CMT/SVN news, Atlas SW/SIT meeting

Christian Arnault, Vincent Garonne
{arnault,garonne}@lal.in2p3.fr

Laboratoire de l'Accélérateur Linéaire

April 6, 2006

Outline

CMT news

CMT *news (1)*

New CMT *release (version v1r18p20060301)*

No major improvements but fixed several problems with the existing CMT versions, including:

- ▶ Non-existent directories are no longer removed from search paths
- ▶ Add new warning messages in relation with setup checking which are explicitly called with the `--warnings` option
- ▶ The run-time environment for a release can be setup by specifying the "runtime" during setup command, e.g.:
  `> source ∼/cmthome/setup.sh -tag=11.0.4,runtime`
- ▶ New version of the AtlasSettings package (backwards compatible)

CMT *news (2)*

New CMT *release (version v1r18p20060301)*

- ▶ Change the internal system for setting environment variables in a proper way
- ▶ Better support of the (non-)version-directory mode in severals cases:
  - • Option to force package version directories to be created when using "cmt co":
    set CMTSTRUCTURINGSTYLE "with_version_directory"
  - • Resp. "without_version_directory" force a package version directory not to be created.
- ▶ See Changelog [1] for more informations

CMT *feedbacks*

Reports

- ▶ Tested on the nightlies with "no" problems except
  - • Should understand the problem of eternal loop encountered with dictionaries ...
- ▶ Should be the 'stable' v1r18 in the futur
- ▶ Problems encountered with the cmtcvs plugin in the way cvs tags are printed :
  - • Thanks to D. Chamont, this is fixed and should be available for the next plugin release

CMT *Future*

Future improvements planned for the next CMT release :

- Request for having a macro that reflects all active tags , i.d. we should introduce a link between the macro and tag concepts

- Improve the log message mechanism for message classification within *CMT*, e.g. "à la" Python's logging module [2], Apache's logging module [3]

Exploring SVN for Atlas

Scenario for Atlas

Roadmap

1. Convert the Atlas CVS repository into a Subversion one
2. Provide tools to provide synchronization operations between *CVS* and *SVN* repositories
   - For a certain period, we will have both system running at the same time
   - remark: iterative transition is less painful for the developers :)
3. Incorporate *SVN* in the complex Atlas software management chain, it supposes impact to :
   - CMT
   - Tag Collector
   - Nicos
4. If successful, we could do the permanent switch; if not ... :(

Converting CVS Repository to Subversion

► We have developed our own tool "cvs2svn.py"

► We have selected the following organization for each project/package/tag:
  `<project>/.../<package>/trunk/...`
  `<project>/.../<package>/tags/<tag>/...`

► The complete Atlas cvs repository has been converted to a svn one. The svn url is :
  file:///afs/cern.ch/atlas/software/svnrepository/repository

► It takes a looong time to do that the first time
  • Try different *SVN* backends: Berkley DB, *SVN* file system
  • It takes me one week to complete the procedure, e.g. 8 hours for the AtlasCore project
  • but update will only concern recent tags

Conversion validation and management tools

Conversion validation tools

► We have implemented several tests to check the quality of the conversion:
  - Comparing the sources at each tag with diff tests
  - Comparing the list of tags for a package
  - Check global structure of the *SVN* repository
  - /afs/cern.ch/atlas/scripts/testcvs2svn.py

Management tools

► We have also implemented a script which re-organizes the SVN repository:
  ```
  <project>/.../<package>/trunk/...
  <project>/.../<package>/<tag>/...
  ```

Performance and scalability

Test stress scenario

We define the following **action** for an user:

▶ User selects randomly a package

▶ He extracts and modifies it, then registers it until the operation is successful

▶ Then he sleeps n seconds with n sort randomly $\in [1, T_{max}]$

Experiment parameters

▶ We have done experiments with different:
  • Total number of users (10, 100, 1000)
  • Total number of packages (10, 1000, 10.000)

▶ Concurrent user actions were launched by a multi-threaded program (/afs/cern.ch/user/v/vgaronne/svnTestStress)

Performance and scalability - Results

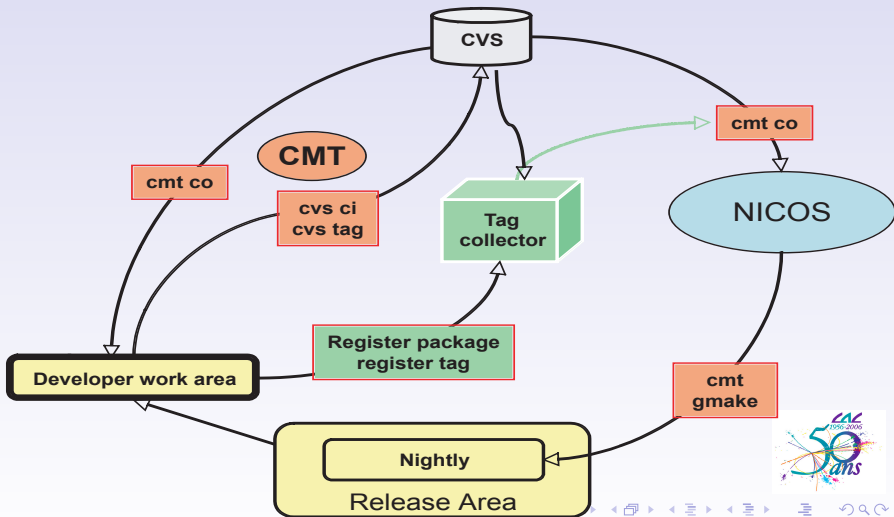▶ Let $\mathcal{R}atio$ be the ratio $\frac{Total\ number\ of\ users}{Total\ number\ of\ packages}$

We obtained :

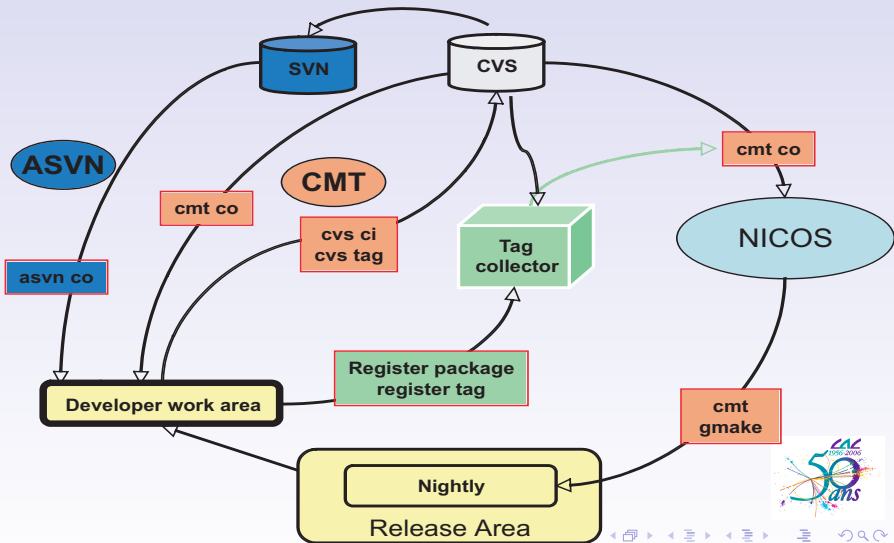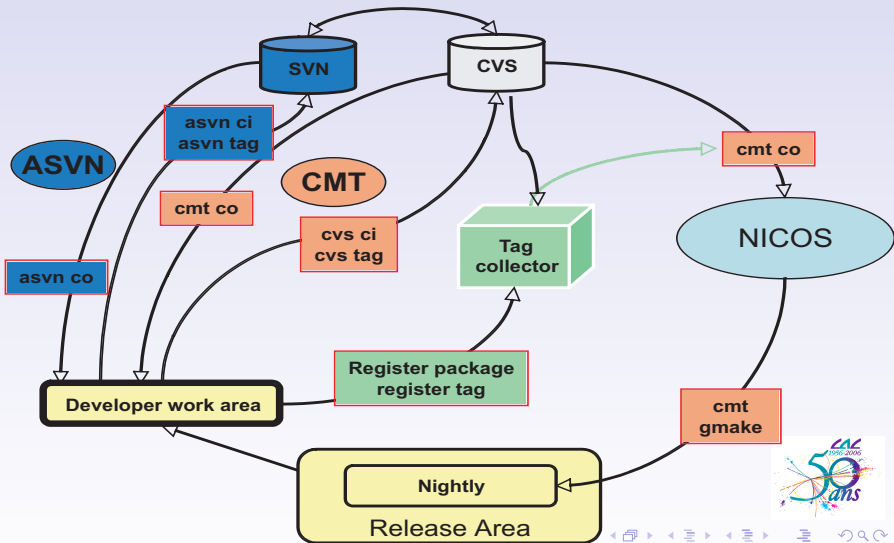| $\mathcal{R}atio$	T_{max}	Average action time (s)
0.01	3600.0	1.34
0.01	100.0	1.06
0.01	10.0	1.38
0.1	3600.0	5.77
0.1	100.0	4.26
0.1	10.0	6.96
1.0	3600.0	7.42
1.0	100.0	7.06
1.0	10.0	8.88

**In blue, the values relevant for Atlas

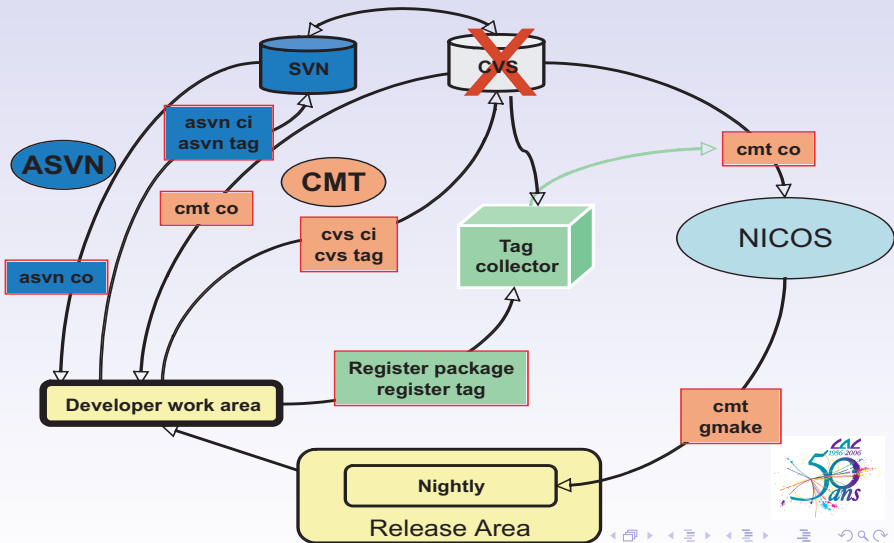# A brief summary by pictures
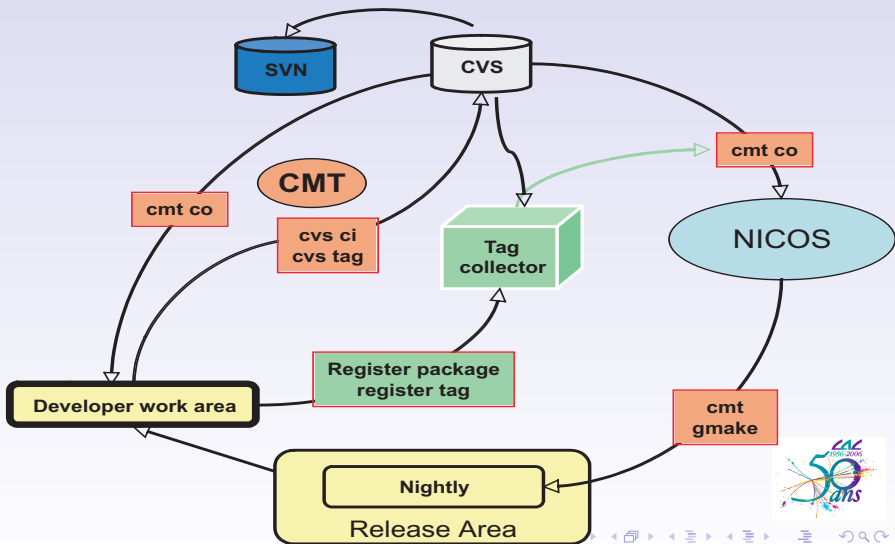
A brief summary by pictures
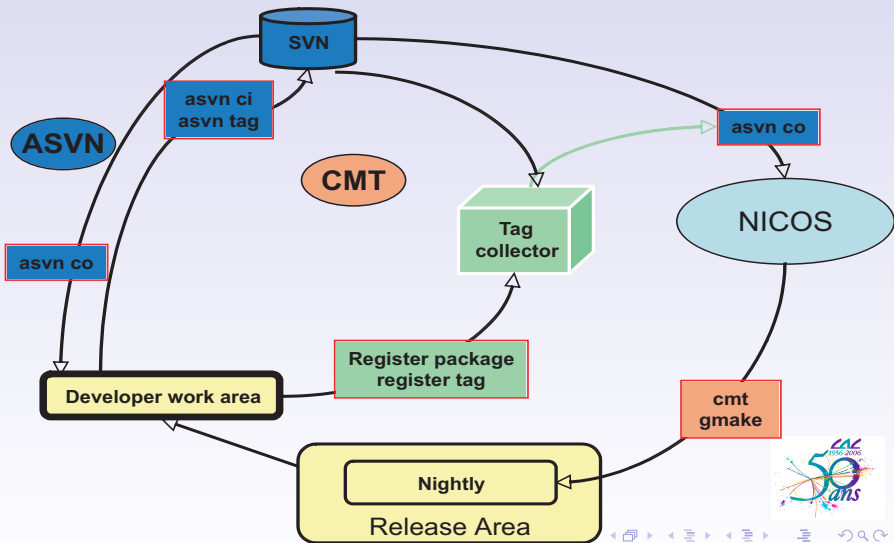
A brief summary by pictures

A brief summary by pictures

A brief summary by pictures

A brief summary by pictures

Synchronization operations

Mainly at the user level:

- ▶ We proposed a script "asvn" that mimics the CVS syntax for SVN, including the synchronization operations on tags:
  - Non exhaustive commands are: checkin, checkout , tag, ...
  - Provides also cmt specific commands working with SVN, e.g. recursive checkout, svntags, svnsubprojects, ...
  - see /afs/cern.ch/atlas/scripts/asvn

- ▶ We assume that when users switch completely to svn for a package, they will not work anymore with *CVS* for this package, otherwise we will have integrity problems

- ▶ The synchronization operation should be triggered at the user/developer level during tag operations
  ```
  > asvn --sync tag Package-00-00-01 Package
  ```

Impact to the Atlas software management chain and tools

Impact to the work model

- ▶ No real change is expected

Impact to the tools

- ▶ CMT: The "asvn" script previously presented covers these aspects
- ▶ Tag Collector: Needs an interface to SVN, but should be much simpler than the interface to CVS. If not possible immediately, we could go on temporally with the synchronized Atlas CVS repository
- ▶ NICOS: Normaly only concerned with checkout operations ?

Conclusion

- ▶ All the necessary tools are in place to start using SVN in Atlas
  - ▶ Doing more tests to validate the atlas svn repository
- ▶ Testing by real users, will help to tune, customize and improve these tools
  - ▶ We need friendly users !
- ▶ After, we could define milestones for the all conversion
- ▶ We should clarify the migration policy
  - ▶ Management of the "CVSROOT/commitavail" file for each package to guaranty synchronization between CVS and SVN repository
- ▶ Implementing the interface to SVN in Tag Collector, depends on TC team availability
- ▶ SVN should be installed at cern and supported by IT

References I

📄 CMT Changelog.
http://www.cmtsite.org/ReleaseNotes.html.

📄 Python's logging module.
http://docs.python.org/lib/module-logging.html.

📄 Logging Services.
http://logging.apache.org/log4cxx/.

📄 Subversion and CVS comparison.
http://wiki.gnuarch.org/SubVersionAndCvsComparison.