Mass production with ILCDIRAC
GEAR Extension: 3D Volume Tree

S. Poss[1] A. Münnich[1]

[1] CERN, Switzerland

19/10/2010 / IWLC, Geneva

Introduction

Aim

- Give an overview of the CERN LCD group activities in software development

Content

- ILCDIRAC: GRID solution for ILC and CLIC
- GEAR extension

Other topics not discussed here, but in other sessions

Part I

Mass production with ILCDIRAC

S. Poss

ILCDIRAC

ILCDIRAC is an implementation of DIRAC

- DIRAC is the GRID solution for the LHCb community, used for mass production of MC simulations events and real data reconstruction and analysis
- Designed to simplify user access to the GRID and overcome GRID middleware deficiencies
- Based on pilot job principles

Mass production with ILCDIRAC (1)

Aim:

- Provide production system for the LC physics studies

Constraints:

- Minimal user effort to:
  1. generate;
  2. simulate;
  3. reconstruct

  the data needed
- Register the data produced by each step and keep track of relation between files
- Must work for both detector concepts (ILD and SiD)

Mass production with ILCDIRAC (2)

Idea:

- Automatically submit jobs based on request

Request:

- Physics channel (e.g. ee_h_mumu)
- Number of events / luminosity
- Input files (e.g. SLCIO files' list)

Implementation

Principles:

- Divide in independent steps that can be run one after the other

Steps:

- Whizard: generate the events according to desired channel and number of events. Produces STDHEP files
- Mokka / SLIC: simulate the detector behavior using given detector description. Produces single SLCIO file
- Marlin / LCSIM: reconstruct the events. Produces several SLCIO files
- Uploading of data: stores the data in at least 2 places (CERN and CC-IN2P3), logs are at CERN
- Register in FileCatalog: set metadata flags for each file

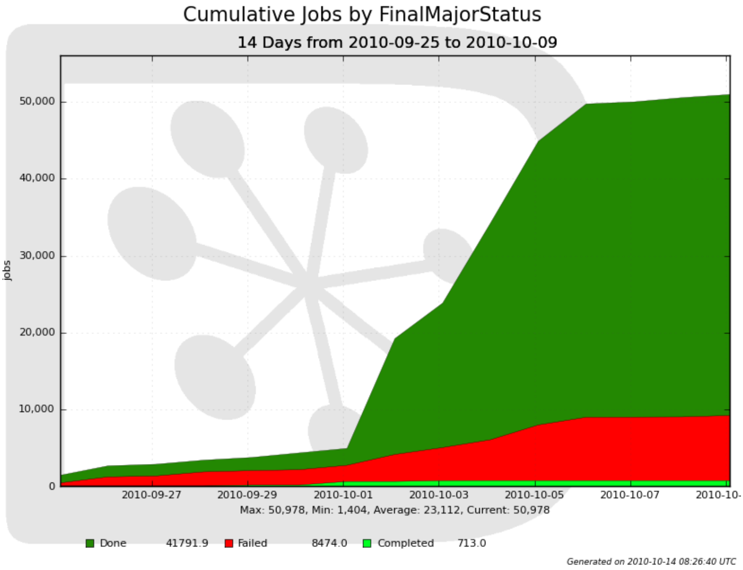
Results for ILCDIRAC

Full production chain available and working!

Individual steps available to users jobs, and already massively used, even by users outside CERN (e.g. LAL)

Please find more info at
https://twiki.cern.ch/twiki/bin/view/CLIC/DiracUsage

Results for ILCDIRAC

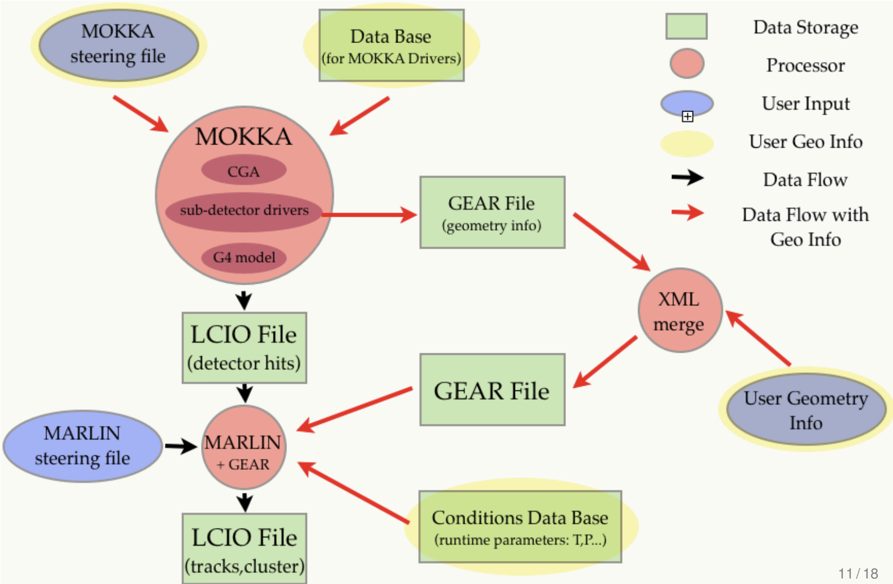


Part II

GEAR Extension

A. Münnich

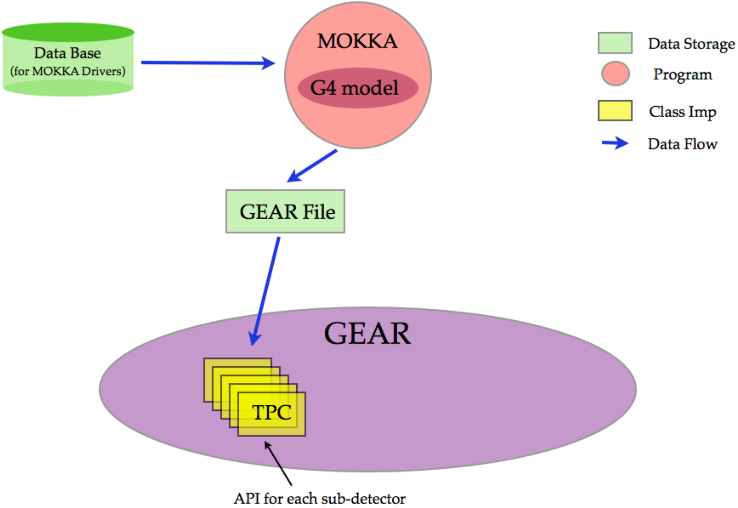
Current status of geometry description



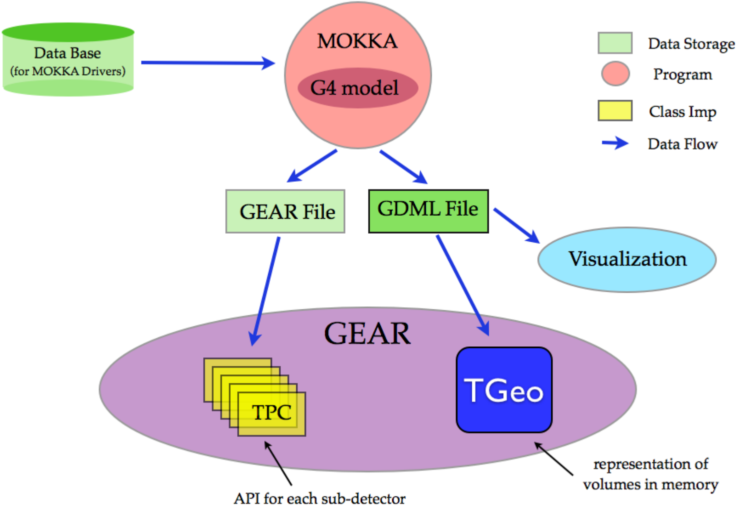
Motivation for new Geometry System

- No central place for geometry information:
    - Geometry can be changed in various places (inconsistencies)
    - Changing geometry requires changes in C++ code, XML, DB. . .
- No representation of geometrical volumes for complex questions, e.g. radiation length, misalignment, no information for coordinate transformation (global $\leftrightarrow$ local) etc.
- No easy visualization at later stage (after MOKKA)
- GEAR interface needs extension, too many user-parameters written from MOKKA
- Software maintenance aspects: e.g. MOKKA code contaminated with GEAR

Geometry Information Flow



Geometry Information Flow



Data Base
(for MOKKA Drivers)

MOKKA

G4 model

GEAR File

GDML File

Visualization

GEAR

TPC

TGeo

API for each sub-detector

representation of volumes in memory

🟩 Data Storage

● Program

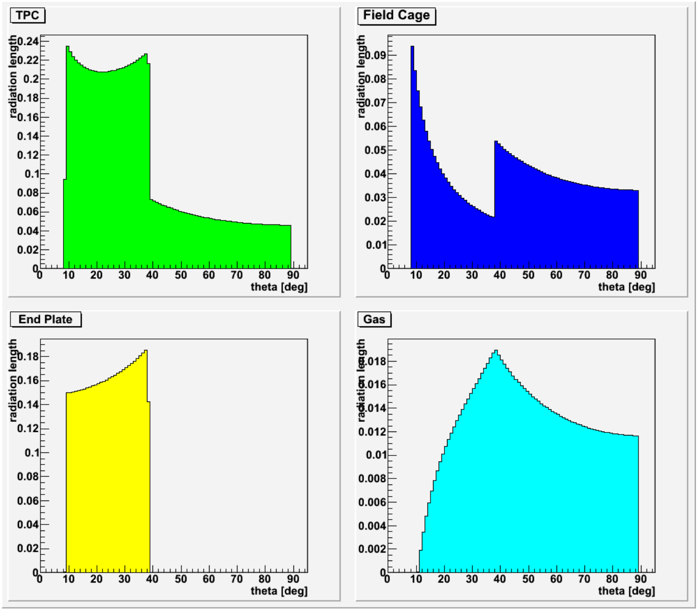
🟨 Class Imp

→ Data Flow

Application

- Tracking code: Material budget between two points
- Misalignment: shift volumes or coordinates based on global to local coordinate transformation
- Access to detailed geometry info that goes beyond abstraction level in GEAR API (but requires knowledge of volume hierarchy or some convention to browse efficiently)
- Local to global coordinate transformations
- Material budget for full detector
- . . .

Example: Material Budget TPC using new GEAR



Summary on GEAR extension

Allows a better geometry handling and description, plus gives access to information not easily available before

Users need to get latest version of ilc-soft and at least ROOT 5.27.06 to benefit

One extra line in GEAR xml file:

```
<GDMLFile name="World.gdml">
```

Everything else stays the same, changes are transparent to user!

Part III

Conclusion

Conclusion

The CERN LCD group exclusive activities in software development are:

- ILCDIRAC: provides user friendly access to grid resources
- GEAR extension: better geometry interface

Many other activities are not described here:

- detector model implementation (C. Grefe et al.)
- Pandora development (P. Speckmayer et al.)