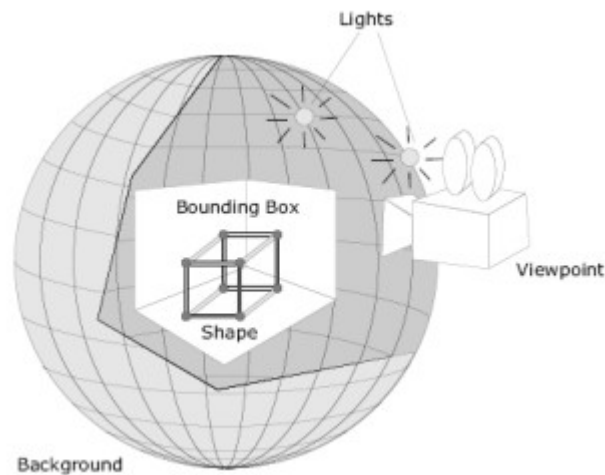


Daniel Adler  
Project Portfolio  
2002-2006

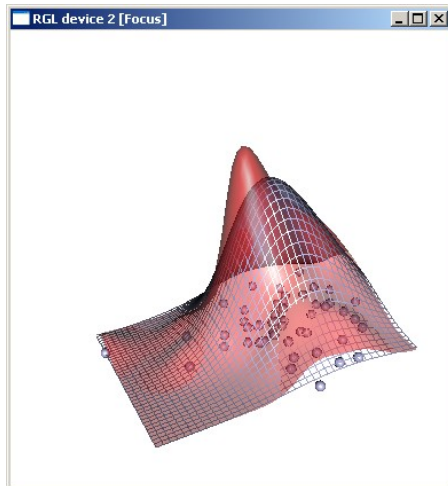
## Scientific Application "RGL": Award-winning 3D Visualization System for R

Based on Diplom Thesis: "Visualization of Multi-Dimensional Data in R using OpenGL". RGL extends R with a real-time visualization device system. It is delivered as a plugin package, portable across major R platforms such as Windows, Mac OS X Carbon and Unix/X11 using C++ and OpenGL. It features a simple API of 20 core functions for developing 3D visualizations interactively in the R Programming Environment. The user navigates in real-time using the mouse.

*A typical Scene:*



*Example:*



*In R Console:*

```
n<-50; ngrid<-40; zscale<-20
x<-rnorm(n); y<-rnorm(n)
# estimate non-parametric density surface (kernel smoothing)
denobj<-kde2d(x, y, n=ngrid)
den.z <-denobj$z
# generate parametric density surface of a bivar normal dist
xg <- denobj$x
yg <- denobj$y
bi.z <- dnorm(xg)%*%t(dnorm(yg))

# RGL comes into play now

# Draws the simulated data as spheres on the baseline
spheres3d(x,y,rep(0,n),radius=0.1,color="#CCCCCCF")
# Draws non-parametric density
surface3d(xg,yg,den.z*zscale,color="#FF2222",alpha=0.5)
# Draws parametric density
surface3d(xg,yg,bi.z*zscale,color="#CCCCCCF",front="lines")
```

*Status:* The Software was released under the GNU Public License and is actively maintained.

*Project Homepage:* <http://rgl.neoscientists.org>

*R Homepage:* <http://www.r-project.org>

*Award:* John Chambers Statistical Software Award 2003 given by the ASA

## Java Swing Application Development at B&N Software AG

### Frameworks:

- Multi-document framework
- Improved localization and translation to english
- Asynchronous notification of filesystem changes in pure java

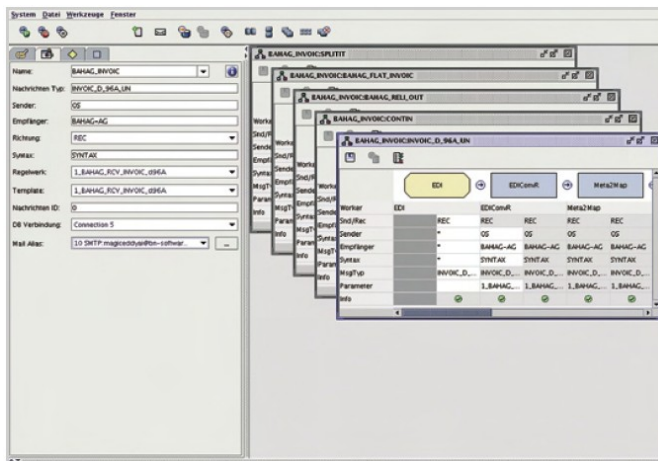
### Applications:

- Workflow Studio for defining workflows using drag'n'drop
- Async/soft real-time database transaction monitor
- Integrated Development Environment for inhouse programming language with function completion and on-screen manual popup
- C++ mini-IDE for easy development/compilation/deployment of C++ workers accessible from the java user-interface

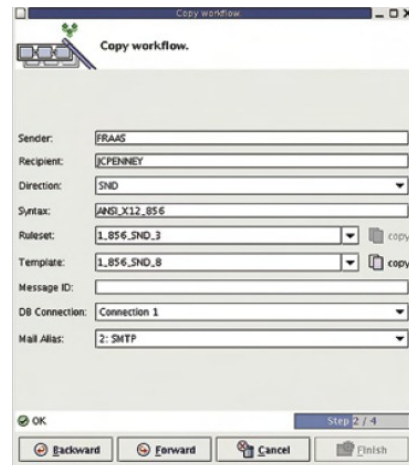
### Misc:

- CVS Repository administration and backup configuration
- Build management using Ant make tool

### ScreenShots:



Workflow Studio



Copy Wizard

Source: [M@gicEddy AI Product Sheet PDF](#), Copyright by B&N Software

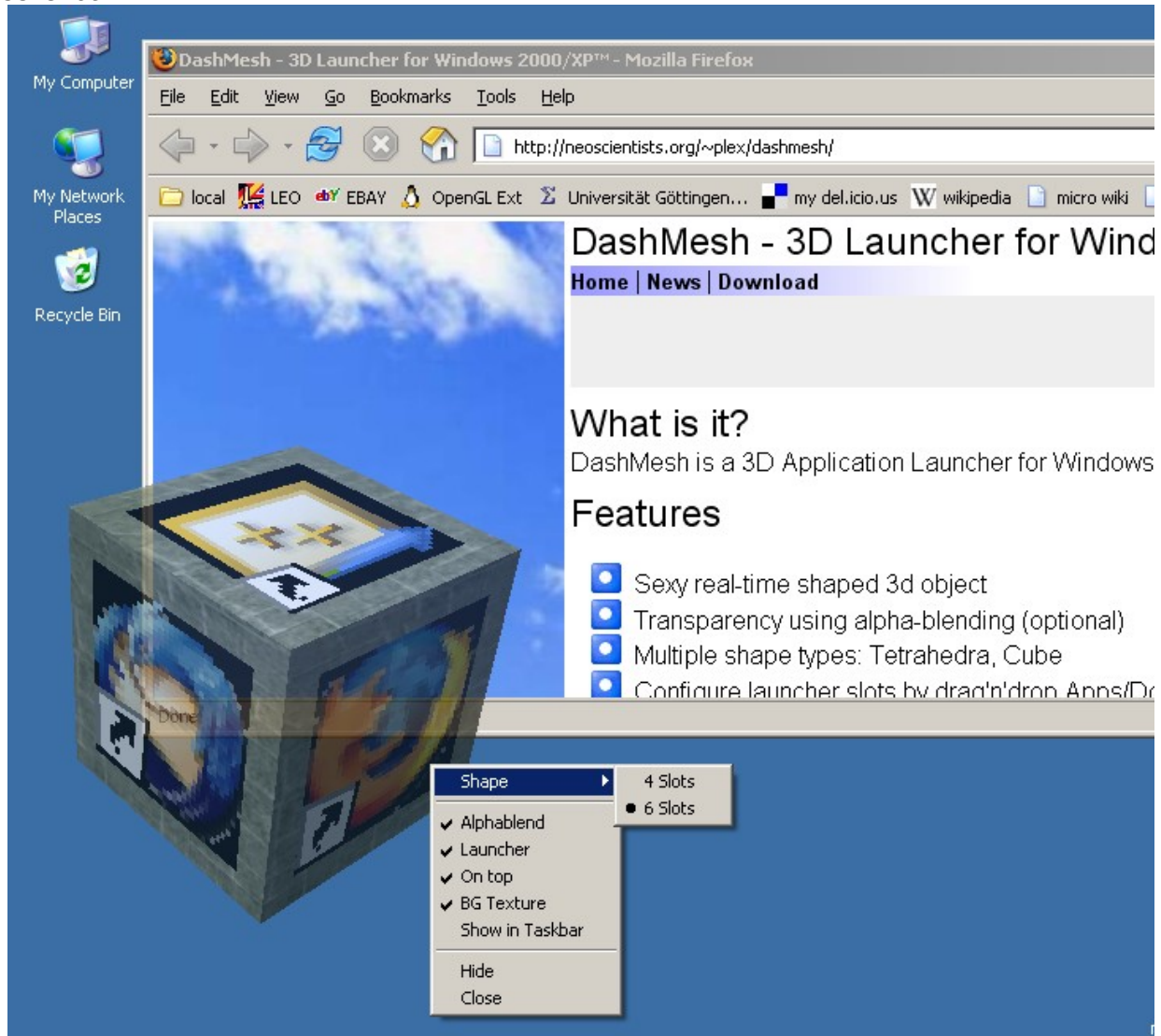
Note: Product Sheet available at [misc/bnsoftware.pdf](#)

Reference: Christian Göhler, [cgoehler@bn-software.de](mailto:cgoehler@bn-software.de)

## Windows Utility Application "DashMesh"

3D Desktop toy using "render to layer" technique. Application launchers can be configured by context menus and drag'n'drop operations. 3D Objects are positioned and spin rotated using the mouse. Multiple launchers can be created using the system tray menu.

Screenshot:



Project Homepage: <http://neoscientists.org/~plex/dashmesh>

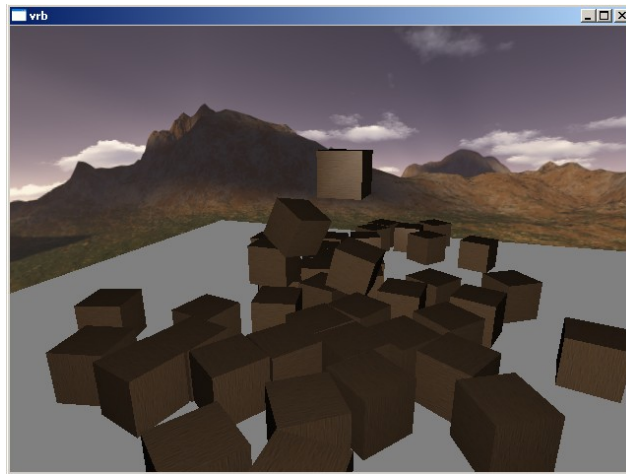
## VRB X3D toolkit (technology preview)

Flexible run-time engine modelled after the X3D architecture. Designed for real-time interactive applications, games, scientific visualizations and easy extensibility in C++ and scripting languages.

### Features:

- Modular components implementation design
- VRML97/X3D/DOM nodes building blocks authoring design
- Advanced C++ implementation kit
- Virtual file-system
- Physics engine (ODE)
- Auto-generated node reference manual
- Abstract graphics/sound rendering interface
- Core elements such as user-input navigation exposed to the authoring space (e.g. XML)
- Functional "node" evaluation using scripting languages such as 'R'

### Example:



*demo.xml:*

```
<Scene>
  <Inline url="skybox1.xml"/>
  <Inline url="player.xml"/>
  <Shape id="cube">
    <Appearance>
      <Material color="0.5 0.5 0.5" smooth="false"/>
      <TextureStage func="mul" source0="diffuse" source1="texture" coords="position">
        <CubeTexture all="wood2.jpg" mipmap="true"/>
      </TextureStage>
    </Appearance>
    <Box friction="0.0"/>
  </Shape>
  <Proto name="cube">
    <Body add_child="cube" position="0 10 0"/>
  </Proto>
  <World gravity="0 -9.81 0">
    <Emitter type="cube" rate="1" max="50"/>
    <Shape>
      <Appearance>
        <Material color="0.5 0.5 0.5" lit="false"/>
      </Appearance>
      <Box size="20 1.5 20" friction="0.5"/>
    </Shape>
  </World>
</Scene>
```

## VRB X3D toolkit (technology preview) [continue]

### Overview:

#### Node Index

<a href="#">Action</a>	<a href="#">Appearance</a>	<a href="#">AxisAction</a>	<a href="#">Background</a>	<a href="#">BallJoint</a>	<a href="#">Bindable</a>	<a href="#">Body</a>
<a href="#">BoolFilter</a>	<a href="#">Box</a>	<a href="#">ButtonAction</a>	<a href="#">Child</a>	<a href="#">CubeTexture</a>	<a href="#">Document</a>	<a href="#">Element</a>
<a href="#">Emitter</a>	<a href="#">Eval</a>	<a href="#">Geometry</a>	<a href="#">Group</a>	<a href="#">Hinge2Joint</a>	<a href="#">HingeJoint</a>	<a href="#">ImageTexture</a>
<a href="#">IndexedTriSet</a>	<a href="#">Inline</a>	<a href="#">IntInterpolator</a>	<a href="#">IntTrigger</a>	<a href="#">Joint</a>	<a href="#">Light</a>	<a href="#">Material</a>
<a href="#">NormalCubeTexture</a>	<a href="#">Object</a>	<a href="#">OutputChannel</a>	<a href="#">OutputSink</a>	<a href="#">Plane</a>	<a href="#">Proto</a>	<a href="#">Real3Interpolator</a>
<a href="#">Real3Scaler</a>	<a href="#">Real3Trigger</a>	<a href="#">Real4Interpolator</a>	<a href="#">RealInterpolator</a>	<a href="#">RealScaler</a>	<a href="#">RealTrigger</a>	<a href="#">RotationModifier</a>
<a href="#">Route</a>	<a href="#">Scope</a>	<a href="#">Script</a>	<a href="#">Shape</a>	<a href="#">Skybox</a>	<a href="#">SliderJoint</a>	<a href="#">Sphere</a>
<a href="#">TextNode</a>	<a href="#">Texture</a>	<a href="#">TextureStage</a>	<a href="#">TimeSensor</a>	<a href="#">TimeTrigger</a>	<a href="#">Transform</a>	<a href="#">TranslationModifier</a>
<a href="#">UniversalJoint</a>	<a href="#">Viewpoint</a>	<a href="#">World</a>				

### Details: C++ Implementation Kit

Use of modern C++ template techniques and syntactic sugar to ease development.

### Example: Background scheme driven by implementation

```
Background
Super Class
  Bindable
Fields
  init SFString id
  exposed SFBool bind
  exposed SFReal3 color
Derived Classes
  Skybox
```

```
Header background.hpp:
#ifndef VRB_BACKGROUND_HPP
#define VRB_BACKGROUND_HPP

#include "../core/bindable.hpp"

namespace vrb {
class RenderContext;
class Background : public Bindable
{
public:
    virtual void render(RenderContext& rc);
    static void def_fields(Class& c);
private:
    color mColor;
};
}

#endif // VRB_BACKGROUND_HPP
```

```
Implementation background.cpp:
#include "background.hpp"
#include "../core/class.hpp"
#include "../engine/renderer.hpp"

namespace vrb {
// type definition
void Background::def_fields(Class& c)
{
    c
        .def_constructor<Background>()
        .def( field("color", &Background::mColor ) )
    ;
}

VRB_DEF_CLASS(Background, Bindable)

// implementation
void Background::render(RenderContext& r)
{
    // render plain background
    r.graphics().clear_color( mColor );
}
}
```

Note: Node Reference available at [vrbdoc/ref.html](http://vrbdoc/ref.html)  
X3D Homepage: <http://www.web3d.org>