Design of the non-uniform magnetic Field in CEPCSW

Tao Lin

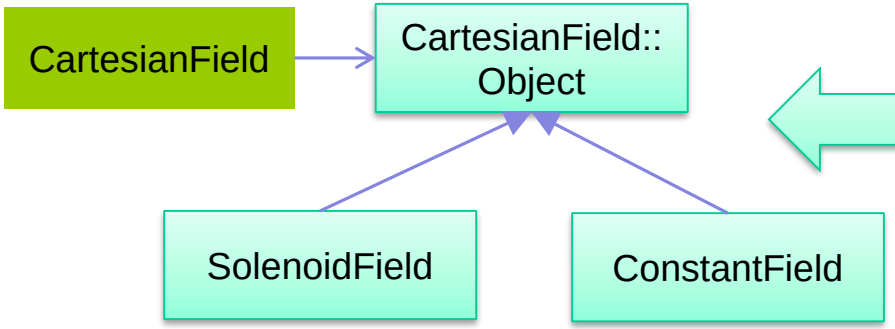
May 2021

B-field in DD4hep

- DD4hep provides a solution on the management of B-fields.

```
<fields>
  <field name="InnerSolenoid" type="solenoid"
         inner_field="Field_nominal_value"
         outer_field="0"
         zmax="SolenoidCoil_half_length"
         inner_radius="SolenoidCoil_center_radius"
         outer_radius="Solenoid_outer_radius">
  </field>
  <field name="OuterSolenoid" type="solenoid"
         inner_field="0"
         outer_field="Field_outer_nominal_value"
         zmax="SolenoidCoil_half_length"
         inner_radius="Solenoid_outer_radius"
         outer_radius="Yoke_barrel_inner_radius">
  </field>
</fields>
```

Define the field type and properties in XML file



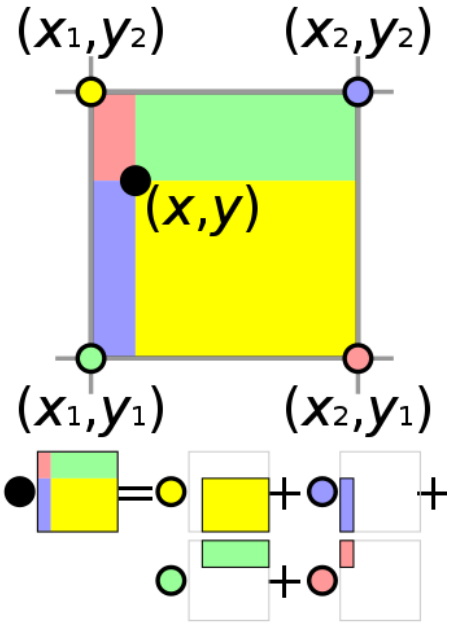

```
static Ref_t create_SolenoidField(Detector& description, xml_h e) {
  xml_comp_t c(e);
  bool has_inner_radius = c.hasAttr(_U(inner_radius));
  bool has_outer_radius = c.hasAttr(_U(outer_radius));

  if (!has_inner_radius && !has_outer_radius) {
    throw_print("Compact2Objects[ERROR]: For a solenoidal field at least one of the "
                " xml attributes inner_radius of outer_radius MUST be set.");
  }
  CartesianField obj;
  SolenoidField* ptr = new SolenoidField();
  //
  // This logic is a bit weird, but has it's origin in the compact syntax:
  // If no "inner_radius" is given, the "outer_radius" IS the "inner_radius"
  // and the "outer_radius" is given by one side of the world volume's box
  //
  if (has_inner_radius && has_outer_radius) {
    ptr->innerRadius = c.attr<double>(_U(inner_radius));
    ptr->outerRadius = c.attr<double>(_U(outer_radius));
  }
  else if (has_inner_radius) {
    Box box = description.worldVolume().solid();
    ptr->innerRadius = c.attr<double>(_U(inner_radius));
    ptr->outerRadius = box.x();
  }
  else if (has_outer_radius) {
    Box box = description.worldVolume().solid();
    ptr->innerRadius = c.attr<double>(_U(outer_radius));
    ptr->outerRadius = box.x();
  }
  if (c.hasAttr(_U(inner_field)))
    ptr->innerField = c.attr<double>(_U(inner_field));
  if (c.hasAttr(_U(outer_field)))
    ptr->outerField = c.attr<double>(_U(outer_field));
  if (c.hasAttr(_U(zmax)))
    ptr->maxZ = c.attr<double>(_U(zmax));
  else
    ptr->maxZ = description.constant<double>("world_side");
  if (c.hasAttr(_U(zmin)))
    ptr->minZ = c.attr<double>(_U(zmin));
  else
    ptr->minZ = -ptr->maxZ;
  obj.assign(ptr, c.nameStr(), c.typeStr());
  return obj;
}
DECLARE_XMLELEMENT(SolenoidMagnet,create_SolenoidField)
// This is the plugin required for slic: note the different name
DECLARE_XMLELEMENT(solenoid,create_SolenoidField)
```

Parse the XML file and create Field Object

Handling non-uniform B-Field in lcgeo

- In the lcgeo project, there is an example to read the non-uniform B-Field from ROOT file.
  - https://github.com/iLCSoft/lcgeo/blob/master/detector/other/FieldMapBrBz.cpp
  - Use bilinear interpolation to calculate the field at the given position
- The examples of field maps can be found in
  - https://github.com/iLCSoft/lcgeo/tree/master/fieldmaps

```
root [2] ntuple->Print()
******************************************************************************
*Tree    :ntuple    : data from ascii file                                      *
*Entries :     5751 : Total =           94620 bytes  File  Size =      44239 *
*       :          : Tree compression factor =   2.12                         *
******************************************************************************
*Br    0 :rho_mm    : rho_mm/f                                                  *
*Entries :     5751 : Total  Size=      23568 bytes  File Size  =        572 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=  40.35     *
*.............................................................................*
*Br    1 :z_mm      : z_mm/f                                                    *
*Entries :     5751 : Total  Size=      23558 bytes  File Size  =        524 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=  44.04     *
*.............................................................................*
*Br    2 :Brho      : Brho/f                                                    *
*Entries :     5751 : Total  Size=      23558 bytes  File Size  =      21391 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.08     *
*.............................................................................*
*Br    3 :Bz        : Bz/f                                                      *
*Entries :     5751 : Total  Size=      23548 bytes  File Size  =      21133 *
*Baskets :        1 : Basket Size=      32000 bytes  Compression=   1.09     *
*.............................................................................*
```



bilinear interpolation

Software design in CEPCSW

- Requirements
  - Easy to extend and support different sources, including files and database.
  - Consistent with the design of DD4hep.