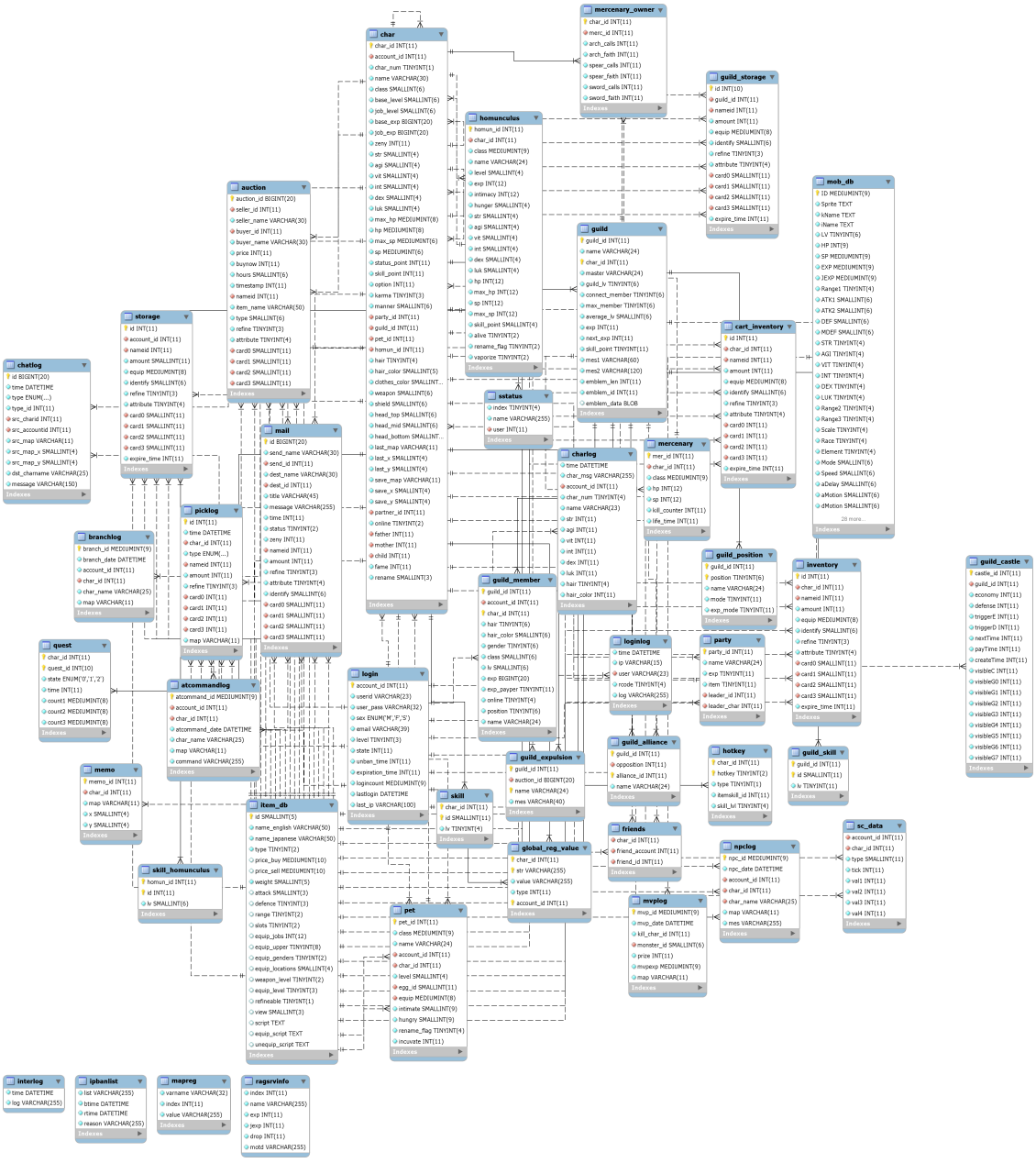
```
128 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (1, 71);
129 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 2);
130 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 15);
131 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 23);
132 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 30);
133 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 43);
134 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 50);
135 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 52);
136 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 65);
137 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (2, 71);
138 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 4);
139 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 12);
140 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 21);
141 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 22);
142 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 26);
143 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 32);
144 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 40);
145 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 48);
146 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 49);
147 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 54);
148 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (3, 62);
149 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 7);
150 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 13);
151 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 18);
152 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 35);
153 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 41);
154 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 46);
155 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 57);
156 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 63);
157 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (4, 68);
158 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 5);
159 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 9);
```



```
160 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 16);
161 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 33);
162 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 37);
163 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 44);
164 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 55);
165 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 59);
166 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (5, 66);
167 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 2);
168 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 8);
169 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 14);
170 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 30);
171 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 36);
172 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 42);
173 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 52);
174 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 58);
175 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (6, 64);
176 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 5);
177 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 9);
178 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 16);
179 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 33);
180 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 37);
181 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 44);
182 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 55);
183 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 59);
184 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (7, 66);
185 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 3);
186 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 10);
187 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 17);
188 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 31);
189 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 38);
190 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 45);
191 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 53);
```

```
192 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 60);
193 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (8, 67);
194 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 5);
195 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 9);
196 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 16);
197 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 33);
198 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 37);
199 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 44);
200 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 55);
201 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 59);
202 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (9, 66);
203 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 6);
204 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 11);
205 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 19);
206 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 20);
207 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 34);
208 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 39);
209 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 47);
210 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 56);
211 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 61);
212 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 69);
213 INSERT INTO interop_classes(wow_class, ragnarok_class) VALUES (10, 70);
214
215 CREATE VIEW interop_classes_view AS
216 SELECT wow_classes.class_name AS "warcraft", ragnarok_classes.class_name AS "ragnarok",
      interop_classes.second_life_class AS "second life" FROM ((interop_classes INNER JOIN
      wow_classes ON interop_classes.wow_class = wow_classes.class_surrogate) INNER JOIN
      ragnarok_classes ON interop_classes.ragnarok_class = ragnarok_classes.class_surrogate);
```

G Ragnarok Online Database Schema



The source to this database layout can be found at

`http://svn.eathena.ws/svn/ea/trunk/sql-files/main.sql`