20 May 99, CERN

Graphics for Analysis

Architecture

First Principles

Top-level Design:

- SubDomain Decomposition
- Control Structure
- Use Cases

Selected Pieces:

- Democracy of Scenes
- XML/DOM

Status

Requirements

Design

Implementation:

- Core System
- Scenes (Views)
- Plottables (Data)

Documentation

Domain Interfaces

Evolvability

You need to build a system that is futureproof; it's no good just making a modular system.
You need to realize that your system is just going to be a module in some bigger system to come, and so you have to be part of something else, and it's a bit of a way of life.

Tim Berners-Lee at the WWW7 Conference

Architecture / First Principles

The aim of the Atlas Graphics is to enable visual representation of the objects existing in the Atlas software. The Design of the Atlas Graphics is based on the believe that both requirements and graphics software abilities will be very broad at any time and will constantly evolve. The Atlas Graphics should be able to accommodate all that diversity and change. This can be accomplished only by extreme flexibility and modularity of the core control structure. The Atlas Graphics is part of the full Atlas software, it covers its grahical components (Histograming, EventDisplay, GUI,...).

Graphics interacts both with the data and with the reconstruction package. Graphics consists of a set of views showing geometrical representation of various real objects via graphical objects. Operations are performed on the views and contained graphical objects as well as to original real objects. Any real object (which can be any object, candidate for being displayed) has the potential to be displayed. All objects can be displayed in some way, some objects will be displayable in more ways than others.

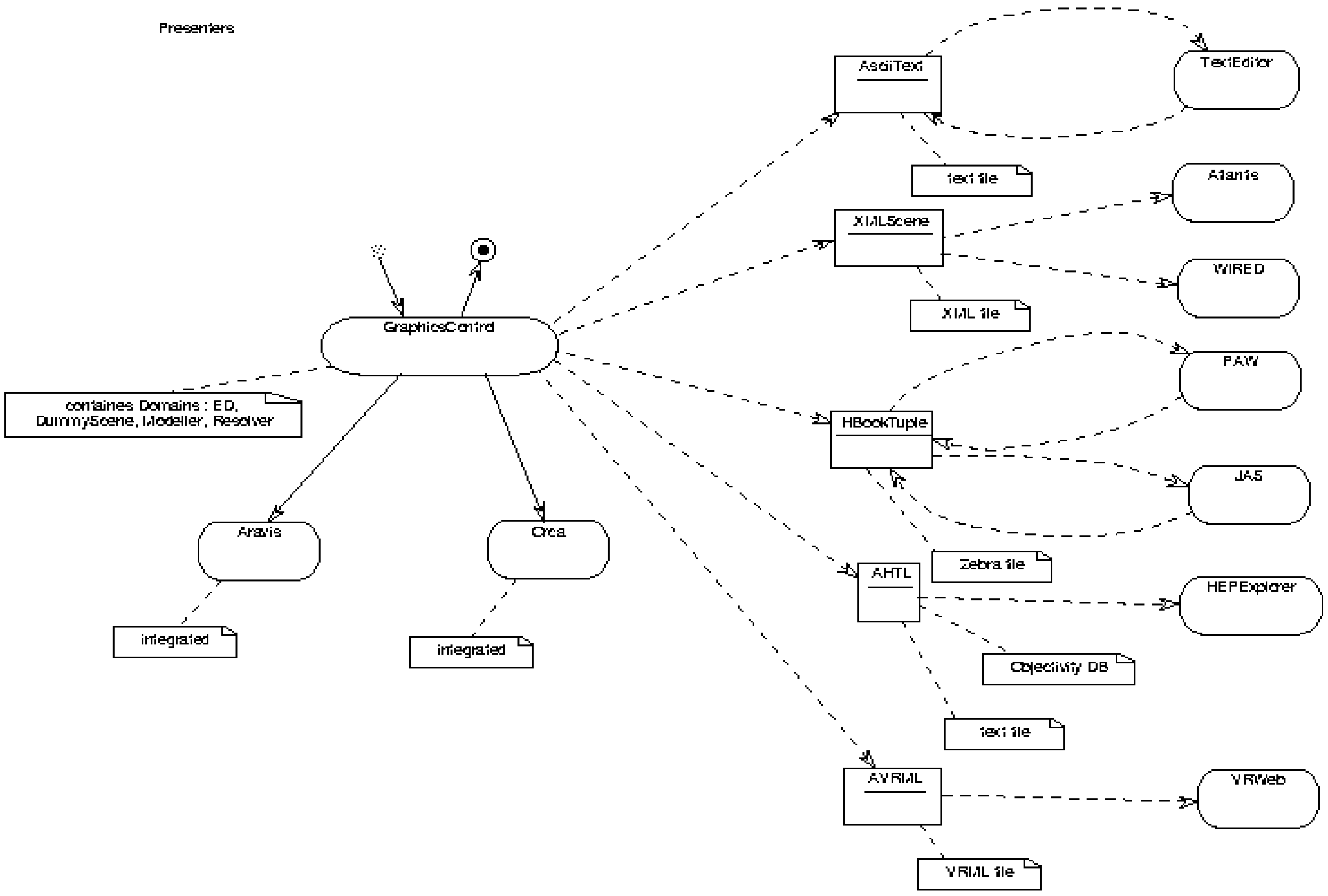
Architecture / First Principles

The major architectural principles of the Graphics are:

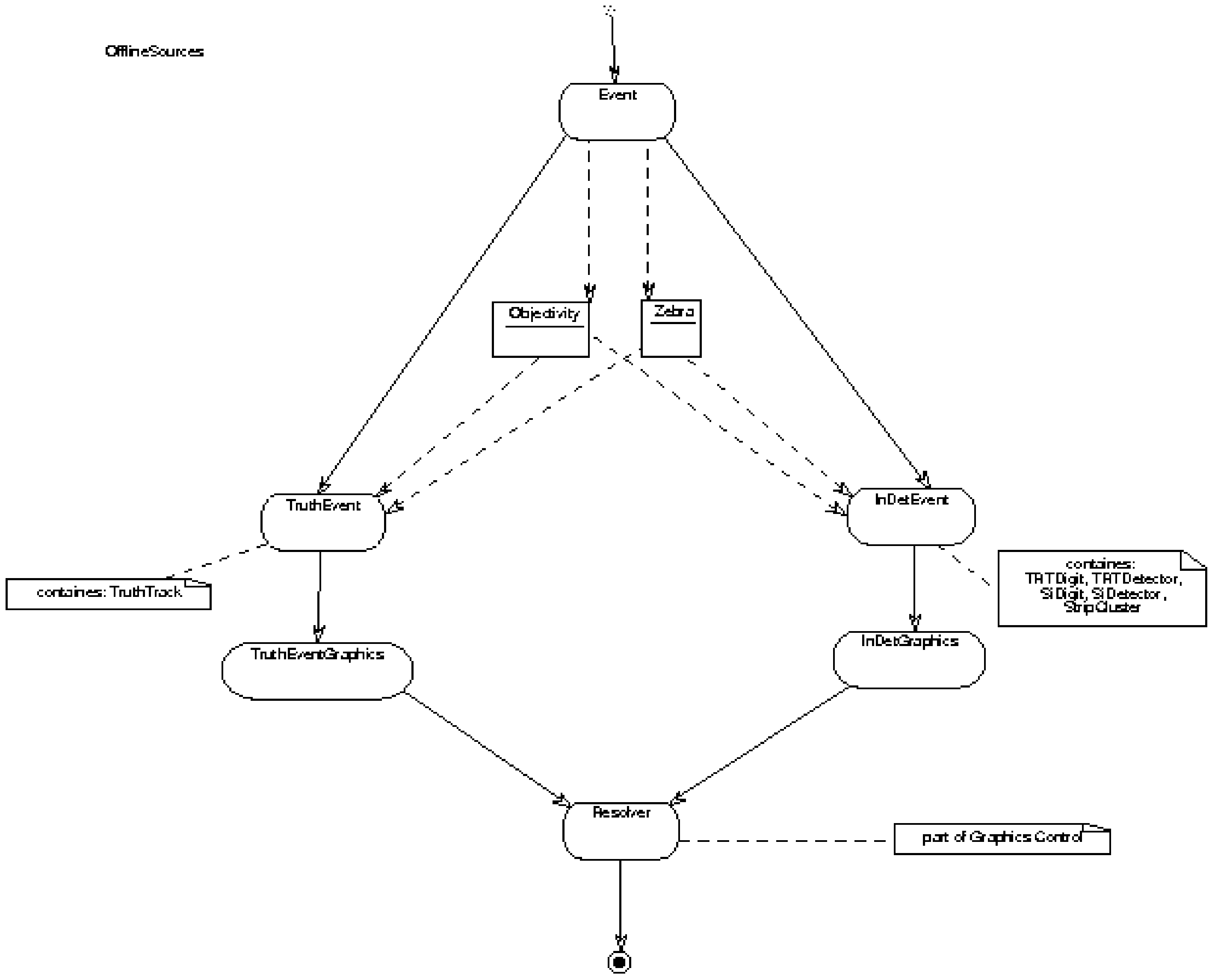
The fact, that any object is visualised should not influence the design of that object.

The design of the graphics should not depend on the any particular visualisation software.

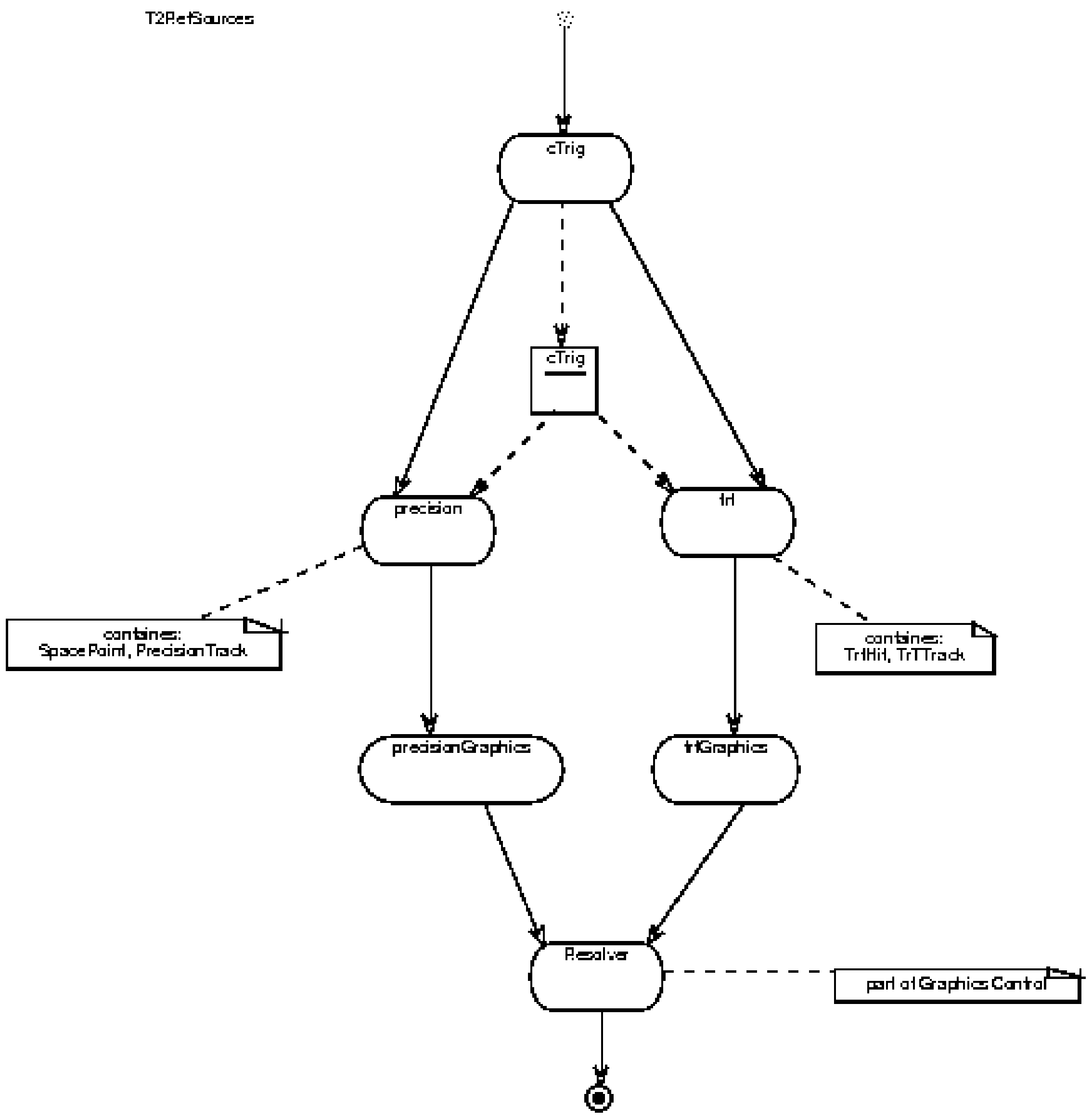
Architecture / Top-level Design - SubDomain Decomposition



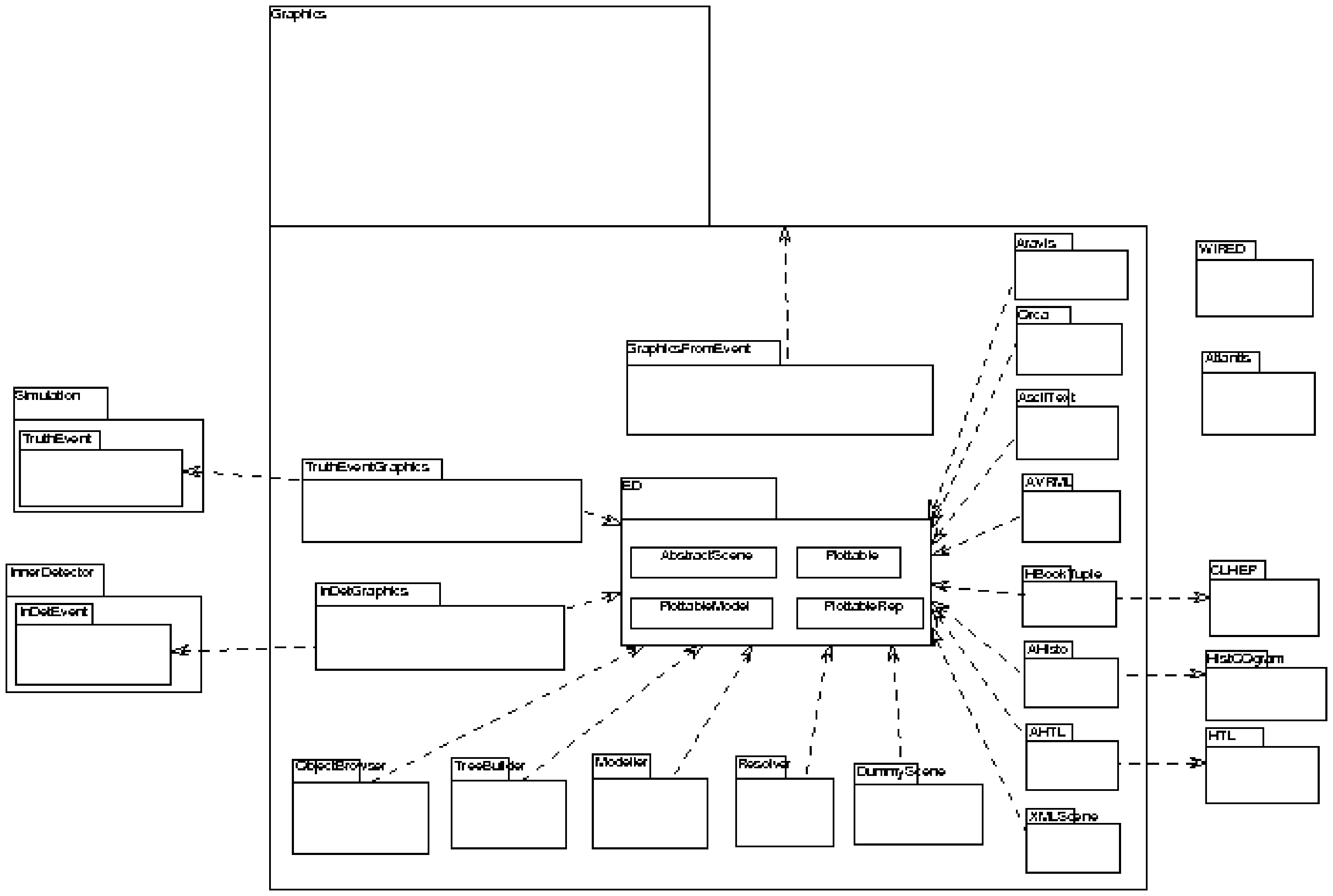
Architecture / Top-level Design - SubDomain Decomposition



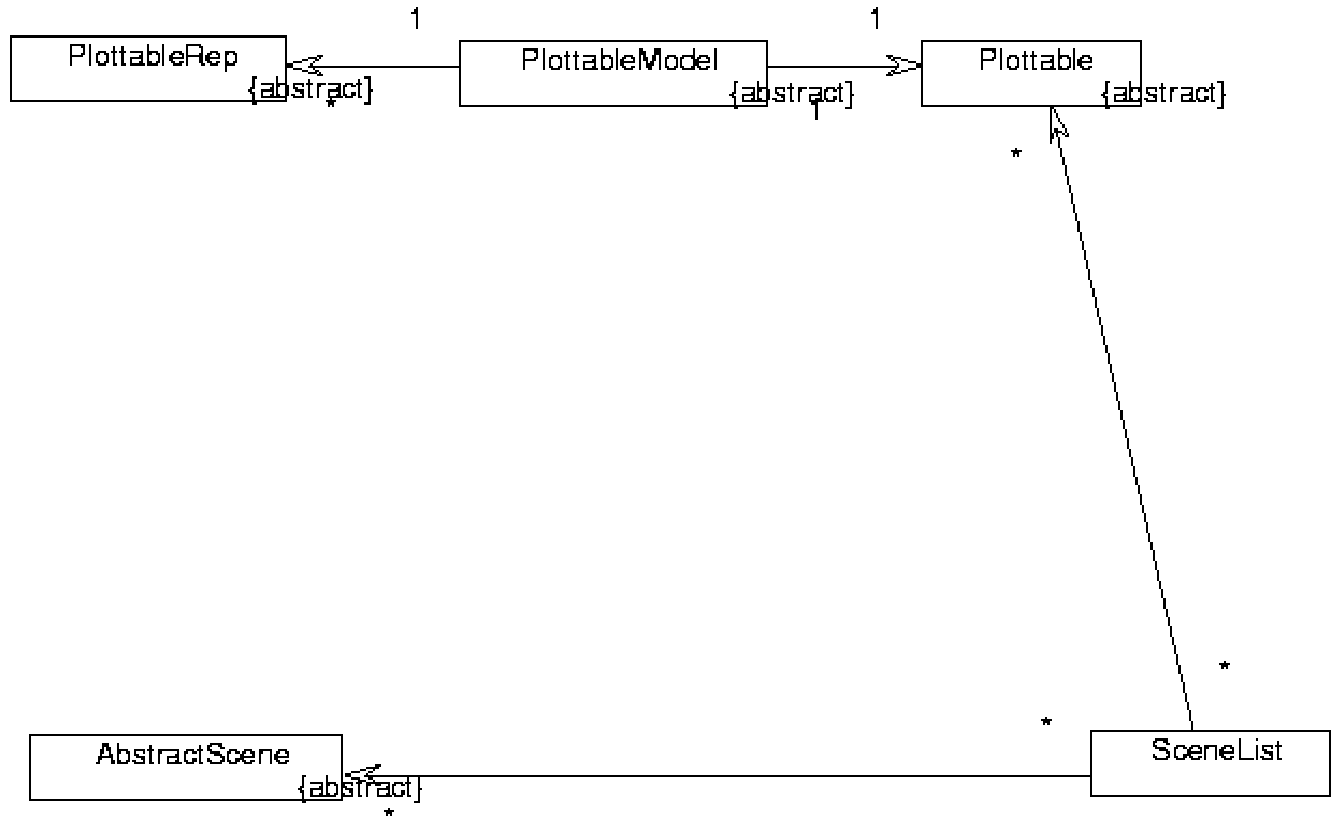
Architecture / Top-level Design - SubDomain Decomposition



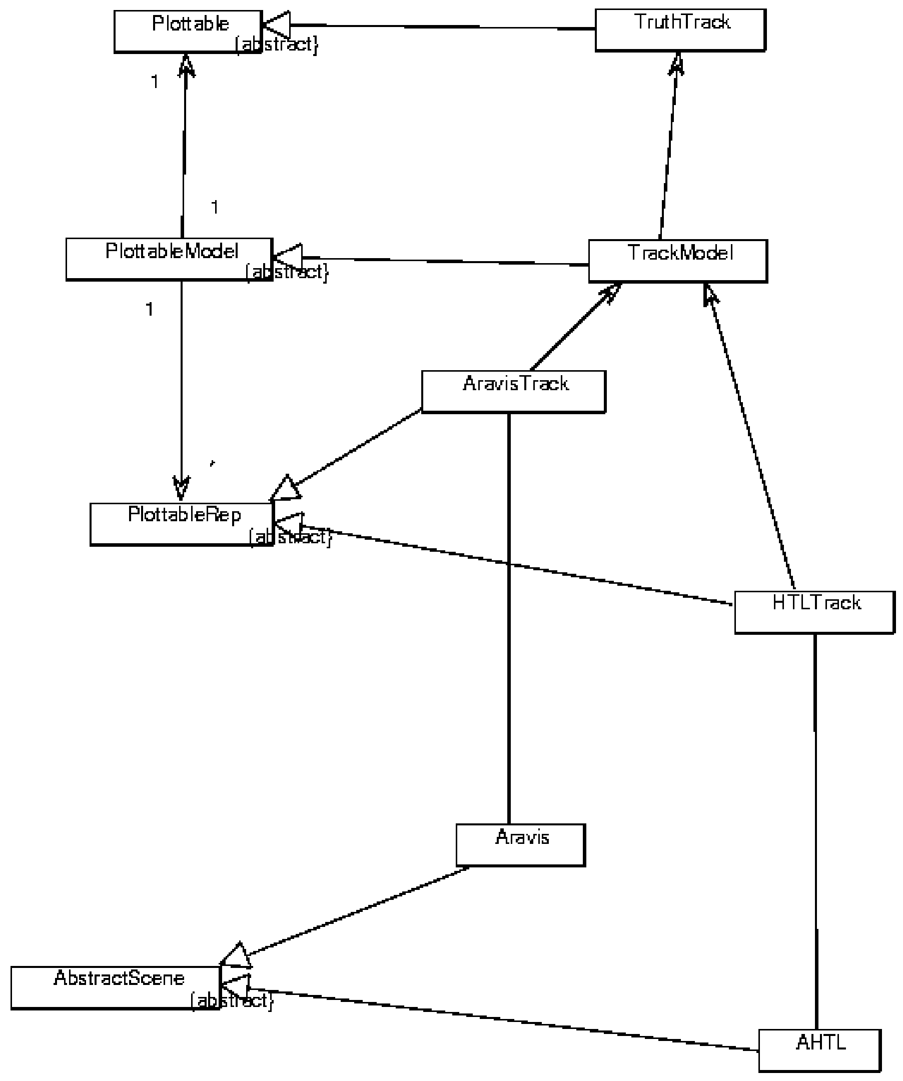
Architecture / Top-level Design - SubDomain Decomposition



Architecture / Top-level Design - Control Structure



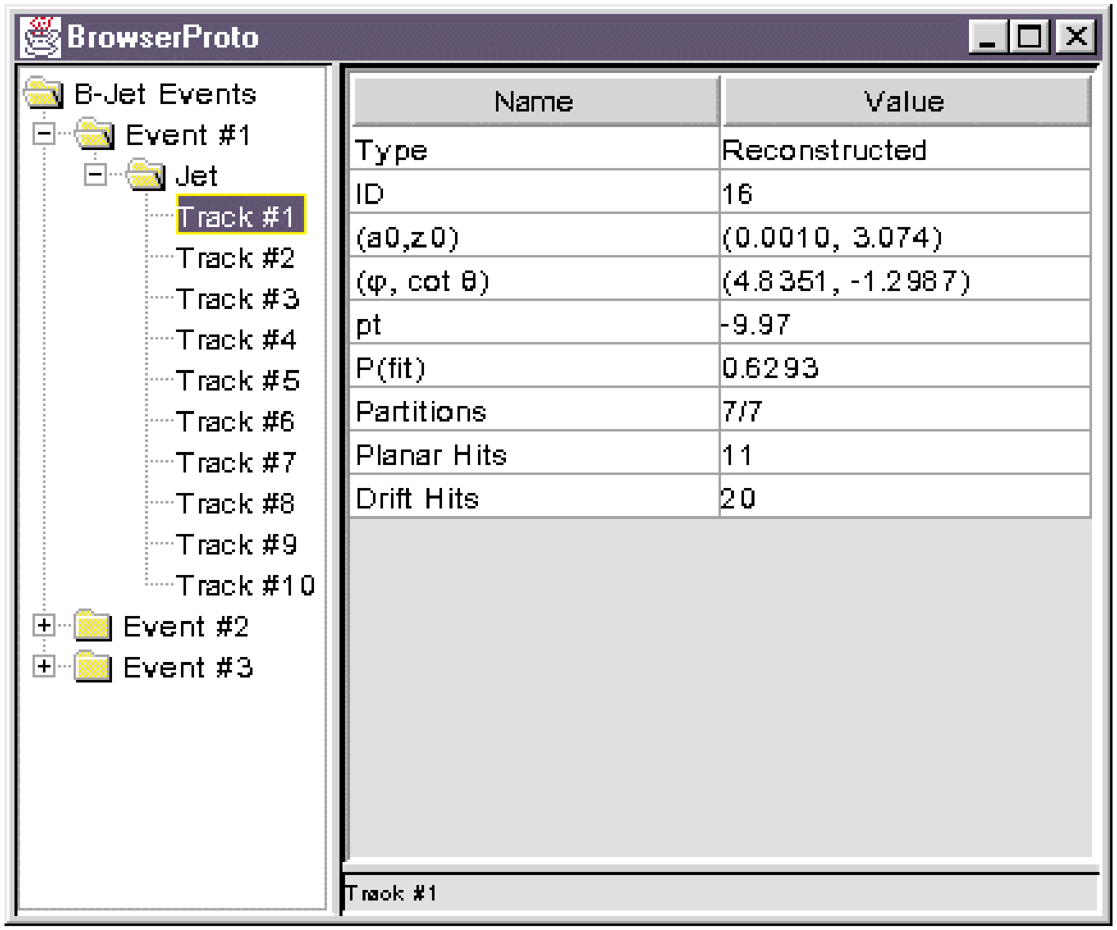
Architecture / Top-level Design - Control Structure



Architecture / Top-level Design - Use Cases

End User

. ObjectBrowser



BrowserProto

Name	Value
Type	Reconstructed
ID	16
(a0,z0)	(0.0010, 3.074)
(φ, cot θ)	(4.8351, -1.2987)
pt	-9.97
P(fit)	0.6293
Partitions	7/7
Planar Hits	11
Drift Hits	20

Track #1

Architecture / Top-level Design - Use Cases

End Programmer

```
SceneList sl;                        // create control
XMLScene xml("MyFile");              // open XML file
AVRML vrml("MyFile");                // open VRML file
HBookTuple hbook("MyFile");          // open HBook file
Aravis aravis();                     // open Aravis window
sl.add(xml);                         // register xml
sl.add(vrml);                        // register vrml
sl.add(hbook);                       // register hbook
sl.add(aravis);                      // register aravis


// .... create or get TruthTrack

sl.show(TruthTrack);                 // send TruthTrack to xml, vrml, hbook
                                     // show TruthTrack on aravis
```

then look

at MyFile.xml with WIRED or Atlantis

at MyFile.wrl with VRWeb

at MyFile.hbook with Paw or Jas

Architecture / Top-level Design - Use Cases

Plottable Developer

```
$ createPlottableModel TruthTrack          # create sceletons for all classes needed
$                                          # for TruthTrack visualisation
$                                          # they will compile and link, but without
$                                          # any real nontrivial efect
$
$ nedit VRMLTruthTrack.cxx                # define VRML behaviour
$ nedit HBTupleTruthTrack.cxx             # define HBook behaviour
```

```
// pTrack is available
// ...
aVertex = pTrack->vertex();        // get Vertex
HepPoint3D pos(aVertex);           // create position
HepPoint3D dir(pTrack->p());       // create direction
Line aLine(pos, dir);              // construct line
add(aLine);                        // add line
// ...
```

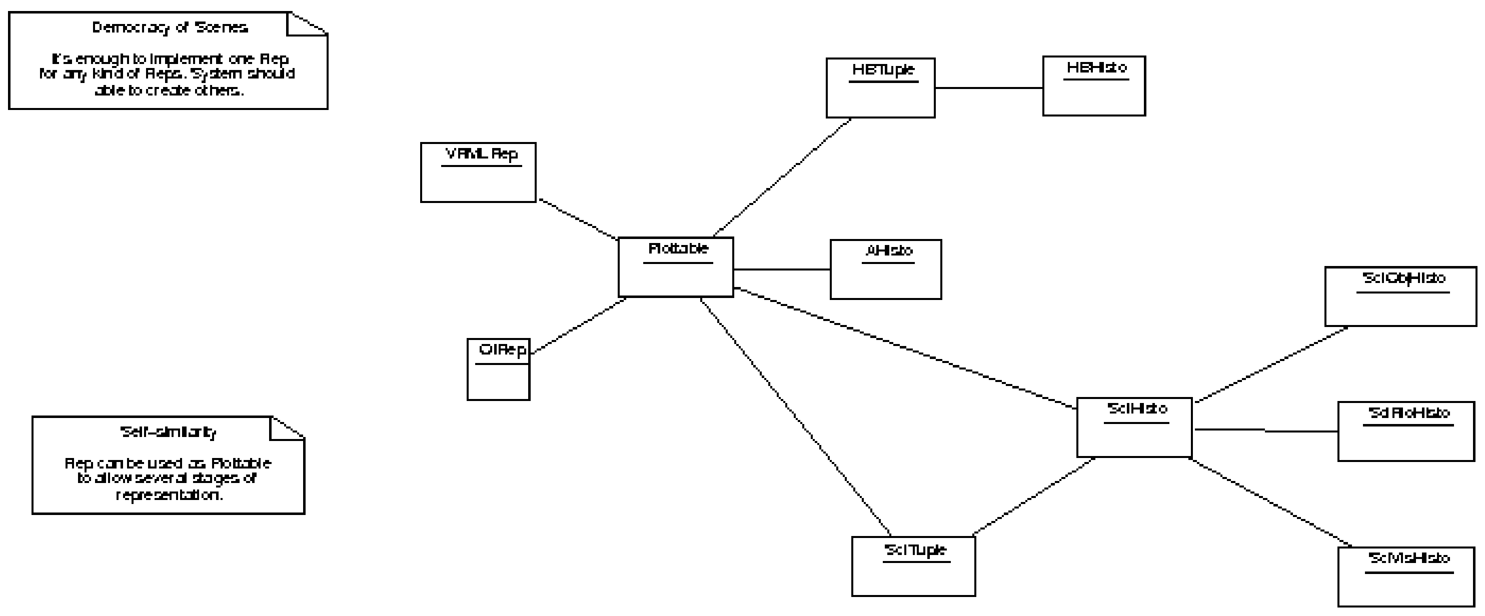
```
// pTrack is available
// ...
tuple->column("px", pTrack->p_x(), 0);  // add px
tuple->column("py", pTrack->p_y(), 0);  // add py
tuple->column("pz", pTrack->p_z(), 0);  // add pz
// ...
```

Architecture / Top-level Design - Use Cases

Scene Developer

1) implement Scene, conforming to the standard interface, which is connected to the Scene and Rep classes
2) write documentation
3) include sceletons for automatic creation of Reps
4) include test
5) implement Reps for existing Plottables

Architecture / Selected Piceses - Democracy of Scenes



Democracy of Scenes

It's enough to implement one Rep for any Kind of Reps. System should able to create others.

Self-similarity

Rep can be used as Plottable to allow several stages of representation.

Architecture / Selected Pieces - XML/DOM

today - just using standard Graphics interface (XMLScene),
XML files are simplified representation of data,
used by WIRED, Atlantis, XML browsers

in future - using also Event mechanisms,
XML files are full representation of DetectorDescription
and Event data,
DTD generated dynamicaly
- statistical XML objects with inlined DTD,
interface to other XML tools

Status / Requirements

Definitions:

- Real and Graphical Objects
- Operations, Operations on Real Objects, Operations on Graphical Objects
- Views
- Static Objects (geometry,...)
  Streaming Objects (statistical, acumulative,...)
  Removable Objects (event,...)

Requirements:

- General (should be fulfilled everywhere)
  Existence (should be fulfilled somewhere)
  Environment (environment should be provided)
- Functional Requirements, System Properties, Constraines

Status / Design

constantly evolving
constatly beeing implemented

passed 2 ASP Reviews

current version available on WWW

Status / Implementation - Core System

ED, Modeler, Resolver - functional, simple programming access
to graphics (via SceneList)
- will be upgraded into new
Design/Implementation (multimethods,...)

ObjectsBrowser - user-friendly & powerfull GUI (L.Tuura)
- foundation and prototype exist

TreeBuilder - temporary implementation of tree structures

Status / Implementation - Scenes (Views)

Event Display:

AVRML - 3D view

fully implemented, viewable by VRWeb, MSIE, Netscape

Aravis - integrated, simple (T.Burnett, R&D.Candlins)

ramp-up-ed Arve graphics

system implemented

next: Reps

Atlantis - sophisticated physicists Event Display (H.Drevermann & comp.

well implemented

problem with access to data (C++ - F77)

next: better XML parser

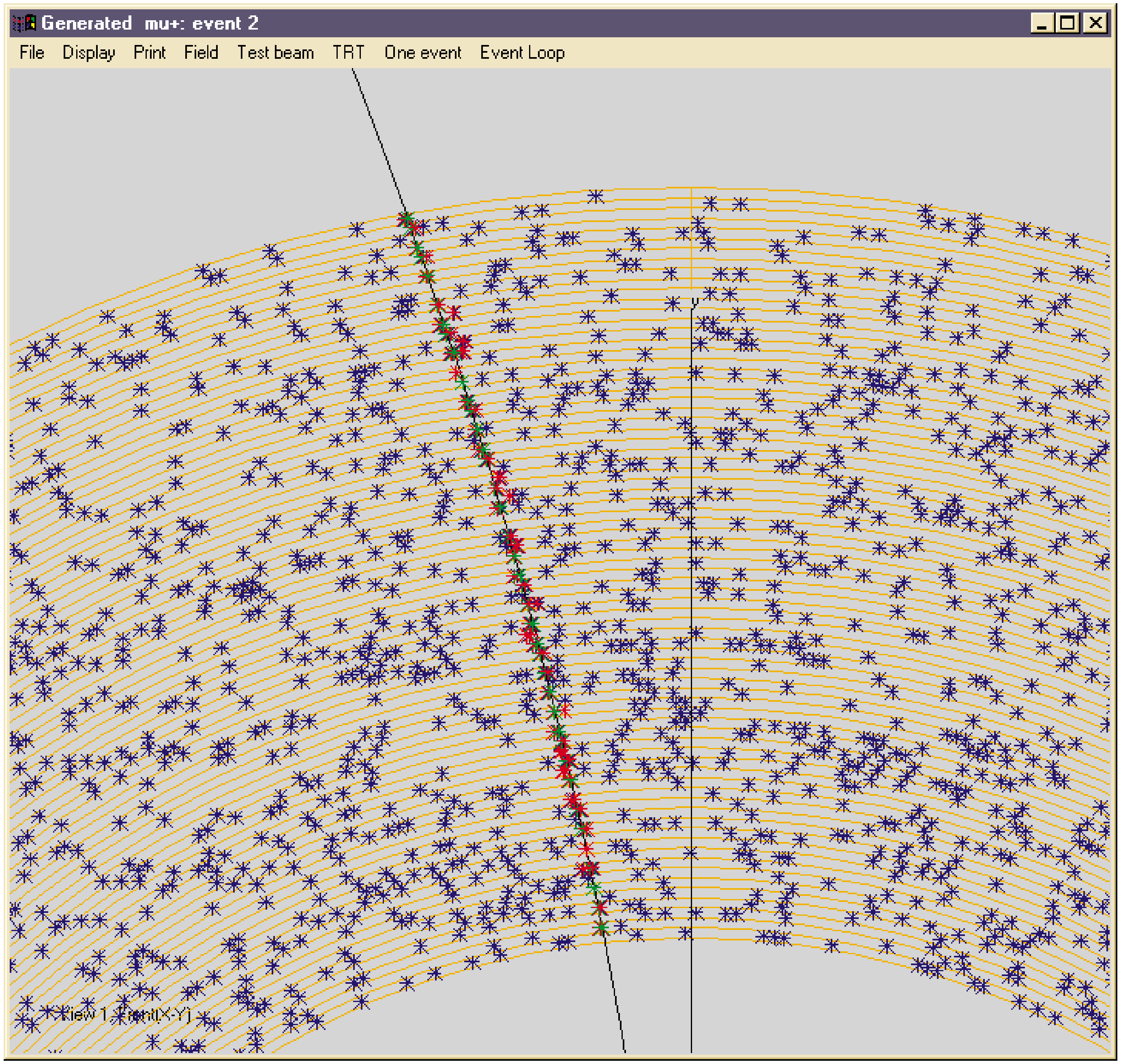
Wired - full Event Display in Java (M.Donszelmann & comp, D.Koper)

well implemented

next: feedback

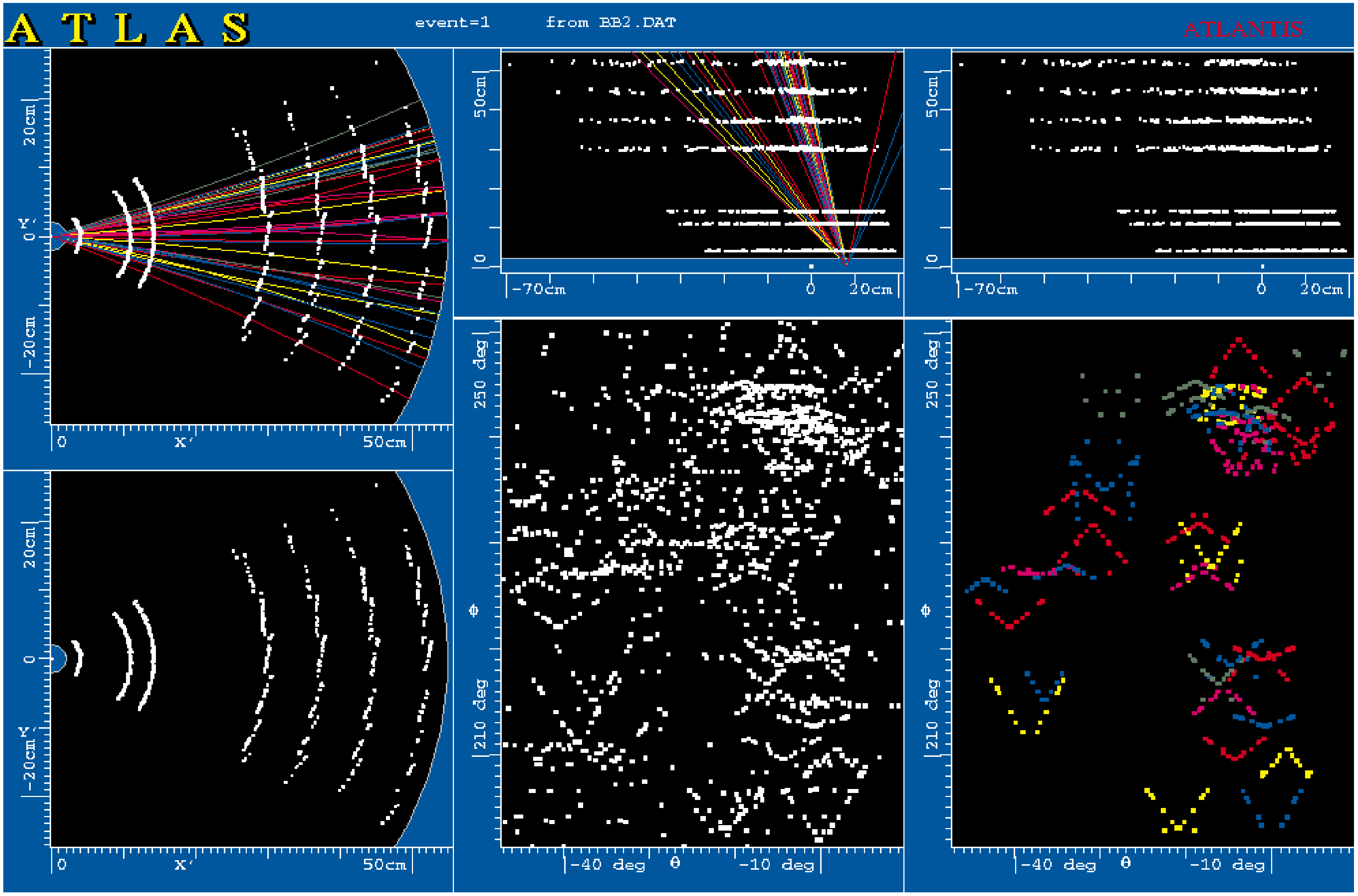
Status / Implementation - Scenes (Views)

Aravis



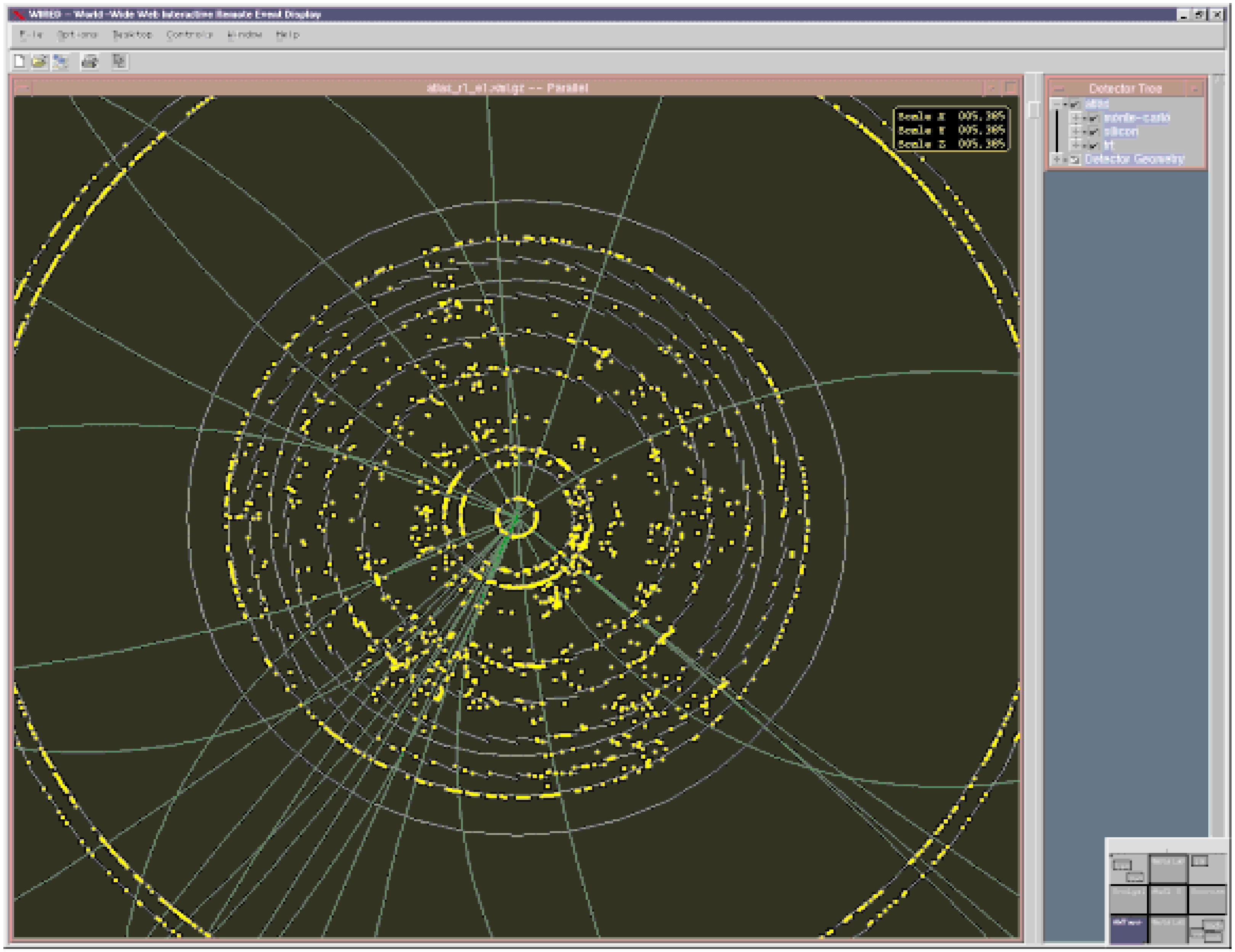
Status / Implementation - Scenes (Views)

Atlantis



Status / Implementation - Scenes (Views)

Wired



Status / Implementation - Scenes (Views)

Statistics:

HBookTuple - writes HBook files
implemented, quite obsolete

AHisto - writes HistOOgrams into Objy
simple implementation, quite obsolete

AHTL - creates HTL histograms
simple implementation

AOS - creates Open Scientist histograms
just plans

AJas - interface to Jas
just plans

Orca - simple integrated environment (T.Burnett)
works on NT, not clean interface

Status / Implementation - Scenes (Views)

Misc:

AsciiText - just textual output
fully implemented

XMLScene - output into XMLFiles
well implemenented
used by Wired, Atlantis,...
next: will expand to more general text interchange
file format

Command - using Plottable-Model-Rep pattern for G(UI)
just initial design

Status / Implementation - Plottables (Data)

Offline:

SiDetector, SiDigit, TRTDetector, TRTDigit ✓
Muon***
LArg***
Tile***
StripCluster ✓
SpacePoint
TruthTrack ✓
OutputTrack

subsystem involvment
urgently needed
(cca 10% FTE per subsystem)

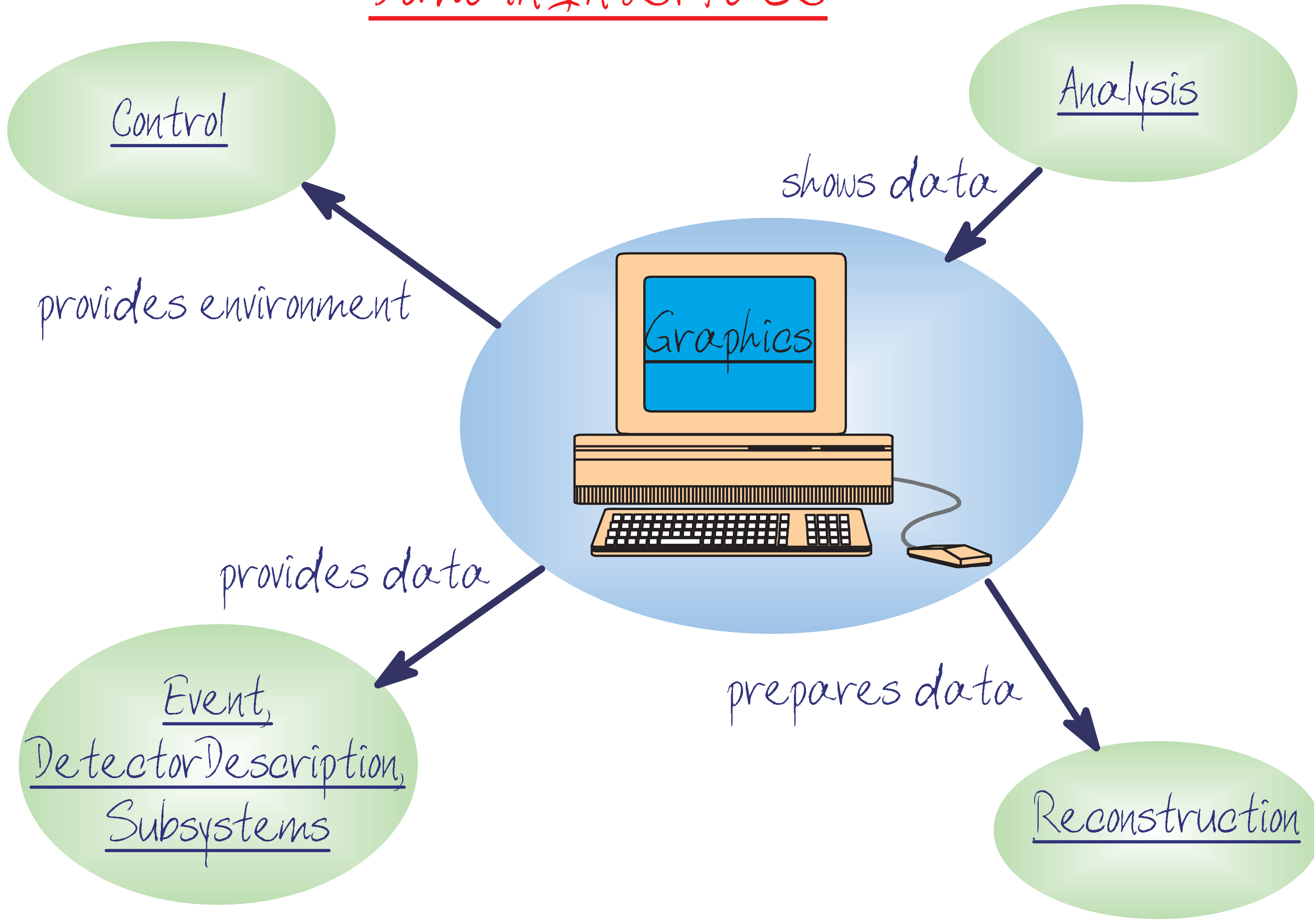
Trigger Ref:

SpacePoint, TrtHit ✓
PrecisionTrack, TrtTrack ✓

Status / Implementation - Documentation

- Implementation Guidlines (for both offline and t2ref)
- Frequently Asked Questions (automaticaly created from DB)
- Design (StP)
- Packages Documentation (automaticaly extracted to WWW)

Domain Interface



18 May 99, CERN

Atlas Graphics

1) Status and Plans

Data
- Offline
- TriggerRef

→

Core
- Control
- ObjectBrowser
- TreeBuilder

→

Scenes
- Aravis
- Atlantis
- Wired
- XMLScene
- AHisto
- (G)UI

2) XML

3) AOB

Data

Offline:

SiDetector, SiDigit, TRTDetector, TRTDigit ✓

Muon***

LArg***

Tile***

StripCluster ✓

SpacePoint

TruthTrack ✓

OutputTrack

Trigger Ref:

SpacePoint, TrtHit ✓

PrecisionTrack, TrtTrack ✓

Core

Control:

Feedback from Aravis

Simple extensions

ObjectBrowser:

Core mechanism + Prototype installed

TreeBuilder:

Mechanism functional

Not all combinations of features available

Temporary solution

"Standard Tree" implemented

Scenes

Aravis:
Core installed

next: Reps

XMLScene:
Works fine

next: use Expat + ExpatInterface

Atlantis:
Installed

next: change XML parser

AHisto:
Hist00grams obsolete

next: OpenScientist Histo, HTL

Wired:
Installed

First Feedback

G(UI):
nothing yet

XML

Now

- standard way of creation (XMLScene)
- three files (generaly): Detector + Event + Relations
- relations via id + rid
- user definable name attribute
- TagName = ClassName
- AttributeName != MemberName

Future

- also Event Domain mechanims of creation
- also full replica of Data (DTD created from DDL ?)
- use for Hustogram objects (with inlined DTD)