Detector - Beam Delivery Interfacing

John Carter[1], Adrian Vogel[2]

1. Royal Holloway University of London

2. DESY, FLC Group

- Motivation
- Mokka
- BDSIM
- Geometry Model of the IR
  - A first step towards unifying our descriptions
- Conclusions & Outlook

22nd June 2005

ILC - BDIR

Motivation

General Motivation

- Many simulation tools exist - to model different aspects of the ILC
  - Linac tracking
  - Beam Delivery System tools
  - Detector modelling
  - Others (e.g. beam-beam interactions)
- These tools generally overlap in some form with at least one other tool…
- The passing and sharing of information between these tools needs to converge
  - To make sure we're working from the same parameter set, layout, etc.
  - To save time

Personal Motivation

- Mokka & BDSIM are both based on Geant4 and so both use similar geometry descriptions - so it makes sense to work towards a common input.

Mokka

- Mokka is a framework for full detector simulation
  - Based on Geant 4
  - Made for ILC studies - but can be used for other setups as well (prototypes, test beam installations, etc.)
- Mokka provides
  - A user-level interface to the Geant 4 toolkit
  - A common approach to define detector components ("Drivers")
  - A full persistency (= data storage) system (LCIO - Linear Collider Input/Output)
- Mokka geometry drivers
  - Construct single detector components
  - Read the actual geometry data from MySQL databases
  - Are plugged together to form "geometry models"
  - Drivers belonging to the chosen geometry model are invoked and read the geometry data at runtime

BDSIM

- BDSIM is a tool for simulating the Beam Delivery System
  - Also based on Geant 4
  - Designed to not only provide full optics tracking, but also secondary production
- BDSIM provides
  - Optimised and fast tracking solutions for particles in the BDS - similar to MAD
  - "Hands over" to Geant 4 to deal with showers and particle interactions
  - Takes MAD formatted optic lattices as input for general beamline components
- Now requires another form of component and geometry description to model the interaction region
  - Mokka's use of a MySQL database format seems a sensible start
    - As both tools have Geant 4 based requirements
- So a MySQL wrapper has been written to enable BDSIM to read Mokka-style input
  - A first step towards a common format!!

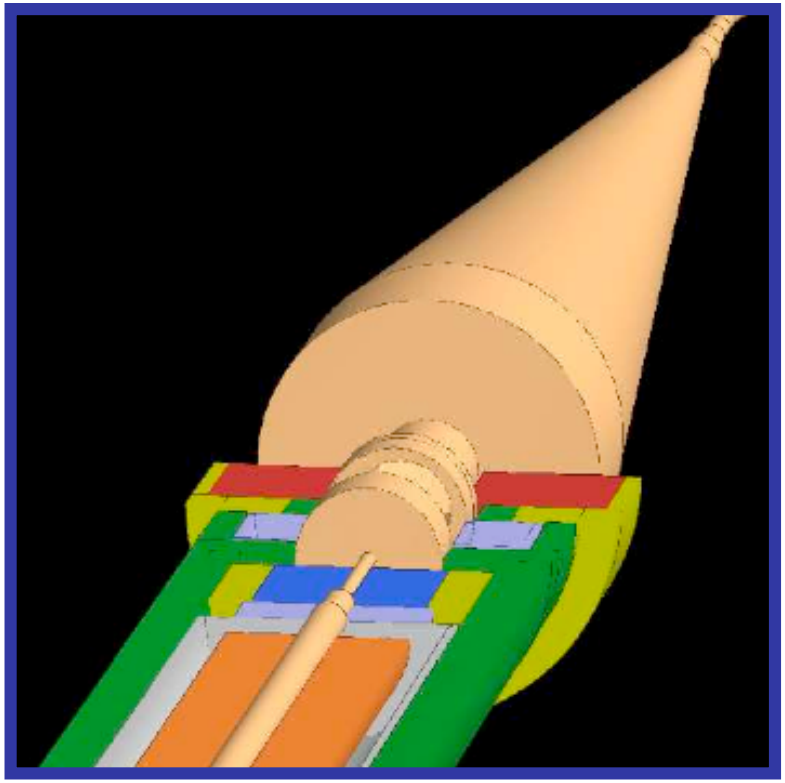
MySQL

- Typical Example of MySQL dumped output
  - This could be parsed in 'offline mode'
  - Accessed directly by Client/Server MySQL server structure
- Creating a complex structure by defining multiple solids
  - Specifying dimensions, positioning, materials - and can be extended to any properties needed, such as field strengths for magnets

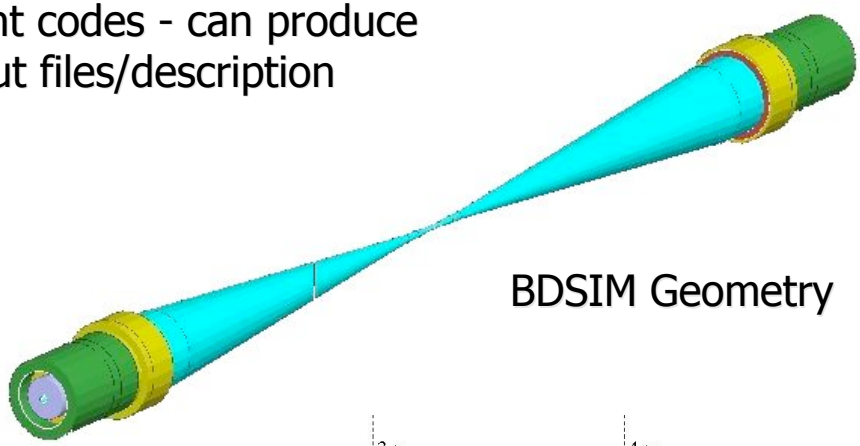
```
CREATE TABLE tungsten (
    zStart        DOUBLE(10,3),
    zEnd          DOUBLE(10,3),
    rInnerStart   DOUBLE(10,3),
    rInnerEnd     DOUBLE(10,3),
    rOuterStart   DOUBLE(10,3),
    rOuterEnd     DOUBLE(10,3),
    material      VARCHAR(32),
    name          VARCHAR(32)
);
# between LumCal and pump
INSERT INTO tungsten VALUES (3250.0, 3340.0, 110.0, 110.0, 160.0, 160.0, "Tungsten", "");
# shield around pump
INSERT INTO tungsten VALUES (3340.0, 3510.0, 210.0, 210.0, 250.0, 250.0, "Tungsten", "");
# shield behind pump
INSERT INTO tungsten VALUES (3510.0, 3580.0, 100.0, 100.0, 250.0, 250.0, "Tungsten", "");
# main support tube
INSERT INTO tungsten VALUES (3580.0, 3960, 180.0, 180.0, 250.0, 250.0, "Tungsten", "");
```

Geometry Model of the IR

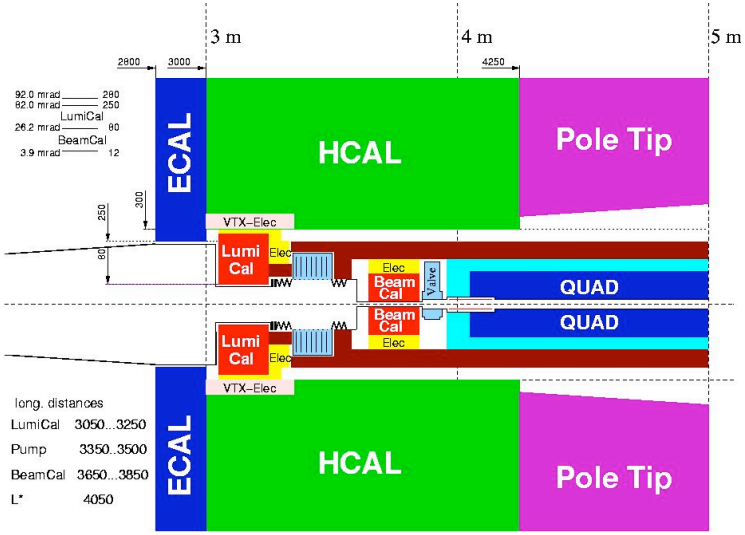
- Both BDSIM and Mokka - two independent codes - can produce the same geometries using the same input files/description



Mokka Geometry



BDSIM Geometry



Stahl Design used to produce the MySQL Geometry

Conclusions & Outlook

- There has been a tentative start in the direction of sharing geometry descriptions
  - So modifications and improvements to the geometries can be propagated quickly
  - Migration of BDSIM to MySQL input was a relatively straight forward and painless process
    - BDSIM still uses MAD-style inputs, but now "knows" how to handle MySQL
- It is often commented that standard and common output and input files need to be agreed upon…
  - We're not proposing that MySQL should be this format - but it works for us and is a step in the right direction!
- As inspiration we could look to the standards used in the detector community
  - LCIO - which is supported by all major detector simulation and analysis tools