

Configuration Management

Geant4 Review

19 June 2001

Gunter Folger

Configuration Management

- Software components can be identified
- Software is built from a consistent set of components
- Software components are available and accessible
- Software components never get lost
- Every change to the software is approved and documented
- Changes do not get lost
- It is always possible to go back to a previous version
- A history of changes is kept, so that it is always possible to discover who did what and when

from Software Engineering Guides, C.Mazza et.al., (ESA guides to SE)

Identification

- Decomposition in domains and sub-domains results in directory structure for source code, developer controls individual classes
- Category coordinator has responsibility for a domain: design, source code, and documentation
- TSB decides on content and timing of releases.
- Release coordinator plans the release
- Release coordinator and STT responsible for testing and QA for a release

Consistent sets

- Source files (code, documentation, ...) are stored in a CVS repository
- CVS Tags used to form sets of files, each file has its specific version. Set ranges from single file to complete source code tree.
- A Geant4 release is identified by a specific tag.
- The release is built out of category tags.
- Testing proposed category tags towards a release is incremental
- The order of testing is given by the class category diagram

Availability and Accessibility

- Developers have access to all files and versions in CVS
- Access to the CVS repository is through
 - the AFS file system
 - a CVS server using the CVS protocol
- In general, a developer shall only change his files, and a category coordinator shall only changes files in his category
- Monthly development releases (reference tags) made available within the collaboration (CVS tag, source code and object libraries)

Change Management

Relying on CVS features:

- Complete history kept:
 - can go back to any previous version by version ID, date, or tag
 - can get differences between any two versions
 - can find out who changed what and when
 - comments on changes made kept with the file itself
 - CVS assists in merging (conflicting) changes
- No file ever gets deleted, can only be removed from current set
- History file per sub-domain for tracking changes

Building

Build system based on GNU make

- one library per sub-domain
- global libraries, i.e. one for each domain possible
- Dependency files allow minimal recompilation after a change
- Dependency files serve to identify needed sub-domain libraries in correct linking order
- Makefile rules in central place (config directory)
- Platform specific settings separate

Release

- Two major releases per year
- Minor releases or patches to release bug fixes
- Releases are available
 - on the web as source code and as compiled object libraries
 - In the collaboration in addition through CVS, as source code and compiled object libraries
- Documentation released 2-3 weeks after code release

Summary

- Configuration management stable since first release
- Rely on CVS and GNU-make