

Some "Trivialities"

- some rather general remarks:
- We should try to reuse as much as possible existing software packages
- We donnot have enough personpower to support ourselves large new packages
- The LC studies are not yet collaborations:
 - ➔ we should be as open as possible (only fix something if it is really necessary)
 - ➔ we have to rely on free software if at all possible
 - ➔ we have to support a wide variety of p latforms
- This does not mean that we should not design the software properly, a good design and a clear definition will only help

Discussion on Interfaces

- Interfaces and a clean definition of interfaces are very important

- The geometry definition interface

- functionality (following yesterdays presentations)

- ➔ define the shapes of objects

- ➔ define the materials

- ➔ define the dimensions

- ➔ define the hierarchies

Do this and keep track of geometry versions

- interface this ideally to both GEANT3 and GEANT4 versions of the simulation

Geometry Interface

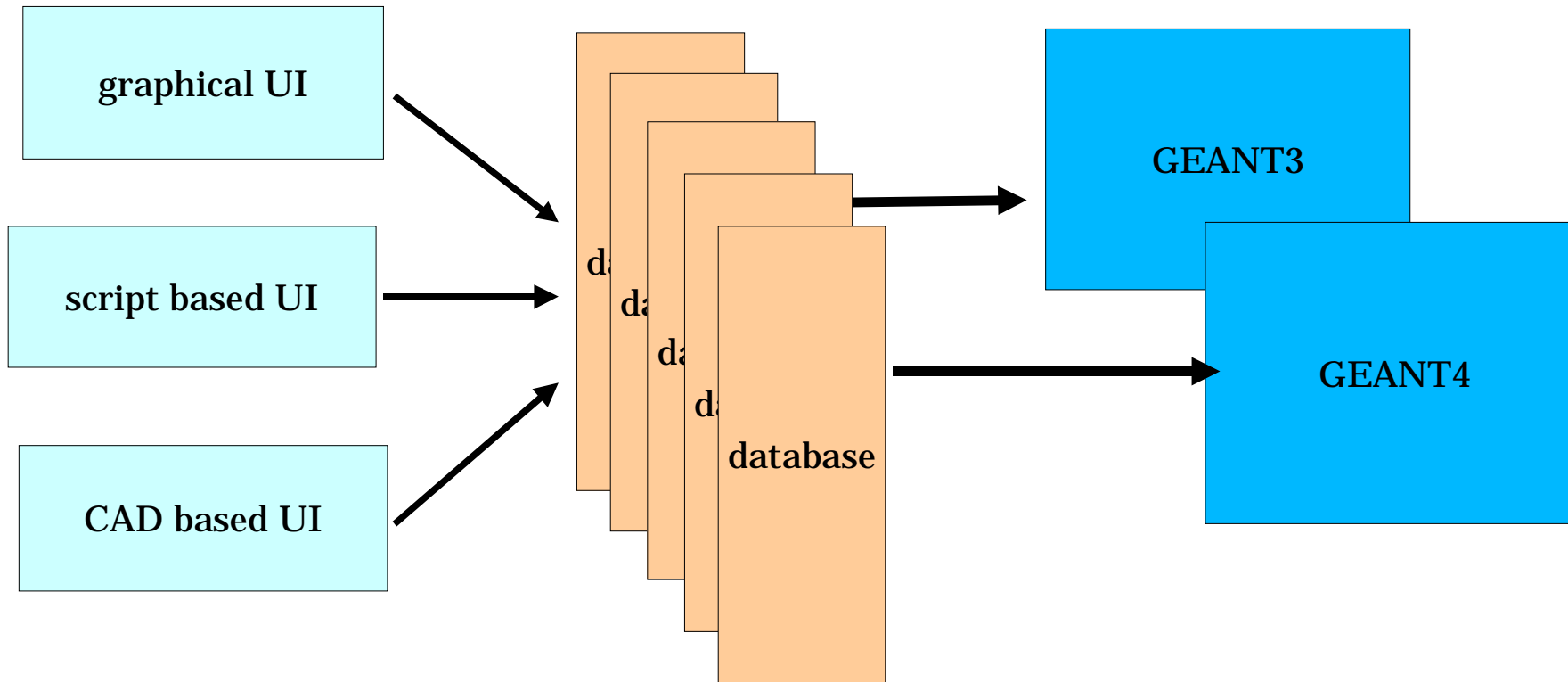
- What do we have:

- **BRAHMS**
 - xxgeom: dimensions
 - xxbld shapes and hierarchies
 - version management through program version management
- **MOKKA**
 - MYSQL dimensions, some hierarchies
 - xxdriver shapes and hierarchies
- **NLCSIM**
 - XML file dimensions, shapes, and hierarchies
 - general parser to translate into GEANT4 volumes

missing for all systems: a full versioning of the geometries (MOKKA comes closest)

the goal: marry the database approach and the XML approach somehow.

The ideal thing?



- we should study in a bit more detail what's available on the "market"
- the ATLAS implementation seemed quite close to our goal?
- the MySQL experience from MOKKA is very important

Persistency

- the "official" GEANT 4 persistency model is based on Objectivity:

- we should think very carefully whether we want to use this commercial, expensive software (this basically excludes anaphe?)

- root offers an alternative solution

- its free
 - its fairly powerful
 - it is used by a number of current experiment
 - requires intimate connection with root

- use a special system:

- MOKKA: structured directories
 - SIO ala NLCsim
 - simple ascii

} data management and data handling is not solved by these systems: interface to some data handling scheme is needed

Contents

- need to design and define the content of the "data"

- hit structure

- ➔ tracking

- ➔ calorimetry

- final event output structure

- ➔ event structure a la SIMDET?

- ➔ other structures?

- ➔ use root self defining structures?

We should try to solve these questions fairly soon, even if we do not fix the format

Analysis Tools/ Reconstruction modules

- less critical
- we should make sure we remain open: the user should use whatever he or she wants
- do we want to maintain PAW compatibility?
- ROOT / JAS can live side by side quite easily

- reconstruction: once the interfaces are defined, there should not be a real problem.....
 - ➔ multi language support
 - ➔ maintain existing programs
 - ➔ make implementation of new systems easy