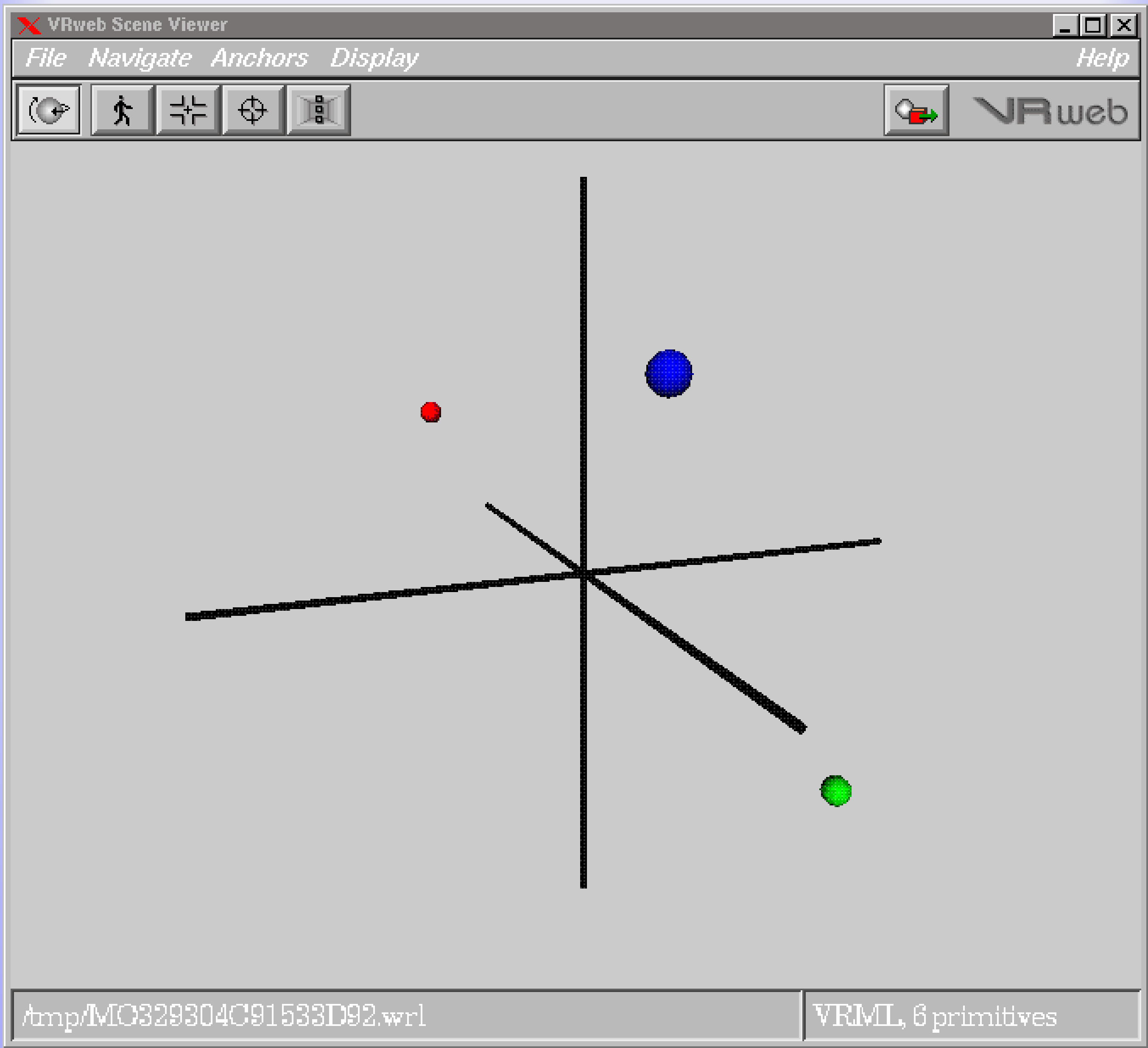
Event Display
VRWeb Browser

- What is VRWeb
  - Scenes
  - Graphical Objects
- Interfacing VRWeb to ATLAS
  - Interface Object
  - Helper Objects
  - Plottable Representant
- Example Application
  - Code
  - Running Program
- Conclusion
- Plans

VRWeb Browser

What is VRWeb

(1)



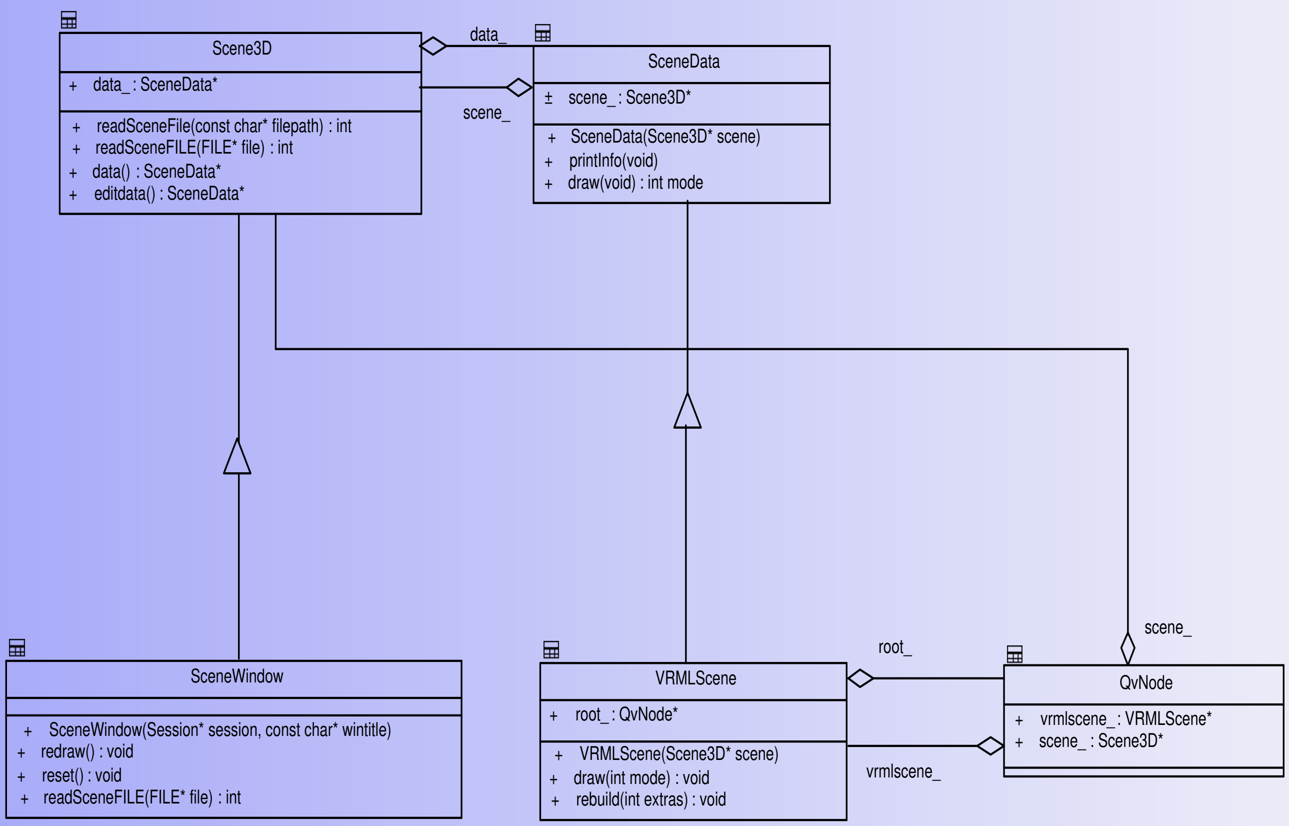
VRWeb Browser

What is VRWeb

(2)

- full VRML 1.0
- Graz University of Technology under **GNU**
- g++ and MSVC++, Interviews
- SGI Irix, Sun Solaris and OS, DEC Alpha and Ultrix, HP-UX, IBM AIX, Linux, FreeBSD, Windows NT and 95 and 3.x
- changing into Java to enable VRML 2.0

VRWeb Browser
Scenes

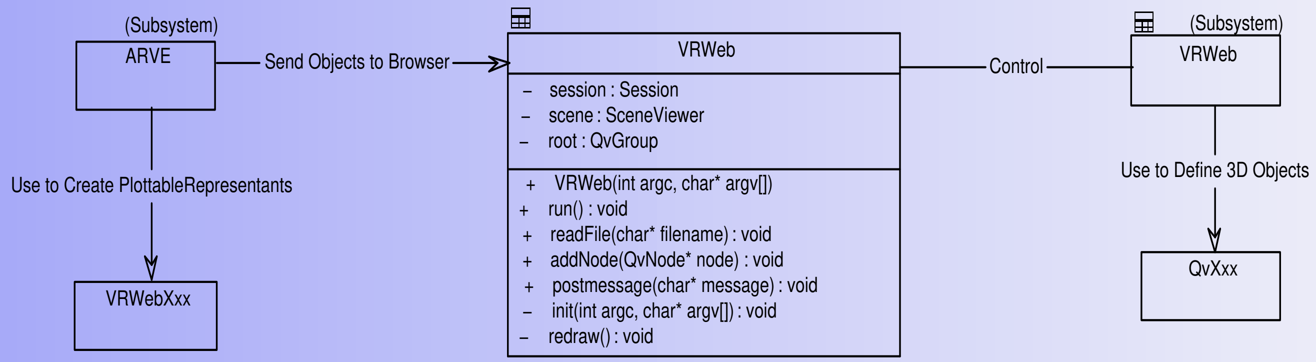


VRWeb Browser
Graphical Objects

- VRML 1.0: Cone
- Open Inventor: SoCone
- VRWeb: QvCone
- HEP-VRWeb: Cone

VRWeb Browser

Interfacing VRWeb to ATLAS



VRWeb Browser
Interface Object

```
class VRWeb
{
  public:
    VRWeb(int argc, char* argv[]);
    void run(); // Interview loop
    void readFile(char* filename);
    void readURL(char* url);
    void addNode(QvNode* node);
    void postMessage(char* message);
  private:
    void init(int argc, char* argv[]);
    void redraw();
    Session session; // Interview Session
    SceneViewer scene;
    QvGroup root; // top-level container
};
```

VRWEb Browser
Helper Objects

```
class VRWebCube : public QvCube
{
public:
    VRWebCube();
    VRWebCube(float w, float h, float d);
};

class VRWebObjectInSpace : public QvSeparator
{
public:
    VRWebObjectInSpace(QvNode& node,
        float t1, float t2, float t3,    // translation
        float r1, float r2, float r3); // rotation
};
```

VRWeb Browser
Plottable Representant

```
class VRWebPlottableRepresentant
{
  void plot(Scene&);
  void remove(Scene&);
//void update(Scene&); --- remove, plot
  void notify(PlottableModel*);
}

class VRWebScene
{
  void display(PlottableRepresentant*);
  void remove(PlottableRepresentant*);
//void update(PlottableRepresentant*); --- remove, plot
}
```

VRWeb Browser
Example Application

```
#include <vrweb.h>
#include "VRWebSphere.h"
#include "VRWebCube.h"
// include other helper classes
```

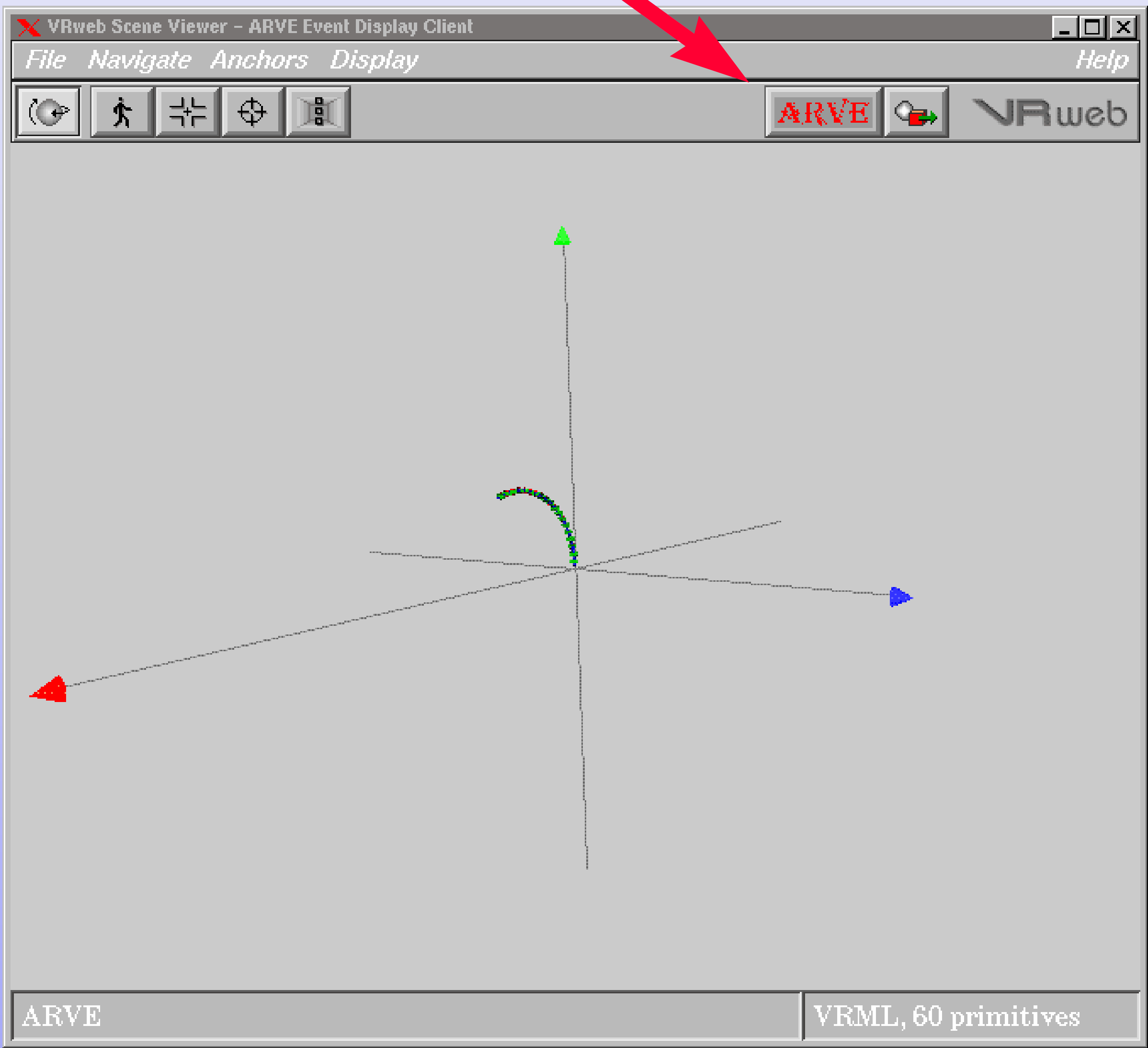
- link with libvrweb.a

VRWEb Browser
Code

```
int main(int argc, char* argv[])
{
  VRWeb aVRWeb(argc, argv);
  VRWebSphere sphere(2);
// create other objects, colors, transformations here
  QvSeparator funnyObject;
  funnyObject.children->append(&red);
  funnyObject.children->append(&sphere);
// add other objects, colors, transformation here
  aVRWeb.postMessage("ARVE");
  aVRWeb.addNode(&translation);
  aVRWeb.addNode(&funnyObject);
  aVRWeb.run();
}
```

VRWeb Browser
Running Program

can interact with running ARVE



VRWeb Browser

Conclusion

- it's *quite easy* to interface graphical application
- the strict interface should be defined to prevent mess
- <u>VRWeb</u> seems suitable for 3D Event Display graphics (but others can exist simultaniously)
- there are good non-comercial packages

VRWeb Browser
Plans

- create full VRWebPlottableRepresentant, VRWebScene
- implement threads
- create suitable HEP-VRWeb library
- make shared library
- use with real ARVE
- compile on WNT
- use VRWeb 2.0, when available