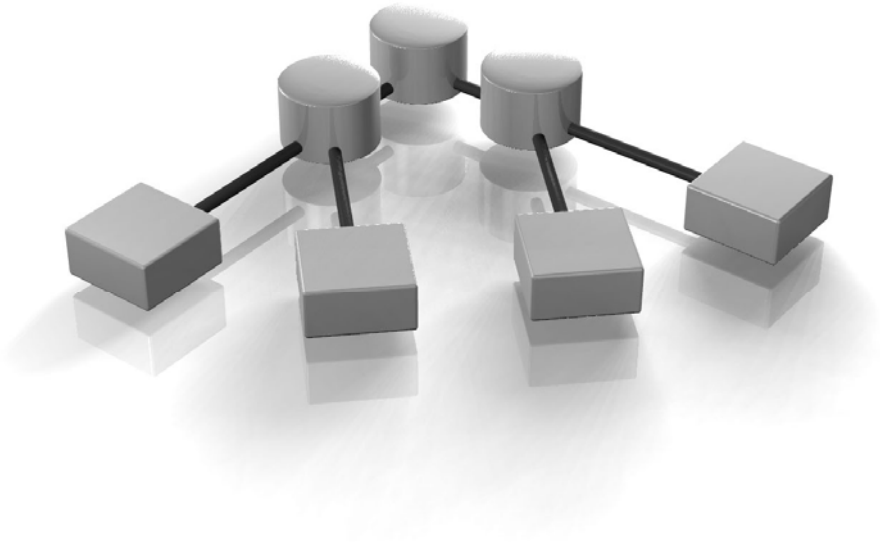
SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2002-2.3.1.8
Networked Audiovisual Systems

Uni-Verse Project

D5.4 Dynamic Mesh Generation

June 2007

Distribution: Public



STREP project
Project acronym: Uni-Verse
Project full title: A Distributed Interactive Audio-Visual Virtual Reality System
Proposal/Contract no.: 002228

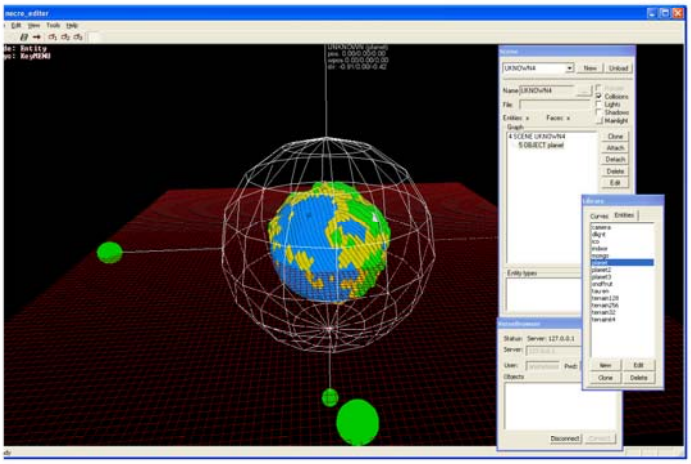
This document describes the Uni-Verse project aim to research and develop a distributed system with generative models for real-time visualization. This system utilizes and integrates with the Verse protocol and the goal is it to provide basis for fully dynamic 3D simulation software such as 3D Game-engines.

Generative models here refer to 3D models that are described not only through their geometry (mesh) and texture, but models that are described fully or partly by algorithms. The focus is to utilize different fractal algorithms such as Fault Formation and L-Systems[1] to create 3D models of structures from nature (landscapes, tree's etc).

The use of for example fractals to create landscapes is more or less standard in current game content creation. There are several commercial and non-commercial production tools[2] that use these types of techniques, but they normally convert the models to meshes for the export. Our approach is to enable fully dynamic objects, through keeping models in their original syntax. Thereby they could easily be manipulated and changed later in a production or even during real-time game performance.

The original goals of WP5.4 were

- to research and create working and high-performance modules utilizing the Verse protocol for creating and distributing fractal-based 3D models and changes in them.
- to research and develop real-time handling of the transition between algorithm and mesh to enable rendering of models in real-time.



Martin Schlingmann
The Interactive Institute - Sweden

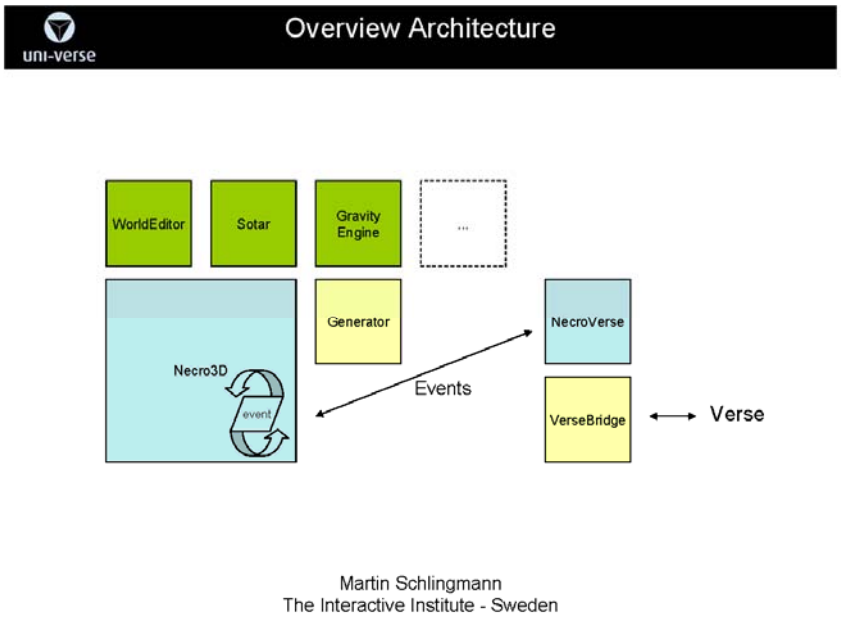
PROJECT

The basis of the project is an open source C++ game engine library, Necro3D, which was developed by Martin Schlingmann et al at the Interactive Institute Game Studio, in Visby, Sweden. (webb..)

Integration with the verse protocol required a review and redesign of the Necro3D architecture. Achieving this means that all models in a game world based on verse-enabled Necro3D can be dynamic at any point (lock-unlock).

Second a separate library, NVerse library, was built for Necro3D-Verse integration. This handles export/import operations and dynamic relation between objects in the game world, through the verse protocol.

Third, a simple game demonstrator has been developed, consisting of tools for generating and editing of dynamic objects. This consists of a planetary environment, a dynamic solar system with a flexible number of planets, each with surface structure, circling a central star.



This lead to the final structure of the deliverables of this workpackage

1. enable standardized dynamic assets (3dmodels, sound etc) in Online games through a redesigned and verse enabled Necro3D open source game engine library
2. enable generate, evolve/change and use of algorithmically described models for game objects, BioModels, such as terrain and trees (fractals, L-systems etc) through NecroGenerator, Ngen library
3. building a NecroVerse Library and a VerseBridge, between Necro3D and verse server, thereby integrating a game engine with a verse system
4. provide a demonstrator of real-time online 3D game functionality's, by integrating verse into a Game-engine library

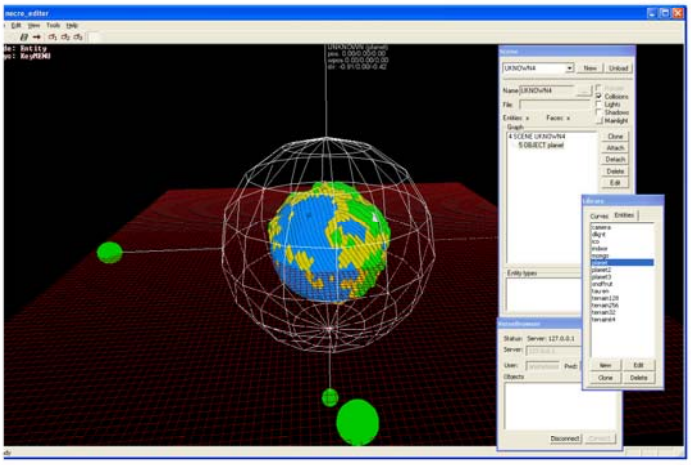
NECRO3D ARCHITECTURE REVIEW/REDESIGN

NecroVerse (NVerse namespace) is an implementation of the verse multimedia network protocol. It is built on top of the verse standard c-library. We have divided the implementation into two parts, a generic abstraction library for verse (VerseBridge) and an implementation for the Necro3D engine (NecroVerse).

Geometry, textures and transformation is implemented which makes it possible to distribute standard Necro3D models and their transformations (to and from) a verse server/network.

GENERATOR

The NGEN NecroGenerator library provides the fundamental support for algorithmically generated objects, based on fractals and L-systems. This is the BioModel generator of terrain, trees, flowers etc.

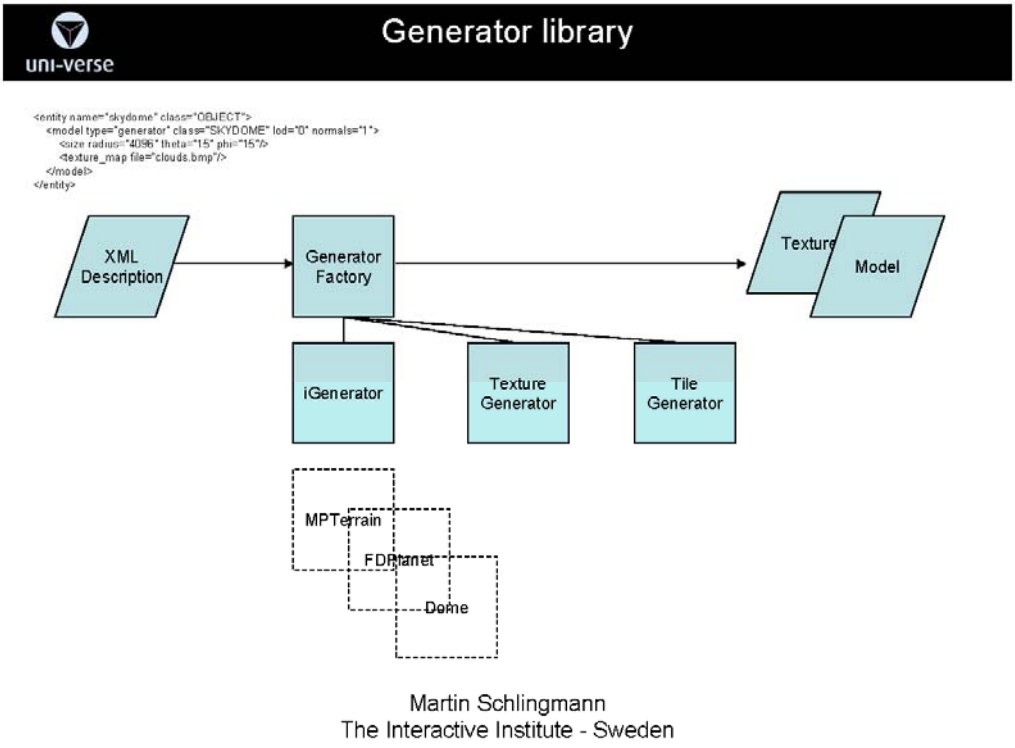


Martin Schlingmann
The Interactive Institute - Sweden

NGEN Generator library handles

- Architecture (Generator factory and base classes)
- Basic Terrain generators (Mesh and Textures)
- (FaultDeformation, MidPointDisplacement)
- Basic Skydome generator (half-speheres, cloud generator(perlin))
- LSystem (for trees and flowers)

The workflow and structure of the library is outlined below:



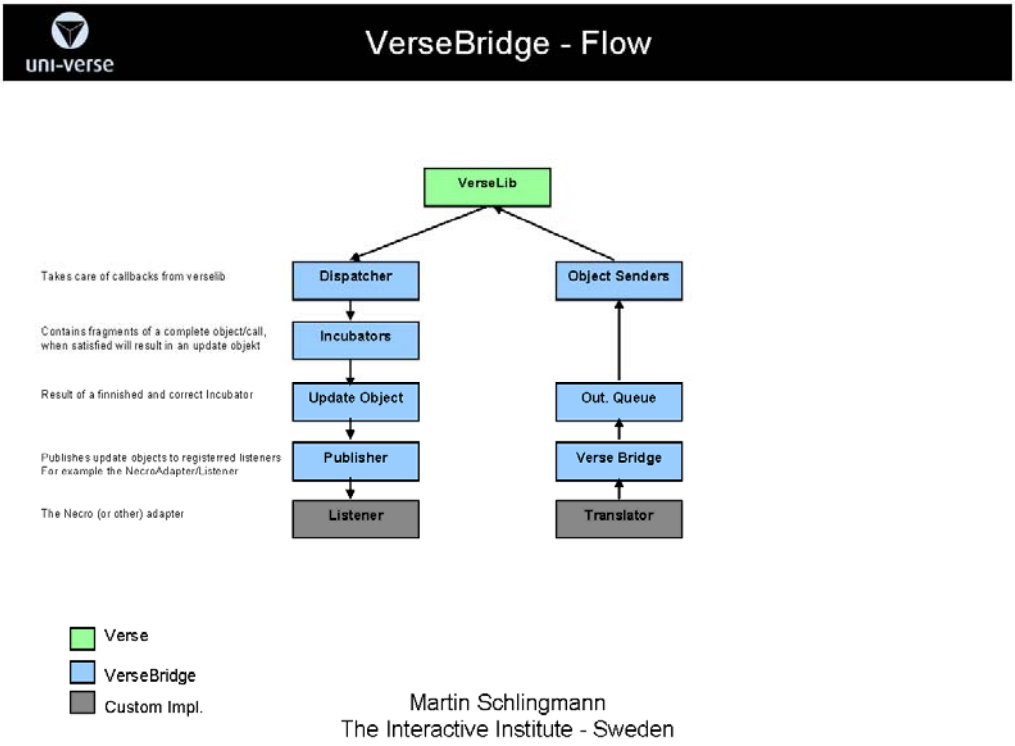
Neditor, the NecroVerse editor is a Win32 Scene editor with verse integration (objects browser etc)

- Creating, placing, configurating objects
- Loading & Saving Scenes
- Import & export of objects

VERSE BRIDGE

Objectives of this library is

- To provide a higher level interface for clients, to hide underlying low-level details in protocol and library implementation
- To guarantee that data delivered to objects is complete and renderable
- To provide an object model closer to representation and object exchange in common 3d-engines and modelers
- To minimize the amount of code necessary to connect external systems with a verse system



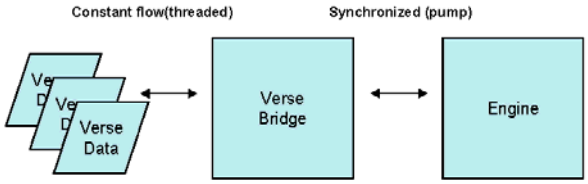
Importance and success of NecroBridge is demonstrated by the fact that less than 1K lines of code is needed for none-trivial integration with verse.

The VerseBridge in addition to the standard verse library provides:
1. C++ interface
2. Thread Safety implemented
3. Object oriented approach (Objects not vertices etc)

3D models support lock-unlock. Also multiple rendering options are provided (static, dynamic, progressive). Progressive Mesh rendering (subdivision) is however not implemented.

ENGINE INTEGRATION

A 3D Engine consists of several sub-systems. Normally these communicate via events, and the verse-bridge implementation listens to events, like other sub-systems. The tricky part is the synchronization versus background networking, which is here implemented by pumping.

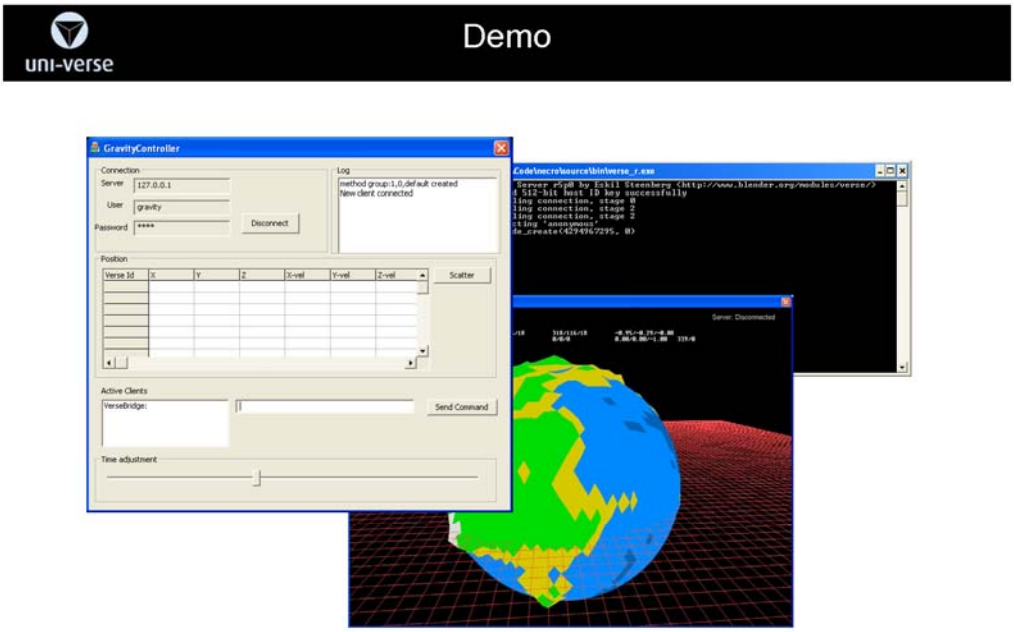


Martin Schlingmann
The Interactive Institute - Sweden

DEMONSTRATOR

A simple game demonstrator has been developed, showing a planetary world, with fully dynamic 3D objects, based on fractal generated geometry and 3D motion. The world consists of a planetary environment, a dynamic solar system with a flexible number of planets, each with surface structure, circling a central star.

Through an editing window changes can be made in a set of parameters, including the number of planets.



Martin Schlingmann
The Interactive Institute - Sweden

State Of The Art Analysis for WP5.4 Uni-Verse project
Generative 3D distributed system (G3D)

Author: Martin Schlingmann, The Interactive Institute, 2005

This document requires the reader to have basic understanding of 3D visualization and networked system.

Background
This document describes the core parts of the aim to research and develop a distributed system with generative models for real-time visualization. It compares these with the current public research and software found.
This system utilizes and integrates with the Verse protocol and the goal is it to be used by mainly 3D simulation software's such as 3D Game-engines.
In this context, when mentioning Generative models, I refer mainly to 3D models that are described not only by direct geometry (mesh), but models that are described fully or partly by readable mathematical syntax (formula, code etc).
In this part of the Uni-Verse project the focus is mainly to utilize different Fractal algorithms such as Fault Formation and L-Systems[1] to create 3D models of structures from nature (landscapes, tree's etc).

Goals
Goals of WP5.4 are
- to research and create working and high-performance modules utilizing the Verse protocol for creating and distributing fractal-based 3D models and changes in them.
- to research and develop real-time handling of the transition between algorithm and mesh to enable rendering models in real-time.

Generative Systems
The use of for example fractals to create landscapes is not a new technique; rather this is more or less standard in current game content creation. There are several commercial and non-commercial tools[2] that uses this type of techniques, but these normally convert the models to meshes for the export. By keeping them in their original syntax they could easily be manipulated and changed later in a production or in real-time.

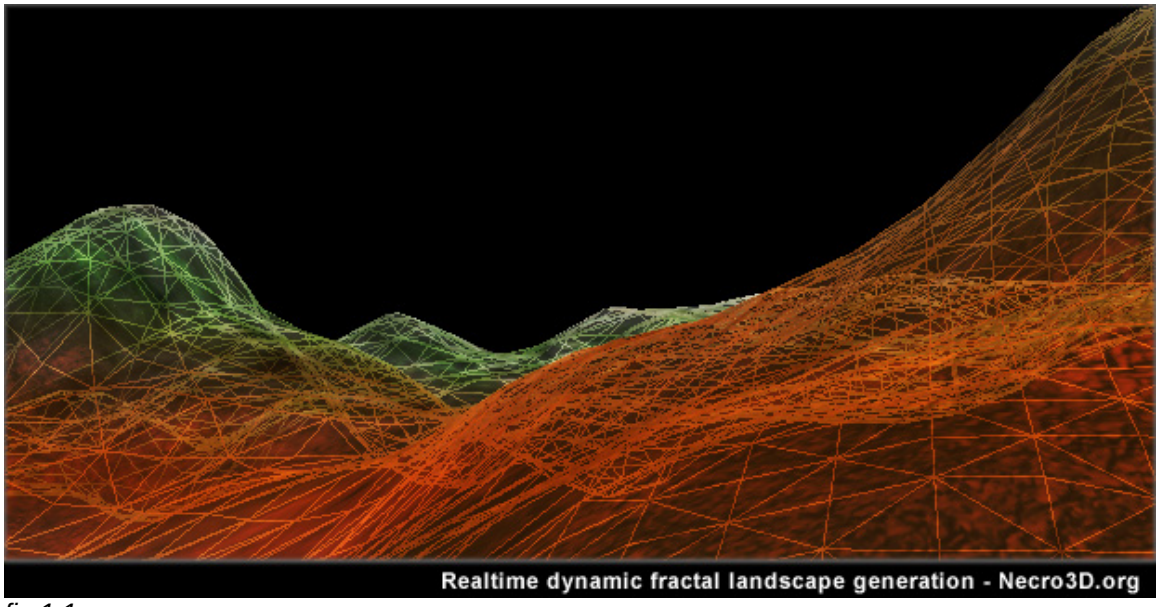


fig 1.1

The algorithms are normally only used for the content creation and not in run-time (converted to some kind of mesh), the actual source of the model is lost and it cannot be changed by using for example algorithms (genetic algorithms etc) in run-time, which would have been the case if it was an algorithmic-based model.

For real-time 3D visualization systems such as Game-engines, it is harder to find software that utilizes fractal techniques. Some use comparable techniques with templates and random seeding (for creating variations in real-time) and some use fractal algorithms in real-time[3].

To compare the G3D system with current tools and engines on the market is not really possible, they both use fractals and algorithms to create models, but in contrast to G3D they convert models into standard mesh formats. So the models cannot be distributed and changed in real-time.
One of the future goals is to integrate the G3D system to existing Game-engines and content creation tools.

There is no ground-breaking research performed on the actual algorithms in this project. All fractal algorithms are already being used by different software's and 3D engines.
The uniqueness and efforts rather lies in the combination of techniques as described in next section (distributed generative system).

Distributed Generative Systems

The uniqueness of G3D lies in combining algorithmic based model description with distribution of these in real-time.
This means that the clients of such a system are responsible to generate the meshes (and even textures in some cases) from the algorithms and descriptions. Only the actual algorithm and/or description are distributed over the network.

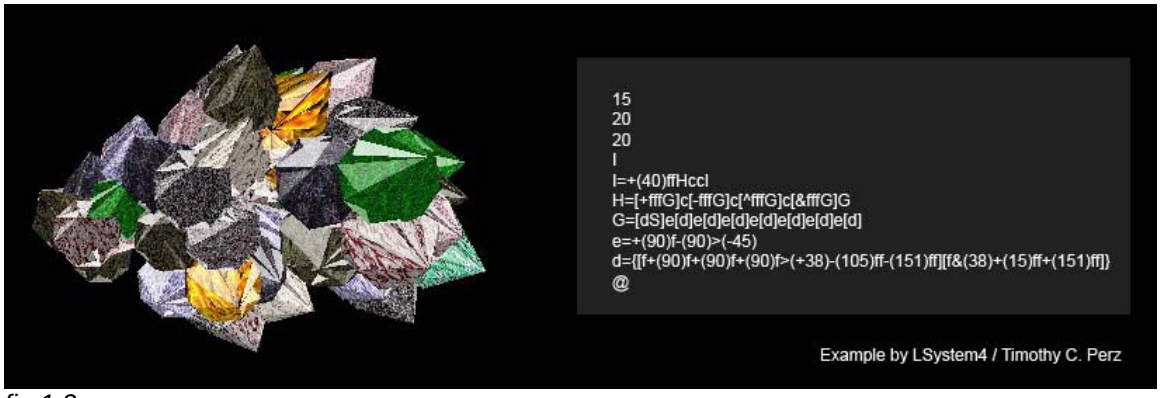


fig 1.2

The main benefits of this approach are:

- Enabling to distribute new models in runtime. A description for a model can be less than 1k, while distributing the actual mesh could weight several megabytes.
- Enabling to change and update parameters of the descriptions and thus allowing for usage of for example genetic algorithms to change the models.
- Enabling to easily and rapidly create high resolution and complex models to be used by 3D visualization engines using the Verse protocol.

The author has not found any software aiming at these goals. Prior there has been some work performed in this field[4], but none fully utilizing a distributed network protocol.
Rather there have been research and software to utilize L-Systems for a-life systems.

Conclusion

The G3D system aims at enabling the usage of real-time distributed algorithmic models in 3D visualization engines and tools.
It is based on the Verse standard for distributed audio and visual data.
The competitors in this field are mainly existing commercial Game-engines with proprietary and limited implementations.

References

1. Lindenmayer A (1968). Mathematical models for cellular interaction in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology* 18:280-289

   L-systems are a mathematical formalism proposed by the biologist Aristid Lindenmayer in 1968 as a foundation for an axiomatic theory of biological development. More recently, L-systems have found several applications in computer graphics (Smith 1984; Prusinkiewicz and Hanan 1989; Prusinkiewicz and Lindenmayer 1991). Two principal areas include generation of fractals and realistic modeling of plants

2. Bryce, http://bryce.daz3d.com/55index.php

   XFrog by GreenWorks Organic Software, http://www.xfrog.com

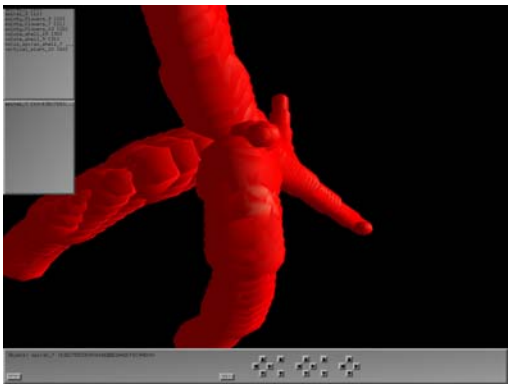
   WorlBuilder by Digital Element, http://www.digi-element.com

   LSystem4 by Timothy C. Perz, http://www.thinkpiece.com

3. Ryzoom by Nevrax, http://www.ryzom.com

   Elite by David Braben and Ian Bell 1984 , using pseudo random generators and simple fractals, http://www.frontier.co.uk/games/elite

4. IIWorlds application by Martin Schlingmann, utilizing L-System parser and genetic mutations by Prof.Craig Lindley, Trans-Reality Game Laboratory, Institution for Technology, Art and New Media Gotland, http://trgl.tii.se

   A software using fractals and especially L-Systems for generating and evolving models in real-time.