



A Quick Marlin Field Implementation

Jason Abernathy
University of Victoria



Basic Features

- Query the magnetic / electric field at a point in the detector
- A global field is the sum of smaller fields
- Data driven
 - **Field classes are registered with the global field manager (much like Marlin)**
 - **Required fields are defined in Marlin steering file**
 - **Field data (bins or just parameters) are stored in LCGenericObjects**



GlobalFieldProcessor

- Singleton class (one instance)
- Interface:
 - HepVector get_global_magnetic_field(const HepVector& x)
 - HepVector get_global_electric_field(const HepVector& x)
 - double get_main_crossing_angle()
 - bool register_field(Field*);
- Steering:

```
<processor name="MyConditionsProcessor" type="ConditionsProcessor">  
  . <parameter name="SimpleFileHandler" type="StringVec">LCDv2_z_Bz LDCv2_14mradX.slcio LCDv2_z_Bz|  
  DID14_01_z_Bx LDCv2_14mradX.slcio DID14_01_z_Bx</parameter>  
</processor>  
  
<processor name="MyGlobalFieldProcessor" type="GlobalFieldProcessor">  
  . <parameter name="MagneticFields">DIDZMapMagneticField DID14_01_z_Bx SolenoidZMapMagneticField  
  DID14_01_z_Bz</parameter>  
</processor>
```

- Basic field class
- Interface:
 - `virtual Field *new_field(void) = 0;`
 - `virtual bool construct(EVENT::LCCollection *) = 0;`
 - Called whenever the conditions collection is changed (via ConditionsProcessor)
 - `virtual HepVector get_field_value(const HepVector &) = 0;`
 - `std::string get_typename();`
- Base constructor registers typename with GlobalFieldProcessor
- Doesn't do any rotating / translating yet...



Example Implementation

- SolenoidZMapMagneticField
 - Taken from Mokka fieldX01 (authors Paulo Mora de Freitas, Adrian Vogel)
- Data: Mokka database -> .txt file -> LCIO

```
jabrnth@prophet:~$ mysql -hpollin1.in2p3.fr -uconsult -p -e "SELECT * from fieldmaps00.LDCv2_z_Bz;"
```

- Constructing:

```
// grab the bounds from the collection
_r_min = collection->getParameters().getFloatVal("r_min");
_r_max = collection->getParameters().getFloatVal("r_max");
_dz = collection->getParameters().getFloatVal("dz");

for( int map_counter = 0; map_counter < collection->getNumberOfElements(); map_counter++ ) {
    LCGenericObject *field_map_object = dynamic_cast<LCGenericObject*>(
    collection->getElementAt(map_counter) );
    assert(field_map_object);

    FieldMapEntry map_entry;

    map_entry._z = field_map_object->getDoubleVal(0);
    map_entry._Bz = field_map_object->getDoubleVal(1);

    _field_map.push_back(map_entry);
}
```