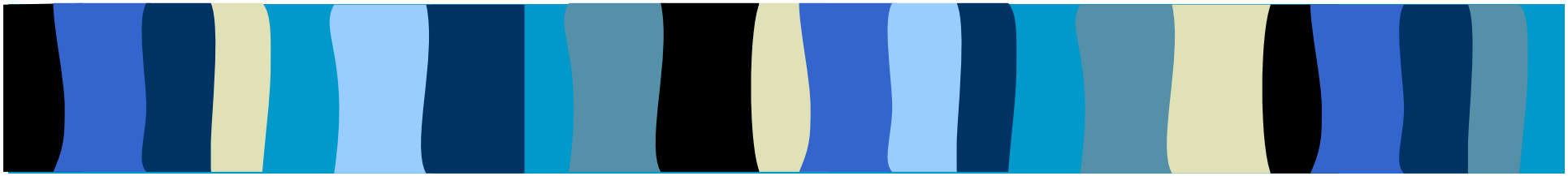
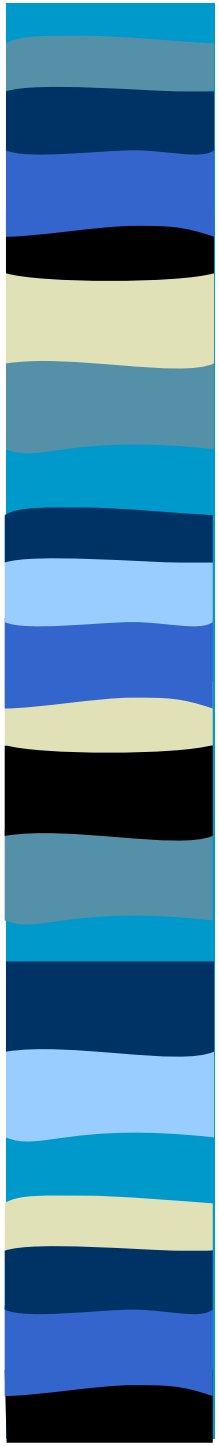
Software Process in Geant4



Gabriele Cosmo

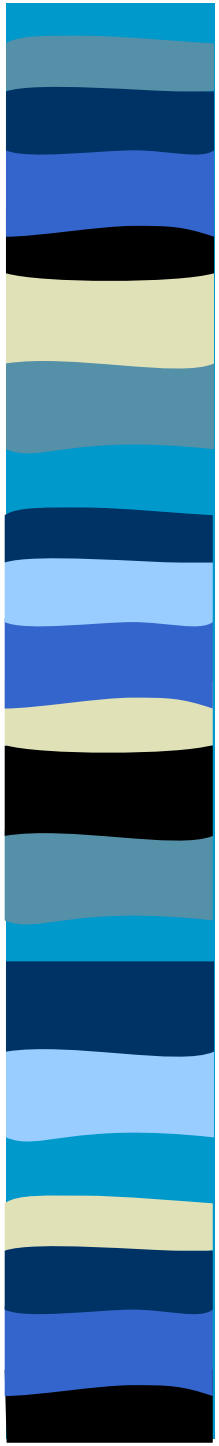
CERN IT/API-SI

Gabriele.Cosmo@cern.ch



Outline

- Overview on Software Processes
- The area of application
- Life-cycle processes in Geant4
- Software Process Improvement
  - Future evolutions
- Conclusions



Definitions...

- Software Process
  - A set of interrelated activities, which transform inputs into outputs (*ISO 12207/8402*)
    - used by an organisation or project to plan, manage, execute, monitor, control and improve any software related activity
  - Life-cycle processes are structured in *dimensions*:
    - Primary processes
      - includes all major functions of software development
    - Supporting processes
      - for supporting other processes with a purpose
    - Organisational processes
      - for corporate level management and improvement

Process Architecture

Customer-Supplier

- CUS.1 Acquisition
  - CUS.1.1 Acquisition Preparation
  - CUS.1.2 Supplier Selection
  - CUS.1.3 Supplier Monitoring
  - CUS.1.4 Customer Acceptance
- CUS.2 Supply
- CUS.3 Requirements Elicitation (*)
- CUS.4 Operation
  - CUS.4.1 Operational Use
  - CUS.4.2 Customer Support (*)

Engineering

- ENG.1 Development
  - ENG.1.1 System Requirements A&D
  - ENG.1.2 Software Requirements Analysis
  - ENG.1.3 Software Design (*)
  - ENG.1.4 Software Construction (*)
  - ENG.1.5 Software Integration
  - ENG.1.6 Software Testing
  - ENG.1.7 System Integration & Testing (*)
- ENG.2 System & Software Maintenance (*)

Support

- SUP.1 Documentation (*)
- SUP.2 Configuration Management (*)
- SUP.3 Quality Assurance
- SUP.4 Verification
- SUP.5 Validation
- SUP.6 Joint Reviews
- SUP.7 Audit
- SUP.8 Problem Resolution

Management

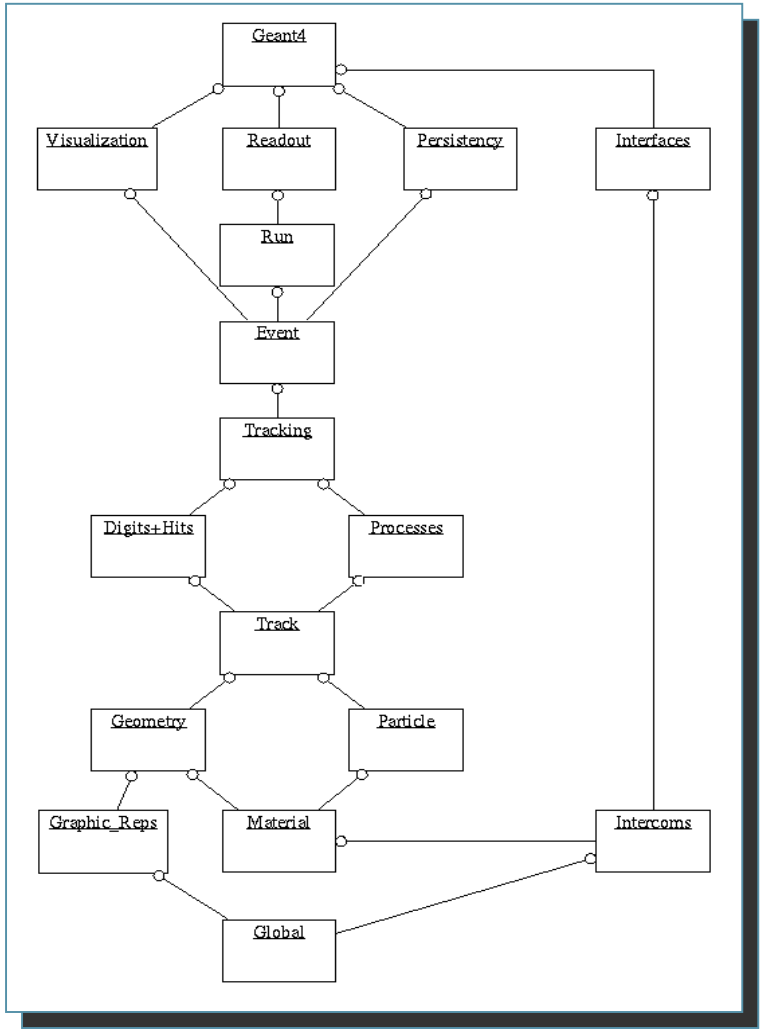
- MAN.1 Management
- MAN.2 Project Management
- MAN.3 Quality Management
- MAN.4 Risk Management

Organisation

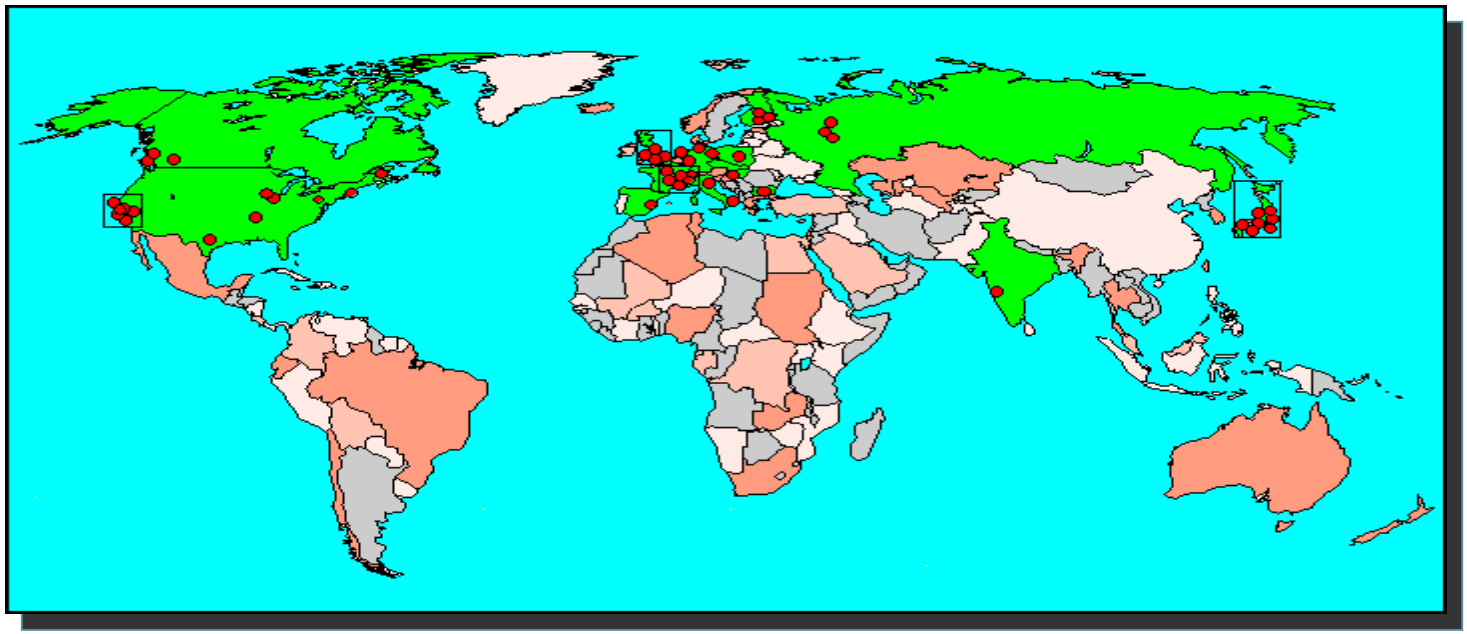
- ORG.1 Organisational Alignment
- ORG.2 Improvement
  - ORG.2.1 Process Establishment
  - ORG.2.2 Process Assessment
  - ORG.2.3 Process Improvement (*)
- ORG.3 Human Resource Management
- ORG.4 Infrastructure
- ORG.5 Measurement
- ORG.6 Reuse

The area of application: Geant4

- More than 1200 classes distributed in 17 Categories
  - software components in the Booch terminology
  - complex Categories organised in a hierarchical structure
- Decomposition to domain Categories derived from the design Category diagram
  - one development team associated to one Category domain



The area of application: Geant4



- Development teams distributed world-wide
  - domain decomposition <> geographical location of teams
  - centralized coordination of domain Categories
  - local coordination of each Working Group
    - assignment of responsibilities and support
  - distributed resources and funds in a *dynamic* environment
- Coordination for a coherent development
  - computing environment, methods and tools

Requirements Elicitation

- General User Requirements (UR) collected during the R&D phase of the project (RD44)
  - GEANT3 user community involved
  - URD generated according to the ESA PSS-05 software engineering standard
  - regular update and versioning of the URD along the development process
- Change-management based on *CVS*
  - general URD currently under revision
  - maintenance and tracking of specific detailed URDs under responsibility of WG coordinators
- New requirements approval: by the TSB
  - ongoing process improvement

Software Design

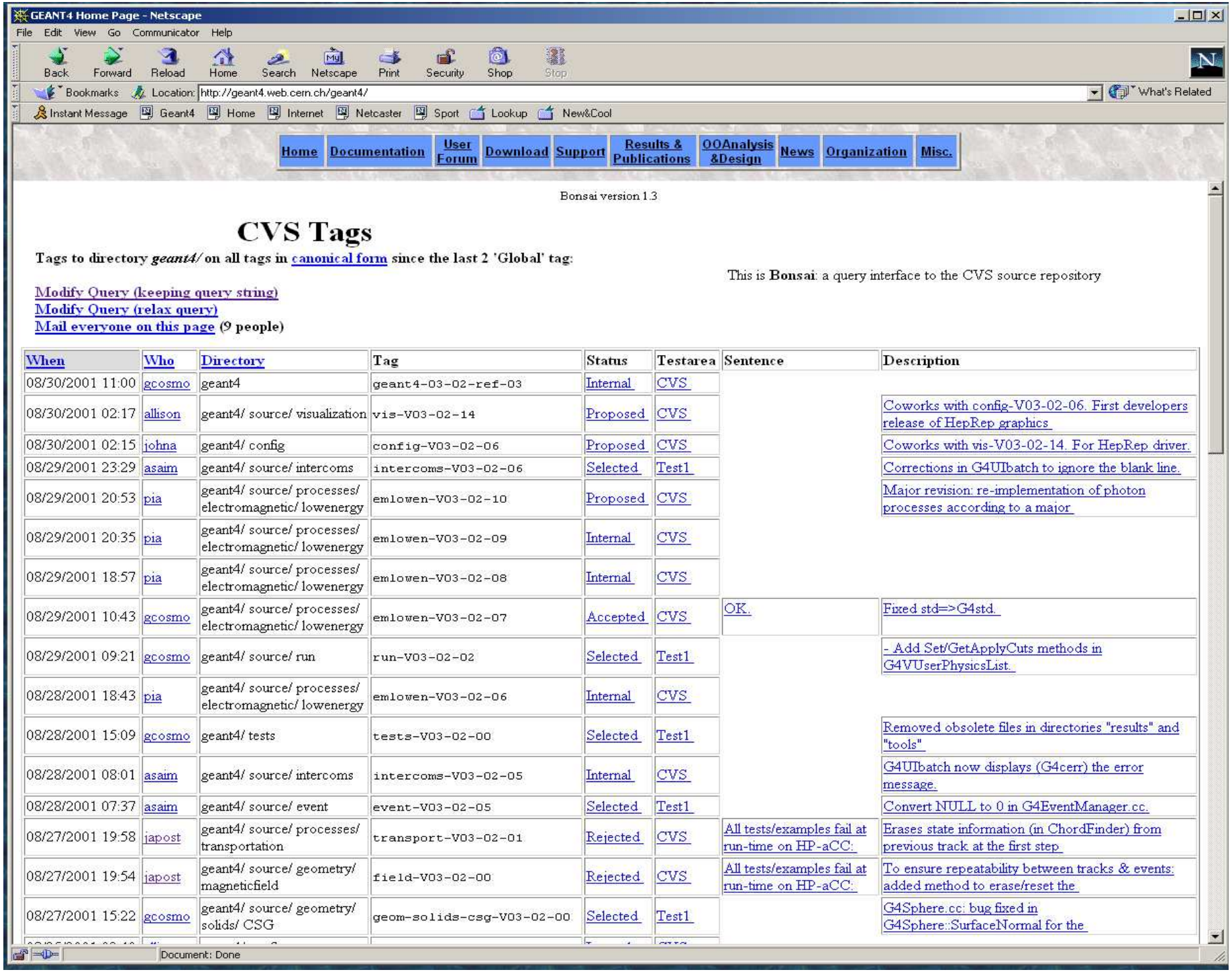
- Adoption of the *Booch* methodology for OOAD since the R&D project start
  - chosen after deep evaluation of the existing methodologies (‘94)
  - tailored to project specific needs
  - supported by CASE tools (*Rational-Rose*)
  - UML notation adopted for design documents
    - Category diagrams, Class diagrams, Scenario diagrams, Class specifications
    - ongoing process improvement
- Software development structured in *macro* and *micro* processes showed very effective
  - *iterative & incremental* approach (*spiral* model)
  - loose domain coupling led to efficient WG structure

Software Construction

- Software packaging reflects the domain decomposition in Categories
  - Packaging of Categories and sub-Categories in relation to definition of abstract and concrete interfaces (*frameworks*)
    - Provide a set of services in a *re-usable* way
    - Software *toolkit* approach
- Essential and flexible guidelines for coding
- Code filtering with specialised tools
  - *Code Wizard*
  - both in the global and unit context
    - tool accessible from Web

System Testing

- Activity deployed to a specialised team (STT)
  - based on defined procedures
    - *CVS* tagging policy
    - automated through Web tools and scripts
      - Bonsai, *LXR*, *Tinderbox*
      - ongoing process improvement
  - test applications used also for system integration
    - run & tested on every supported platform/compiler
    - ongoing process improvement
  - user example applications used for acceptance
- Category tags submitted to testing in sequence according to the dependency flow dictated by the design category diagram
- Close collaboration with the release manager

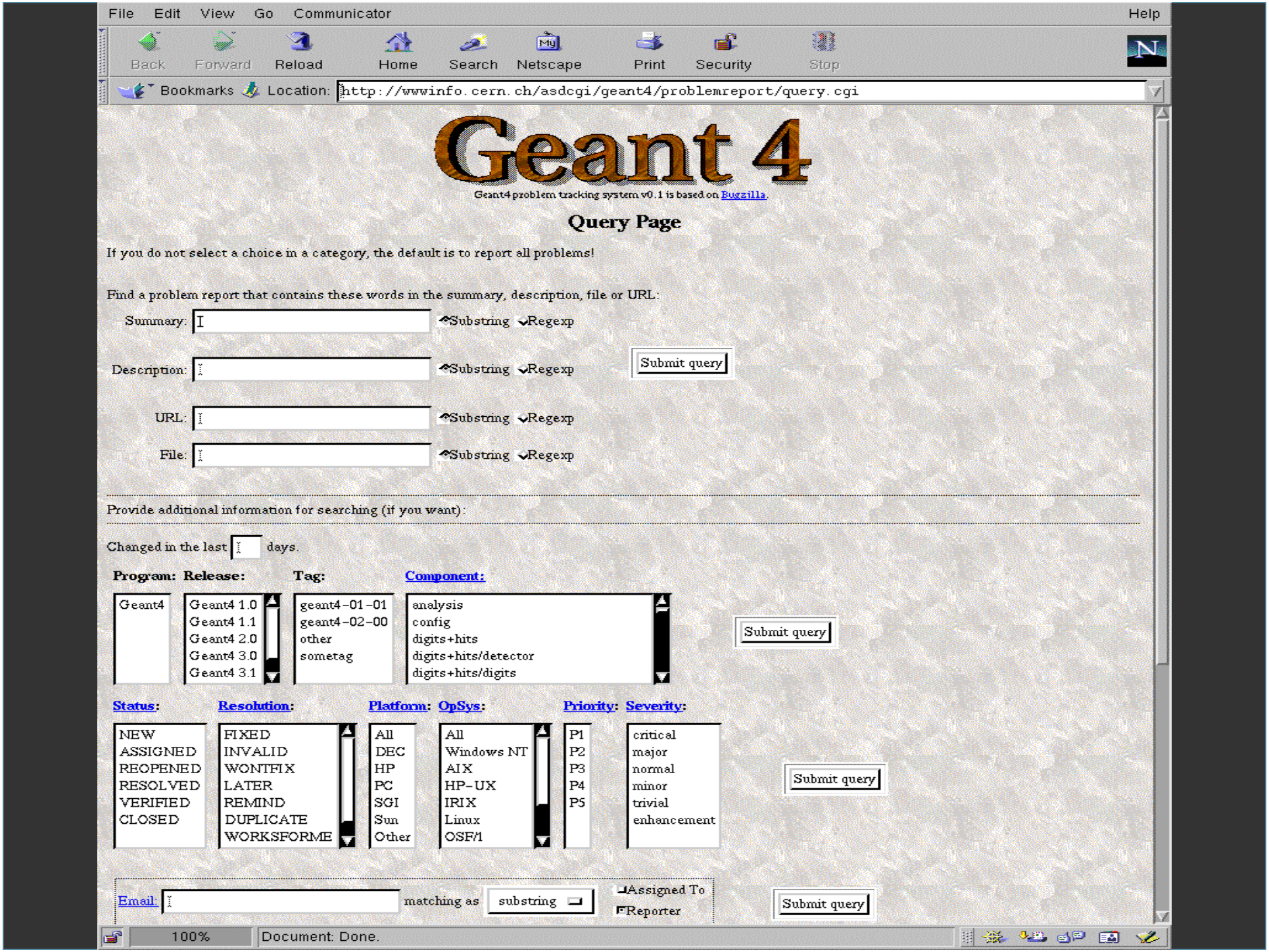

GEANT4 Home Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://geant4.web.cern.ch/geant4/
What's Related
Instant Message Geant4 Home Internet Netcaster Sport Lookup New&Cool
Home Documentation User Forum Download Support Results & Publications OOAnalysis &Design News Organization Misc.
Bonsai version 1.3
CVS Tags
Tags to directory geant4/ on all tags in canonical form since the last 2 'Global' tag:
This is Bonsai: a query interface to the CVS source repository
Modify Query (keeping query string)
Modify Query (relax query)
Mail everyone on this page (9 people)
When Who Directory Tag Status Testarea Sentence Description
08/30/2001 11:00 gcosmo geant4 geant4-03-02-ref-03 Internal CVS
08/30/2001 02:17 allison geant4/ source/ visualization vis-V03-02-14 Proposed CVS Coworks with config-V03-02-06. First developers release of HepRep graphics
08/30/2001 02:15 johna geant4/ config config-V03-02-06 Proposed CVS Coworks with vis-V03-02-14. For HepRep driver.
08/29/2001 23:29 asaim geant4/ source/ intercoms intercoms-V03-02-06 Selected Test1 Corrections in G4UIbatch to ignore the blank line.
08/29/2001 20:53 pia geant4/ source/ processes/ electromagnetic/ lowenergy emlowen-V03-02-10 Proposed CVS Major revision: re-implementation of photon processes according to a major
08/29/2001 20:35 pia geant4/ source/ processes/ electromagnetic/ lowenergy emlowen-V03-02-09 Internal CVS
08/29/2001 18:57 pia geant4/ source/ processes/ electromagnetic/ lowenergy emlowen-V03-02-08 Internal CVS
08/29/2001 10:43 gcosmo geant4/ source/ processes/ electromagnetic/ lowenergy emlowen-V03-02-07 Accepted CVS OK. Fixed std=>G4std.
08/29/2001 09:21 gcosmo geant4/ source/ run run-V03-02-02 Selected Test1 - Add Set/GetApplyCuts methods in G4VUserPhysicsList.
08/28/2001 18:43 pia geant4/ source/ processes/ electromagnetic/ lowenergy emlowen-V03-02-06 Internal CVS
08/28/2001 15:09 gcosmo geant4/ tests tests-V03-02-00 Selected Test1 Removed obsolete files in directories "results" and "tools"
08/28/2001 08:01 asaim geant4/ source/ intercoms intercoms-V03-02-05 Internal CVS G4UIbatch now displays (G4cerr) the error message.
08/28/2001 07:37 asaim geant4/ source/ event event-V03-02-05 Selected Test1 Convert NULL to 0 in G4EventManager.cc.
08/27/2001 19:58 japost geant4/ source/ processes/ transportation transport-V03-02-01 Rejected CVS All tests/examples fail at run-time on HP-aCC: Erases state information (in ChordFinder) from previous track at the first step
08/27/2001 19:54 japost geant4/ source/ geometry/ magneticfield field-V03-02-00 Rejected CVS All tests/examples fail at run-time on HP-aCC: To ensure repeatability between tracks & events: added method to erase/reset the
08/27/2001 15:22 gcosmo geant4/ source/ geometry/ solids/ CSG geom-solids-csg-V03-02-00 Selected Test1 G4Sphere.cc: bug fixed in G4Sphere::SurfaceNormal for the
Document: Done

Software Maintenance

- Adoption of standards
- Encapsulation of components
  - minimise coupling to reduce software complexity
  - regular monitoring of architectural dependencies
- Avoid system-dependent solutions in the source code as much as possible
  - centralise system configuration management
  - modular structure for architecture setups
- Avoid use of too “advanced” language features to maximise porting
- Traceability of updates
  - history files & regular tagging
  - disentangle development from bug-fixes

Customer Support

- Terms of the User Support are defined in the Memorandum of Understanding (MoU)
- Effort shared among WGs
  - contact persons defined for each WG
  - acting as experts in their specific domain
  - joint meetings with users and developers
- Problem Tracking System (*Bugzilla*) available to users
  - flexible design allowing easy customisation for Geant4
  - tokens automatically assigned to responsible persons
  - 300 reports submitted since tool in production
  - ongoing process improvement
- On-line documentation, training and FAQ on Web
- Source code and binaries available on Web and AFS
- *Hypernews* user forum available (hosted by SLAC)


File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://wwwinfo.cern.ch/asdcgi/geant4/problemreport/query.cgi
Geant 4
Geant4 problem tracking system v0.1 is based on Bugzilla.
Query Page
If you do not select a choice in a category, the default is to report all problems!
Find a problem report that contains these words in the summary, description, file or URL:
Summary: Substring Regexp
Description: Substring Regexp
Submit query
URL: Substring Regexp
File: Substring Regexp
Provide additional information for searching (if you want):
Changed in the last days.
Program: Release: Tag: Component:
Geant4
Geant4 1.0
Geant4 1.1
Geant4 2.0
Geant4 3.0
Geant4 3.1
geant4-01-01
geant4-02-00
other
sometag
analysis
config
digits+hits
digits+hits/detector
digits+hits/digits
Submit query
Status: Resolution: Platform: OpSys: Priority: Severity:
NEW
ASSIGNED
REOPENED
RESOLVED
VERIFIED
CLOSED
FIXED
INVALID
WONTFIX
LATER
REMIND
DUPLICATE
WORKSFORME
All
DEC
HP
PC
SGI
Sun
Other
All
Windows NT
AIX
HP-UX
IRIX
Linux
OSF/1
P1
P2
P3
P4
P5
critical
major
normal
minor
trivial
enhancement
Submit query
Email: matching as substring
Assigned To
Reporter
Submit query
100%
Document: Done.

Documentation

- Six user manuals available on-line
  - Introduction to Geant4
  - Installation Guide
  - User's Guide for Application Developers
  - User's Guide for Toolkit Developers
  - Physics Reference Manual
  - Software Reference Manual
- User examples: novice, extended, advanced
- Training kit: three module-structured courses
- Design documents
- Defined policy for update

Configuration Management - releases

- Defined policy for *major* and *minor* releases
  - 4 major releases, 4 minor releases, 6 patches published since in production (December ‘98)
  - policy periodically revised and updated
- Development releases distributed monthly to collaborators and developers
  - additional development releases if necessary
- Close collaboration with System Testing Team
  - acceptance tests, part also of system tests, are also run independently by the release manager
- Prompt collaboration from developers required during the public release phase

Software Process Improvement (SPI)

- Understand, determine and establish applicable procedures to Software development and maintenance of the software
- Make SPI a Software Process *life-cycle driven*

⇒ Primary life-cycle processes:

  – guarantee that the code quality will not degrade with time: SPI actions associated with a regular QA activity

  – assure that coupling will not increase with the growing complexity of the software

⇒ Improve overall usability and robustness of applications: improve quality, maintainability and reliability of the code

⇒ Assure continuity and integration of regular system testing within the normal Software development activity

Software Process Improvement (SPI)

- (Chosen) Domains of applicability in Geant4:
  - Q/A & Optimisation activity
    - applied to the software product in either global and component domain related context
  - Analysis & Design software cycle
    - identify the well established OOP procedure for development and maintenance – assessment based on ISO-15504
  - Testing
    - assure constant improvement and continuity to system testing
- Action for improvement identified
  - plan for SPI established
  - progressive implementation

Future evolutions

- Make SPI part of the software life-cycle
- Consider monitoring progress of the SPI program
  - regular check-points at *Category-Coordinator meetings*
    - regular update of status:
      - **http://cern.ch/geant4/milestones/software_process**
  - include activities addressing SPI in the Collaboration Workshops
- Iterate new assessments in future
  - extend assessment to uncovered (or partially covered) domains (testing, documentation, Software Management)
  - try improving Capability level

Conclusions

- Geant4, a challenging project for applying Software Processes
- Current strategy demonstrated to be effective and flexible
  - far from being perfect !
    - requires continuous monitoring and improvement
    - SPI must be life-cycle driven
  - organisational alignment