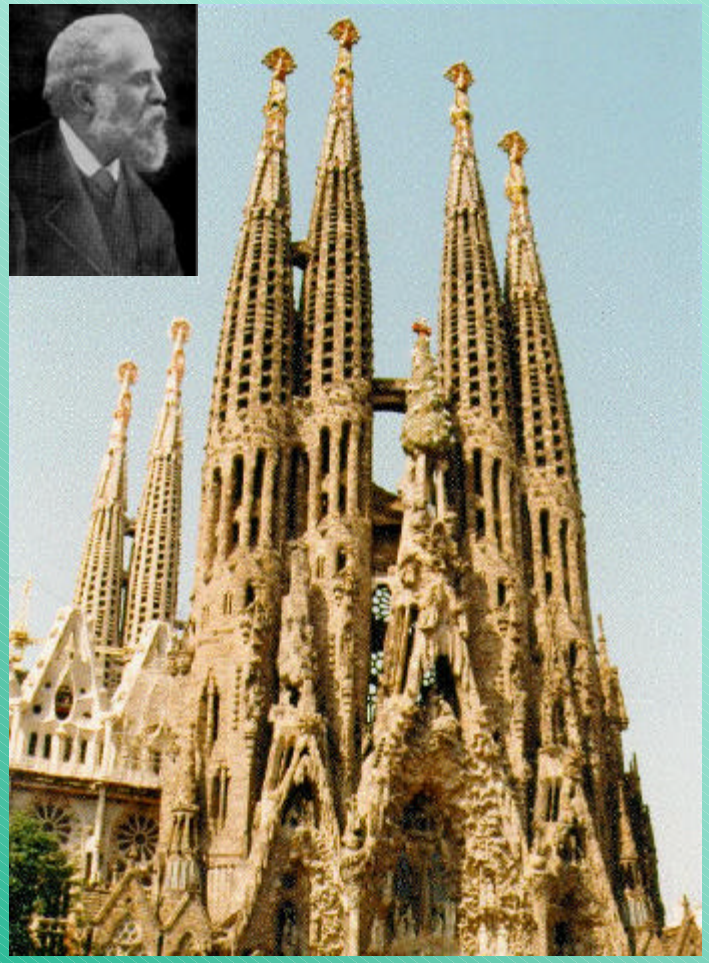
LHCb Software Week



GAUDI

Detector Description News

Radovan Chytracek

What Was Detector Description v2

- Prototype implementation
- Basic DTD covering the minimal set of tags needed to describe detector
- Verbose and complex user defined converters implementation
- Lack of CLHEP units support
- We got plenty of user feedback and motivating criticism

What Is Detector Description v3

- Improved in many ways
- User feed back reflected in the code
- Simplified user defined converters
- Numerical expressions parser
- Compatibility with CLHEP units
- DTD is much more rich and safe
  - Improved positions and rotations
  - Parametric physical volumes available
  - Units entities with the same names as in CLHEP

Numerical Expressions Parser

- Simple parser for evaluation of expressions
- Expressions can be composed of
  - integer and floating point numbers
    - 100 | 100. | .05 | 0.1| 1.34-e12|-23
  - operations: +, -, *, /, unary +|-, exponent ^
  - parenthesized expressions: 1.4 * (23.4-e12 / 1.8)
- Result is always evaluated to double value
- Operator precedence is:
  - [()] [unary +|-] [^] [*|/] [+|-]
- By default checks for units in expressions

CLHEP Units Compatibility

- DTD for XML detector description defines units a la CLHEP
- Units MUST be used where required
  - XML converters assume the use of units
  - In case the units are missing processing stops and an exception is thrown.
- Use expressions parser where needed with check for units enabled
- Examples
  - 23*&cm; | 12*&volt; | 23.6*&g;/&cm3;

MyDetector Example

```
#include "Gaudi/DetectorDataSvc/DetectorElement.h"

extern const CLID& CLID_MyDetector;

class MyDetector: public DetectorElement {
public:

 int cellSize( x,y );
 int setCellSize( double size );

 inline const CLID& clID()   { return MyDetector::classID(); }
 static const CLID& classID(){ return CLID_MyDetector; }

private:
  int m_cellSize;
};
```

MyDetector Structure In XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DDDB SYSTEM "xmldb.dtd" [
    <!--Number of stations in Vertex detector-->
    <!ELEMENT CellSize EMPTY>
    <!ATTLIST CellSize n CDATA #REQUIRED>
]>
<DDDB>
  <detelem classID="7001" name="MyDetectorModule">
    <author>RCH</author>
    <version>0.1</version>
    <geometryinfo>
      <lvname  name = "/dd/Geometry/MyDetector/lvMyDetector" />
      <support name = "/dd/Structure/LHCb"> <npath value = "MDM_0" /> </support>
    </geometryinfo>
    <detelemref classID="7002" href="mysubmodule.xml#MySubModule01"/>
    <detelemref classID="7002" href="mysubmodule.xml#MySubModule02"/>
    <detelemref classID="7003" href="myanothersubmodule.xml#MyAnotherSubModu
    <specific>
      <CellSize n="56.7*&cm;" />
    </specific>
  </detelem>
<DDDB>
```

User defined XML tag

Associated geometry data

Detector specific data

Units inside numerical expression

MyDetector XML Converter

```
#include "DetDesc/XmlCnvSvc/XmlUserDeCnv.h"
#include "MyDetector.h"
class XmlMyDetectorCnv : public XmlUserDeCnv<MyDetector> {
public:
  XmlMyDetectorCnv(ISvcLocator* svc);
  ~XmlMyDetectorCnv()  {}
  virtual void uStartElement( const char* const name, XmlCnvAttributeList& attributes);
};
```

```
static CnvFactory<XmlMyDetectorCnv> myde_factory;
const ICnvFactory& XmlMyDetectorCnvFactory = myde_factory;
XmlMyDetectorCnv::XmlMyDetectorCnv(ISvcLocator* svc)
: XmlUserDeCnv<MyDetector>( svc, "XmlMyDetectorCnv" ){}
void XmlMyDetectorCnv::uStartElement( const char* const name,
                                      XmlCnvAttributeList& attributes) {
  MsgStream log( msgSvc(), m_msId );
  std::string tagName( name );
  if( tagName == "CellSize" ) {
    std::string nval = attributes.getValue( "n" );
    m_dataObj->setCellSize( xmlSvc()->eval(nval) );
    }
  } else {
    // Unknown tag, a warning message can be issued here
  }
}
```

MyDetector Geometry In XML

```
<?xml version="1.0"?>
<!DOCTYPE DDDB SYSTEM "xmldb.dtd">
<DDDB>
  <catalog name="MyDetector">
    <logvolref href="#lvMyDetector" />
    <logvolref href="#lvMyDetectorSubModule" />
    <logvolref href="#lvMyDetectorAnotherSubModule" />
  </catalog>
  <logvol name="lvMyDetector" material="Vacuum">
    <box name="lvMyDeBox" sizeZ="800*&mm;" sizeY="10000*&mm;" sizeX="10000*&mm;" />
    <paramphysvol number="2">
      <physvol name="ppvMySM" logvol="/dd/Geometry/MyDetector/lvMyDetectorSubModule">
        <posXYZ  x="0*&mm;" y="0*&mm;" z="-300*&mm;" />
      </physvol>
      <posXYZ  x="0*&mm;" y="0*&mm;" z="100*&mm;" />
      <rotXYZ  rotX="0*&degree;" rotY="0*&degree;" rotZ="90*&degree;" />
    </paramphysvol>
    <physvol name="pvMyAnotherSM" logvol="/dd/Geometry/MyDetector/lvMyAnotherSubModule">
      <posXYZ x="0*&mm;" y="0*&mm;" z="200*&mm;" />
    </physvol>
  </logvol>
</DDDB>
```

Conclusions

- DetDesc v3 works on supported platforms
- Documentation is updated (being updated)

- New XML DB structure has been defined
  - provides better support for parallel development
  - gives unified hierarchical structure
  - uses the latest features
  - is made as a CMT package
  - is in CVS