A Taxonomy of Scientific Software Applications

Peter Elmer - Princeton University

HEP's Place in the World

- Software and Computing in HEP has been, and still is, a major enterprise
- Participation by a cast of thousands: physicists, software and computing professionals
- Large international conferences (CHEP, ACAT, IGSC, HEPiX, many other smaller conferences/workshops). Even some of our software tools have their own workshops.

HEP's Place in the World - practical questions

- This is a small hobby project. For entirely *pragmatic* reasons I am primarily interested in:
- Understanding better what other scientific software projects look like, and how they get adopted by others and evolve.
- Identifying places where we have something to offer the world (and why we do or don't when we do have something) and when others might have something to offer us. Where can we potentially collaborate?
- Understanding better how and when projects succeed in "leaving the ghetto" of a particular group, experiment or the field is potentially interesting.

Distributed Computing (The Grid)

- I am not going to spend much time on Grid Computing here.
- It is obviously an area where we attracted a lot of attention (and funding), built an amazing worldwide system to do what we needed to do, with no other realistic solutions. We also played a role in enabling other sciences to do things they could not otherwise have done.
- Panel discussion on Thursday "*The end of HEP-specific Computing as we know it?*" Hint of my attitude: nobody is worried today that we aren't all still using CERN httpd.

SLOCCount

- Simple tool to measure "size" of a software package in Source Lines of Code (SLOC)
- http://www.dwheeler.com/sloccount/
- SLOC is not a measure of "value", e.g. zlib/deflate is only 29k SLOC, however it has been of great importance to us
- SLOC is however a crude estimator of the effort required for the initial creation of some piece of software. (Subsequent maintenance, non-technical "physics tuning", etc. are probably not well measured by the number.)

An aside: Scientific Software Production, Cyberinfrastructure, Ecosystems

- While my aims are actually very pragmatic, while looking into this I see that "Scientific Software Production" is actually an academic area of research. See, for example, James Howison (http://howison.name/) and follow references from there. For example:

- http://howison.name/pubs/IncentivesAndIntegration-p459-howison.pdf

- See also (another aside): "When Authorship Isn’t Enough: Lessons from CERN on the Implications of Formal and Informal Credit Attribution Mechanisms in Collaborative Research ", J. Birnholtz

- http://quod.lib.umich.edu/j/jep/3336451.0011.105?rgn=main;view=fulltext

What is HEP software?

- Before I start looking at other fields I looked at the software we build and use.

- I used CMS as a concrete example, and (mostly) focus on the *software* applications, *not* "computing" in the sense of data and workflow management tools.

- What kinds of things are we using and building?

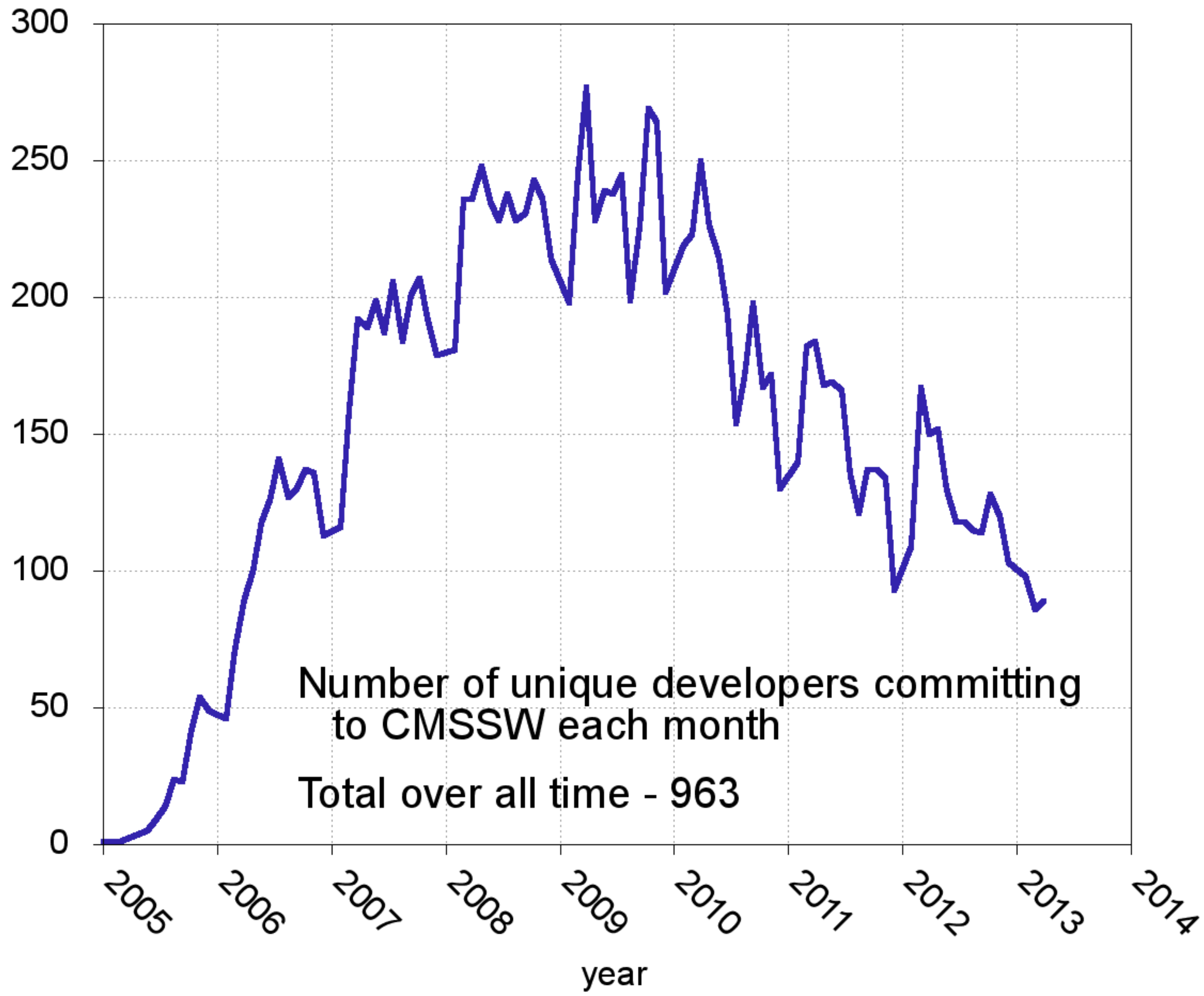
CMSSW Software Release

- First, the CMSSW software release, written by CMS people, is the largest software project we have: 3.6 MSLOC C++, with contributions by 960 people over the years, up to ~250/month.

- Includes all "common" code for the experiment: the core event processing Framework, CMS specific simulation, trigger, reconstruction code, data quality monitoring, validation, conditions mgmt, analysis tools, etc. (See lots of CHEP talks over the years.)

- How much of this is reusable even between HEP experiments is clearly open for discussion. *What is clear is that this scale of software development itself pushed us into a regime of software integration and testing more like large open source projects and companies than most scientific software projects.*

CMSSW Code Contributions (people/month)



See also "The Life Cycle of HEP Offline Software", P.Elmer, L. Sexton-Kennedy, C.Jones, CHEP 2007

Physics Generators

- alpgen (187k), cascade (35.9k) charybdis (2.9k) jimmy (5.4k) LHAPDF (79.6k) Rivet (77.9k) Pythia6 (78k) Pythia8 (75k) Tauola (21.8k) Tauola++ (58.4k) ThePEG (69k) toprex (33k) Sherpa (297.5k) MCDB (1.2k) libHepML (2k) HepMC (9.3k) HepPDT (13.1k) Herwig (120k) Herwig++ (189.9k) Photos (69.k) Professor (14.5k) EvtGenLHC (38.7k)

- Total of about 1.4 MSLOC, approx. half C++, half Fortran, clearly HEP-specific, not of interest to others.

- Starting to become more computationally intensive. (And incentives for theorists are different...)

System and Software Engineering

- General open source: boost bz2lib ccache curl cppunit distcc DMTCP doxygen expat jemalloc gcc gccxml gdb git gmake llvm libjpg libpng libtiff libungif libuuid libxml2 opengl openldap openssl oracle SLOCCount TBB Google-Perftools Valgrind xerces-c xz zlib libSigC++ python python-ldap protobuf ipython sqlite gdbm lcov cvs2git pacparser pcre rpm apt glimpse

- Developed in HEP: Castor classlib dcap dpm xrootd IgProf

Data Analysis, Math and Graphics

- General open source: gnuplot matplotlib numpy scipy GSL meschach lapack fftw3 CGAL graphviz Qt PyQt SIP
- Developed in HEP: CLHEP VDT ROOT (RooFit RooStats) PyMinuit2
- Several potentially interesting tools for the rest of the world.

Other HEP misc

- CORAL FastJet fftjet frontier-client Geant4 KtJet Hector TKOnlineSW

- Mostly not useful outside of HEP, with the exception of Geant4 (later slide) and (perhaps) Hector

ROOT

- analysis tools, visualization, Framework, Math libraries, data persistency, etc.

- 1.7 MSLOC, with many subpackages: cint (263k) graf2d (212k) math (126k) gui (114k) core (113k) graf3d (112k) roofit (108k) hist (93k) tutorials (78k) proof (76k) geom (61k) tmva (60k) test (55k) tree (53k) net (46k) io (42k), etc.

- Ubiquitous in HEP, and used in related fields, some use elsewhere

- Core team at CERN, effort at FNAL and large community input

- Has arguably served as a sort of "distribution" for other tools (xrootd, TMVA, RooFit/RooStats, etc.)

Geant 4

- Monte Carlo simulation of the passage of particles through matter
- 1.22 MSLOC (1.05M C++, 142k Fortran)
- Collaboration structure, effort at many large labs: CERN, FNAL, SLAC, KEK, ESA/ESTEC, etc.
- Used wider than HENP, also accelerator, medical physics, space science. As a physics-based tool, less obviously useful for other fields beyond these.

Sampling of other scientific software

- The following is a sampling of other scientific software

- I do not pretend this is a complete picture (it is not). I've not yet spent much time interacting with original authors, but instead used their documentation

- Most of these were suggestions from a random sampling of people, a few are just plain random.

- This selection is thus a bit "anecdotal", bit perhaps that is an appropriate way to form a "taxonomy"

BLAST+

- Implementation of an algorithm for approximate sequence matching of nucleotides in DNA or amino acids in proteins

- (See paper by Howison earlier for notes on the evolution of the BLAST algorithm and its implementations), rewrite in 2009

- From National Center for Biotechnology Information (NCBI), Penn State, U. Arizona: Biology/Computer Science collaboration

- 757 kSLOC C++/C

- "language and software environment for statistical computing and graphics"
- An implementation of S (Bell Labs), from 1993, by U.Auckland, with core group of ~20 developers with repository write access.
- 423 kSLOC (311k C, 89.6k fortran) + ~150k R
- Worldwide list of R User Groups: estimate "2 million R users", important tool not only in the academic world, but also in industry. Support through dedicated company Revolution Analytics, etc.

HDF5

- Hierarchical filesystem-like data format, data model and file format
- Originally from NCSA (1987), since 2006 supported by a not-for-profit corporation (HDF group). Very widely used by a quite diverse number of users from many fields.
- Not strictly tied to any particular set of tools, used by a large number of tools.
- 456 kSLOC (388k C, 35k F90, 19k sh, 12k C++)

Lammps

- Molecular Dynamics
- Originally written by Sandia/LLNL + Companies (Cray, Birstol Myers Squibb, Dupont)
- Dates from mid 1990's, first version in F90, rewrite in C++ in 2004 (made open source at that point: GPL)
- Designed for distributed MPI-style parallelism, support for some features with GPU's (CUDA, OpenCL), OpenMP
- 488 kSLOC (434k C++, 19.1k Python, 16k Fortran)

astropy

A Community Python Library for Astronomy

- "While there have been a number of efforts to develop Python packages for astronomy-specific functionality, these efforts have been fragmented, and several dozens of packages have been developed across the community with little or no coordination. This has led to duplication and a lack of homogeneity across packages, making it difficult for users to install all the required packages needed in an astronomer's toolkit."
- Includes things like Units/Coordinate-Systems, FITS/ASCII I/O, Astro computations and statistics, Message logger, etc. Tools that should complement more general packages like NumPy and SciPy
- Begun in 2011, it currently consists of 232 kSLOC (150k C, 60k Python, including 140k C for embedded expat)

NAMD
Scalable Molecular Dynamics

- "Not Another Molecular Dynamics" program, "NAnoscale Molecular Dynamics"

- 134 kSLOC (106k C++, 25k C), including Charm++ Message Passing parallel language and runtime 410 kSLOC (220k C++, 167k C)

- Previous codes rewritten from scratch in C++, with parallelism in mind, in the 1990's

- Typical parallel supercomputer application. MPI and selected features CUDA GPU accelerated.

CCP4

- a "suite of programs that allows researchers to determine macromolecular structures by X-ray crystallography and other biophysical techniques", "the CCP4 suite is a set of separate programs which communicate via standard data files"

- programs are modified to use CCP4 data format when they are integrated into the suite

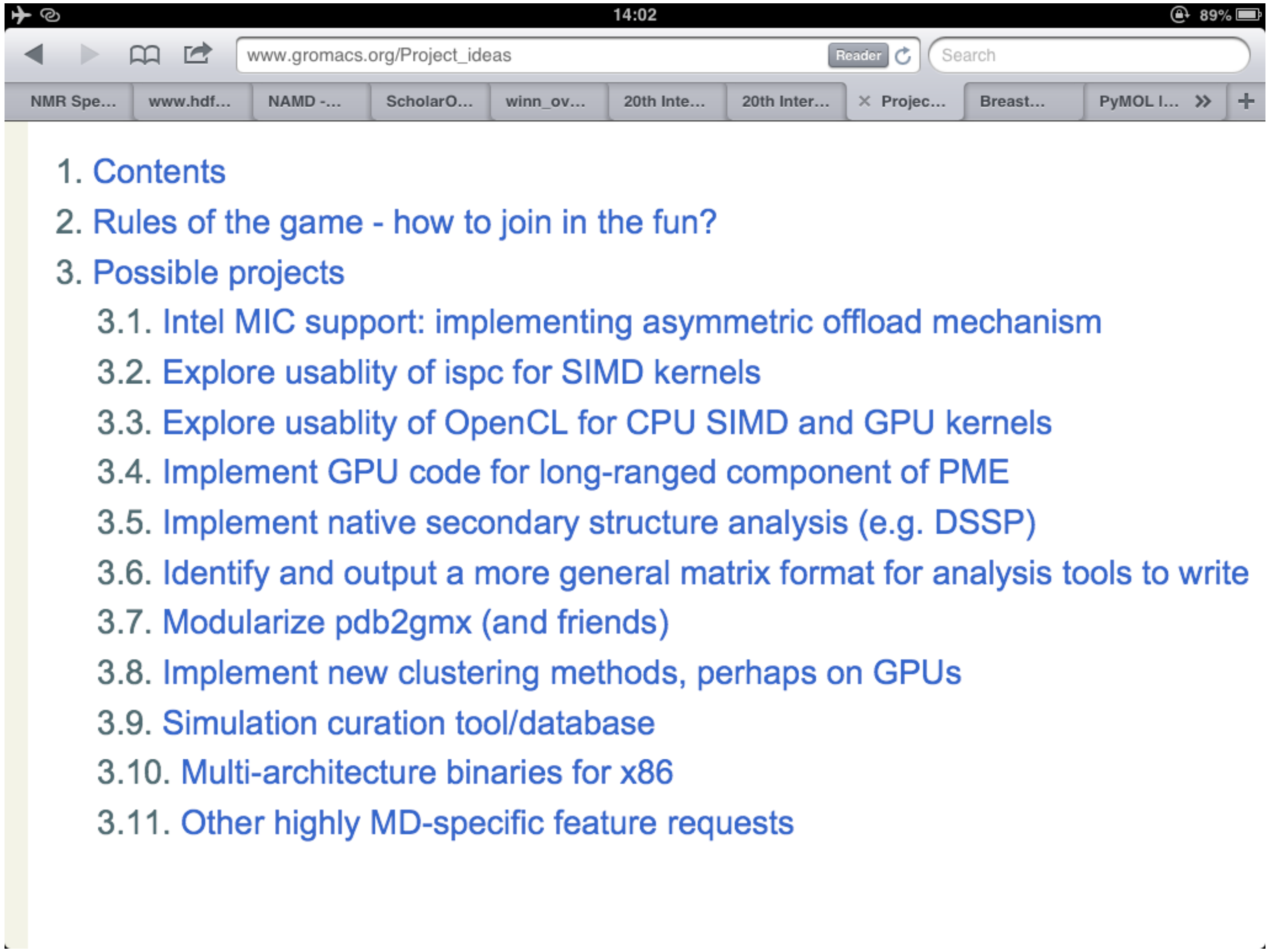
- Collaborative Computational Project Number 4 in Protein Crystallography, created in 1979 and coordinated from RAL

- 8.5 MSLOC (2.64M C, 1.99M C++, 1.90M Fortran, 1.38M Python), distribution of things incl. many like CMS externals (zlib, Python, lapack, etc.)

GROMACS FAST. FLEXIBLE. FREE.

- Molecular Dynamics (again), 1.33 MSLOC C
- Uses MPI, with possibility of CUDA-based acceleration
- Started by group at Groningen University, now appears to have support and developers in Sweden, Germany, US

- (Example) Project ideas:



1. Contents
2. Rules of the game - how to join in the fun?
3. Possible projects
   - 3.1. Intel MIC support: implementing asymmetric offload mechanism
   - 3.2. Explore usablity of ispc for SIMD kernels
   - 3.3. Explore usablity of OpenCL for CPU SIMD and GPU kernels
   - 3.4. Implement GPU code for long-ranged component of PME
   - 3.5. Implement native secondary structure analysis (e.g. DSSP)
   - 3.6. Identify and output a more general matrix format for analysis tools to write
   - 3.7. Modularize pdb2gmx (and friends)
   - 3.8. Implement new clustering methods, perhaps on GPUs
   - 3.9. Simulation curation tool/database
   - 3.10. Multi-architecture binaries for x86
   - 3.11. Other highly MD-specific feature requests

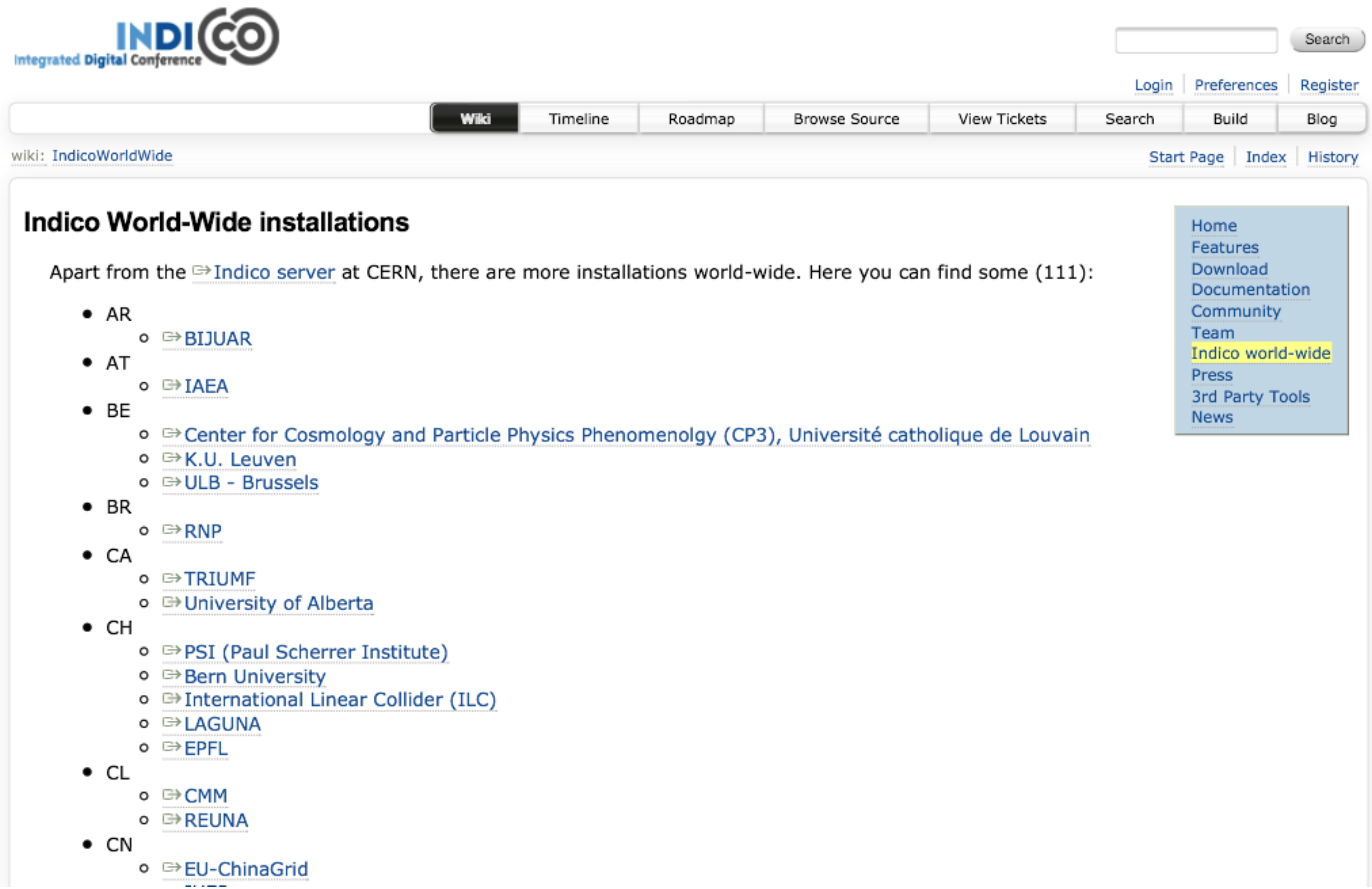
Now back to HEP....

INDICO

Integrated Digital Conference

- Originally an EU funded project: SISSA (Trieste), U.Udine, TNO (NL), UvA (NL), CERN. Open Source, GPL.
- Version 1.1, 151k Python.
- Long list of installations (next slide). Many are HENP collaborating institutions, from which (it appears) sometimes the tool begins to be used by others. (Network effect!)

Worldwide list of sites (111 total)

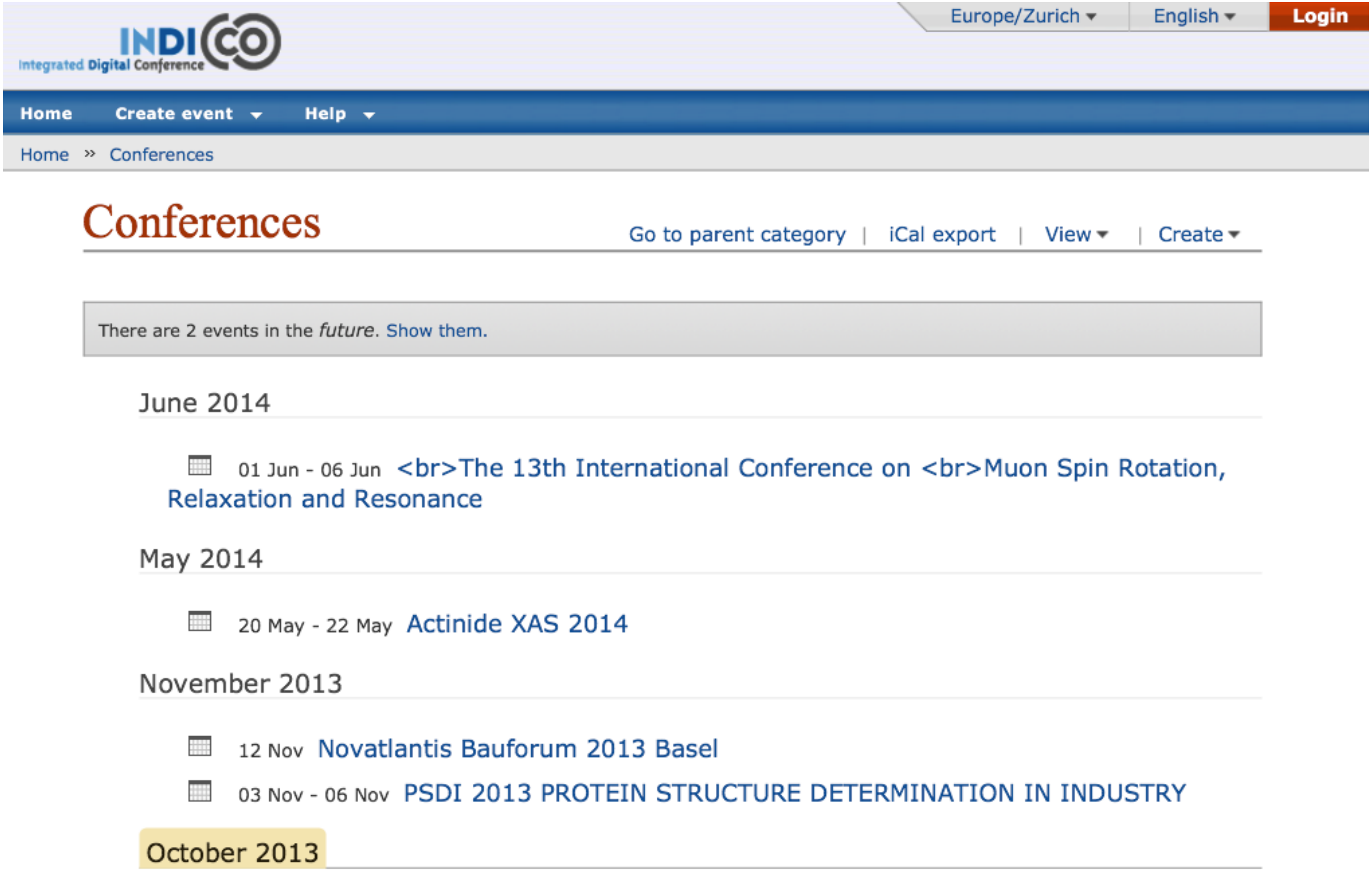


Indico World-Wide installations

Apart from the Indico server at CERN, there are more installations world-wide. Here you can find some (111):

- AR
  - BIJUAR
- AT
  - IAEA
- BE
  - Center for Cosmology and Particle Physics Phenomenolgy (CP3), Université catholique de Louvain
  - K.U. Leuven
  - ULB - Brussels
- BR
  - RNP
- CA
  - TRIUMF
  - University of Alberta
- CH
  - PSI (Paul Scherrer Institute)
  - Bern University
  - International Linear Collider (ILC)
  - LAGUNA
  - EPFL
- CL
  - CMM
  - REUNA
- CN
  - EU-ChinaGrid

PSI



Europe/Zurich ▾
English ▾
Login
INDICO
Integrated Digital Conference
Home
Create event ▾
Help ▾
Home » Conferences

Conferences

Go to parent category | iCal export | View ▾ | Create ▾

There are 2 events in the *future*. Show them.

June 2014

01 Jun - 06 Jun
The 13th International Conference on
Muon Spin Rotation, Relaxation and Resonance

May 2014

20 May - 22 May Actinide XAS 2014

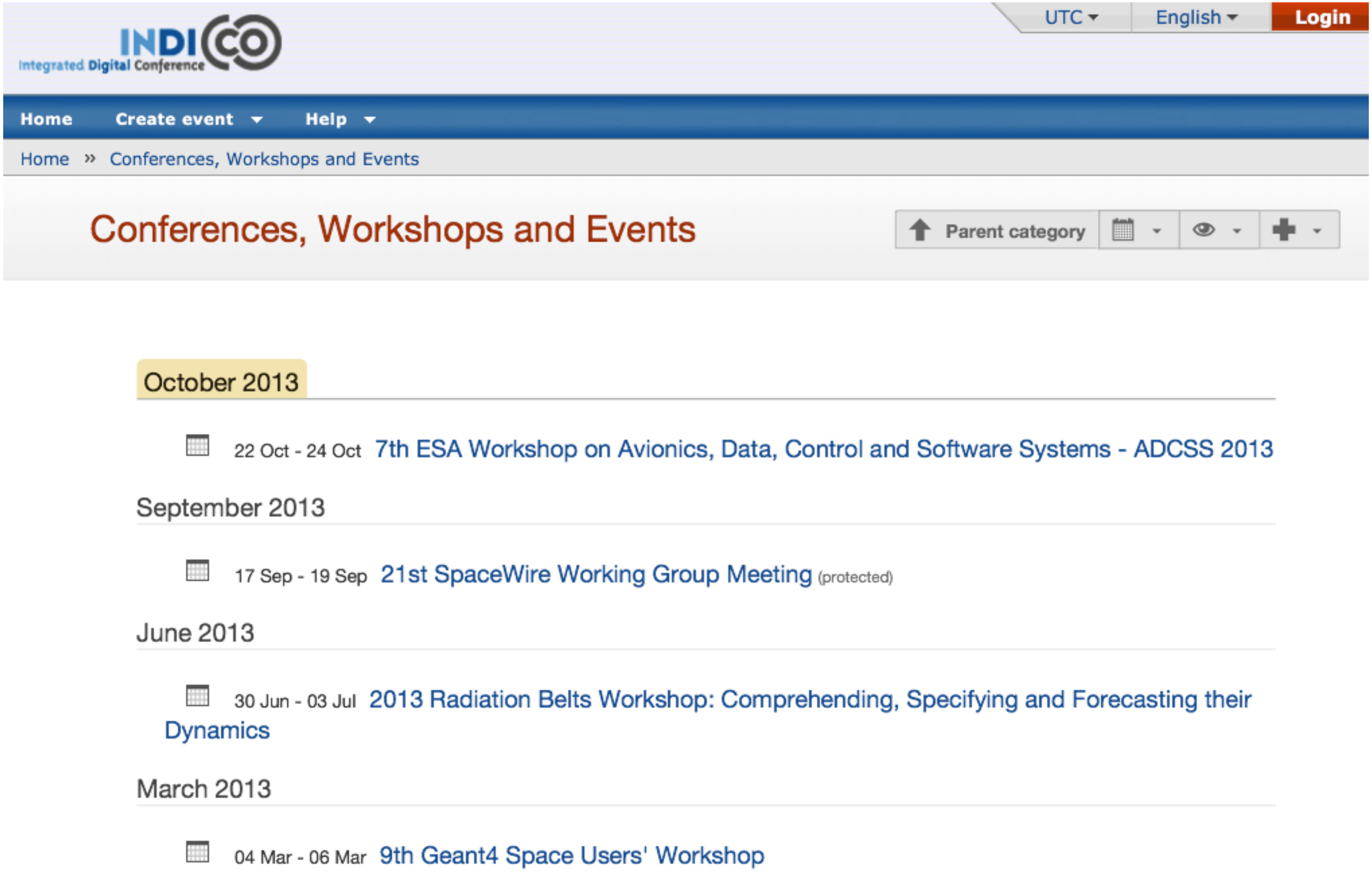
November 2013

12 Nov Novatlantis Bauforum 2013 Basel

03 Nov - 06 Nov PSDI 2013 PROTEIN STRUCTURE DETERMINATION IN INDUSTRY

October 2013

ESA/ESTEC



Home » Conferences, Workshops and Events

Conferences, Workshops and Events

October 2013

22 Oct - 24 Oct 7th ESA Workshop on Avionics, Data, Control and Software Systems - ADCSS 2013

September 2013

17 Sep - 19 Sep 21st SpaceWire Working Group Meeting (protected)

June 2013

30 Jun - 03 Jul 2013 Radiation Belts Workshop: Comprehending, Specifying and Forecasting their Dynamics

March 2013

04 Mar - 06 Mar 9th Geant4 Space Users' Workshop

Classifying Scientific Software - Use Models

- Roughly three categories:
    - Groups that use canned packages to do the heavy lifting
    - Groups that write their own software from the bottom up
    - Groups that are part of larger communities with Frameworks, etc.
- This is from F.Wuerthwein, characterizing what he saw as applications on the OSG.

Lots of interesting models

- Multiple methods of "Distribution" creation (AstroPy, CCP4)
- Tools with much more widespread use by a diverse community (R, HDF5)
- Wider use via "network" effects (InDiCo)
- Large (parallel) C++ packages (NAMD, LAMMPS) and C packages (gromacs)
- Large C++ (and other) codes (CCP4)

Strengths of HEP

- Very large collaborations with a clear notion of "common software and computing". Significant engineering effort, managed coherently. We can, and need to, be quite ambitious!

- Tradition of pushing software and computing limits.

- Result is a scale for coherence that is impressive: adoption of ideas or technology by an experiment can rapidly create very large user communities. We "natively" handle distributed collaboration and collaboration at a distance

Summary

- HEP has a lot to offer the world at large, and not just for "distributed computing". To the extent that this is expected of us, we can probably deliver if we learn how.
- The rest of the world also has a lot offer us, we should embrace that, too...
- This "bug hunt" surely missed a lot of things: if you know of other interesting software packages, fields with interesting development models or have ideas as to where collaborations as above might be beneficial, I would be interested to hear them...

Thanks for listening....