

Belle II

Framework and PXD DAQ meeting

February 21 to 22

Max-Planck-Institut für Physik



Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)



MAX-PLANCK-GESELLSCHAFT

Belle II Framework and PXD DAQ meeting

2011/02/21

Andreas Moll

Max-Planck-Institut für Physik

Geometry workpackage

- UserInfo and optical surfaces
- On-demand loading
- Overlap handling
- Isotopes and Atoms

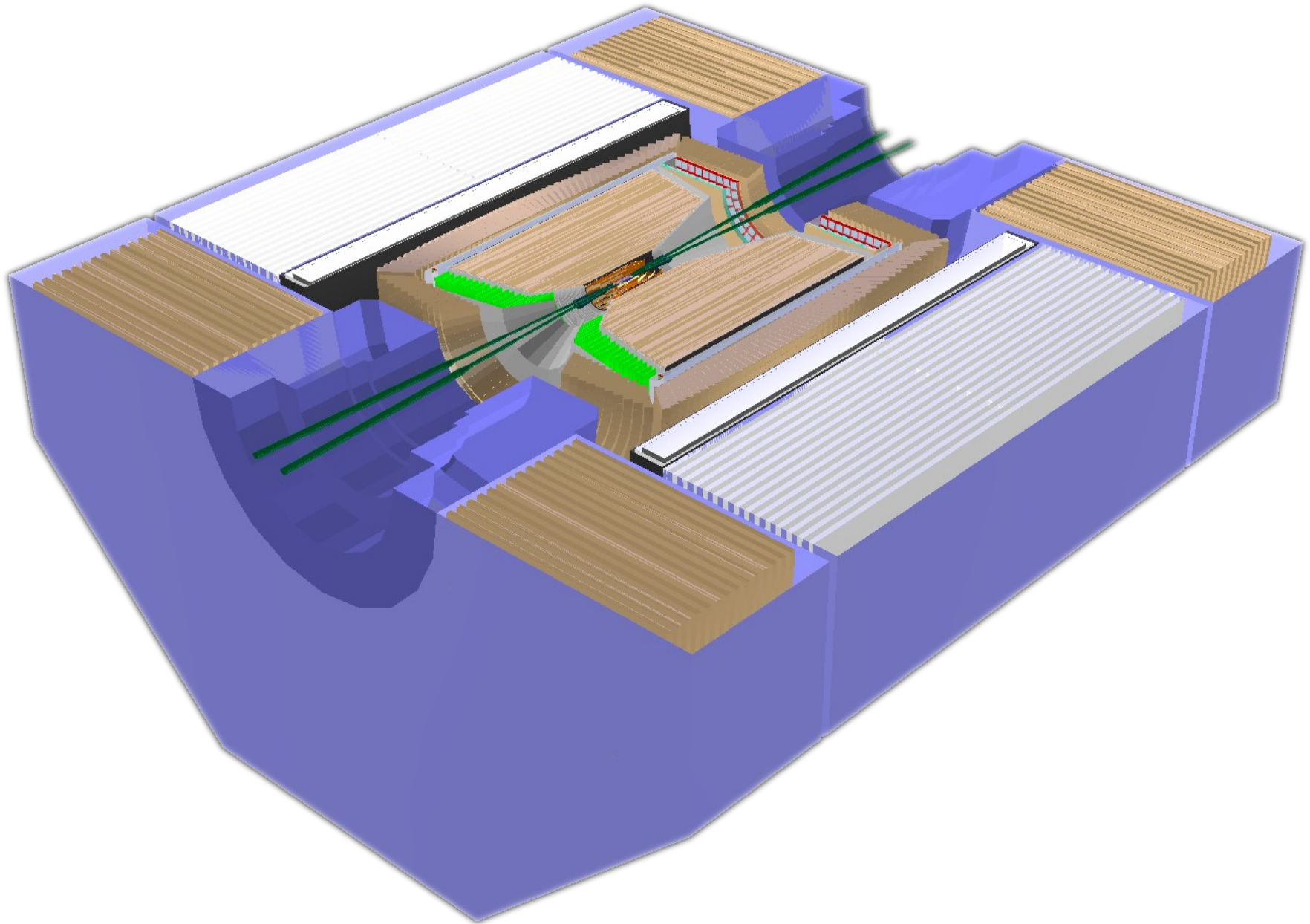


Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)



MAX-PLANCK-GESellschaft





Overlap check (1mm tolerance):

CDC backward cover overlaps ECL crystals:	144
CDC left tube overlaps CDC backward endplate:	3
CDC right tube overlaps CDC forward endplate:	1
CDC left mid tube overlaps CDC backward endplate:	3
CDC forward endplate overlaps middle:	5
CDC backward endplate overlaps middle:	1



How to handle overlaps between subdetectors ?

Introduce Geant4 based overlap checks ?

Add additional information to volumes

Save this information with the volumes to a ROOT file

➔ Layer ID, Ladder ID, Sensor ID, optical surfaces

TVolume

```

virtual void TAttFill::SetFillAttributes () MENU
virtual void TAttFill::SetFillColor (Color_t fcolor)
virtual void TAttFill::SetFillStyle (Style_t fstyle)
virtual void TAttLine::SetLineAttributes () MENU
virtual void TAttLine::SetLineColor (Color_t lcolor)
virtual void TAttLine::SetLineStyle (Style_t lstyle)
virtual void TAttLine::SetLineWidth (Width_t lwidth)
virtual void TDataSet::SetMother (TDataSet* parent = 0)
virtual void TNamed::SetName (const char* name) MENU
virtual void TNamed::SetNameTitle (const char* name, const char* title)
virtual void TObjectSet::SetObject (TObject* obj) ←
virtual TObject* TObjectSet::SetObject (TObject* obj, Bool_t makeOwner)
static void TObject::SetObjectStat (Bool_t stat)
virtual void TDataSet::SetParent (TDataSet* parent = 0)
virtual void TNamed::SetTitle (const char* title = "") MENU
virtual void TObject::SetUniqueID (UInt_t uid)
virtual void SetVisibility (TVolume::ENodeSEEN vis = TVolume::kBothVisible) MENU
virtual void TDataSet::SetWrite ()
    
```

Embedded TObject

➔ **Not available in TGeoVolume**

Solution so far (GeoDetector.h)

```

00042
00043     typedef boost::unordered_map<TGeoVolume*, VolumeUserInfoBase*> GeoVolumeUserInfoMap;
00044
00106         std::list<std::string> m_supportedSections;
00107         std::list<std::string> m_calledCreators;
00108         GeoVolumeUserInfoMap m_geoVolumeUserInfo;

```

```

UserInfoClass & getVolumeUserInfo ( TGeoVolume * geoVolume ) throw ( GeoVolumeIsNULL, UserInfoDynamicCastError)

```

Returns a reference to the volume user information, which is attached to the specified `TGeoVolume`.

If the specified `TGeoVolume` has no user information attached, a new one is created.

Parameters:

geoVolume Pointer to the `TGeoVolume` whose user information should be returned.

Returns:

Reference to the volume user information.

template method

Definition at line 140 of file `GeoDetector.h`.

```

{
    if (geoVolume == NULL) throw GeoVolumeIsNULL();
    UserInfoClass* userInfo = NULL;

    //Check if there has already been created a user info for the TGeoVolume. If yes, return its user info.
    boost::unordered_map<TGeoVolume*, VolumeUserInfoBase*>::iterator mapIter = m_geoVolumeUserInfo.find(geoVolume);
    if (mapIter == m_geoVolumeUserInfo.end()) {
        userInfo = new UserInfoClass();
        m_geoVolumeUserInfo.insert(make_pair(geoVolume, userInfo));
    } else {
        userInfo = dynamic_cast<UserInfoClass*>(mapIter->second);
        if (userInfo == NULL) throw(UserInfoDynamicCastError() << typeid(UserInfoClass).name());
    }
    return *userInfo;
}

```

Use TGeoVolume UserInfo system

Transform optical surface UserInfo automatically for Geant4

Creator:

```
OpticalSurface* optSurf = gearDir.readOpticalSurface([XPath]);
```

XML structure ?

➔ **ArichUserInfo** inherits from **OpticalSurfaceUserInfo** ← ROOT dictionary

```
ArichUserInfo& arichInfo = <ArichUserInfo>getVolumeUserInfo(arichVolume);
arichInfo.setOpticalSurface(optSurf);
```

Simulation (automatically):

OpticalSurfaceUserInfo ➔ G4OpticalSurface (+ MaterialPropertiesTable)

G4OpticalSurface + G4LogicalVolume = G4LogicalSkinSurface

G4OpticalSurface + 2 G4VPhysicalVolume = G4LogicalBorderSurface

Example for the current implementation:

```

12 <Mixture name="PXD_Glue">
13   <Density unit="g/cm3">1.13</Density>
14   <Elements>
15     <Element name="C" weight="0.43902439">
16       <AtomNumber>6</AtomNumber>
17       <MassNumber>12.0107</MassNumber>
18     </Element>
19     <Element name="O" weight="0.073170732">
20       <AtomNumber>8</AtomNumber>
21       <MassNumber>15.9994</MassNumber>
22     </Element>
23     <Element name="H" weight="0.487804878">
24       <AtomNumber>1</AtomNumber>
25       <MassNumber>1.00794</MassNumber>
26     </Element>
27   </Elements>
28 </Mixture>

```

Definition of atoms not necessary
(e.g. Geant4 provides them)

What about isotope composition ?

➔ Common definition for Atoms

Suggestion:

```
<Element name="H" weight="" />
```

➔ Get element from
Geant4 NIST manager for sim.

+

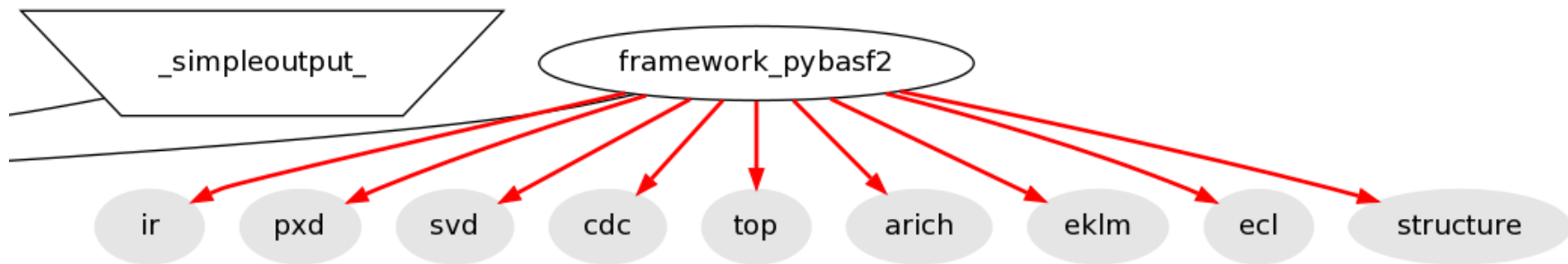
The best accuracy for the most relevant parameters guaranteed:
*Density, Mean excitation potential, Chemical bounds, Element composition
Isotope composition, Various corrections, ...*

+

Consistent radiation length handling (TGeo and Geant4 have both their own formula)

▬

How to feed back the Geant4 elements to TGeo (Avoid TGeo material system ?)



Linking the subdetectors with the framework registers the Creators

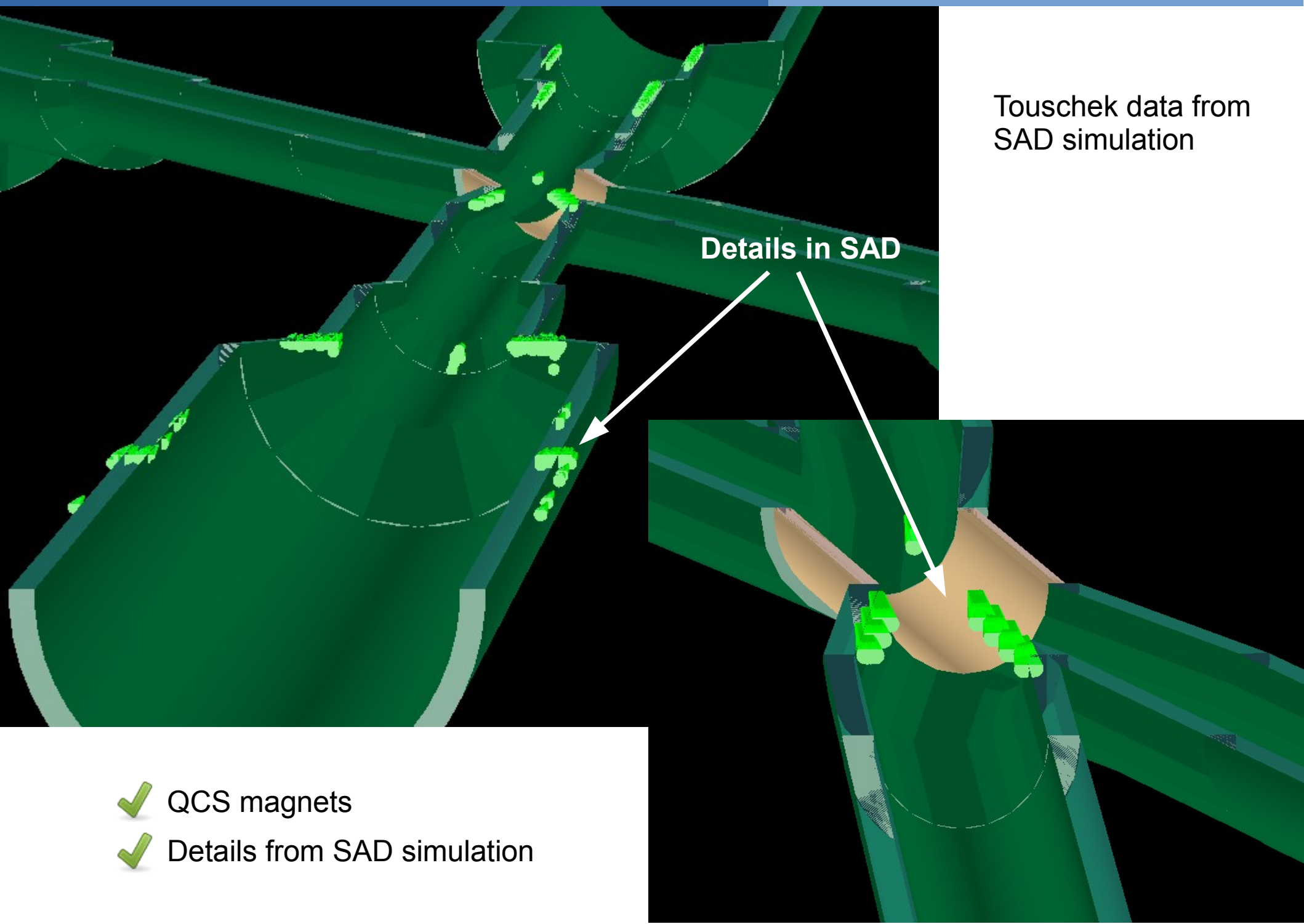
```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ParamSet type="PXD">
3   <Name>PXD BelleII</Name>
4   <Description>The famous RedBull can</Description>
5   <Version>0</Version>
6   <Creator>PXD BelleII</Creator>
  
```

XML file calls creator

On demand loading:

- ✓ No linking of subdetector libraries
- ✓ Specify subdetectors in the steering file (don't use BelleII.xml)
- ❓ Mechanism similar to the module on-demand loading



Touschek data from
SAD simulation

Details in SAD

- ✓ QCS magnets
- ✓ Details from SAD simulation