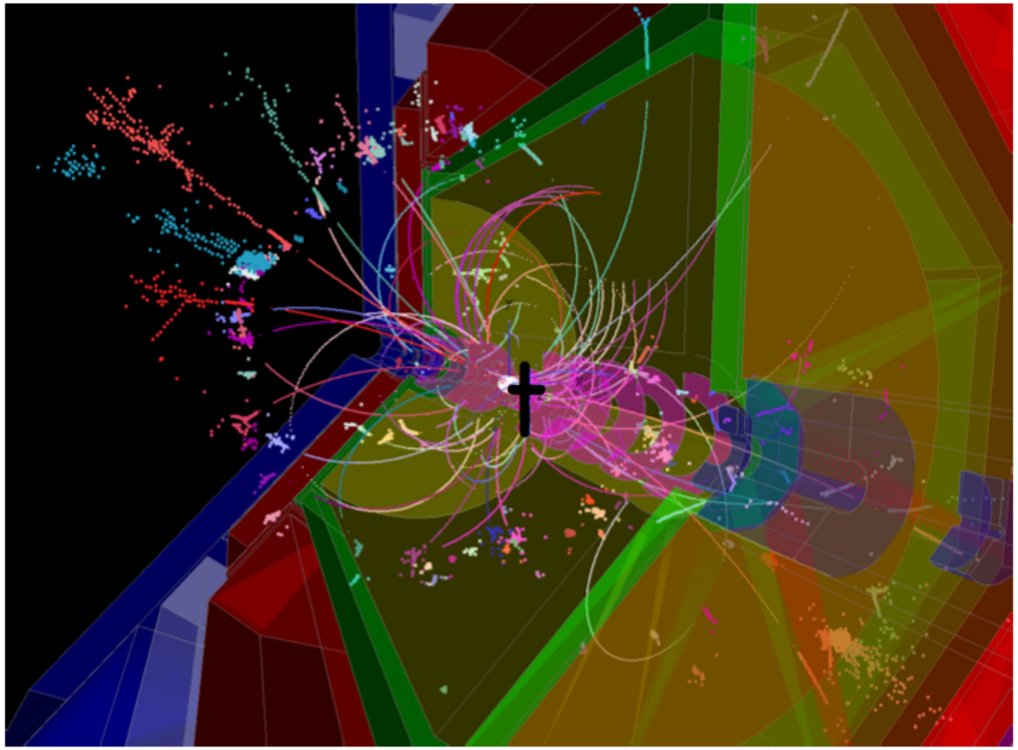
iLCSoft: Status, Plans and Grid Production

Steve Aplin
DESY

ILD Software Pre-meeting 2011 LAL Orsay
22nd May 2011

- iLCSoft v01-11
- LCIO – Evolution of the EDM
- Improvements to the Build System
- Grid Production



iLCSoft v01-11

- **Released and installed in afs and on the Grid:**

  /afs/desy.de/project/ilcsoft/sw

  i386_gcc34_sl4 i386_gcc41_sl5 x86_64_gcc41_sl5

- Mainly targeted at providing software release for the CLIC CDR Monte Carlo mass production – clear benefit here for ILD

Overlay	
Improved Tracking	(split tracks, forward tracks, z-slices in TPC)
Dedicated Digitizers	(timing information)
Background filters	(gamma gamma -> hadrons)

- Plus additional improvement to the core/build tools w.r.t. v01-10

LCIO – Evolution of the EDM

- **LCIO 2.0 (AKNA LCIOv2) is planned for some time now**

- Planned/requested features:

– **direct access to events**	**Done**
– **partial reading of events**	**?**
– **splitting of events over files**	**?**
– **storing of (arbitrary) user classes**	**Currently not planned**
– **simplify using LCIO with ROOT**	**Done**
– **improving the event data model**	**Under Way**

LCIO – Evolution of the EDM

- **Extensions of MCParticle**
  - add spin information: **float[3] getSpin()**
  - add color flow information: **int[2] getColorFlow()**
  - both copied from stdhep/HepEvt4 as written by Whizzard
  - user request: have simProcessId for particles that decayed in simulator
    use lower 16 bits of SimStatus word + collection parameters:
    SimProcessID, SimProcessName: **short getSimProcessID()**
    - need to define details of processIDs
    - need same implementation in Mokka and SLIC

LCIO – Evolution of the EDM

- **Meta data in event**
  - MC truth information from generator:

    processID

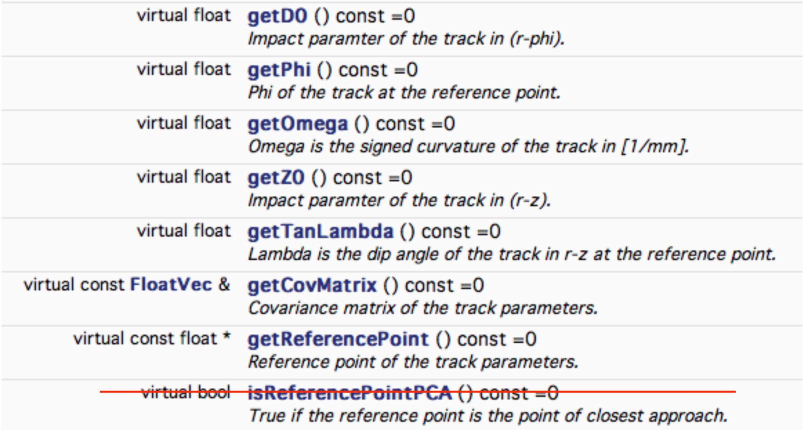
    processName

    alphaQCD

    alphaQED
  - store as collection parameters in LCEvent
    - to be implemented in StdHepReader and Mokka/SLIC
  - To be documented at: https://confuence.slac.stanford.edu/display/ilc/LCIO
  - Other meta information needed ?

LCIO – Evolution of the EDM

- **Track – multiple track states**
  - agreed to store multiple track states for Track:
    @IP, First Hit, Last Hit, Calo Face, other ?
  - Introduce Track State object: **TrackStateVec& getTrackStates()**
  - Track State will contain:



virtual float	**getD0** () const =0 *Impact paramter of the track in (r-phi).*
virtual float	**getPhi** () const =0 *Phi of the track at the reference point.*
virtual float	**getOmega** () const =0 *Omega is the signed curvature of the track in [1/mm].*
virtual float	**getZ0** () const =0 *Impact paramter of the track in (r-z).*
virtual float	**getTanLambda** () const =0 *Lambda is the dip angle of the track in r-z at the reference point.*
virtual const **FloatVec** &	**getCovMatrix** () const =0 *Covariance matrix of the track parameters.*
virtual const float *	**getReferencePoint** () const =0 *Reference point of the track parameters.*
~~virtual bool~~	~~**isReferencePointPCA** () const =0~~ *True if the reference point is the point of closest approach.*

  - original functions **getX()** of track class will return: **trk.getTrackStates()[0].getX()**

LCIO – Evolution of the EDM

- **TrackerHit and CalorimeterHit**
  - Canonical way of accessing layer number:
    getLayerNumber(), setLayerNumber()
  - filled from cellIDs after reading, write to cellID
  - need convention: string “layer” in CellIDEncoding
  - if “layer” not present – layerNum = -1 (deal with this in Marlin/org.lcsim)
  - update SLIC and Mokka accordingly

  - Add cellIDs to TrackerHit:
    getCellID0(), getCellID1() (as in CalorimeterHit)
    use cellID for consistency with CaloHit – (although generally there are no cells)
  - drop old 'type' word and replace **getType()** with access to cellID[“type”]

  - Convention for subdetectorIDs in cellIDs ?
    this will probably have to be done on a per concept (detector) basis
    need convention for ILD for DBD reconstruction

LCIO – Evolution of the EDM

- **1d and 2d Tracker Hits**
  - agreed to introduce six new TrackerHit classes

    PlanarDisk1D
    Planar1D
    Cylindrical1D
    PlanarDisk2D
    Planar2D
    Cylindrical2D
  - have u, du, pos1, pos2 (strip begin end) for 1D
  - have u, v + cov(u,v) + cylinder/plane parameters for 2D
  - details currently defined (Norman Graf)
  - these will likely also implement TrackerHit interface (x,y,z, cov) for backward compatibility

this will make it possible to properly take Si-strip detectors into account in the tracking (as manpower allows)

LCIO – Evolution of the EDM

- **SimCalorimeterHit**
  - Optionally add the position where the energy deposition (step) occurred:

    float[3] getStepPosition(int i) (only if fag LCIO.CHBIT_STEP==1)
  - useful for detailed simulation studies of edge effects in calorimeter cells or MAPS digitization

- **Cluster**
  - Add **float getEnergyError()**

any other requests ?

Improvements to the Build System

- **iLCSoft**
  - **Core packages**
    - LCIO, GEAR, LCCD, CED, …
  - **Marlin plugins**
    - MarlinReco, LCFIVertex, MarlinPandora, CEDViewer, …
  - **External dependencies**
    - ROOT, GSL, CLHEP, CERNLIB, …
  - Previously, dependencies in iLCSoft packages were treated individually between all packages, i.e.:
    - If A->B and B->C and D->A
    - Then: D->A, D->B, D->C
    - Not: D->A->B->C,
    - where D depends on A and therefore also gets B and C

Improvements to the Build System

- **Motivation for improvements**
  - The iLCSoft framework is continuously growing (~25 packages + external dependencies)
  - Initial versions of ilcinstall exclusively implemented for building packages using “traditional makefiles”, later on *upgraded* for using cmake as well.
  - Initial versions of CMake scripts for the iLCSoft packages
    - were designed to be backwards-compatible with *older* “makefiles”: led to use non-standard cmake constructs
  - Checking of package versions did not have high priority
  - BuildSetup.cmake including hard-coded paths proved a pitfall for most users
  - Other pitfalls, e.g. -DStreamlog_HOME=... -DIcio_HOME
  - Inter-package dependencies is a very tricky subject and needs careful handling, otherwise it can easily lead to “dependency-hell”:
    - why do I need to specify LCIO_HOME when Marlin anyways depends on LCIO?
  - **Simplify usage of cmake build tools for both, developers and users**

Improvements to the Build System

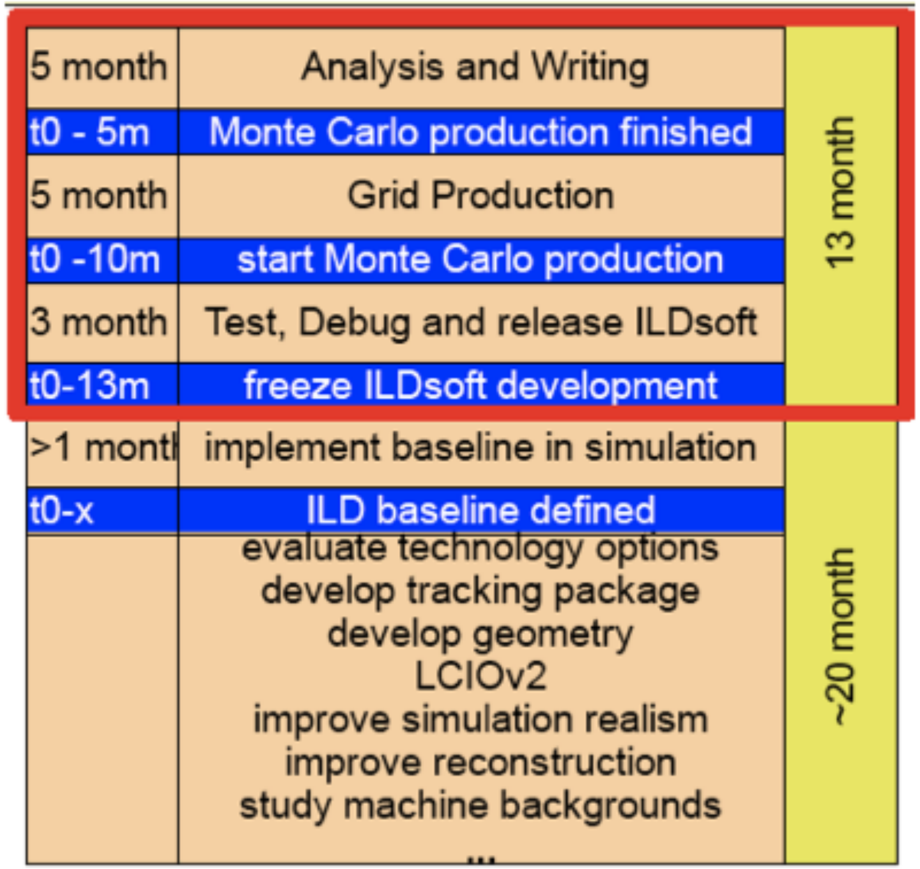
- **Deprecated features:**
  - BuildSetup.cmake files, PKG_HOME and BUILD_WITH cmake variables, macros: LoadPackage.cmake and CheckDeps.cmake
- **Replaced with:**
  - cmake PKG_DIR (or CMAKE_PREFIX_PATH) and standard cmake FIND_PACKAGE
- **New package: ILCUTIL**
  - "meta-package" grouping small utility packages needed by most iLCSoft packages

- **Reduced dependencies**
  - Now D->A->B->C as apposed to D->A and D->B and D->C
- **Building simple Marlin plugin is now much easier:**
  - FIND_PACKAGE(Marlin REQUIRED)
  - INCLUDE_DIRECTORIES(${Marlin_INCLUDE_DIRS})
  - LINK_LIBRARIES(${Marlin_LIBRARIES})
  - cmake -DILCUTIL_DIR=/path/to/ilcutil -DMarlin_DIR=/path/to/Marlin

  - **No need to know about LCIO, GEAR, streamlog or CMAKE_MODULE_PATH**

Mass Production

- Convergence on a simulation model and reconstruction for mass production to be discussed here.
- Need to establish process to realise the mass production.
- Need to identify major production sites.
- Both coordination and technically.
- DESY happy to take on this task. Naturally all offers to help will be gladly received.



5 month	Analysis and Writing	13 month
t0 - 5m	Monte Carlo production finished	
5 month	Grid Production	
t0 -10m	start Monte Carlo production	
3 month	Test, Debug and release ILDsoft	
t0-13m	freeze ILDsoft development	
>1 montl	implement baseline in simulation	~20 month
t0-x	ILD baseline defined	
	evaluate technology options develop tracking package develop geometry LCIOv2 improve simulation realism improve reconstruction study machine backgrounds ...	

Mass Production

- **Proposal naming scheme for centrally produced files:**
  - description keys (one character)+string, separated by “.”
  - no order specified, can be extended with additional keys as needed
  - facilitates well defined parsing in scripts

- **Keys:**
  - r/s : ilcsoft version used for rec/sim
  - m : detector (Mokka) model
  - E : center of mass energy in GeV
  - P : process description string
  - e : electron polarization
  - p : positron polarization
    - e.g. eR80.pL30 , eL.pR (100% polarized)
  - I : generator process ID
  - c : release tag of new svn package ILDConfig; will contain:
    - MokkaDBConfig, StandardConfig packages
    - Custom README and all needed configuration and axillary files)

Comments Welcome

example: rv01-09-03.mILD_00_p01.E500.Paa_udsc.eR80.pL30.Iw123456.cKEK_sciEcal_01.DST0124-0723.000

Mass Production

- **Discussed common DST format**

- ILD DSTs:

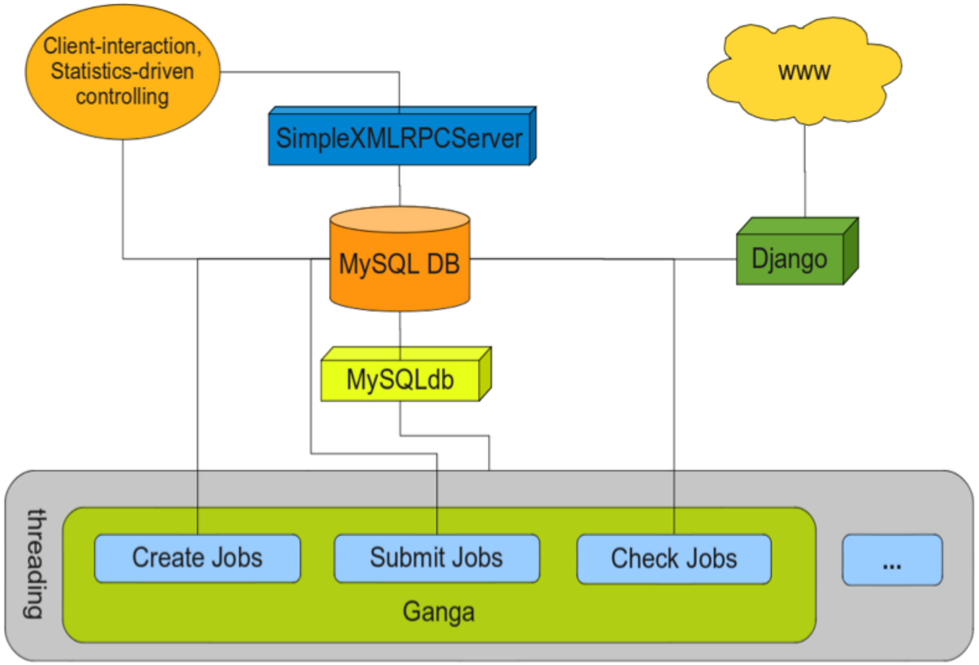
  ReconstrucedParticles (PFOs)
  Tracks
  Cluster
  MCParticlesSkimmed
  LCRelation PFOs ↔ MCTruth (via Tracks and Clusters or directly)
  2-6 JetCollections with Flavour tag

- SID interested in having a common definition of the DST format for the DBD

  will simplify collaboration on physics analyses

Mass Production

- **Mass Production System**
- Initial tests allowed to test the production system (thank you!)
- Bugs fixed and code improved



Mass Production

- **Initial tests were successfully run in the past months**

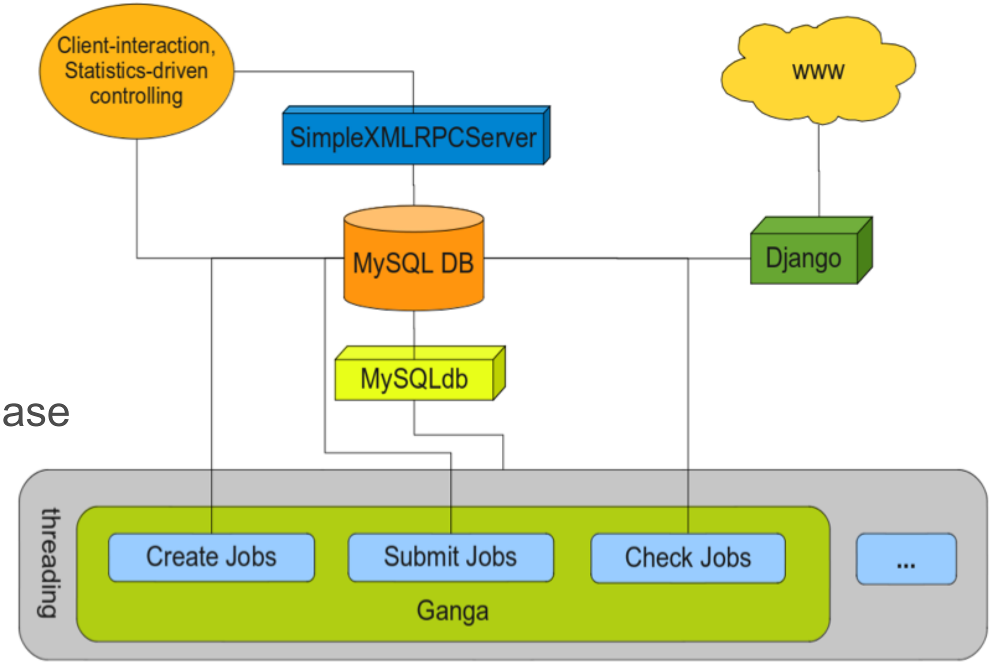
```
A. Miyamoto ~1.000.000 events
+--------+------------+--------------+--------+------------+---------------+--------+--------+------------+------+-----------+--------------+
| gen_id | process    | generator    | energy | luminosity | cross_section | pol_ep | pol_em | series     | cut  | root_name | total_events |
+--------+------------+--------------+--------+------------+---------------+--------+--------+------------+------+-----------+--------------+
|    329 | e1e1e2e2_o | whizard-1.40 |    350 |       1000 |       3912.51 |     -1 |     -1 | KEK_350GeV | null | w100615   |       250000 |
|    331 | e1e1e2e2_o | whizard-1.40 |    350 |       1000 |       3981.99 |      1 |     -1 | KEK_350GeV | null | w100616   |       250000 |
|    333 | e1e1e2e2_o | whizard-1.40 |    350 |       1000 |       3940.24 |     -1 |      1 | KEK_350GeV | null | w100617   |       250000 |
|    335 | e1e1e2e2_o | whizard-1.40 |    350 |       1000 |       3908.03 |      1 |      1 | KEK_350GeV | null | w100618   |       250000 |
+--------+------------+--------------+--------+------------+---------------+--------+--------+------------+------+-----------+--------------+
```

```
M. Terwort ~200.000 events
+--------+------------+-----------+--------+------------+---------------+--------+--------+---------------------+------+---------------------+--------------+
| gen_id | process    | generator | energy | luminosity | cross_section | pol_ep | pol_em | series              | cut  | root_name           | total_events |
+--------+------------+-----------+--------+------------+---------------+--------+--------+---------------------+------+---------------------+--------------+
|      1 | 21n1n1ee   | Whizard   |    500 |            |               |     -6 |      8 | 21n1n1ee_ep-6_em8   | null | 21n1n1ee_ep-6_em8   |        48165 |
|      2 | 21n1n1mumu | Whizard   |    500 |            |               |     -6 |      8 | 21n1n1mumu_ep-6_em8 | null | 21n1n1mumu_ep-6_em8 |        48217 |
|      3 | 31n1n1ee   | Whizard   |    500 |            |               |     -6 |      8 | 31n1n1ee_ep-6_em8   | null | 31n1n1ee_ep-6_em8   |        55716 |
|      4 | 31n1n1mumu | Whizard   |    500 |            |               |     -6 |      8 | 31n1n1mumu_ep-6_em8 | null | 31n1n1mumu_ep-6_em8 |        55961 |
+--------+------------+-----------+--------+------------+---------------+--------+--------+---------------------+------+---------------------+--------------+
```

```
M. Berggren ~10.000 events
+--------+------------+-----------+--------+------------+---------------+--------+--------+---------------+------+--------------+
| gen_id | process    | generator | energy | luminosity | cross_section | pol_ep | pol_em | series        | cut  | total_events |
+--------+------------+-----------+--------+------------+---------------+--------+--------+---------------+------+--------------+
|      1 | bbcsdu     | Whizard   |    500 |    1000.02 |       164.701 |      1 |     -1 | w17785        | NULL |         4200 |
|      3 | bbcse1n1   | Whizard   |    500 |    1000.01 |       57.4803 |      1 |     -1 | w17773        | NULL |         1500 |
|      5 | bbcse2n2   | Whizard   |    500 |    1000.01 |       57.0802 |      1 |     -1 | w17777        | NULL |         1500 |
|     11 | bbcse3n3   | Whizard   |    500 |          0 |       57.0304 |      1 |     -1 | w19389_fixed2 | NULL |         1500 |
|     13 | bbn1e1e2n2 | Whizard   |    500 |    1000.03 |       19.1375 |      1 |     -1 | w17697        | NULL |          500 |
|     19 | bbn1e1e3n3 | Whizard   |    500 |          0 |       19.1229 |      1 |     -1 | w19357_fixed2 | NULL |          500 |
|     25 | bbn3e3e2n2 | Whizard   |    500 |          0 |       19.0238 |      1 |     -1 | w19369_fixed2 | NULL |          500 |
+--------+------------+-----------+--------+------------+---------------+--------+--------+---------------+------+--------------+
```

Mass Production

- **Mass Production System**
- Initial tests allowed to test the production system (thank you!)
- Bugs fixed and code improved



- Work in progress:
  - Template for adding generator files into database
  - Web interface for file search and browsing
  - Improve meta-data in database catalog

Ready for Next Mass Production

Summary

- The increased feedback on software quality is very welcome.
- Improvements to the build system will benefit all users.
- Latest additions to LCIO data-model need to be finalised very soon.
- Need to proceed with the Mass Production procedure.