

SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2002-2.3.1.8
Networked Audiovisual Systems

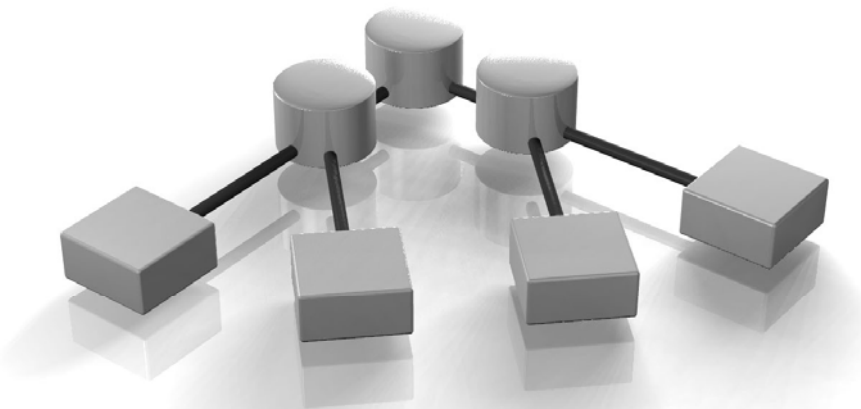


Uni-Verse Project

D5.3 Administration Tools

March 1, 2006

Distribution: Public



STREP project

Project acronym: Uni-Verse

Project full title: A Distributed Interactive Audio-Visual Virtual Reality System

Proposal/Contract no.: 002228

Table of Contents

Table of Contents	2
Introduction	3
Connector	3
UV Editor	4
Geometry Layer Painter	5
Loq Airou	7
Chat	7
“Enough” Storage Library	8
"Persuade" Render Library	9
Saver / Loader	9

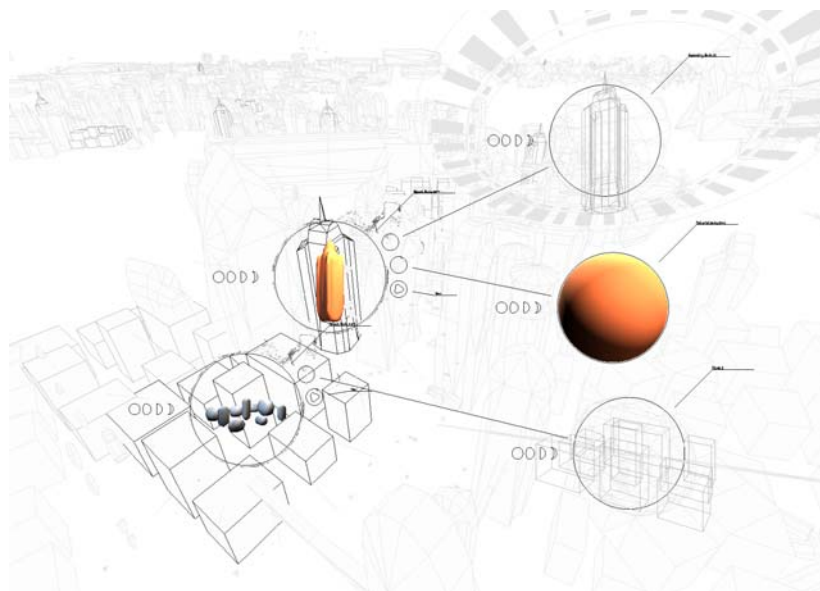
Introduction

The administrative tools are all native tools that let the user interact with Verse. Compared to other tools, the native tools expose Verse functionality fully and let the user control every aspect of Verse data. A non-native tool like 3D Studio Max or Blender may try to hide how Verse works in order to retain the application's user experience. In this case the user has less interest in exactly what is going on under the hood, but is rather interested in simply having their application work. So the administrative tools are more geared towards the user who has a technical interest in Verse, or who wants to create data sets in native Verse format. During this project these tools have proved invaluable for developers both as templates for good Verse programming and as tools that lets developers set up and change data for testing and debugging their own clients.

Although the administrative tools are more technical and may not be a novice's choice of tool, they are very capable and have fast and refined graphical user interfaces. Especially Loq Airou and Connector contain a range of useful tools and graphical polish. These tools are therefore excellent demos as they both demonstrate Verse and are impressive to see in action. Many of the tools also use the same range of code libraries such as "Enough" and "Persuade" making much of their code shareable with future projects.

Connector

Connector is a schematic tool that once connected to a Verse host allows the user to browse and modify any data on the server. All nodes are displayed in a graphics-intensive, schematic way. By clicking a node the user can display all the properties of the node and modify them.



Basically, Connector makes the core Verse command structure available for direct graphical manipulation. This is a very useful tool to e.g. administrate a Verse host, as the user can easily get access to any data element. It is also very important when debugging applications, since the user can set up any dataset and then see how his/her application responds to it. Connector can also be used to generate network commands, to see how applications respond to different commands. When an application that modifies a Verse dataset fails, Connector can be used to inspect the data to see what went wrong.

D5.3 Administration Tools

In many cases Connector is a too “technical” tool that doesn't cater well to an artist's needs. But it is the perfect tool for the power-user or developer who wants to have perfect control over a Verse session. These are some of the things Connector will do:

- ▶ Create and delete nodes.
- ▶ Rename nodes.
- ▶ Create and set tags and the groups that gather them.
- ▶ Edit material properties, and view material previews
- ▶ Create and send method calls.
- ▶ Create and manage layers of all types.

The following table from the specification defines the support for the various node types of Verse in the Connector application.

<i>Node Type</i>	<i>Basic Support¹</i>	<i>Editing Support</i>
Object	Yes	Full
Geometry	Yes	Partial ²
Material	Yes	Full
Bitmap	Yes	None
Text	Yes	None
Curve	Yes	Full
Audio	Yes	None

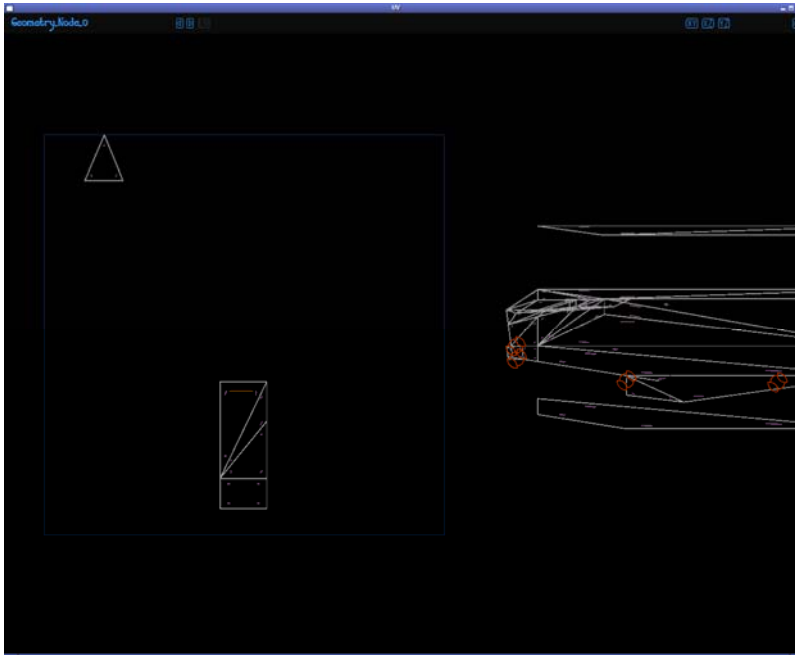
All planned features have been implemented.

UV Editor

The UV Editor is a simple tool that allows the user to edit the mapping of a texture over an object's surface. The editor allows the user to select polygons and vertices, and move them over a texture mapped over the surface. This is a required step in any 3D modelling application, and having it available is good for general demonstrability of the Verse platform. The application has some advanced features that let you project the mapping using any axis or polygon normal, or unfold complex 3D objects into flat projections. It has a full range of snap and divide tools to give the user full control over the precision of the mapping. Additionally the tool is able to create the necessary UV layers and communicate with other modelling tools using shared select layers that can be used to only work on a specific subset of polygons.

¹ Nodes of this type can be created, listed, renamed, and destroyed.

² Layers can be created and managed, but their contents (actual geometry) cannot. This is by design, Connector is more of a data browser than a geometry editor.



This table from the specification shows some of the primary features of the UV editing tool in a summary fashion.

<i>Feature</i>
Display mesh
Allow vertex and polygon selection
Display points in UV space, over bitmap
Unfold mapping
Support shared selection ³

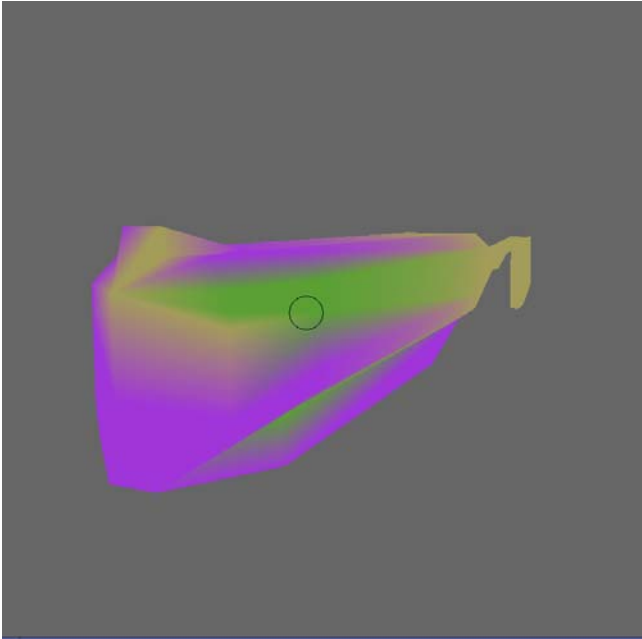
All planned features have been implemented.

Geometry Layer Painter

The geometry layer paint tool allows a user to paint geometry layer values using brushes directly onto 3D surfaces. The tool can display up to three layers as red, green and blue, and will have a variety of brushes and blend modes such as replace, add, subtract, and colorize. The tool can also be used to deform the geometry by moving the vertexes in the direction of the normal. The tool is used for general manipulation of surface properties such as colour, texturing, and acoustic properties. The tool supports all geometry layer types and is able to draw and display high dynamic range.

³ Shared selection simply uses a Verse geometry node layer to store the selection of vertexes, thus making the selection externally visible and distributed to other clients working on the same data.

D5.3 Administration Tools



The following table from the specification summarizes some of the primary features of the Geometry Painter tool.

<i>Feature</i>
Direct-manipulation painting
Paint multiple layers simultaneously
Different “modes” for painting
High dynamic range support

All planned features have been implemented.

Loq Airou

Loq Airou is a 3D modelling/sketch tool. It allows the user to create and model subdivision surface objects. Loq was initially implemented for the first iteration of Verse. The application has been ported to Verse 2 and has been updated to match the updates to the protocol.

The tool has an innovative interface where the user draws the cage in 3D and the application creates the surfaces accordingly. The application only requires one hand to be fully used making it ideal for sketching and prototyping. The application has no visible buttons or menus but is operated by manipulating the geometry. By drawing lines the user creates geometry, by crossing out geometry it is deleted and by circling geometry it is selected. Additionally using context sensitive pop-up menus the user can get access to a range of additional features such as slice flatten, smooth selection and many more. Adding to this the application is beautifully realized with lots of effects and graphics that help by giving you hints and feed back.

The table from the specification below lists some of the primary features of the Loq Airou modelling tool.

<i>Feature</i>
Interactive 3D real-time modelling
Interface optimized for sketching
Direct manipulation interface ⁴
Additional features (slice, flatten, smooth etc)

All planned features have been implemented.

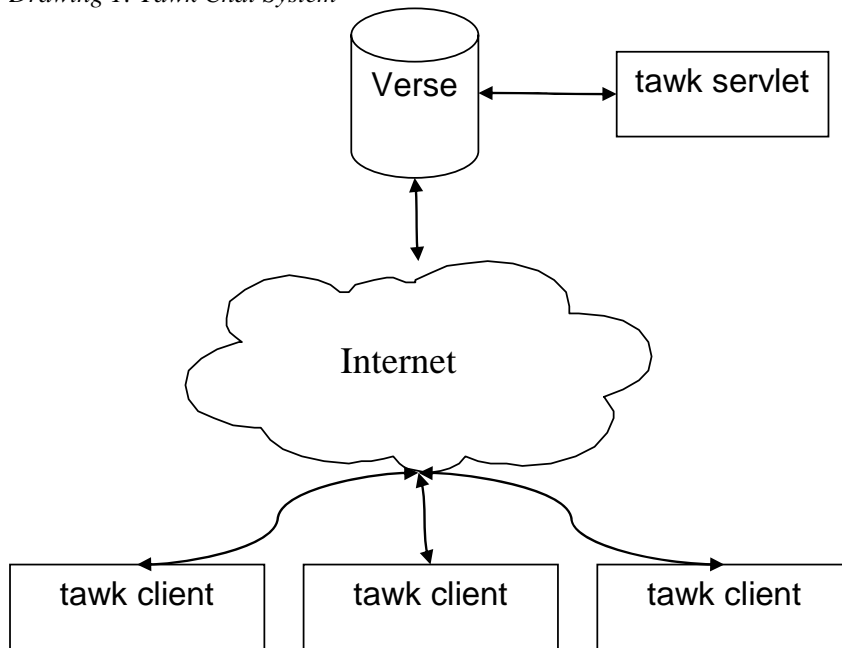
Chat

The Chat utility allows the user to chat over the Verse protocol with users connected to a host.

The chat system, called “Tawk”, is implemented mainly as a proof of concept. It consists, as planned, by two parts: one “servlet” and a client. The “servlet” acts as a provider of Chat services, and all talk goes through it. It is, technically, just another client from the Verse server's perspective. The connections are outlined in Drawing 1, below.

⁴ Operations in Loq Airou are often mapped directly to actions on the geometry, rather than going through buttons and menus. To delete an edge, cross it out in the 3D view, and so on.

Drawing 1: Tawk Chat System



Logically, the servlet maintains a number of chat “rooms”, or channels. All talk is always done in a channel, and whoever has registered an interest in the channel will receive all that is said. This works very much like IRC, and it should be possible to implement a gateway between IRC and the Verse chat system without too much work because of this similarity.

Under the hood, Tawk relies on Verse object methods⁵. The servlet publishes a handful of methods for logging in, creating and destroying channels, and saying something. The client publishes even fewer methods for joining and leaving, and for “hearing” what is being said.

The Tawk chat system was implemented using the PyVerse Verse bindings for the Python programming language, and weighs in at under 500 lines of code. It is thus well suited for acting as a template or illustration of how to implement chat using Verse.

“Enough” Storage Library

“Enough” is a storage library, useful when writing C programs, that shields the application programmer from having to deal with the details of the Verse data model. It takes care of all the callbacks and makes the data available as if it were local to begin with. An application based on Enough does not have to handle callbacks, but can simply ask the Enough API for the current state.

The Enough library is the foundation of many applications such as Connector, Quel Solaar, Loq Airou and the UV editor. Enough is not specifically designed for one type of application and can therefore be a great base for many future Verse applications.

⁵ Methods are introduced by <<http://verse.blender.org/cms/fileadmin/verse/spec/n-object.html#o-methods>> in the Verse specification document.

"Persuade" Render Library

"Persuade" is the core library of the "Quel Solaar" rendering engine developed by Eskil Steenberg. It uses the Enough library and OpenGL. Any application with a GL context and Enough is able to use this library, and it will therefore be integrated into Connector and Loq Airou. It is based on OpenGL 2.0 but also has a 1.x fallback, however this fallback is fairly trivial.

The Persuade lib features all the graphical features of Quel Solaar such as global illumination, displacement mapping, programmable shaders and so on. The Persuade API enables the user to draw entire scenes or individual objects. Persuade also includes code for displaying bitmap nodes as GL textures.

This rendering engine is intended to be very advanced. The heart of the engine consists of an operating system like task manager that divides up all computation to ensure a smooth frame rate even if large sets of data change. The data coming from Verse is processed in real-time and goes through many stages of optimizations in order to be displayed. The geometry pipeline goes through a series of stages, and depending on how the geometry is manipulated it only recomputes the smallest possible number of stages in order to display the changes. The geometry pipeline first subdivides the geometry using Catmull/Clark subdivision surfaces scheme. Then the engine creates Level Of Details (LODs) using a reduction algorithm that removes unwanted geometry. The SDS engine does not actually compute the new shape, but rather how the control mesh relates to the surface. This means that when the control mesh is modified or animated, one can quickly recompute the shape.

The geometry engine also supports displacement mapping, as well as geometry computed for use as stencil shadow geometry. The surfaces are rendered using OpenGL 2.0's shading language and compute the surface shading per pixel. The engine takes the Verse shading tree and converts it into GL's shading language, compile, link and run it in real time. This allows users to see in real-time the changes they make in material editors. The shading language will allow us to replicate the Verse shading tree very closely in real-time. The engine will feature a simple but effective global illumination algorithm. Each object will maintain a list of surrounding objects, and light and draw them into environment cube maps. The shading engine will then use these cube maps to get the ambient term on the lighting calculations. We believe that this engine will be rated unique because it does all computation on the fly, whereas most engines require off-line pre computation of all data. Even if there is no pre computation of data we believe that the feature set of this engine will rival the most advanced engines on the market.

Saver / Loader

A pair of tools to save and load Verse server contents out to disk files has been developed. These tools are very useful during development, since they allow for the quick restoration of a server's contents, if the server has had to be stopped or restarted for some reason. Also exchanging Verse content over e.g. e-mail becomes easy and natural to do. The tools use a XML-based format called VML, which is essentially a 1:1 mapping of the data model to XML. As such, it is not very efficient for some kinds of data (notably bitmaps), but it is very simple to understand and process. The saver uses the Enough data storage library, and represents a very small effort. The loader was implemented using XML parsing code already written for Purple, so it was fairly quick to get going, as well.