

5th Coding Sprint

- 1. Short overview over GammaLib and ctools**
- 2. New features since last coding sprint**
- 3. New development workflow**
- 4. Goals of this sprint**

Jürgen Knödseder (IRAP)

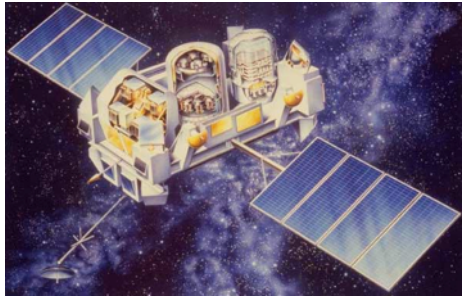
1. Short overview over GammaLib and ctools

The advent of VHE astronomy

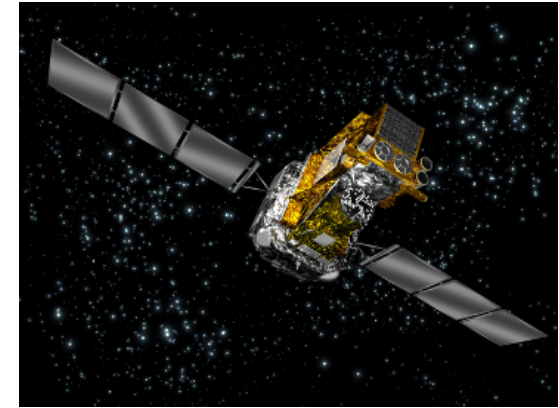


All these telescope produce the same kind of data, but for every telescope a different analysis software is used

The span of gamma-ray astronomy



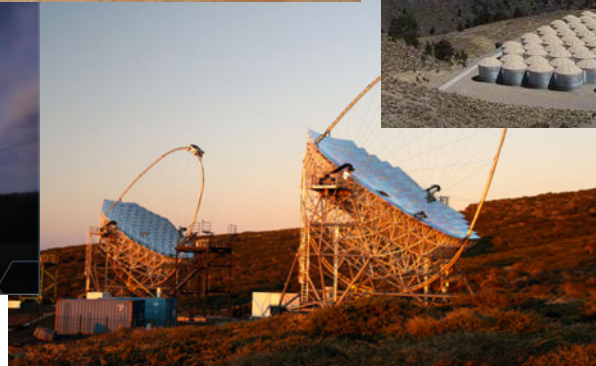
CGRO



INTEGRAL



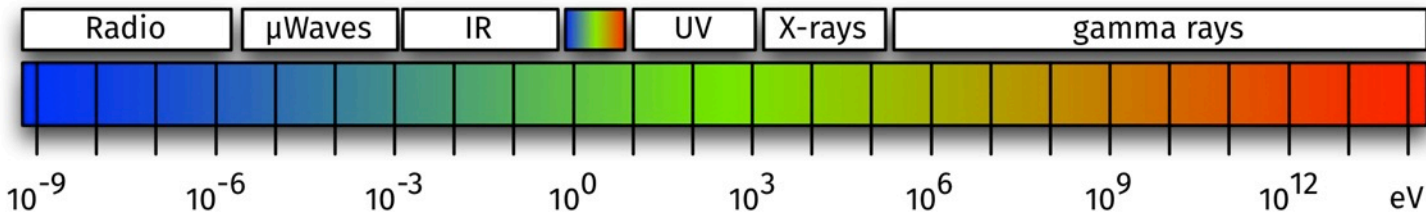
Fermi



HAWC



ASTROGAM



Gamma-ray astronomy commons

All instruments produce events

Events are characterised by an instrument direction, an energy estimate, and a trigger time

FITS is becoming the standard for delivering event data

Space-based telescopes deliver event data in FITS format. Efforts are underway to transform current IACT data into FITS. CTA will deliver event data in FITS. Dedicated workshop on 6-8 April 2016 in Meudon (Paris, France).

Maximum likelihood fitting is widely used for data analysis

Fermi-LAT, INTEGRAL, COMPTEL, EGRET, OSSE, ... and there is no reason not to use it also for VHE astronomy

Many analysis packages implement a modular ftools-style software

Let's do it ...

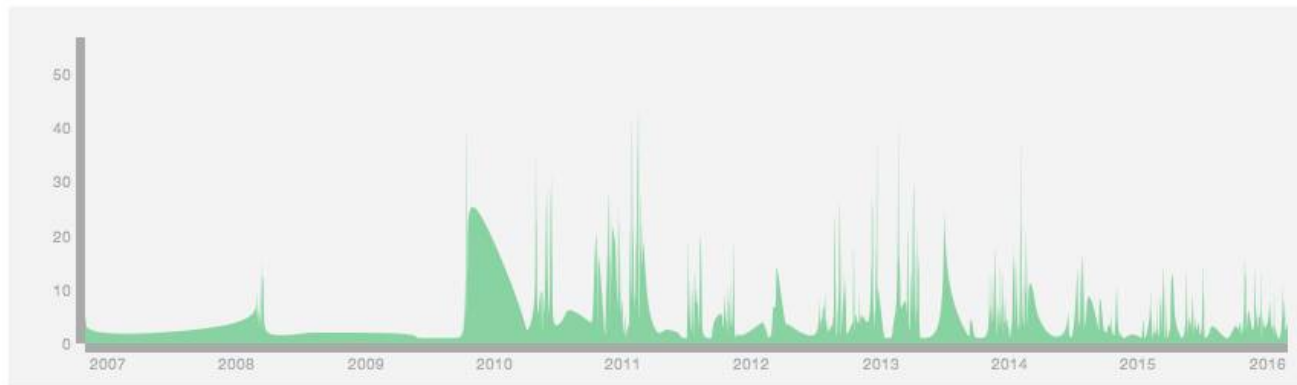
All these considerations led to the idea to develop a common software framework for the analysis of gamma-ray event data



Development started (slowly) in 2006 when I need a sparse matrix implementation to fit more than 10 000 parameters in a maximum likelihood fit for INTEGRAL/SPI data ...
Got with Fermi-LAT the first instrument support in 2008 ...
And took off in 2010 with the implementation of CTA support

October 26 2006 - February 26 2016

Commits to devel, excluding merge commits. Limited by 6,000 commits



29 February - 4 March 2016

5th ctools and gammalib coding sprint
(Jürgen Knödseder)

Let's do it ...

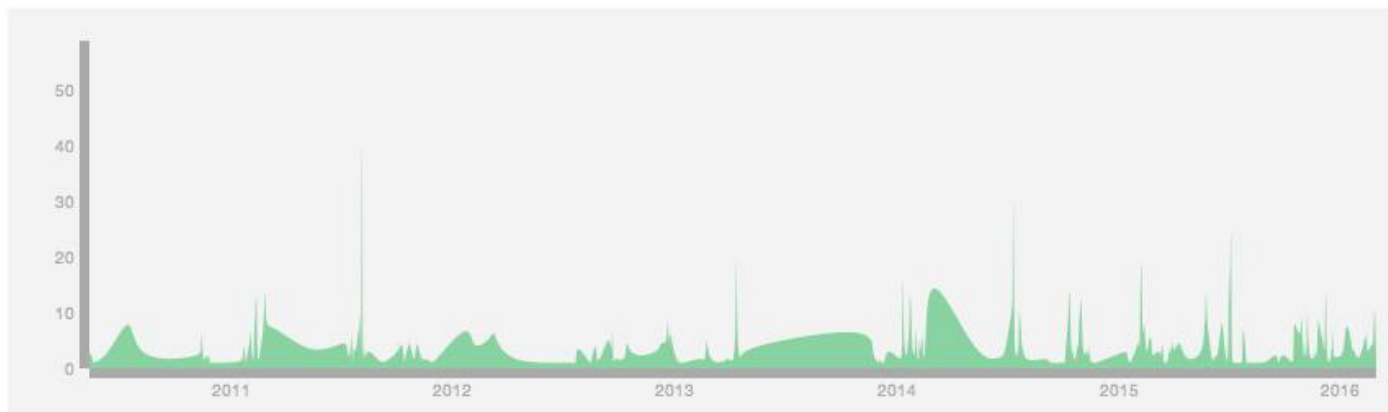
And once all the bricks for managing CTA data were there it was time to create ...



Development started in 2010 as ctatools
First release in 2011 comprises ctobssim, ctselect, ctbin
and ctlike (pure C++ package)
Named ctools in 2011 (with Python support)
Today comprises ~30 tools

May 11 2010 - February 27 2016

Commits to devel, excluding merge commits. Limited by 6,000 commits



29 February - 4 March 2016

5th ctools and gammalib coding sprint
(Jürgen Knödseder)

Code evolution

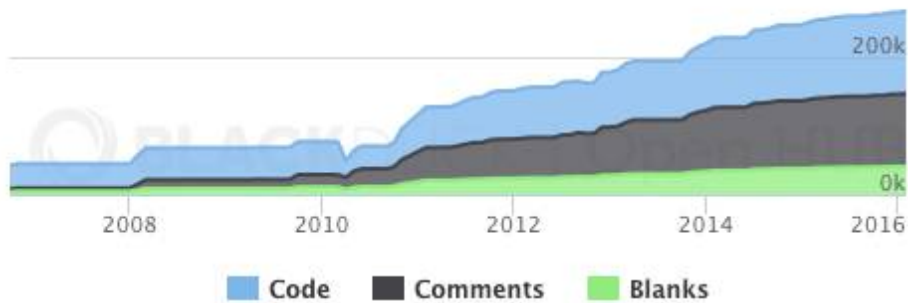
Languages



C++ 92% | 11 Other 8%



Lines of Code



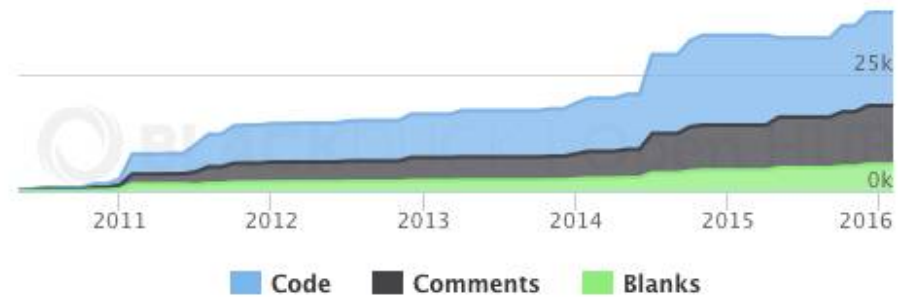
Languages



Python 47% | C++ 38%
Autoconf 5% | 6 Other 10%

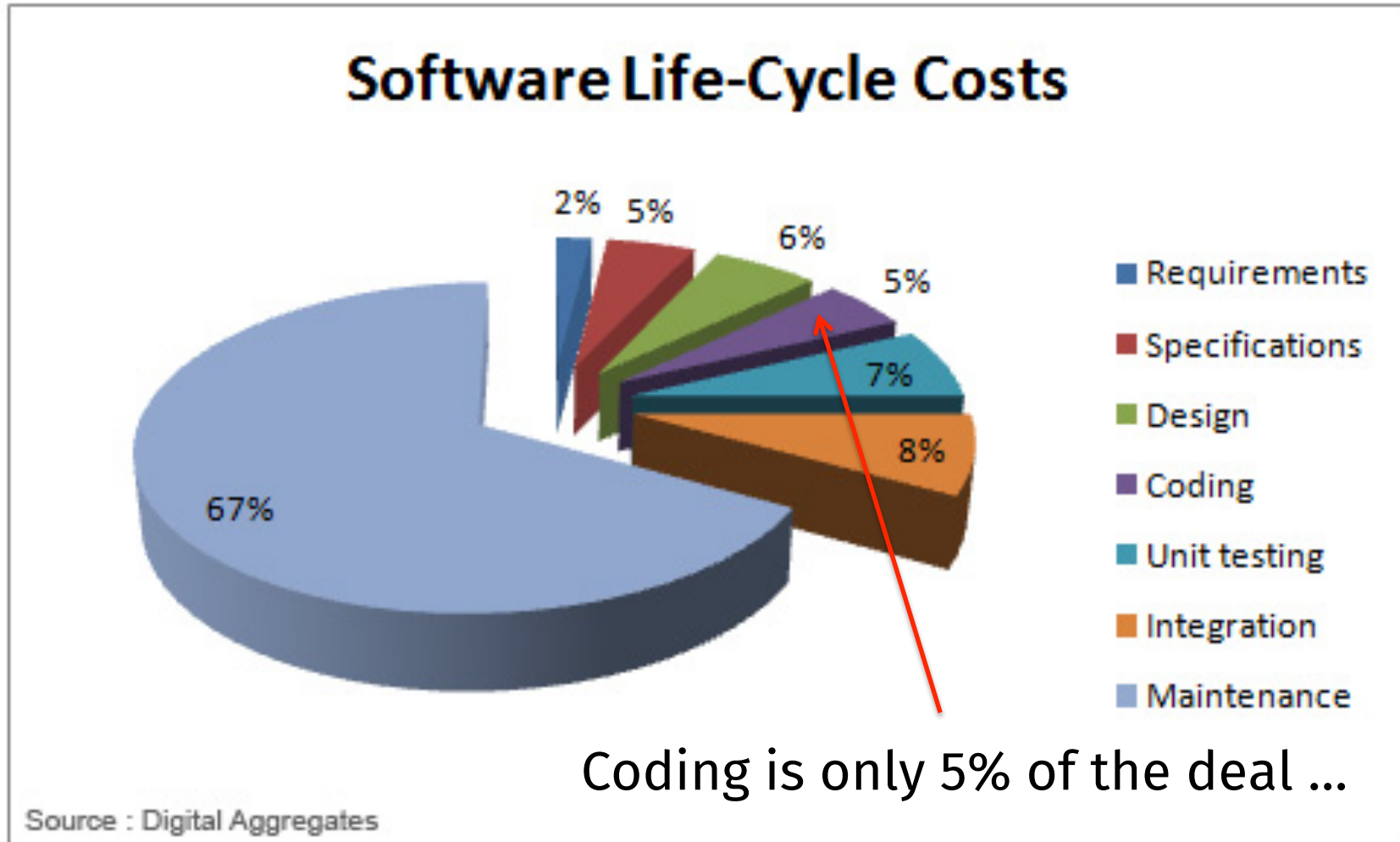


Lines of Code



ctools first language is now Python

Developing sustainable software



Developing sustainable software

Some choices I made:

Define and enforce coding rules (code quality)

Avoid external dependencies (full control over the software)

Do not tie the software to a reference platform

Use continuous integration (building, testing, analysing) and modern code development tools (forge, version control)



Jenkins

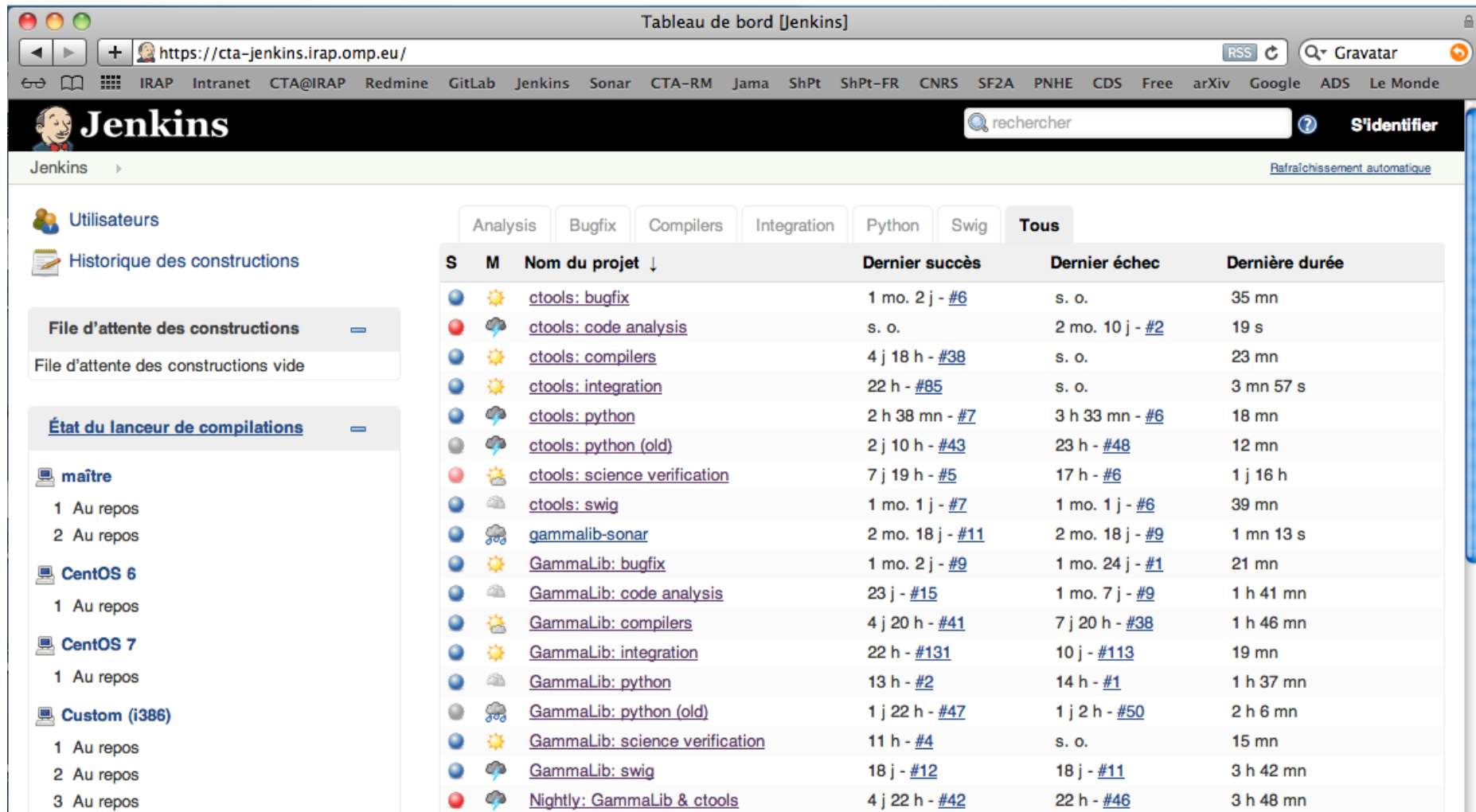


git



Continuous Integration

https://cta-jenkins.irap.omp.eu



The screenshot shows the Jenkins dashboard interface. The browser address bar displays 'https://cta-jenkins.irap.omp.eu/'. The Jenkins logo is visible at the top left, and a search bar is at the top right. The main content area is divided into a left sidebar and a central table of build jobs.

Left Sidebar:

- Utilisateurs
- Historique des constructions
- File d'attente des constructions (vide)
- État du lanceur de compilations
- maître
 - 1 Au repos
 - 2 Au repos
- CentOS 6
 - 1 Au repos
- CentOS 7
 - 1 Au repos
- Custom (i386)
 - 1 Au repos
 - 2 Au repos
 - 3 Au repos

Build Jobs Table:

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée
●	☀	ctools: bugfix	1 mo. 2 j - #6	s. o.	35 mn
●	☁	ctools: code analysis	s. o.	2 mo. 10 j - #2	19 s
●	☀	ctools: compilers	4 j 18 h - #38	s. o.	23 mn
●	☀	ctools: integration	22 h - #85	s. o.	3 mn 57 s
●	☁	ctools: python	2 h 38 mn - #7	3 h 33 mn - #6	18 mn
●	☁	ctools: python (old)	2 j 10 h - #43	23 h - #48	12 mn
●	☀	ctools: science verification	7 j 19 h - #5	17 h - #6	1 j 16 h
●	☁	ctools: swig	1 mo. 1 j - #7	1 mo. 1 j - #6	39 mn
●	☁	gammalib-sonar	2 mo. 18 j - #11	2 mo. 18 j - #9	1 mn 13 s
●	☀	GammaLib: bugfix	1 mo. 2 j - #9	1 mo. 24 j - #1	21 mn
●	☁	GammaLib: code analysis	23 j - #15	1 mo. 7 j - #9	1 h 41 mn
●	☀	GammaLib: compilers	4 j 20 h - #41	7 j 20 h - #38	1 h 46 mn
●	☀	GammaLib: integration	22 h - #131	10 j - #113	19 mn
●	☁	GammaLib: python	13 h - #2	14 h - #1	1 h 37 mn
●	☁	GammaLib: python (old)	1 j 22 h - #47	1 j 2 h - #50	2 h 6 mn
●	☀	GammaLib: science verification	11 h - #4	s. o.	15 mn
●	☁	GammaLib: swig	18 j - #12	18 j - #11	3 h 42 mn
●	☁	Nightly: GammaLib & ctools	4 j 22 h - #42	22 h - #46	3 h 48 mn

Continuous Integration

Multi-platform (Mac OS X, Linux, FreeBSD, OpenSolaris)

gammalib-integrate-os [Jenkins]

https://cta-jenkins.irap.omp.eu/view/Integration/job/gammalib-integrate-os/

Jenkins

rechercher

Jürgen Knödseder | se déconnecter

Jenkins > Integration > GammaLib: integration

Rafraîchissement automatique

Retour au tableau de bord

État

Modifications

Répertoire de travail

Build with Parameters

Supprimer Multi-configuration project

Configurer

Email Template Testing

Log du dernier accès à Git

Historique des builds [tendance](#)

projet GammaLib: integration

Project name: gammalib-integrate-os
Build, check and install GammaLib on various operating systems

[modifier la description](#)

Désactiver le projet

Configurations

[centos6_64](#) [centos7_64](#) [debian6_64](#) [fedora17_64](#) [freebsd9_64](#) [macosx10](#) [macosx11](#)
[macosx7](#) [macosx8](#) [macosx9](#) [mandriva2011_64](#) [opensolaris11_32](#) [opensuse12_64](#)
[sl6_64](#) [ubuntu12_64](#)

[Derniers résultats de test](#) (aucune erreur)

Continuous Integration

Compiler versions

gammalib-compilers [jenkins]

https://cta-jenkins.irap.omp.eu/job/gammalib-compilers/

Jenkins

rechercher

Jürgen Knödseder | se déconnecter

GammaLib: compilers

Retour au tableau de bord

État

Modifications

Répertoire de travail

Build with Parameters

Supprimer Multi-configuration project

Configurer

Email Template Testing

Historique des builds [tendance](#)

find x

#41 23 févr. 2016 03:38

#40 22 févr. 2016 03:38

#39 21 févr. 2016 03:59

#38 20 févr. 2016 03:38

#37 19 févr. 2016 03:39

#36 18 févr. 2016 03:58

#35 15 févr. 2016 03:40

#34 14 févr. 2016 03:38

#33 13 févr. 2016 04:00

projet GammaLib: compilers

Project name: gammalib-compilers
Build GammaLib using different compilers

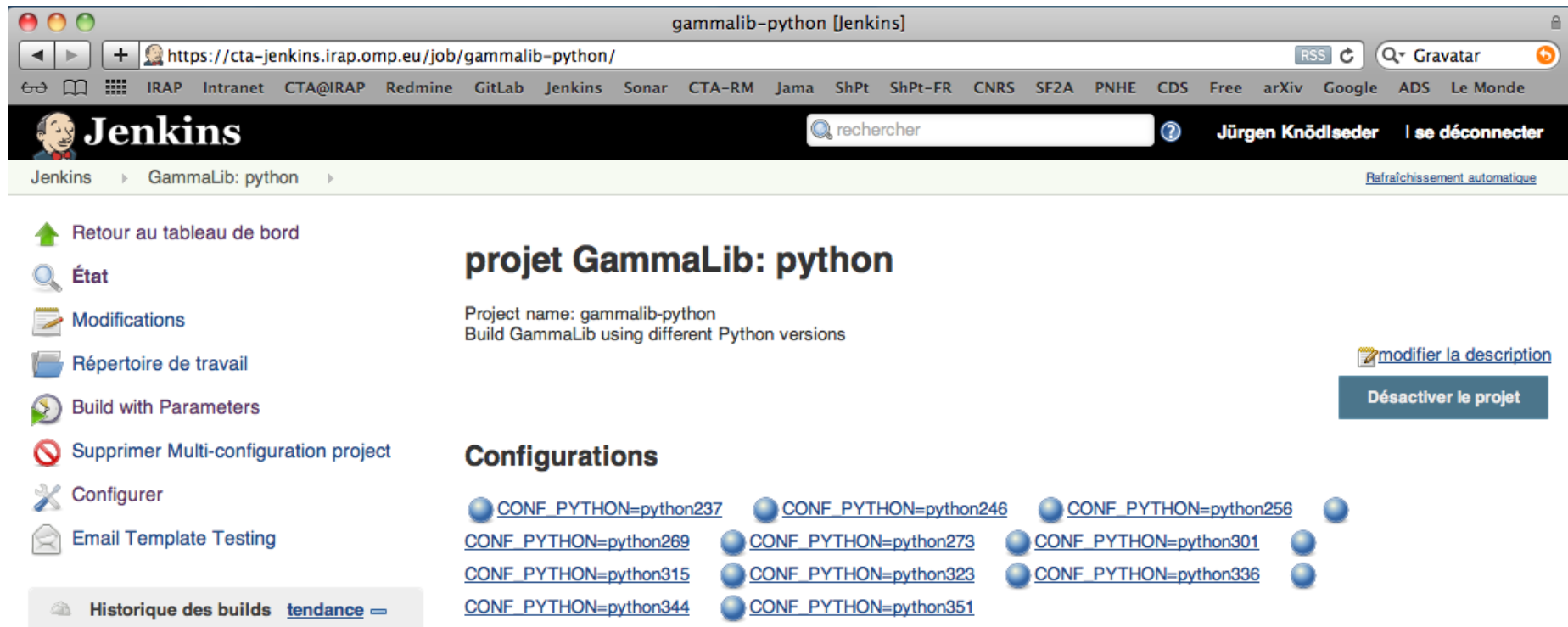
[modifier la description](#)

Désactiver le projet

Matrice de Configuration	i386	x86_64
gcc323	●	●
gcc336	●	●
gcc346	●	●
gcc404	●	●
gcc412	●	●
gcc424	●	●
gcc436	●	●
gcc447	●	●
gcc454	●	●
gcc464	●	●
gcc474	●	●
gcc484	●	●
gcc492	●	●
gcc510	●	●
clang31	●	●

Continuous Integration

Python versions



The screenshot shows the Jenkins web interface for the job 'gammalib-python'. The browser address bar shows the URL 'https://cta-jenkins.irap.omp.eu/job/gammalib-python/'. The Jenkins logo and navigation menu are visible at the top. The main content area displays the project name 'projet GammaLib: python' and the build description 'Build GammaLib using different Python versions'. A list of configurations is shown, each with a radio button and a link to the configuration page. The configurations are:

- [CONF_PYTHON=python237](#)
- [CONF_PYTHON=python246](#)
- [CONF_PYTHON=python256](#)
- [CONF_PYTHON=python269](#)
- [CONF_PYTHON=python273](#)
- [CONF_PYTHON=python301](#)
- [CONF_PYTHON=python315](#)
- [CONF_PYTHON=python323](#)
- [CONF_PYTHON=python336](#)
- [CONF_PYTHON=python344](#)
- [CONF_PYTHON=python351](#)

Note that the code works with Python 2.3 – Python 3.5
(remember: we don't know the environment of the user)

Continuous Integration

swig versions

gammalib-swig [Jenkins]

https://cta-jenkins.irap.omp.eu/job/gammalib-swig/

IRAP Intranet CTA@IRAP Redmine GitLab Jenkins Sonar CTA-RM Jama ShPt ShPt-FR CNRS SF2A PNHE CDS Free arXiv Google ADS Le Monde

Jenkins recherche Jürgen Knödseder se déconnecter

Jenkins > GammaLib: swig Rafraîchissement automatique

Retour au tableau de bord
État
Modifications
Répertoire de travail
Build with Parameters
Supprimer Multi-configuration project
Configurer
Email Template Testing

Historique des builds [tendance](#)

find x

- #12 9 févr. 2016 10:58
- #11 9 févr. 2016 10:56
- #10 9 févr. 2016 10:42
- #9 8 févr. 2016 21:47
- #8 5 févr. 2016 01:20
- #7 27 janv. 2016 16:29
- #6 26 janv. 2016 20:38
- #5 18 déc. 2015 11:21

projet GammaLib: swig

Project name: gammalib-swig
Build, check and install GammaLib using different SWIG versions

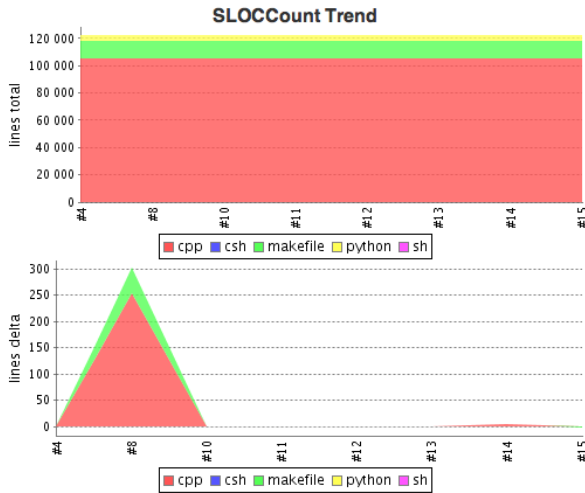
[modifier la description](#)
Désactiver le projet

Matrice de Configuration	python273	python323
swig1340	●	●
swig203	●	●
swig204	●	●
swig205	●	●
swig206	●	●
swig207	●	●
swig208	●	●
swig209	●	●
swig2010	●	●
swig2011	●	●
swig2012	●	●
swig300	●	●
swig301	●	●
swig302	●	●

Continuous Integration

static and dynamic code analysis

lines of code



possible errors

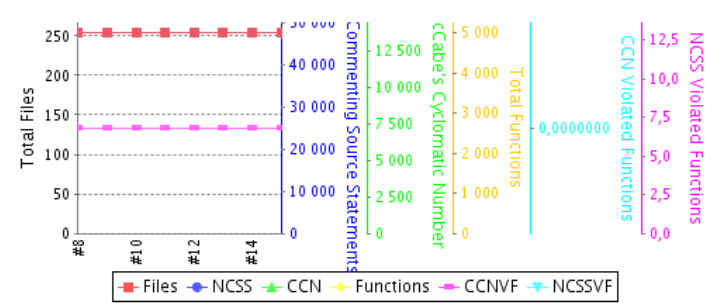


Cppcheck Results

Severity	Count	Delta
Error	1	
Warning	0	
Style	103	
Performance	0	
Portability	0	
Information	0	
No category	0	
Total	104	

code complexity

Cpp NCSS Trend



test coverage

Project Coverage summary

Name	Packages	Files	Classes	Lines	Conditionals
Cobertura Coverage Report	100% 3/3	100% 286/286	100% 286/286	58% 28464/49334	40% 10626/26624

Coverage Breakdown by Package

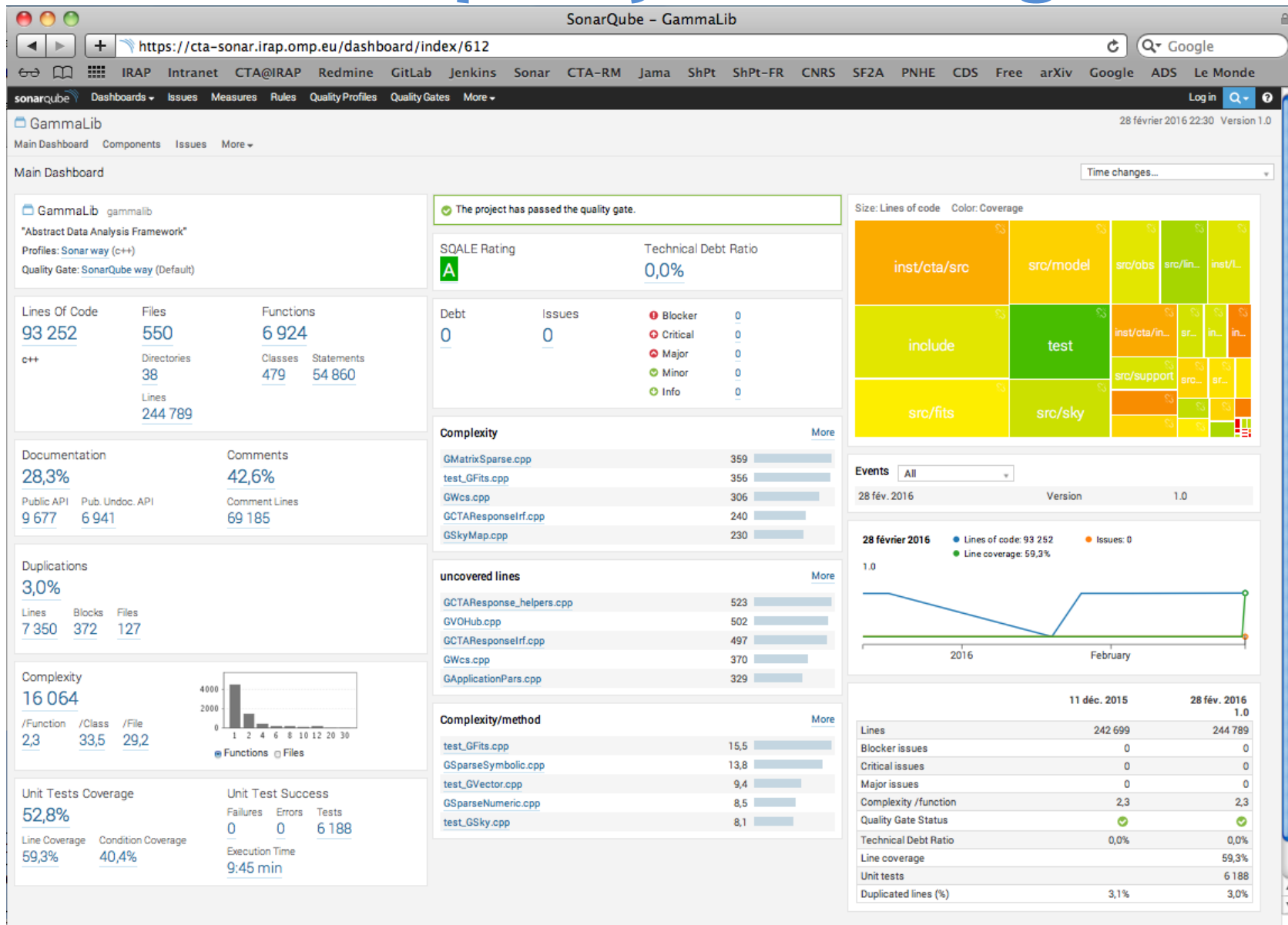
Name	Files	Classes	Lines	Conditionals
<default>	100% 194/194	100% 194/194	63% 22298/35532	45% 8632/19362
src	100% 83/83	100% 83/83	44% 5909/13316	27% 1971/7184
testinst	100% 9/9	100% 9/9	53% 257/486	29% 23/78

memory leaks



Valgrind Results

Code quality monitoring

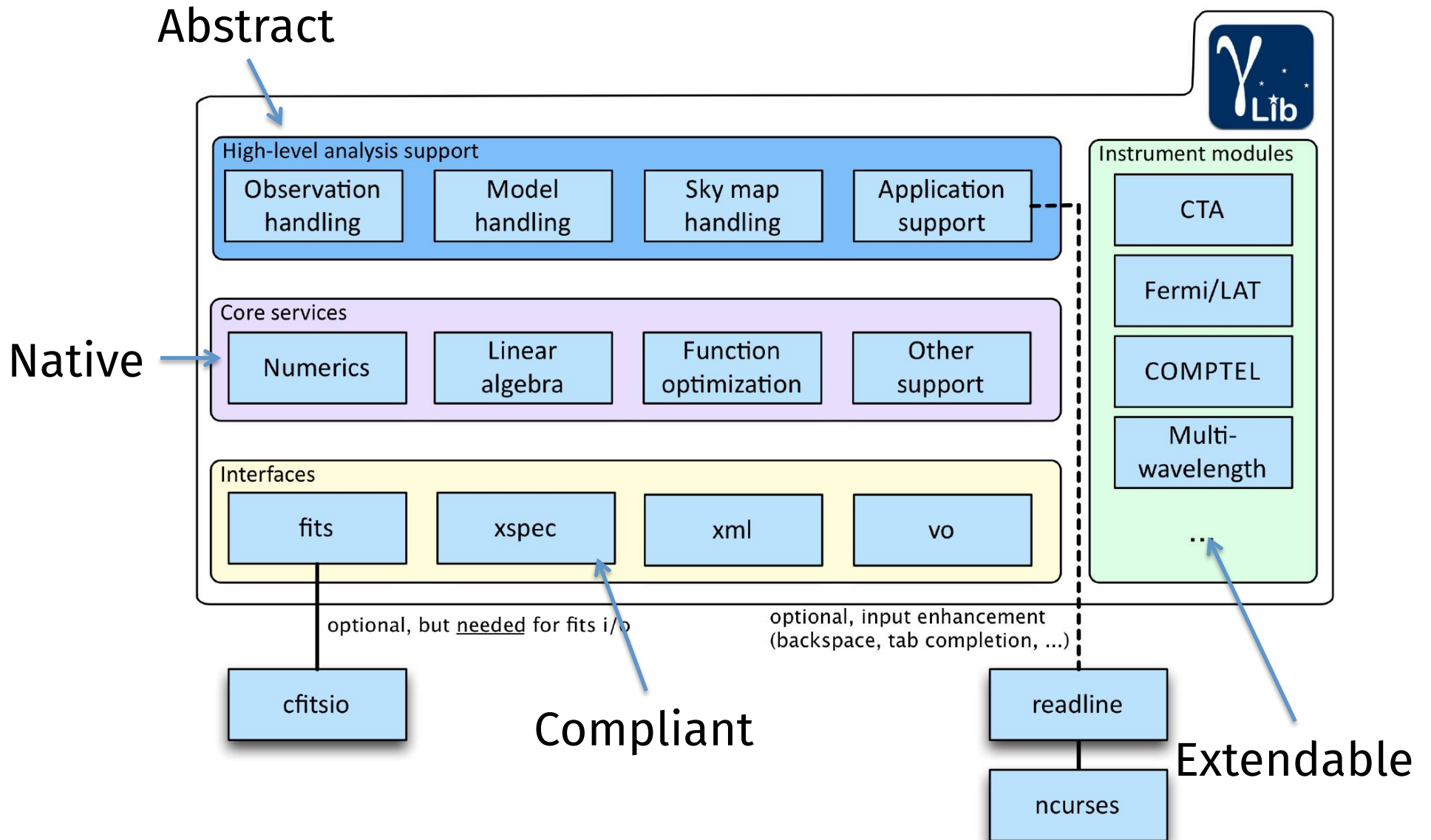


29 February - 4 March 2016

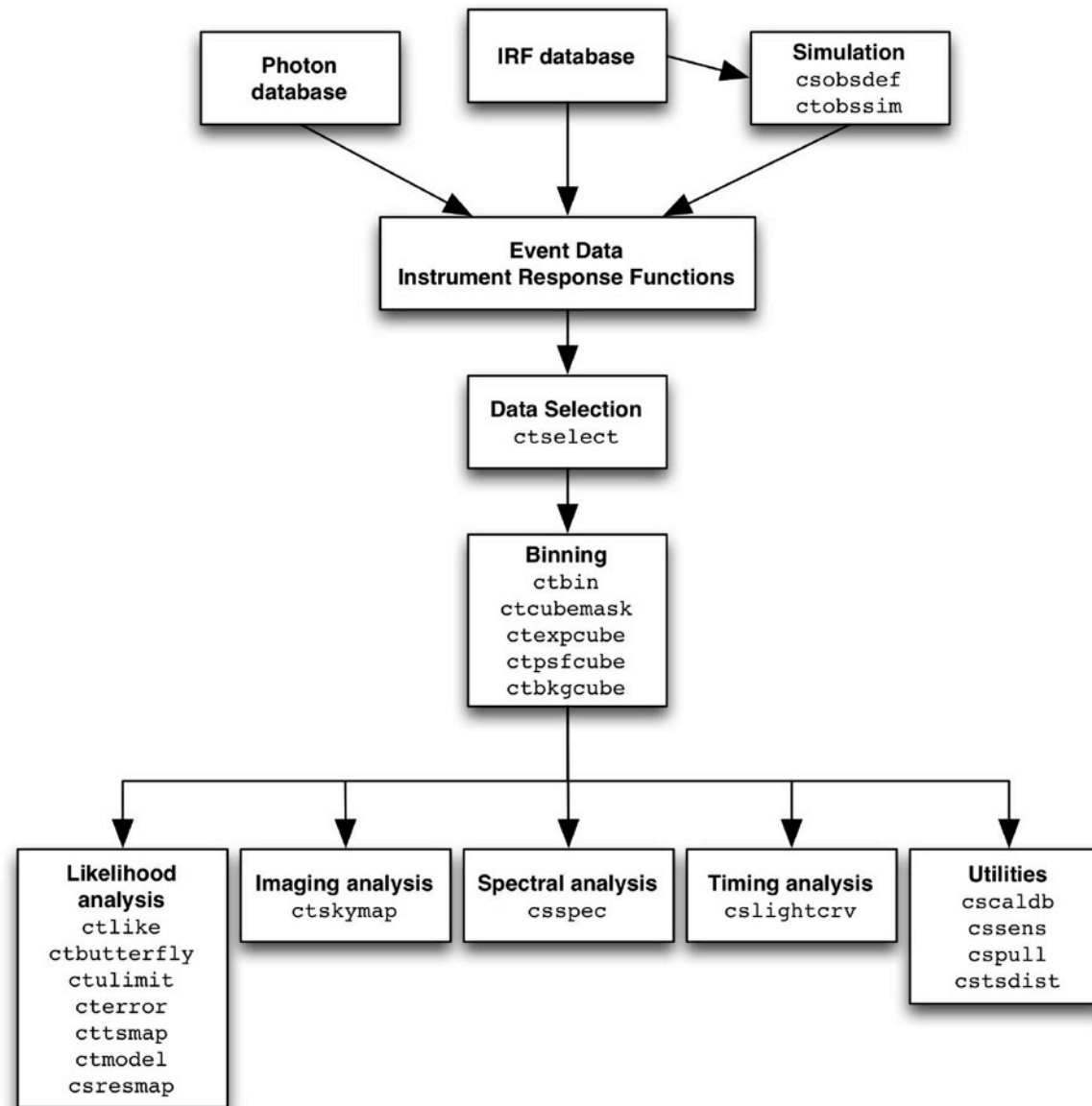
5th ctools and gammalib coding sprint
(Jürgen Knödseder)

17

GammaLib overview



ctools overview

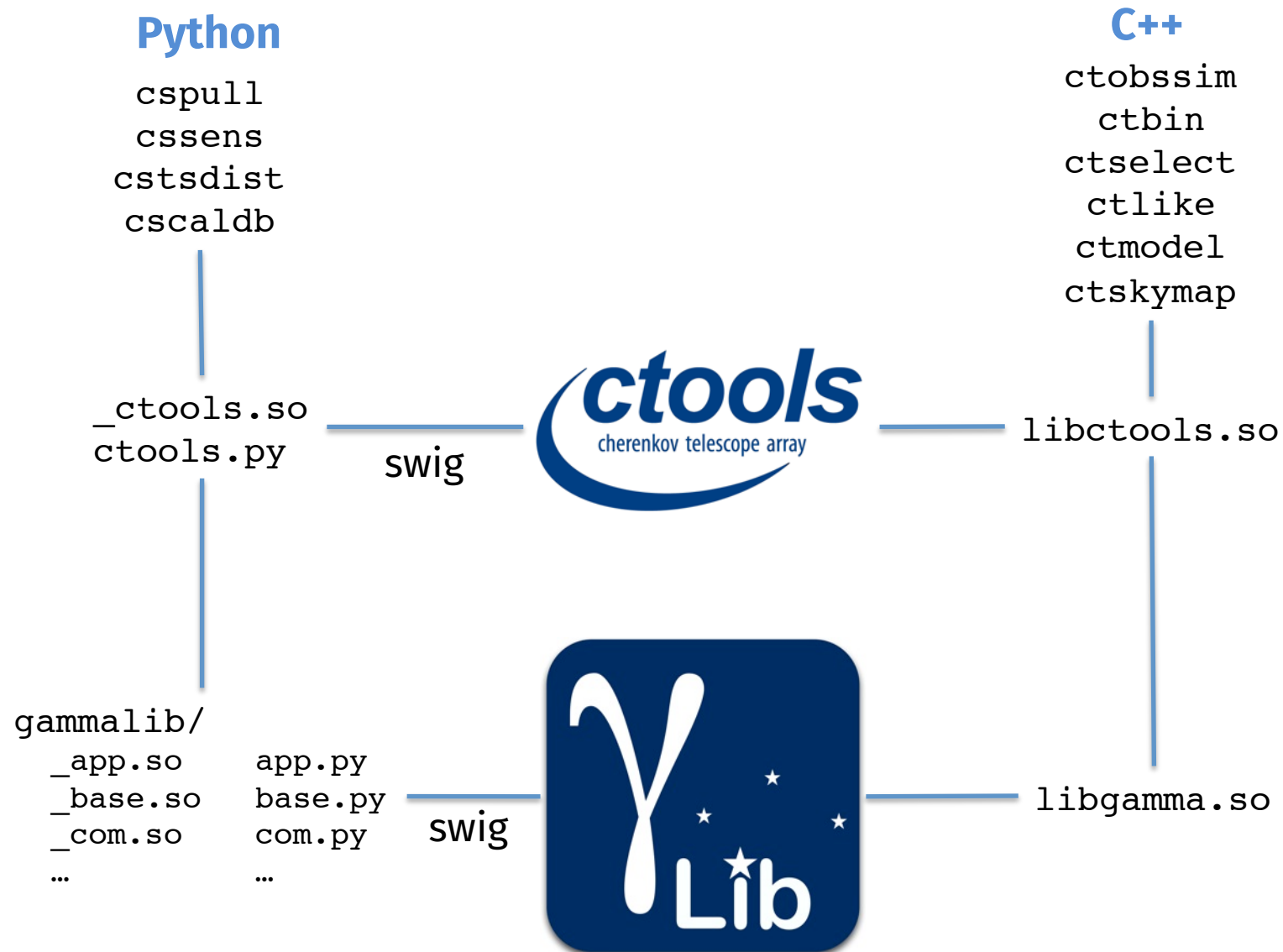


The logo for SWIG, consisting of the letters 'S', 'W', 'I', and 'G' in a white, monospace-style font, centered within a solid black rectangular background.

From <http://www.swig.org/>:

SWIG is a software development tool that connects programs written in C and C++ with a variety of high-level programming languages. SWIG is used with different types of target languages including common scripting languages such as Javascript, Perl, PHP, Python, Tcl and Ruby ...

The overall picture



Getting informed

Sign-in in Redmine, watch anything you're interested in, participate in discussions

Send an e-mail to ctools-subscribe@irap.omp.eu to join the mailing list (news, release info, discussions)

Read the documentation (<http://cta.irap.omp.eu/gammalib/> and <http://cta.irap.omp.eu/ctools/>)

Follow us on Twitter

[@gammalib](#)

[@ctools_software](#)

[#ctools5](#)

Documentation

Code documentation generated using Doxygen

User documentation written using Sphinx

Trunk (devel) documentation

<http://cta.irap.omp.eu/gammalib-devel/>

<http://cta.irap.omp.eu/ctools-devel/>

Release documentation

<http://cta.irap.omp.eu/gammalib/>

<http://cta.irap.omp.eu/ctools/>

Publications

Acknowledgment statement

This research made use of ctools, a community-developed analysis package for Imaging Air Cherenkov Telescope data. ctools is based on GammaLib, a community-developed toolbox for the high-level analysis of astronomical gamma-ray data.

You can cite GammaLib & ctools, it's in the Astrophysics Source Code Library (ASCL) that is indexed by ADS (and soon also Web of Science)

GammaLib: [ascl:1110.007](https://ascl.net/1110.007)

ctools: [ascl:1601.005](https://ascl.net/1601.005)

Release 1.0 paper in advanced stage (all committers are co-authors)

GammaLib and ctools

A software framework for the analysis of astronomical gamma-ray data

J. Knödlse¹, M. Mayer², C. Deil³, J.-B. Cayrou¹, E. Owen³, N. Kelley-Hoskins⁴, C.-C. Lu³, R. Buehler⁴, F. Forest¹, T. Louge¹, H. Siejkowski⁵, K. Kosack⁶, L. Gerard⁴, A. Schulz⁴, P. Martin¹, D. Sanchez⁷, S. Ohm⁴, T. Hassan⁸, and S. Brau-Nogué¹

2. New features since last coding sprint

New GammaLib classes

New cscripts

--help option

User defined HDU names

Virtual Observatory support

New GammaLib classes

GFilename

- Replaces now all string arguments for filenames

- Handles transparently FITS extensions


- Includes methods to check whether a file exists or whether it is a FITS file

CCTAModelAeffBackground

- Uses CTA effective area as background template


New cscripts

csfindobs
csiactcopy
csiactdata
csiactobs
csobs2caldb



existing IACT analysis

csobsinfo
csmodelinfo
csmodelmerge



support scripts

Current IACTs analysis support

csiactcopy

Download IACT data from a remote machine

csiactdata

Explore local IACT database

csfindobs

Creates a run list based on user criteria (pointing direction, zenith angle, etc.)

csiactobs

Creates an observation definition XML file and a model definition XML file containing background models based on a run list

csobs2caldb

Creates a CALDB entry for an observation container (useful for simulations for current IACTs)

Support scripts

csobsinfo

Dumps content of an observation definition XML file

csmodelinfo

Dumps content of a model definition XML file

csmodelmerge

Merge several model definition XML file

Read more on

http://cta.irap.omp.eu/ctools-devel/reference_manual/

--help option

All tools and scripts now accept the `-help` option to display the reference manual text

```
$ ctobssim --help
```

```
ctobssim  
=====
```

```
Simulate event list(s).
```

```
Synopsis  
-----
```

```
This tool simulates event list(s) using the instrument characteristics specified by the instrument response function(s) and an input model. The simulation includes photon events from astrophysical sources and background events from an instrumental background model.
```

```
By default, ctobssim creates a single event list. ctobssim queries a pointing direction, the radius of the simulation region, a time interval, an energy interval, an instrumental response function, and an input model. ctobssim uses a numerical random number generator for the simulations with a seed value provided by the hidden seed parameter. Changing this parameter
```

User defined HDU names

Previously the FITS extension names were hard coded (e.g. “GTI” for Good Time Intervals, “EVENTS” for CTA events list, etc.). Now the user can specify the FITS extension name in the filename

```
gti.save("gti.fits[Good Time Interval]", true)  
obs.save("cta_events.fits[EVENTS3;GTI3]", true)
```



Syntax to specify events and
GTI HDU name

Virtual Observatory Support

From <http://www.ivoa.net/>

The Virtual Observatory (VO) is the vision that astronomical datasets and other resources should work as a seamless whole

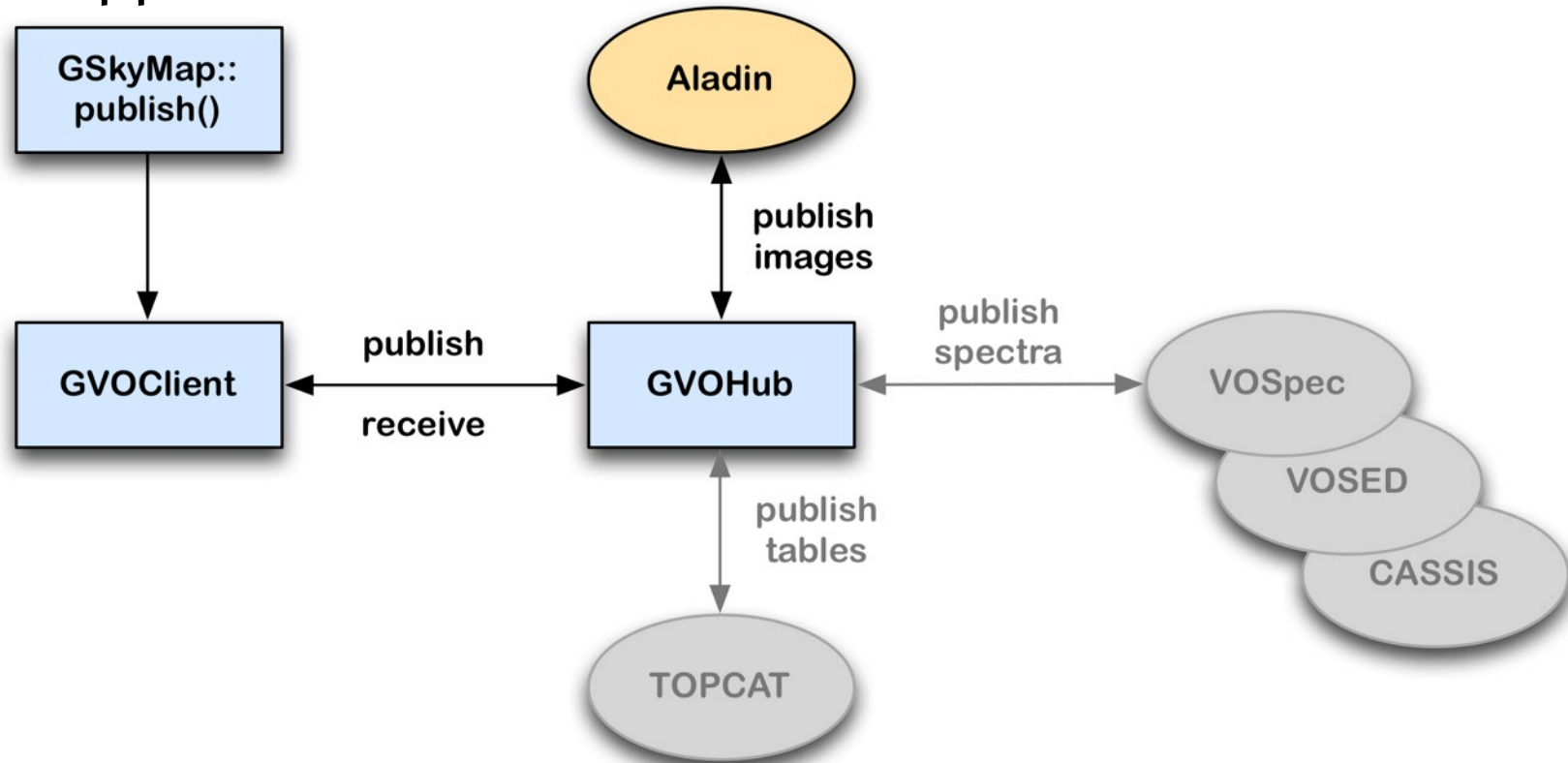
The VO is not a concrete thing, like a data warehouse. Rather, it is more like an ecosystem of **mutually compatible datasets, resources, services, and software tools which use a common set of technologies and a common set of standards**. The idea is to make all these things *inter-operable* - i.e. to make them work nicely together.

From myself

The VO is great if you are not aware that you are using it

Virtual Observatory Support

VO support in a nutshell

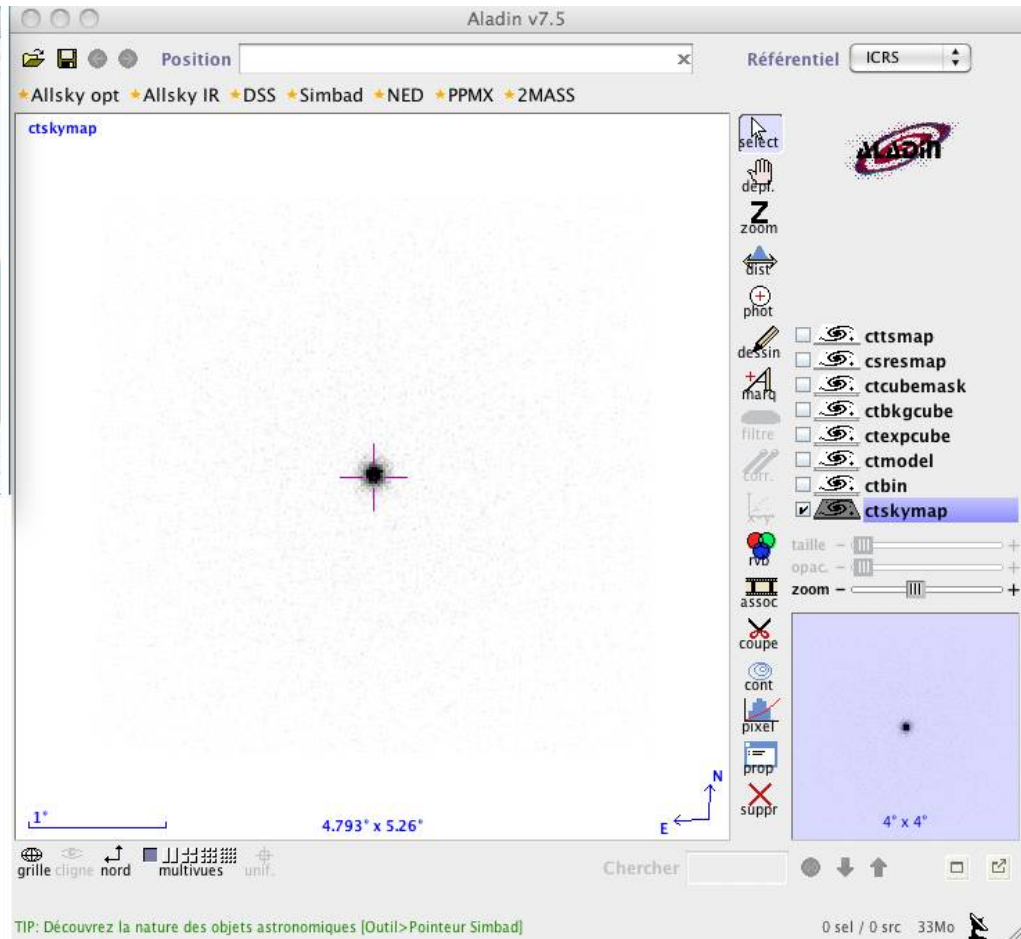


All ctools generating images have now a hidden publish parameter, e.g.

```
$ ctskymap publish=yes
```

Virtual Observatory Support

```
Terminal — bash — 82x23
macp0026:tmp jurgen$ ctskymap publish=yes
Input event list or observation definition XML file [events.fits]
First coordinate of image center in degrees (RA or galactic l) (0-360) [83.63]
Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [22.01]
Projection method (AIT|AZP|CAR|MER|MOL|STG|TAN) [CAR]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [CEL]
Image scale (in degrees/pixel) [0.02]
Size of the X axis in pixels [200]
Size of the Y axis in pixels [200]
Lower energy limit (TeV) [0.1]
Upper energy limit (TeV) [100.0]
Output skymap file [skymap.fits]
macp0026:tmp jurgen$
```

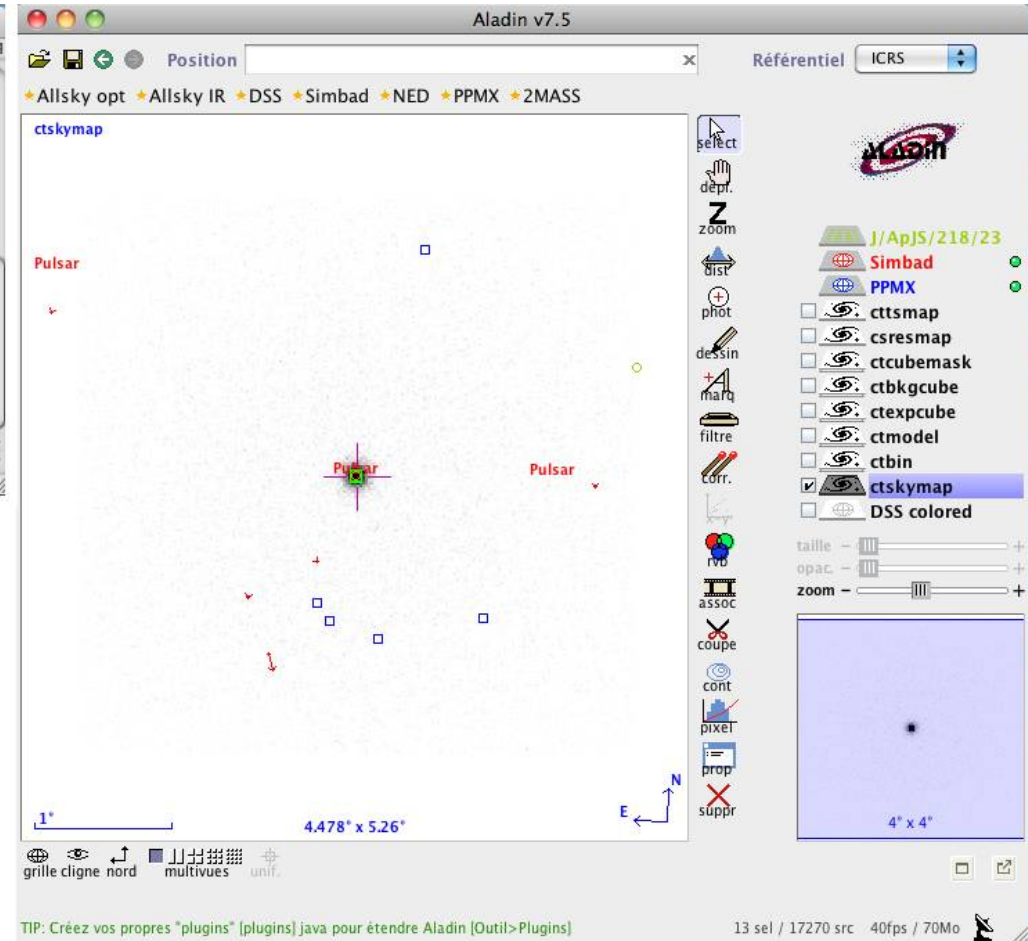


ctskymap sends a sky map via the Hub to Aladin (open Aladin before sending the sky map)

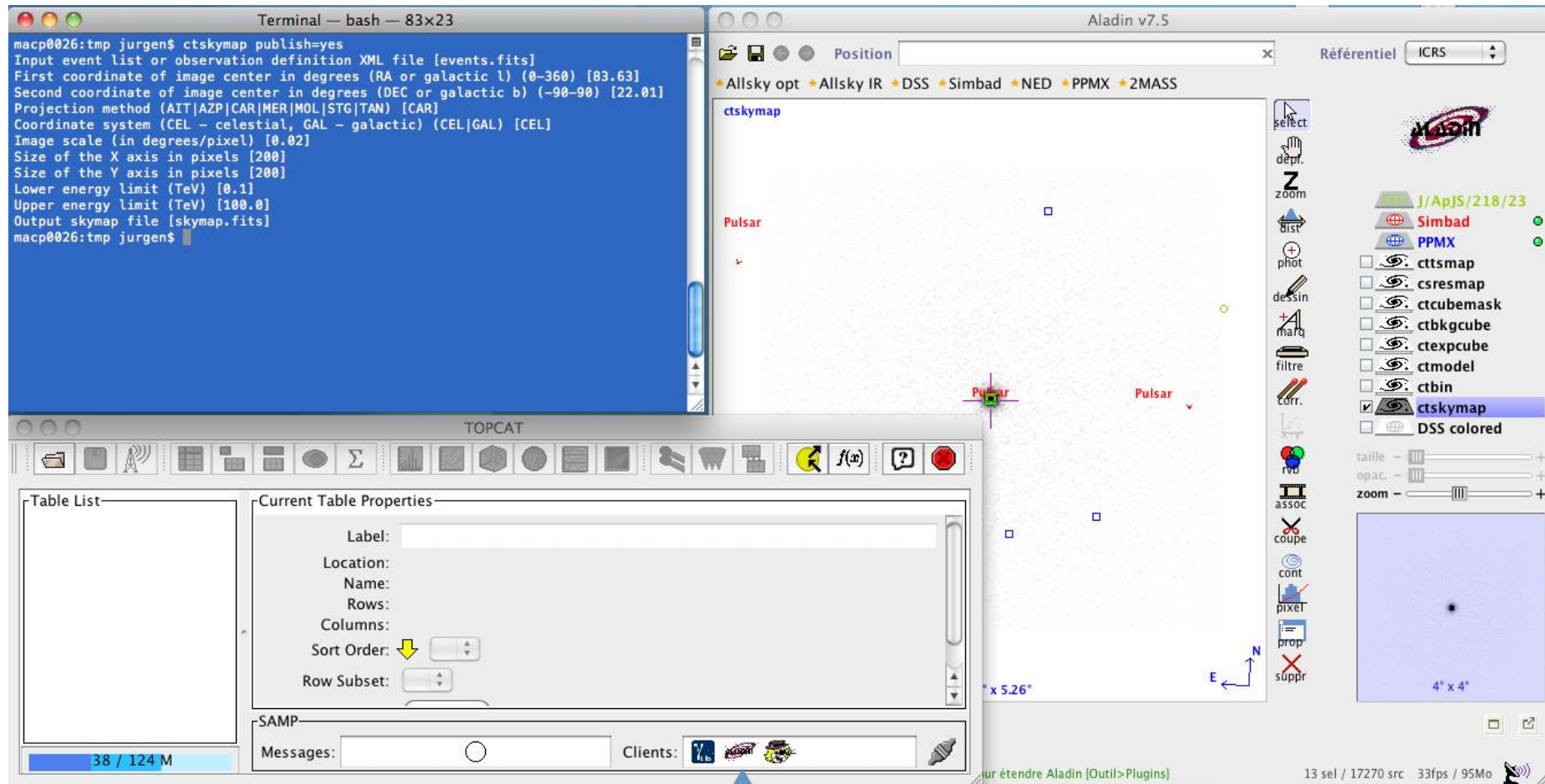
Virtual Observatory Support

```
Terminal — bash — 82x23
macp0026:tmp jurgen$ ctskymap publish=yes
Input event list or observation definition XML file [events.fits]
First coordinate of image center in degrees (RA or galactic l) (0-360) [03.63]
Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [22.01]
Projection method (AIT|AZP|CAR|MER|MOL|STG|TAN) [CAR]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [CEL]
Image scale (in degrees/pixel) [0.02]
Size of the X axis in pixels [200]
Size of the Y axis in pixels [200]
Lower energy limit (TeV) [0.1]
Upper energy limit (TeV) [100.0]
Output skymap file [skymap.fits]
macp0026:tmp jurgen$
```

Aladin can be used to search for catalogues, sky maps obtained at other wavelengths, etc.



Virtual Observatory Support



TOPCAT shows who's connected to the Hub (GVOHub)

Virtual Observatory Support

Next steps

publishing of spectra

publishing of table (e.g. events lists)

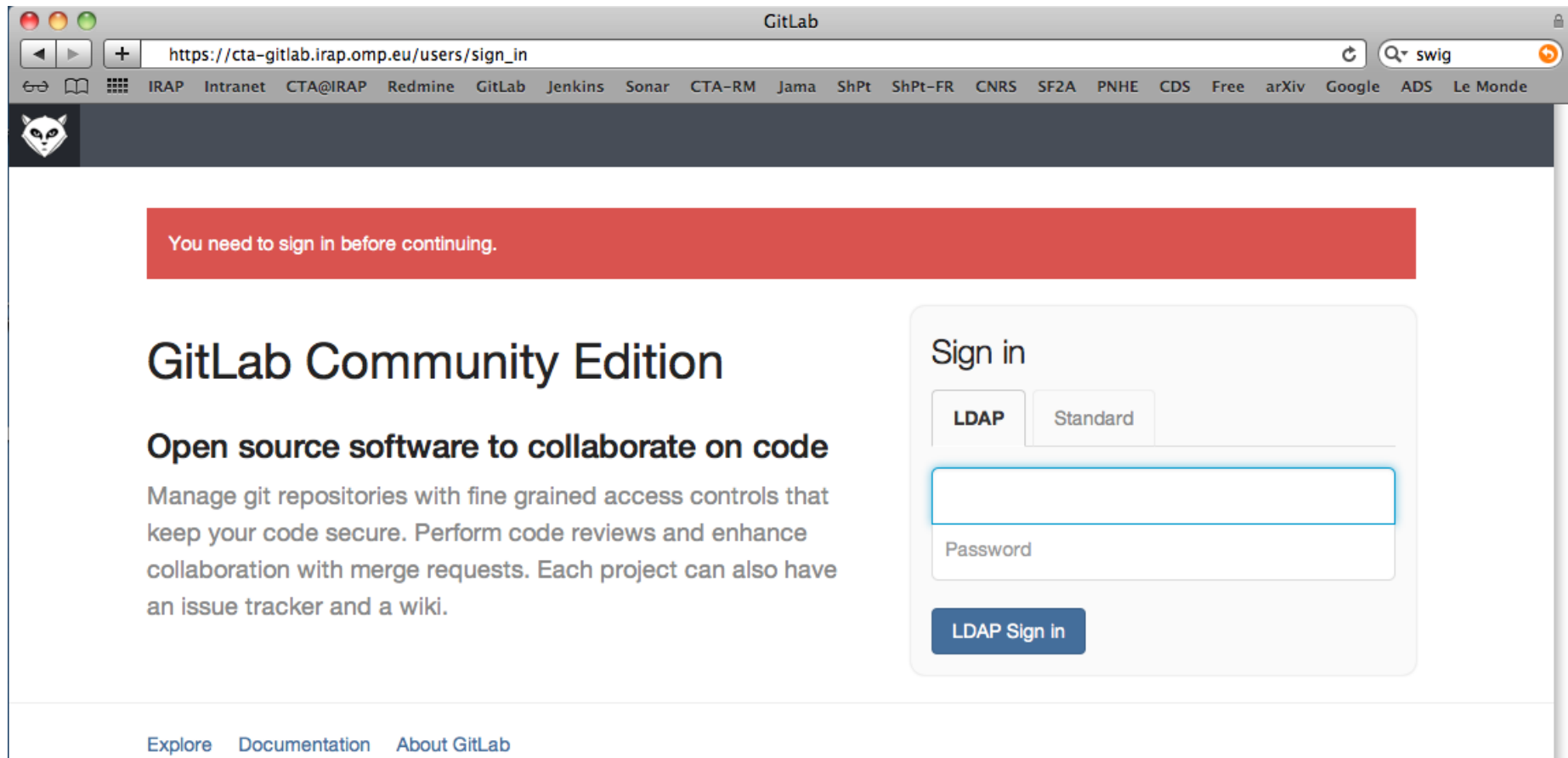
feedback of VO tools to ctools (e.g. source selection)

Note: VO support is still exploratory, but you can now try it out and provide your feedback

3. New Development Workflow

GitLab

https://cta-gitlab.irap.omp.eu



The screenshot shows a web browser window with the URL `https://cta-gitlab.irap.omp.eu/users/sign_in`. The page features a red banner at the top stating "You need to sign in before continuing." Below this, the text "GitLab Community Edition" is displayed, followed by the tagline "Open source software to collaborate on code" and a brief description of the platform's capabilities. On the right side, there is a "Sign in" form with two tabs: "LDAP" (selected) and "Standard". The form includes a text input field for the username and a "Password" label above another text input field. A blue button labeled "LDAP Sign in" is positioned below the password field. At the bottom of the page, there are links for "Explore", "Documentation", and "About GitLab".

A web based Git front-end (à la GitHub)

Redmine

https://cta-redmine.irap.omp.eu

CTA IRAP Project Gateway

Accueil Projets Aide

Recherche:

Accueil

Dernières annonces

ctools: ctools bug fix 1.0.1 release
A bug fix for ctools version 1.0 has been released.
Ajouté par Knödseder Jürgen il y a [29 jours](#)

GammaLib: GammaLib bug fix 1.0.1 release
A bug fix for GammaLib version 1.0 has been released.
Ajouté par Knödseder Jürgen il y a [29 jours](#)


ctools: ctools version 1.0.0 release
The first stable version of ctools has been released.
Ajouté par Knödseder Jürgen il y a [2 mois](#)

GammaLib: GammaLib version 1.0.0 release
The first stable version of GammaLib has been released.
Ajouté par Knödseder Jürgen il y a [2 mois](#)

ctools: ctools version 0.10.0 release
ctools version 0.10.0 has been released. This version is feature complete with respect to the planned 1.0 release.
Ajouté par Knödseder Jürgen il y a [4 mois](#)

[Voir toutes les annonces](#)

Derniers projets

- Clusters of Galaxies (01/08/2012 16:00)
- Primeval_Universe (01/08/2012 15:59)
- ctools (20/02/2012 12:08)
Development of ctools, a ftools-based suite of executables for the scientific analysis of CTA data.
 - User forum
 - Developer forum
- GammaLib (15/02/2012 22:14)

Development of GammaLib toolbox for high-level analysis of astronomical gamma-ray data.
 - User forum
 - Developer forum...

Handles account creation (also for GitLab)

29 February - 4 March 2016

5th ctools and gammalib coding sprint
(Jürgen Knödseder)

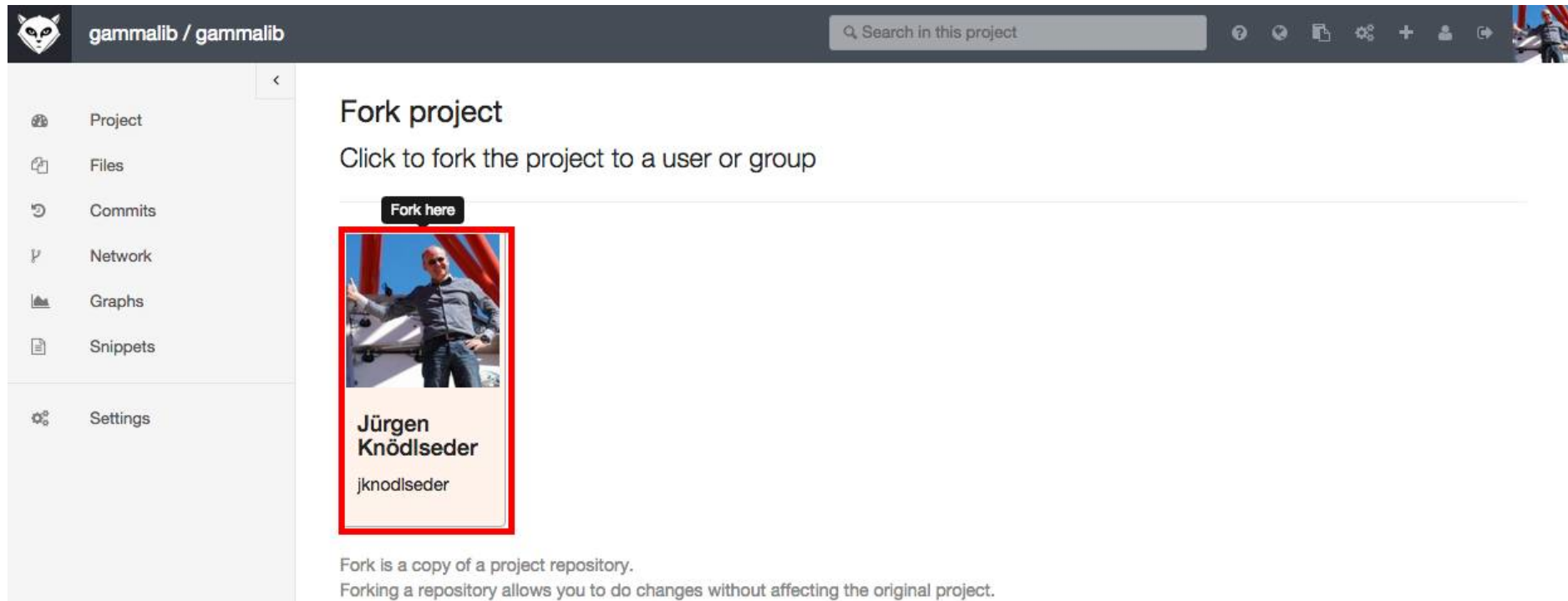


Creating a fork on GitLab

The screenshot shows the GitLab interface for the 'gammalib / gammalib' project. The left sidebar contains navigation options: Project, Files, Commits, Network, Graphs, Snippets, and Settings. The main content area displays the project name, a search bar, and a notification: 'Profile was successfully updated'. Below this, the project title is 'Toolbox for high-level analysis of astronomical gamma-ray data – Edit'. The repository is public, and the 'Fork' button is highlighted with a red box. The repository has 1 star and 0 forks. The SSH and HTTPS URLs are provided, along with the repository size (110.89 MB) and commit/branch/tag counts (5,414 commits, 20 branches, 27 tags). The 'Activity' tab is selected, showing a recent event: 'gammalib imported project gammalib / gammalib' 39 minutes ago. The 'Compare code' and 'Download zip' buttons are also visible.

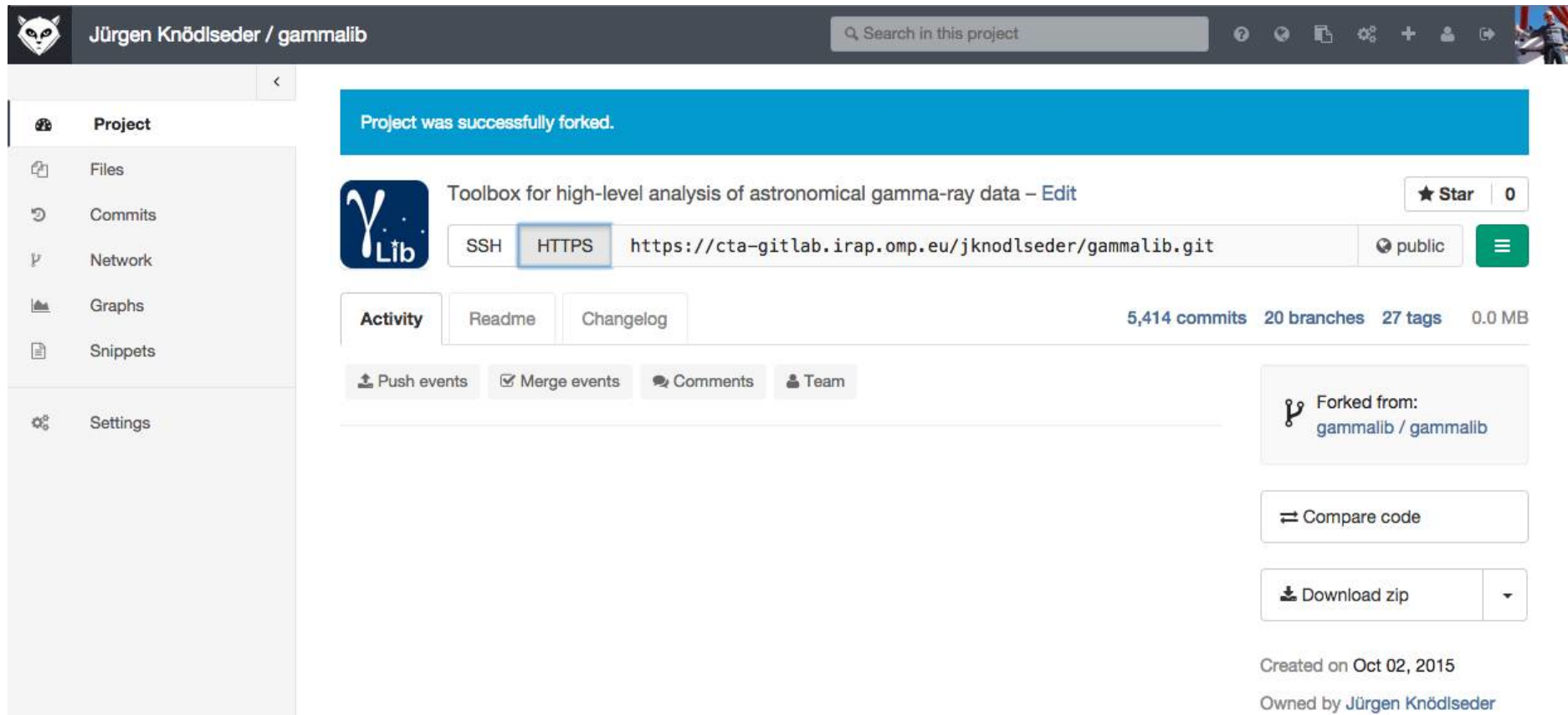
Connect to GitLab with your Redmine user name and password and select the GammaLib or ctools project and click on “Fork”

Creating a fork on GitLab



Now click on your user to fork (copy) the project into your user space (will result in <user>/gammalib)

Creating a fork on GitLab



The screenshot shows the GitLab interface for a project named 'gammalib' by user 'Jürgen Knödseder'. A blue banner at the top of the project page states 'Project was successfully forked.' Below this, the project title is 'Toolbox for high-level analysis of astronomical gamma-ray data - Edit'. The repository is public and has 0 stars. The HTTPS URL is 'https://cta-gitlab.irap.omp.eu/jknodlseder/gammalib.git'. The project has 5,414 commits, 20 branches, 27 tags, and 0.0 MB of size. On the right side, it indicates the project was forked from 'gammalib / gammalib' and was created on Oct 02, 2015, owned by Jürgen Knödseder. The left sidebar shows navigation options like Project, Files, Commits, Network, Graphs, Snippets, and Settings.

Now you have a copy of the project under your user on GitLab

Working with GitLab

Make sure you can access https

```
$ export GIT_SSL_NO_VERIFY=true
```

or

```
$ git config --global http.sslverify "false"
```

Get a clone of the code on your machine

```
$ git clone https://cta-gitlab.irap.omp.eu/jknodlseder/gammalib.git
```

```
$ git init
```

Add main gammalib (or ctools) repository as remote repository to fetch any changes before starting to work on a new feature

```
$ git remote add upstream https://cta-gitlab.irap.omp.eu/gammalib/gammalib.git
```

```
$ git pull upstream devel
```

Contributing code

Create a new branch. Always pull upstream changes first. Always put the Redmine issue number at the beginning of your branch.

```
$ git pull upstream devel
```

```
$ git checkout -b 9101-correct-nasty-bug
```

Code, stage and commit

```
$ git add inst/cta/src/GCTAModellrfBackground.cpp
```

```
$ git commit -m "Fixed the nasty bug (#9101)"
```

Push commit(s) into your GitLab repository

```
$ git push origin 9101-correct-nasty-bug
```

Contributing code

The screenshot shows the GitLab interface for the project 'gammalib' by Jürgen Knödseder. The project title is 'Toolbox for high-level analysis of astronomical gamma-ray data'. The repository is public and has 5,414 commits, 21 branches, 27 tags, and 110.95 MB of code. A recent commit by Jürgen Knödseder is shown, titled '9101-correct-nasty-bug at Jürgen Kn...' and 'Fixed the nasty bug (#9101)'. The commit hash is e0208913. The interface includes a sidebar with navigation options like Project, Files, Commits, Network, Graphs, Snippets, and Settings. There are also buttons for 'Push events', 'Merge events', 'Comments', and 'Team'. On the right, there are options to 'Forked from: gammalib / gammalib', 'Compare code', and 'Download zip'. The project was created on Oct 02, 2015, and is owned by Jürgen Knödseder.

You should see your commit now on GitLab ...

Now go to the Redmine issue to tell me that there is something to merge into the trunk ...

Contributing code

Update

Update Log time Watch Duplicate Copy Move Delete

Change properties (More)

Status New In Progress Resolved Feedback Pull request Closed Rejected

Priority * Low

Assigned To

Target version

Parent task

Start date 2012-12-06

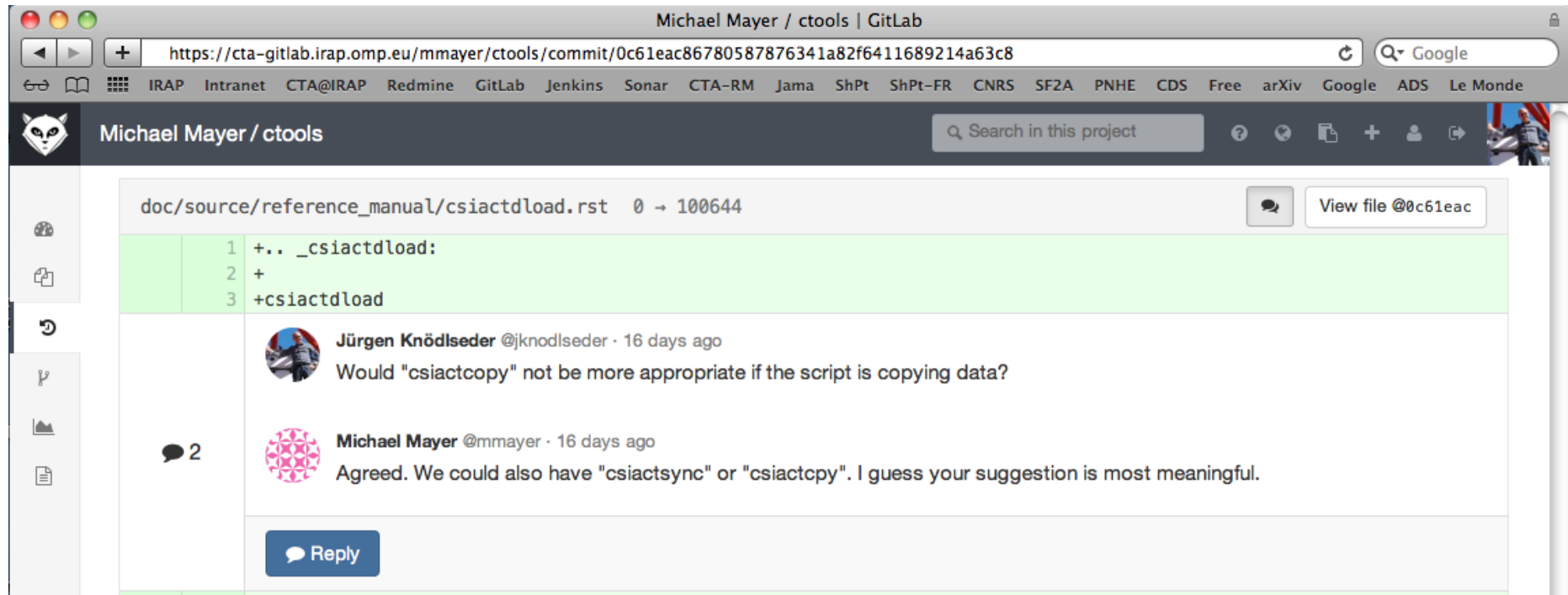
Due date

Estimated time Hours

% Done 100 %

Please always explain in the Redmine issue what you did and what the name of the branch is that you want to get merged

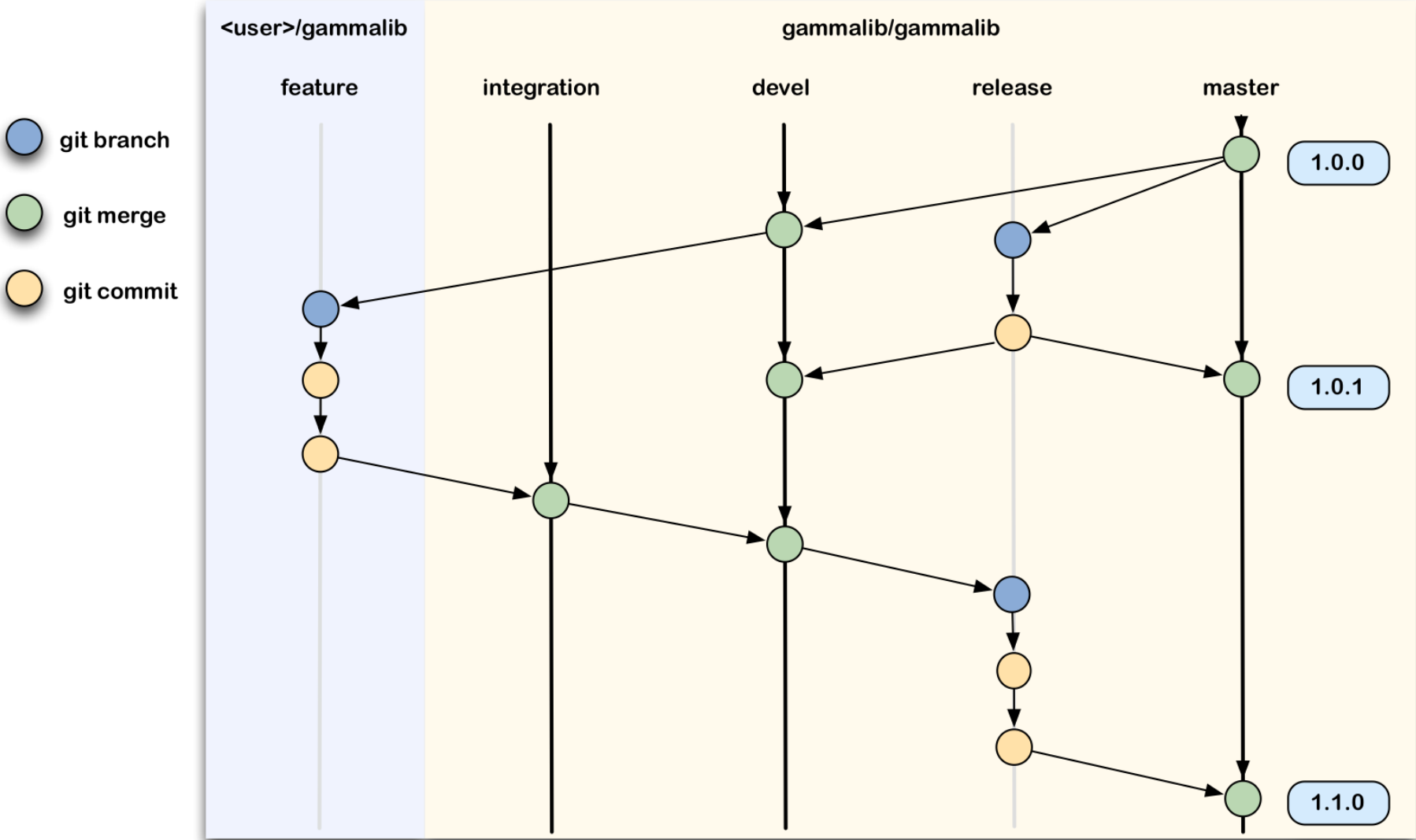
Code review on GitLab



The screenshot shows a web browser window displaying a GitLab pull request. The browser's address bar shows the URL: `https://cta-gitlab.irap.omp.eu/mmayer/ctools/commit/0c61eac86780587876341a82f6411689214a63c8`. The page title is "Michael Mayer / ctools | GitLab". The main content area shows a file named `doc/source/reference_manual/csiactdload.rst` with a commit range of `0 -> 100644`. The file content is highlighted in green and shows three lines of code: `1 +.. _csiactdload:`, `2 +`, and `3 +csiactdload`. Below the code, there is a comment from Jürgen Knödseder (@jknodseder) posted 16 days ago, asking: "Would 'csiactcopy' not be more appropriate if the script is copying data?". A reply from Michael Mayer (@mmayer) is shown below, dated 16 days ago, stating: "Agreed. We could also have 'csiactsync' or 'csiactcopy'. I guess your suggestion is most meaningful." A "Reply" button is visible at the bottom of the comment thread.

Once you issued a pull request, I will inspect your code on GitLab and eventually make some comments (you should get an e-mail notification of that). You can of course reply and engage a discussion 😊

Git workflow



4. Goals of this sprint

Goals of this sprint

Collection of issues to be addressed during the sprint

I hope by the time of the coding sprint we have a good draft of our release paper ready. We may work on analysis examples for the release paper. But we can also address new features needed and discuss the next steps.

Just list below what you would like to do during the sprint:

- Have fun
- Implement analysis workflows (#1508, see also <https://cta-redmine.irap.omp.eu/boards/14/topics/237>)
- Python Function to Convert GObservations to GCTABackground3D (#1530)
- Implement a PSF table format
- Finalize the classical analysis (I know, we have this pending since a long time, but we should terminate the work which is almost done)
- Implement smoothing, oversampling and denoising of images (and skymaps), mentioned in (#1530)
- tool to compute systematic errors (#1712)
- Finish a GModelSpatialRadial for dark matter halos (#1520)
- ...

Workflow implementation

To make a proof of principle for an XML-driven analysis workflow I pushed a branch `1508-implement-workflow` into the `ctools` repository. It contains a script `esworkflow.py` to execute a workflow described by the `test/data/workflow.xml` file.

The format of the XML file is:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<workflow>
  <actor name="input">
    <output>
      <parameter name="inmodel" value="$CTOOLS/share/models/crab.xml" />
      <parameter name="caldb" value="prod2" />
      <parameter name="irf" value="South_50h" />
      <parameter name="edisp" value="no" />
      <parameter name="deadc" value="0.95" />
      <parameter name="ra" value="83.63" />
      <parameter name="dec" value="22.01" />
      <parameter name="target" value="Crab" />
      <parameter name="emin" value="0.1" />
      <parameter name="emax" value="100.0" />
      <parameter name="tmin" value="0.0" />
      <parameter name="tmax" value="1800.0" />
      <parameter name="chatter" value="2" />
      <parameter name="clobber" value="yes" />
      <parameter name="debug" value="no" />
      <parameter name="mode" value="ql" />
    </output>
  </actor>
  <actor name="ctobssim" tool="ctobssim">
    <input>
      <parameter name="inobs" value="NONE" />
      <parameter name="inmodel" value="inmodel" actor="input" />
      <parameter name="caldb" value="caldb" actor="input" />
      <parameter name="irf" value="irf" actor="input" />
      <parameter name="edisp" value="edisp" actor="input" />
      <parameter name="prefix" value="sim_events_" />
      <parameter name="seed" value="1" />
      ...
    </input>
  </actor>
</workflow>
```

The file contains a single `<workflow>` element that is composed by a number of `<actor>` elements. Each `<actor>` has optional `<input>` and `<output>` elements. The syntax of using an output element of an actor as input of another actor is:

```
<parameter name="inmodel" value="inmodel" actor="input" />
```

where `value` is the name of the parameter of the actor, and `actor` is the name of the actor.

Agenda

Tentative agenda

- Monday, 29 February:
 - 14:00 – 16:00: Introduction, meeting goal, status of CTA developments & analysis (Jürgen)
 - 16:00 – 16:30: HESS analysis progress report (Michael)
 - 16:30 – 17:00: VERITAS analysis progress report (Nathan)
 - 17:00 – 18:00: Slots for more progress reports, analysis results, etc. (just enter your proposal)
- Tuesday, 1 March:
 - 9:00–18:00: Coding, Testing, Documenting
- Wednesday, 2 March:
 - 9:00–18:00: Coding, Testing, Documenting
- Thursday, 3 March:
 - 9:00–18:00: Coding, Testing, Documenting
- Friday, 4 March:
 - 9:00 – 12:00: Sprint wrap up

What day is best to organise a social dinner (add your wishes or thoughts)?