Example 5:
Graphics

- Concepts

- Part 0:
  - Scenes, Plottables, SceneList

- Part 1:
  - Graphics as Algorithm
  - Aravis
  - XML (WIRED, Atlantis, GraXML)

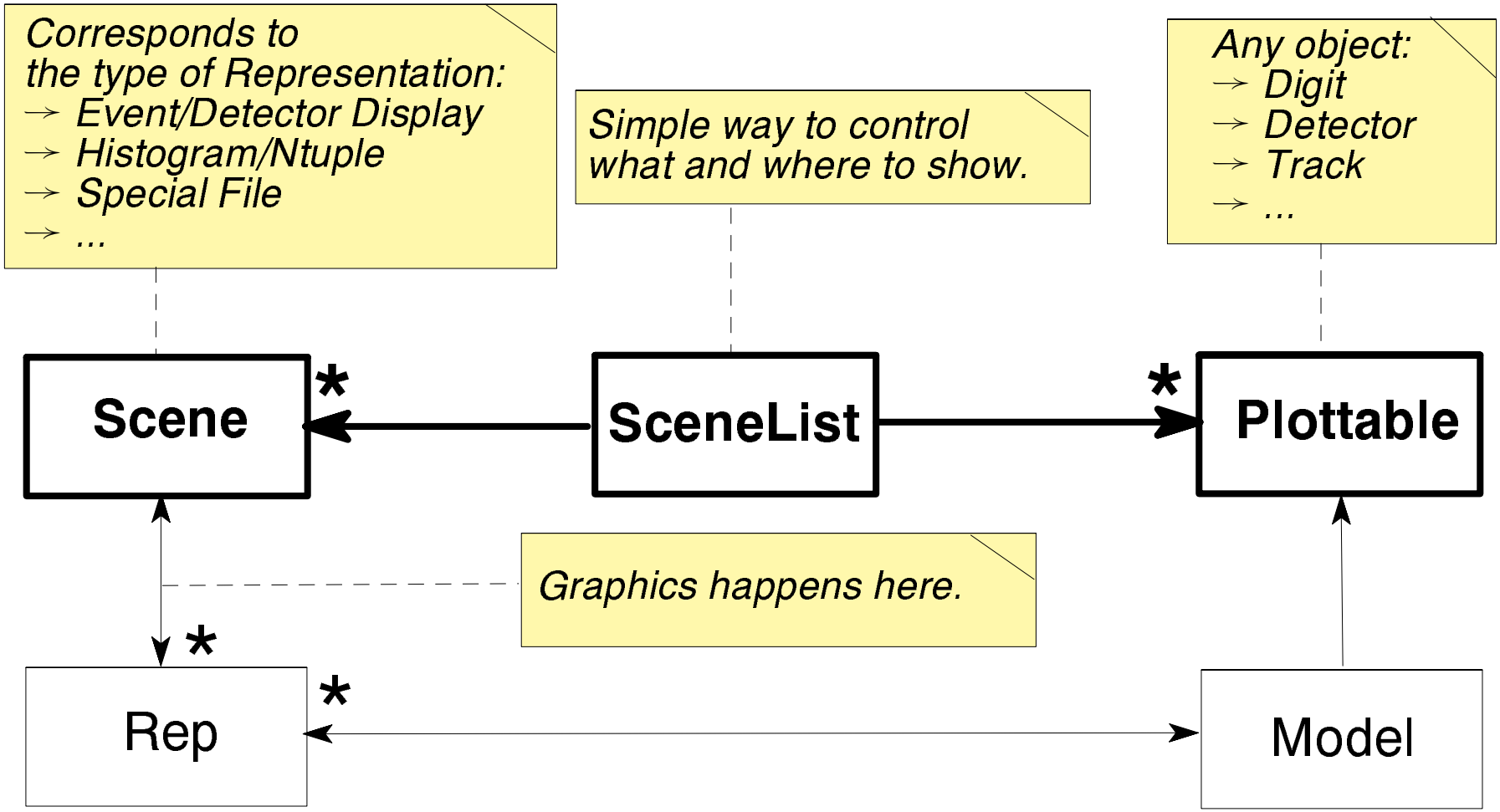
- Part 2:
  - Other Scenes

- Files

- Graphics.h, Graphics.cxx

Example 5: Part 0

Scenes, Plottables, SceneList



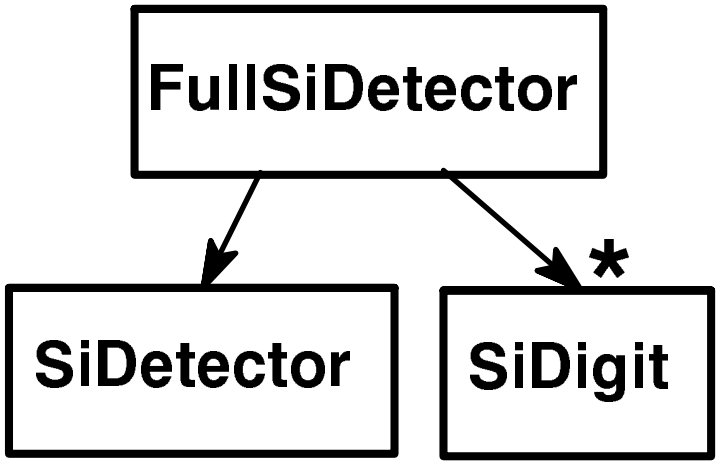
Example 5: Part 1
Graphics as Algorithm

- Include **SceneList** and reguired **Scenes:**

```
// Graphics.h

#include "Modeller/SceneList.h"

#include "AsciiText/AsciiText.h"
#include "XMLScene/XMLFile.h"
#include "Aravis/Aravis.h"
```

- Include **Plottables** to be displayed:

```
// Graphics.h

#include "InDetEvent/SiDetector.h"
#include "InDetGraphics/FullSiDetector.h"
#include "GeneralPlottable/Caption.h"
```



Example 5: Part 1
Graphics as Algorithm (cont)

- Declare *LevelOfDetails*, **SceneList** and **Scenes**:

```
// Graphics.h

int   m_txt_level;
int   m_xml_level;  // One XML file for all Events
int   m_aravis_level;
int   m_xml0_level; // One XML file per Event
int   m_pe_level;   // Pixel SiDetector
int   m_pi_level;   // Pixel SiDigit

SceneList   m_list;
AsciiText*  mp_txt;
XMLFile*    mp_xml;
Aravis*     mp_aravis;
```

Bigger the number is, more details are shown. Usually: LevelOfDetails < 9.

Example 5: Part 1
Graphics as Algorithm (cont)

- Get *LevelOfDetails* Properties:

```
// Graphics.cxx / Graphics::Graphics(...)

declareProperty("TXT",              m_txt_level);
declareProperty("XML",              m_xml_level);
declareProperty("Aravis",           m_aravis_level);
declareProperty("XMLperEvent",      m_xml0_level);

declareProperty("PixelDetectors", m_pe_level);
declareProperty("PixelDigits",      m_pi_level);
```

Example 5: Part 1
Graphics as Algorithm (cont)

- Create and register requested global **Scenes**:

```
// Graphics.cxx / Graphics::initialize()

if (m_txt_level > 0) {
  mp_txt    = new AsciiText("index", m_txt_level);
  m_list.add(*mp_txt);
  }
if (m_xml_level > 0) {
  mp_xml    = new XMLFile("index", m_xml_level);
  m_list.add(*mp_xml);
  }
if (m_aravis_level > 0) {
  mp_aravis = new Aravis();
  m_list.add(*mp_aravis);
  }
```

Example 5: Part 1
Graphics as Algorithm (cont)

- Get **Event**:

```
// See previous Examples
```

- Create and register requested per-**Event Scene**s:

```
// Graphics.cxx / Graphics::execute()

XMLFile* xml;
if (m_xml0_level > 0) {
  xml = new XMLFile(name, m_xml0_level);
  m_list.add(*xml);
  }
```

- Get **Pixel SiDetector**s:

```
// See previous Examples
```

Example 5: Part 1
Graphics as Algorithm (cont)

- Loop over **Pixel SiDetector**s,
- create **FullSiDetector**s,
- add them into **SceneList**:

```
// Graphics.cxx / Graphics::execute()

for(; first != last; ++first) {
  FullSiDetector* silicon =
    new FullSiDetector(**first, m_pe_level, m_pi_level);
  m_list.add_to_scene(silicon, 9);
  }
```

Using directly SiDetector and SiDigit works too.

- Show it all:

```
// Graphics.cxx / Graphics::execute()

m_list.show();
```

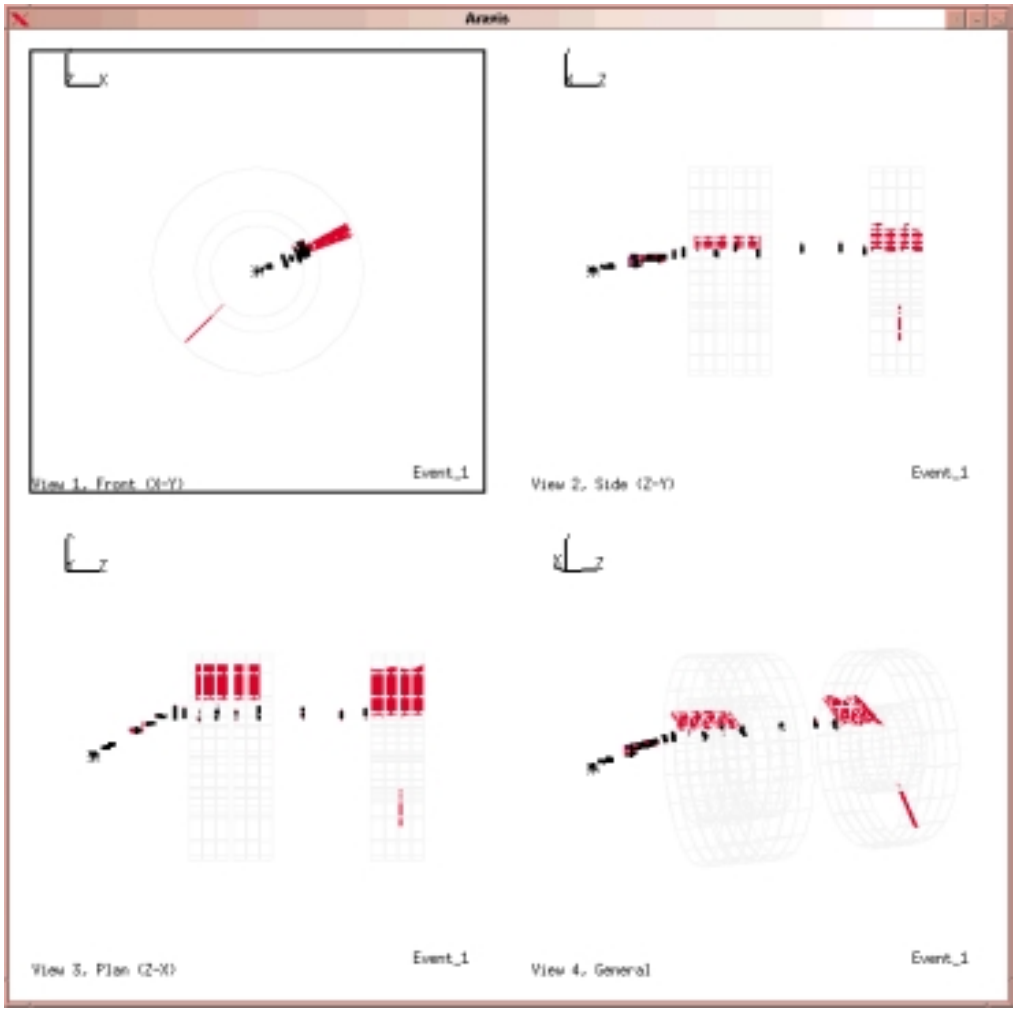
Example 5: Part 1
Graphics as Algorithm (cont)

- Only Pixels shown,
- Curerntly supported:
  - Full ID
  - TruthTracks
  - MDTs, RPCs
  - CaloDigits, CaloRegion

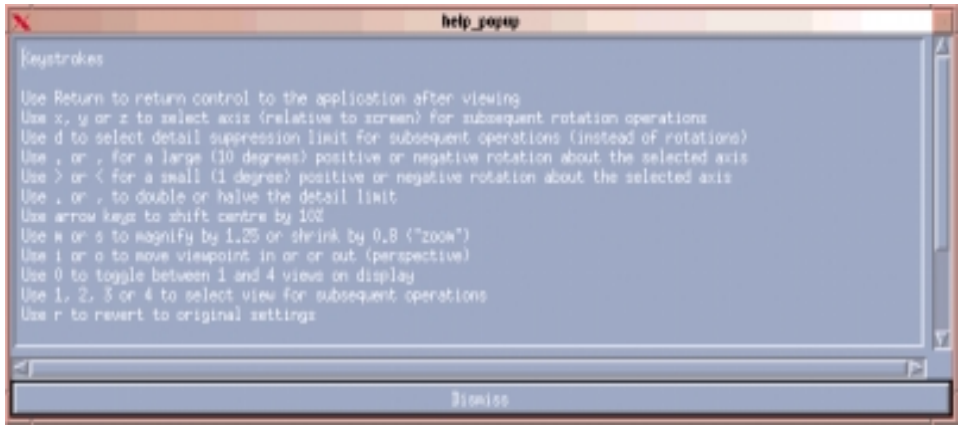
- Only the simplest way of using Graphics shown.

- Not shown:
  - Code, covered in previous Examples.
  - Code, solving C++ problems (memory management,...).
  - Some details.

Example 5: Part 1
Aravis

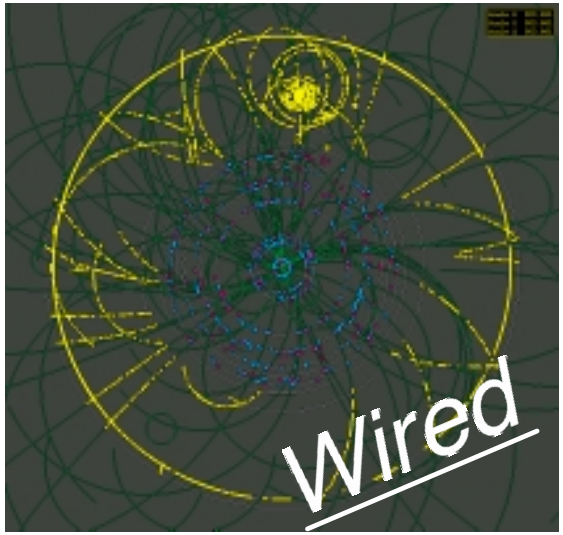


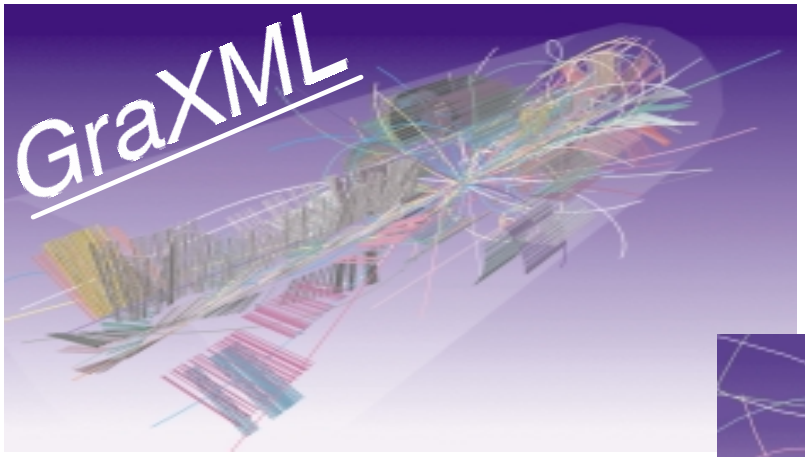
Integrated in the C++ Framework.



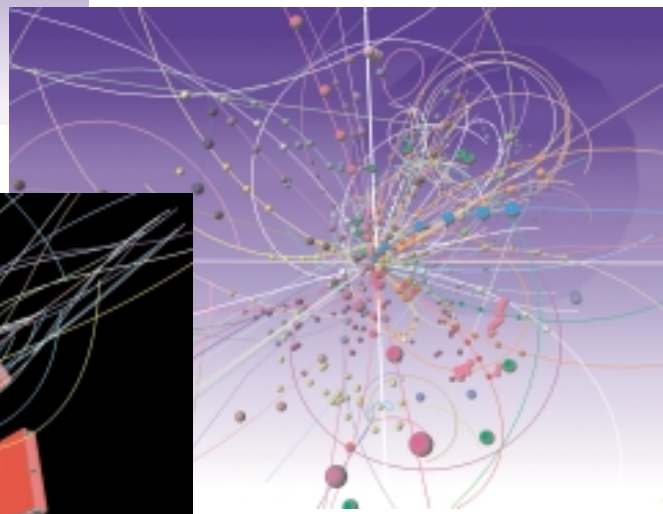
http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/Aravis/

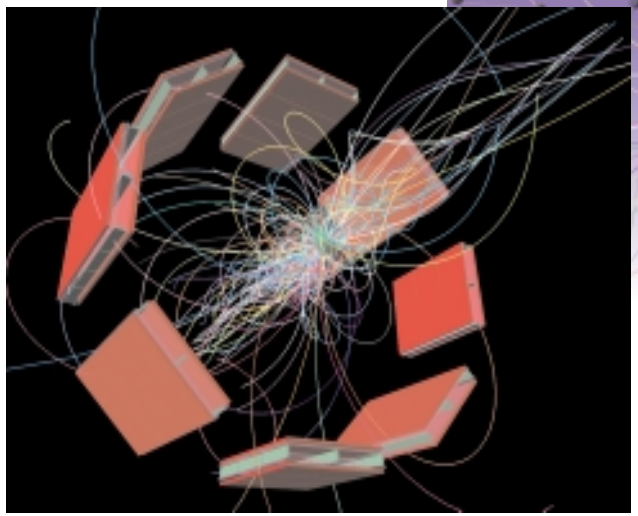
Example 5: Part 1
XML (WIRED, Atlantis, GraXML)





Interfaced via XML files.





http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/WIRED/
http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/Atlantis/
http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/Documentation/GraXML/

Example 5: Part 2
Other Scenes

- C++
- AsciiText
- Aravis
- XMLScene
- Atlantis (via XML)
- WIRED (via XML)
- GraXML (via XML)
- HBookTuple - obsol ete
- AHTL - obsolete

- Java
- Text
- XML
- AtlasEvent
- X3D
- PlotML
- jAtlantis
- WIRED
- GraXML
- JAS

Access to Java Scenes is done in the transparent way to the C++ user, who just has to use different SceneList.

Java Scenes will be available as soon as Java is supported by the Atlas.

http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/