Geant4 Review

Part II - Focus on Primary Lifecycle Processes

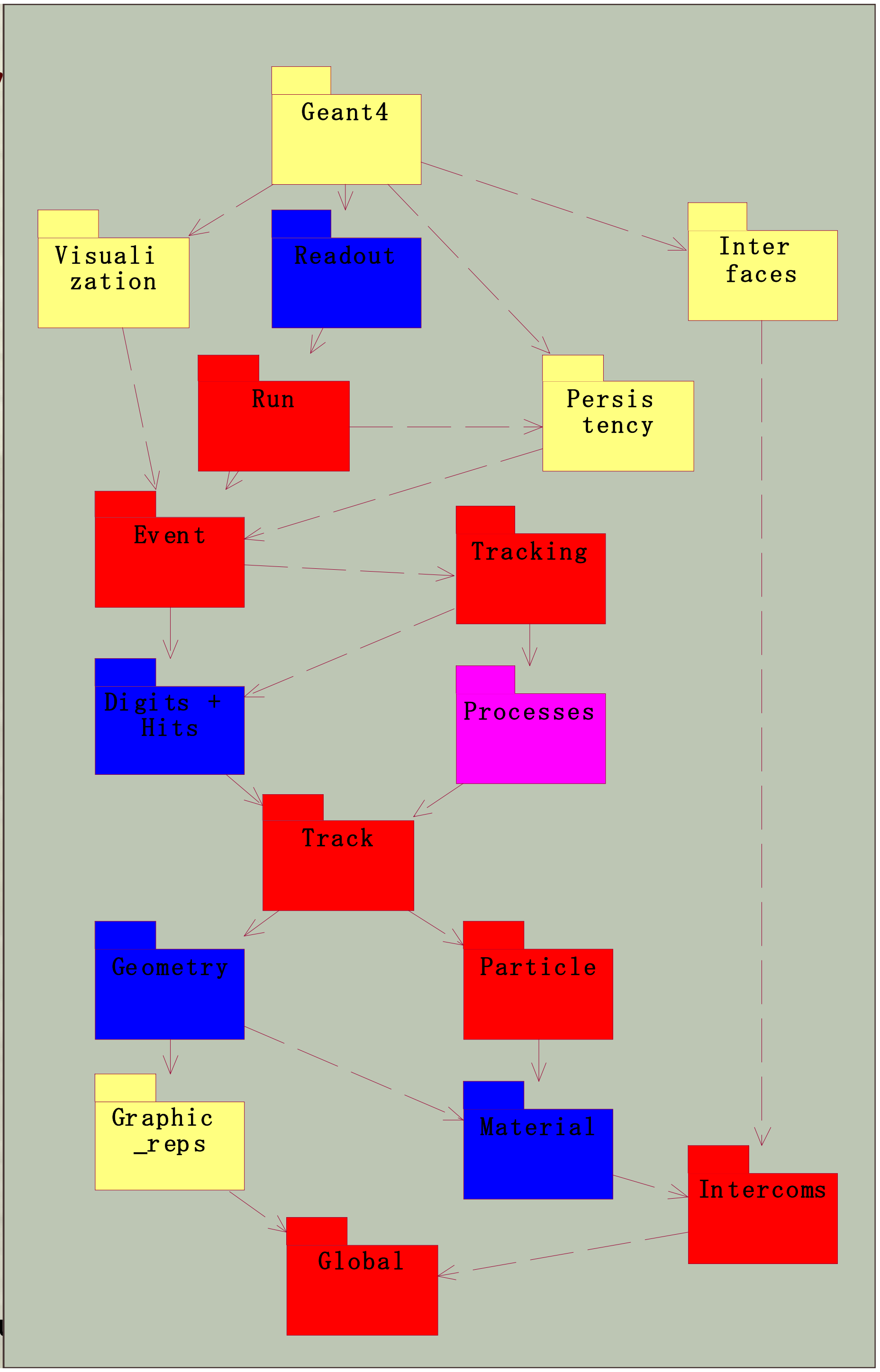
[2] Software Process : Kernel

June 19th, 2001

Makoto Asai (SLAC)

Categories covered by this talk

- Kernel categories
  - Run, Event, Tracking
  - Track, Particle
  - Intercoms, Global
- Detector description
  - Material
  - Geometry
  - Readout, Digits+Hits
- Process management



Maintenance of kernel components

- *Kernel components have been "stable" in the past two years*
  - *Only some minor bug fixes and plug-in modules were introduced.*
- *Major drivers contributing to stability:*
  - *architectural design success*
    - *effective domain encapsulation and abstraction of components*
    - *well defined interfaces*
  - *adoption of standards*

Handling of updates

- Handling enhancement requests
  - integration of new requirements
  - interaction with users
  - ongoing process improvement
    - also activity at the next Workshop
- Handling possible evolutions of the design
  - architecture-design Working Group
  - role of the Technical Steering Board
- Use of the Problem Tracking System (Bugzilla)
  - useful for tracking updates in the code
  - there were quite a few user's problems which caused "minor" design changes (e.g. adding const-ness)
  - all reports are traced till well-understood and resolved / fixed / rejected
- Adoption of History files

Encapsulation & Abstraction of kernel components

- Localization of required changes/fixes
  - fast identification of affected areas
  - easy localization of design decisions which are likely to change in future
  - reliable application with minimum effort
  - e.g. pre-assignment of lifetime of individual primary (required by ATLAS) was achieved by touching to just three classes
- Well defined interfaces of components
  - cross-release compatibility
  - efficient integration of new development
  - e.g. new physics processes are continuously developed and merged without touching to others

Adoption of Standards

- *To facilitate portability of the software in a variety of systems configurations and compilers*
- *To guarantee long life-time to the final product*
  - *ISO C++*
    - *migration of kernel code to ISO/ANSI C++*
    - *migration from Rogue-Wave Tools.h++ to STL*
  - *CLHEP, ODMG for persistency*