Outline

• Introduction
• Current work
• Near term goals
• Looking ahead
• Conclusion
Introduction

- Common understanding, that the simulation and Analysis Software for the TESLA detector will:
  - be based on Geant4
  - implemented in OO languages, i.e. C++, Java
  - be flexible for changes (e.g. geometry, reconstruction algorithms,...)
- Until recently only full simulation and reconstruction f77 based BRAHMS
  - this needs to be changed -> done now!

Interfacing Mokka and BRAHMSreco

- BRAHMS has been split up into a simulation and a reconstruction part (c.f. Ties’ talk)
- Interface based on:
  - flat files (cio and zio)
    that can be read from C++, Fortran, Java, ...
- Start using the existing Geant4 simulation Mokka
- Mokka output:
  - ASCII files (one per event and detector)
  - incompatible with BRAHMS
- Need additional output format in Mokka
  -> brahmsio
brahmsio

- additional library to be included in Mokka:
  - keep changes to Mokka small
    - Mokka.cc, Control.hh[cc], GNUmakefile
    - #ifdef BRAHMSIO -> gcc -DBRAHMSIO
    - Mokka -f BRAHMS_RECO -o outdir ...
  - foresee usage by other Geant4-based application
- using KDevelop as IDE
- not in code repository (CVS) yet
- will be available soon

brahmsio classes

- file handling
- output format
- Mokka specifics
Full-Simulation Software Chain

- **Gen. evts (STDHEP)**
- **Simulation**: Geant4/C++
  - **Mokka**
  - **Code Change**
  - **Param. Change**
  - **GEOM (MySQL)**
- **Reconstruction**: Geant3/f77
  - **Brahms**
  - **Hits, flat files (brahmsio)**
  - **Code Change**
  - **Param. Change**
  - **GEOM f77**
- **Events [Jas, Root,..]**

Next steps I

- **finish work on brahmsio**
  - -> have first full-simulation reconstruction chain based on Geant4 in place soon
- **start to do physics with it !**

- **need to gain further experience with Geant4**
  - possibly use Geant4 for testbeam simulations
  - -> understand the simulated physics

- **start to look ahead:**
  - -> at the bigger picture
The bigger picture I

Persistency

- STDHEP
- Simulation
- Digitization
- Reconstruction
- Analysis

Geometry

- root, JAS, ...

A TESLA Detector SW-Framework

- a lot of questions are open if on thinks of a complete simulation software framework for Tesla detector and physics studies:
  - interfaces, data models
  - persistency formats
  - tools
- how to answer these questions:
  - find dedicated people working on the issue
  - go on a ‘shopping tour’ at experiments that have already solved (some of) the problems: Atlas, CMS, BaBar, LHCb
- Start this rather sooner than later!
Next Steps II

- define a data model for persistency that will be used by all (most) groups involved in LC simulations:
  - generated particles
  - hits and digies
  - tracks, clusters, jets
  - links (pointers) between all of the above
- define a first implementation format
- get interested and experienced people involved (US, France,...)

The bigger picture II
Conclusion

• The first full-simulation-reconstruction for the TESLA detector using Mokka and BRAHMSreco will be in place soon

• Further work and verification needed
  • physics analysis

• Start thinking ahead towards a more general simulation framework
  • define abstract persistency layer and first implementation