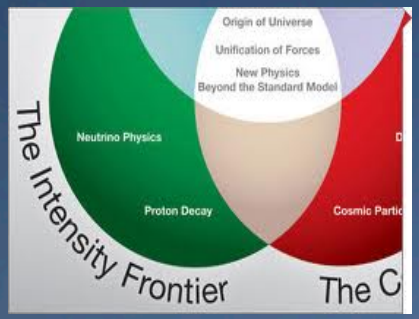
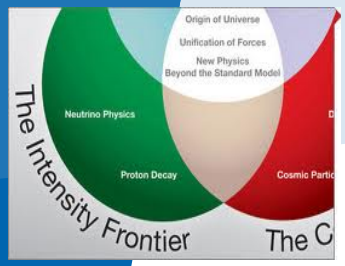


ART based package to monitor physics relevant to intensity frontier physics experiments

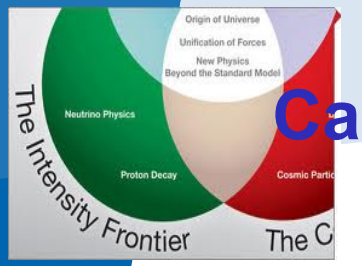
Hans Wenzel

Jul 18th 2013



Charge

- Extension of geant 4 validation effort but geared specifically towards the needs of the intensity frontier community.
- develop an ART based package for monitoring of all identified physics plots relevant to intensity frontier experiments at the model and physics list level.
- port elements of Julia's (stand alone) tests to ART and in addition integrate both EM and HAD plots into this package.
- explore the possibility to use the G4-ART interface.
- Develop tools to facilitate tests and customization of physics lists.
- The plots to monitor in this package will be associated with individual models (compared with thin target experiments aka first interaction) or physics lists, as well as with quantities to be validated with results from test beam and real experiments.
- For more complex validation, simplified geometry may be used or real configurations from the experiments could be imported to this ART application.



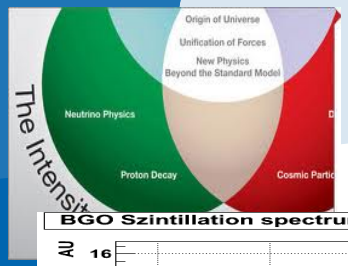
CaTS: Calorimeter and Tracker Simulatio

Calorimeter and Tracker Simulation

CaTS is a flexible and extend-able framework (based on geant4 and ROOT) for the general simulation of calorimeter and tracking detectors.

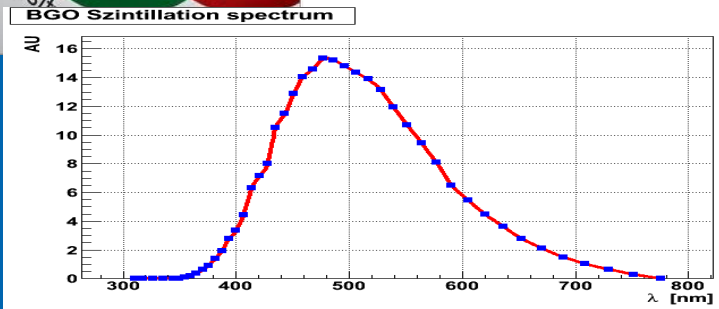
In the following look at CaTS to:

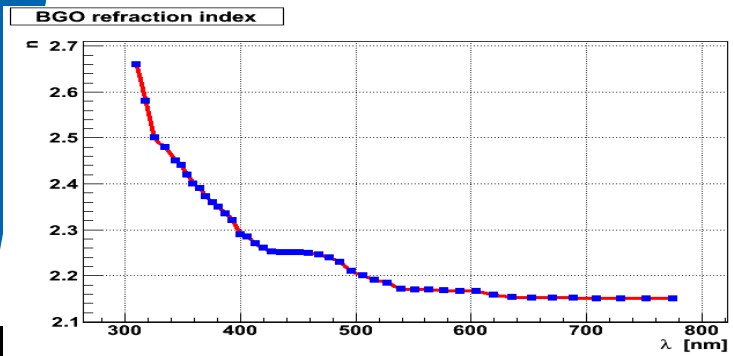
- **identify the features that we want,**
- **features that should be replaced by services of Art,**
- **extensions ??**

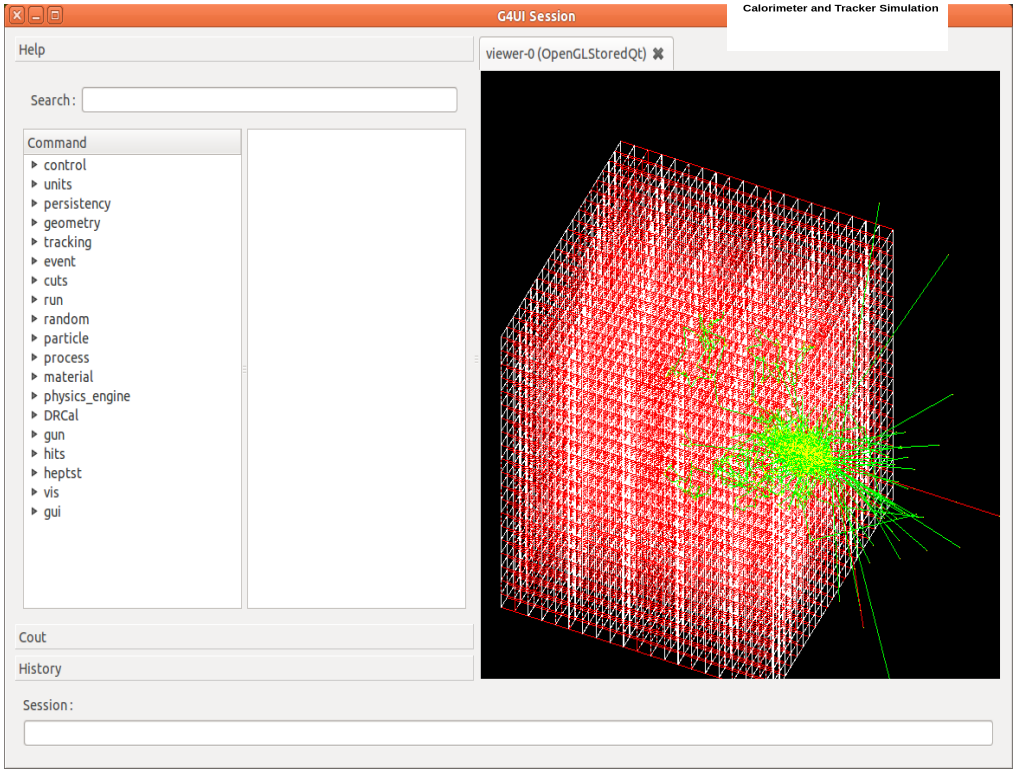


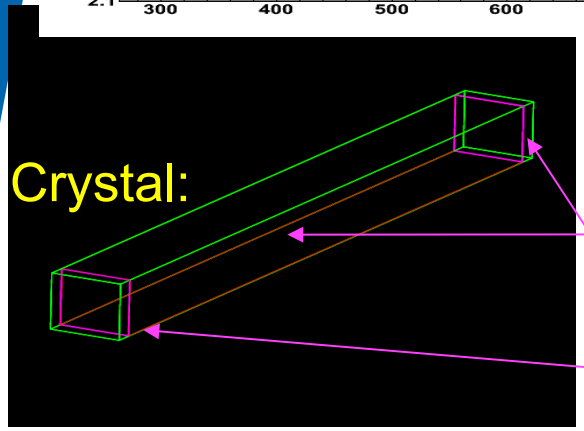
CaTS in Action: inputs and results

Calorimeter and Tracker Simulation



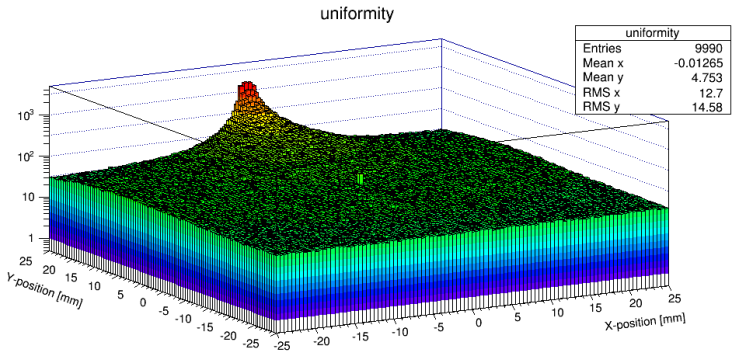


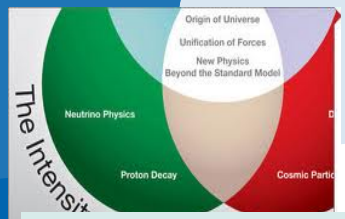




DRCalorimeterSD

PhotonSD

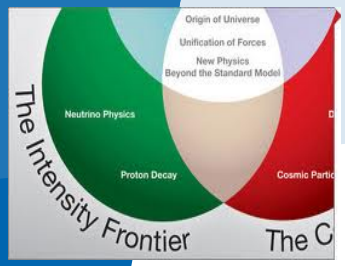




Elements of CaTS

Calorimeter and Tracker Simulation

Detector Description:	Xml based gdml input file (e.g. crystalcal.gdml) (Geometry, Materials, optical properties, sensitive detector), we provide working examples, no recompilation necessary (keep)
Persistency	uses Root reflexion (gccxml) to automatically, create dictionaries for all classes we want to write out (e.g. Hits) (Art)
Input modules:	GPS, Particle Gun, HEPMC (Pythia) (keep, extend to more file formats?)
Physics Lists:	choice of all Reference Physics Lists which can be extended to include optical physics processes (Cerenkov, Rayleigh, Scintillation etc.) (keep, allow it to be more modular)
Sensitive Detectors and Hits:	TrackerSD, CalorimeterSD, DRCalorimeterSD (also registers Cerenkov photons), DRTSCalorimeterSD (time slices), StoppingCalorimeterSD, PhotonSD: sensitive detector that registers optical photons. (keep)
User Actions:	examples of user actions (EventAction, RunAction, StackingAction,SteppingAction...) are provided (keep but need additional ones)
CVS Code repository, Redmine project & Instructions:	http://cdcvs.fnal.gov/cgi-bin/public-cvs/cvsweb-public.cgi/?hidenonreadable=1&f=h&logsort=date&sortby=file&hideattic=1&cvsroot=ilcdet http://home.fnal.gov/~wenzel/cvs.html#Optical
Histogram manager	Deals with analysis histograms created by various classes (e.g. sensitive detectors), Analysis enabled/disabled via compiler flag. (Art)



Remarks

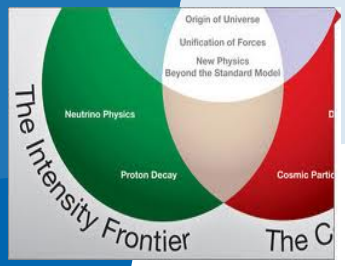
clear separation of detector description (gdml) and run time configuration

Clear separation of physical volumes and sensitive detector classes
Sensitive Detector:

- Knows how to create Hits and how to add hitlist to the event.
- Is attached to a physical volume (as specified in gdml file)

User Actions: configurable at runtime.

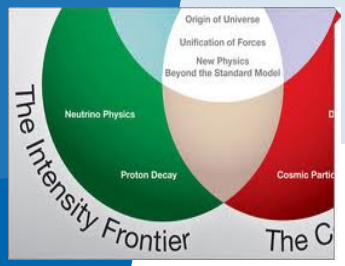
But: no scheme for segmentation (needed?)



gdml file (header+ definitions)

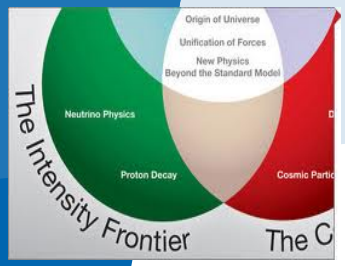
```
<?xml version="1.0" encoding="UTF-8" ?>
<gdml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"http://service-spi.web.cern.ch/service-spi/app/releases/GDML/schema/gdml.xsd">

  <define>
    <variable name="i" value="0"/>
    <variable name="j" value="0"/>
    <variable name="k" value="0"/>
    <constant name="numlay" value="499"/>
    <constant name="numcol" value="8"/>
    <constant name="numrow" value="8"/>
    <constant name="scalex" value="300"/>
    <constant name="scaley" value="300"/>
    <constant name="scalez" value="5"/>
    <constant name="absoffsetz" value="-0.5"/>
    <constant name="szoffsetz" value="2.0"/>
  </define>
```



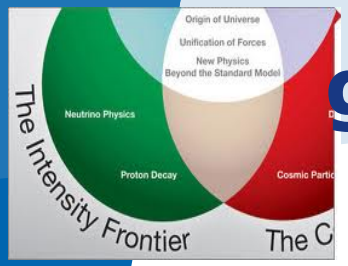
gdml file (materials)

```
<materials>
    <element name="Oxygen"   formula="O"  Z="8."> <atom value="16.0"/>  </element>
    <element name="Nitrogen" formula="N"  Z="7."> <atom value="14.01"/> </element>
    <element name="Lead"     formula="Pb" Z="82."> <atom value="207.20"/> </element>
    <element name="Carbon"   formula="C"  Z="6."> <atom value="12.01" unit="g/mole"/>
    </element>
    <element name="Hydrogen" formula="H"  Z="1."> <atom value="1.01" unit="g/mole"/>
    </element>
    <material name="Air">
      <D value="1.290" unit="mg/cm3"/>
      <fraction n="0.7" ref="Nitrogen"/>
      <fraction n="0.3" ref="Oxygen"/>
    </material>
    <material name="Scintillator">
      <D value="1.032" unit="g/cm3"/>
      <composite n="9" ref="Carbon"/>
      <composite n="10" ref="Hydrogen"/>
    </material>
    <material name="metalPb">
      <D value="11.340" unit="g/cm3"/>
      <composite n="1" ref="Lead"/>
    </material>
</materials>
```



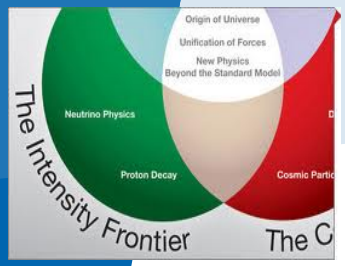
gdml file (solids)

```
<solids>
    <box name="WorldBox" lunit="mm" x="5000" y="5000" z="5000"/>
    <box name="CalorimeterCell"  lunit="mm" x="300" y="300" z="4"/>
    <box name="ScintillatorCell"  lunit="mm" x="300" y="300" z="1"/>
 </solids>
```



gdml file (structure,placing)

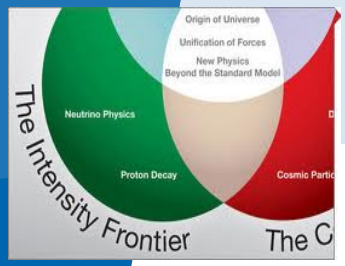
```
<structure>
   <volume name="CaloVol">
     <materialref ref="metalPb"/>
     <solidref ref="CalorimeterCell"/>
       <auxiliary auxtype="SensDet" auxvalue="DRTSCalorimeter"/>
       <auxiliary auxtype="Color" auxvalue="Red"/>
   </volume>
   <volume name="ScintVol">
     <materialref ref="Scintillator"/>
     <solidref ref="ScintillatorCell"/>
       <auxiliary auxtype="SensDet" auxvalue="DRTSCalorimeter"/>
       <auxiliary auxtype="Color" auxvalue="Blue"/>
   </volume>
   <volume name="TOP">
     <materialref ref="AIR"/>
     <solidref ref="WorldBox"/>
     <loop for="i" from="0" to="numrow" step="1">
     <loop for="j" from="0" to="numcol" step="1">
     <loop for="k" from="0" to="numlay" step="1">
       <physvol>
           <volumeref ref="CaloVol"/>
         <position name="posijk" x="scalex*(i-numrow/2)" y="scaley*(j-numcol/2)" z="absoffsetz+scalez*(k-numlay/2)"/>
       </physvol>
         <physvol>
           <volumeref ref="ScintVol"/>
         <position name="posijk2" x="scalex*(i-numrow/2)" y="scaley*(j-numcol/2)" z="szoffsetz+scalez*(k-numlay/2)"/>
       </physvol>
    </loop>
       </loop>
     </loop>
   </volume>
 </structure>
```



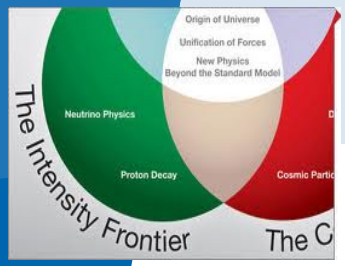
gdml file (setup)

```
<setup version="1.0" name="Default">
    <world ref="TOP"/>
  </setup>

</gdml>
```

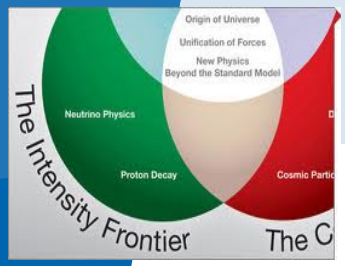

Origin of Universe
Unification of Forces
New Physics
Beyond the Standard Model
Neutrino Physics
Proton Decay
The Intensity Frontier

backup



Technical

- Framework: Art → worked with it in the past
- Code Repository: Redmine and Git since this is used by ART and artg4 → requested to create the project (CaTS).
  artg4:
  https://cdcvs.fnal.gov/redmine/projects/artg4
- Detector description: options gdml e.g. used by nova/CaTS extension of Geant4., fhicl: used by artg4, custom: used by mu2e
- release management: relocatable ups???
- build system: cmake (used by Geant 4, CaTS….)
- environment setup: custom shell script
- development machine: (something with art and artg4 installed) gm2gpvm → got an account still waiting for instructions to set up the environment.
- Execution: for now use Geant 4 VO and (limited) grid resources to execute jobs



Technical (cont.)

- Display of results: use geant 4 web application and database hosted here at fermilab (just create a new category)
- Configuration of physics lists/ processes: → discuss with Robert, look how it's done in G4