

McStas 13 years after - lessons learned?

Peter Willendrup¹

Emmanuel Farhi^{2,3}

Erik Knudsen¹

Uwe Filges⁴

Kim Lefmann^{5,2}

¹Materials Research Division, RISØ DTU, Roskilde, Denmark

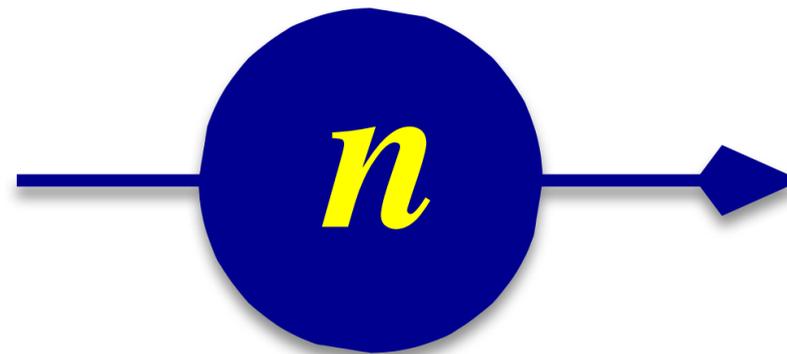
²Institut Laue-Langevin, Grenoble, France

³European Spallation Source AB, Lund, Sweden

⁴Paul Scherrer Institut, Villigen, Switzerland

⁵Niels Bohr Institute, University of Copenhagen, Denmark

McStas



McStas project <http://www.mcstas.org>

Agenda

- Short McStas introduction
- Lessons learned
 - Development/design
 - Infrastructure
 - User contribs
 - Documentation
 - Dependencies
- Conclusion

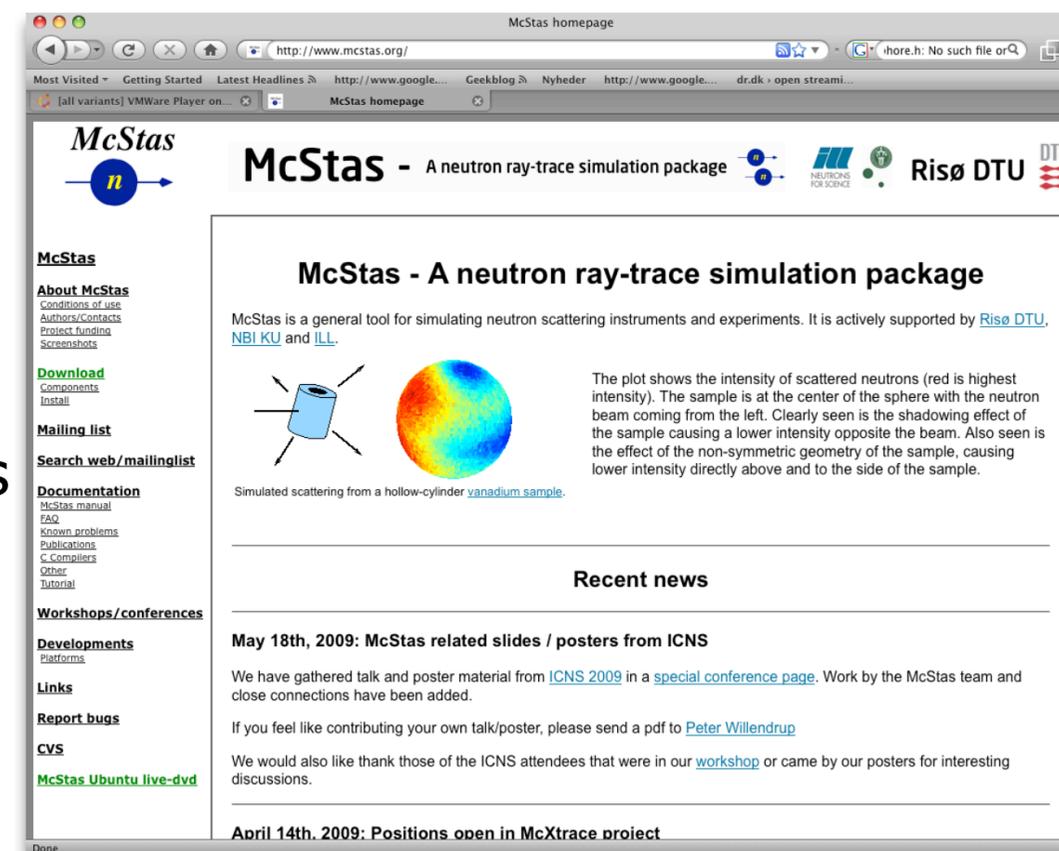
McStas Introduction

- Flexible, general simulation utility for neutron scattering experiments.
- Original design for Monte carlo Simulation of triple axis spectrometers
- Developed at RISØ DTU, KU, PSI and ILL
- V. 1.0 by K Nielsen & K Lefmann (1998)
- Currently 5 people (2 full time, 3 part), students
- International users/contributors

GNU GPL license
Open Source

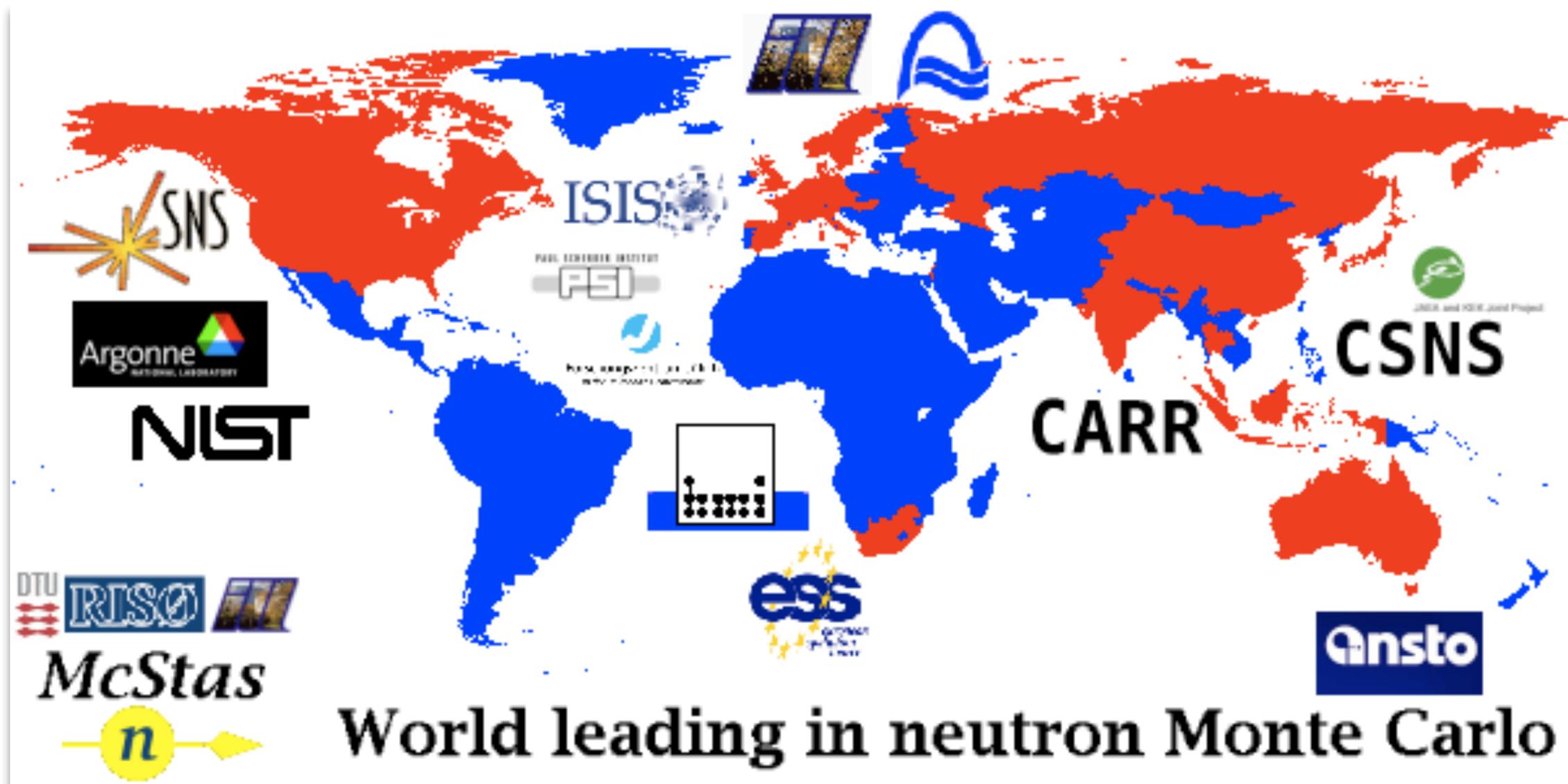
Project website at
<http://www.mcstas.org>

mcstas-users@mcstas.org mailinglist



McStas Introduction

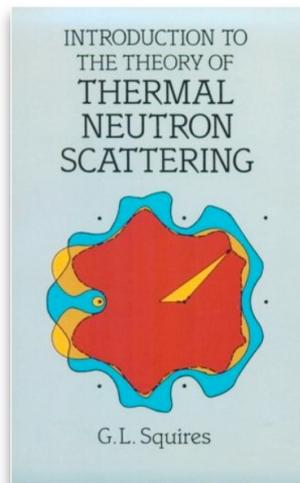
- Used at all major neutron sources (or instrumentation efforts)



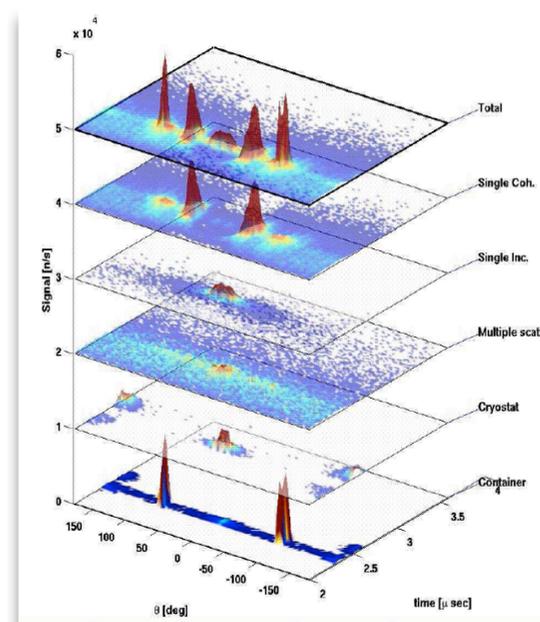
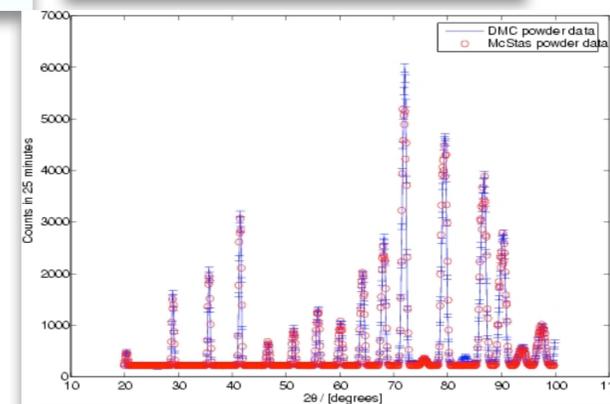
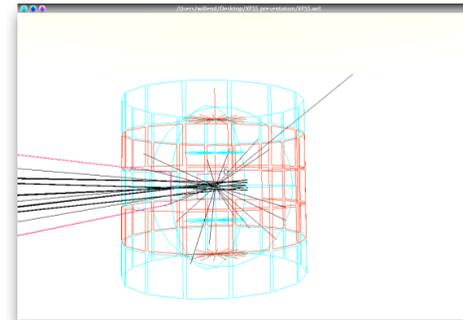
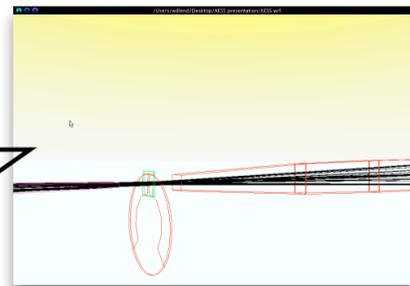
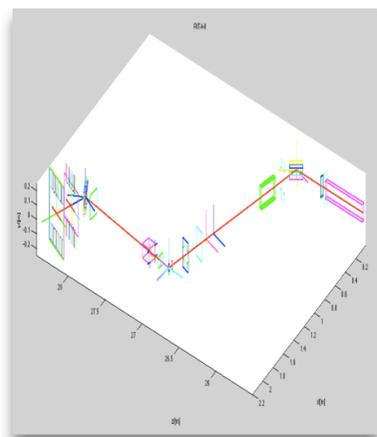
What is McStas used for?

- Instrumentation
- Virtual experiments
- Data analysis
- Teaching

(KU 2005-2010)

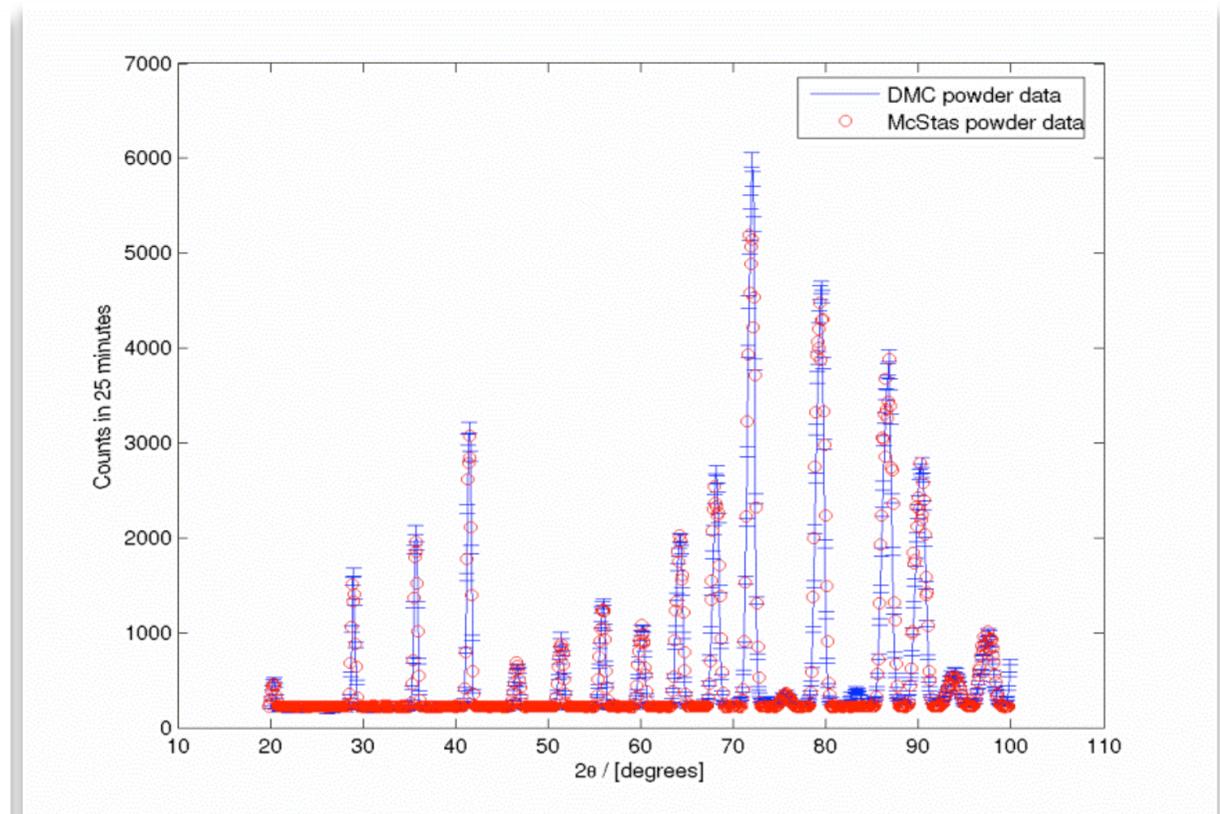
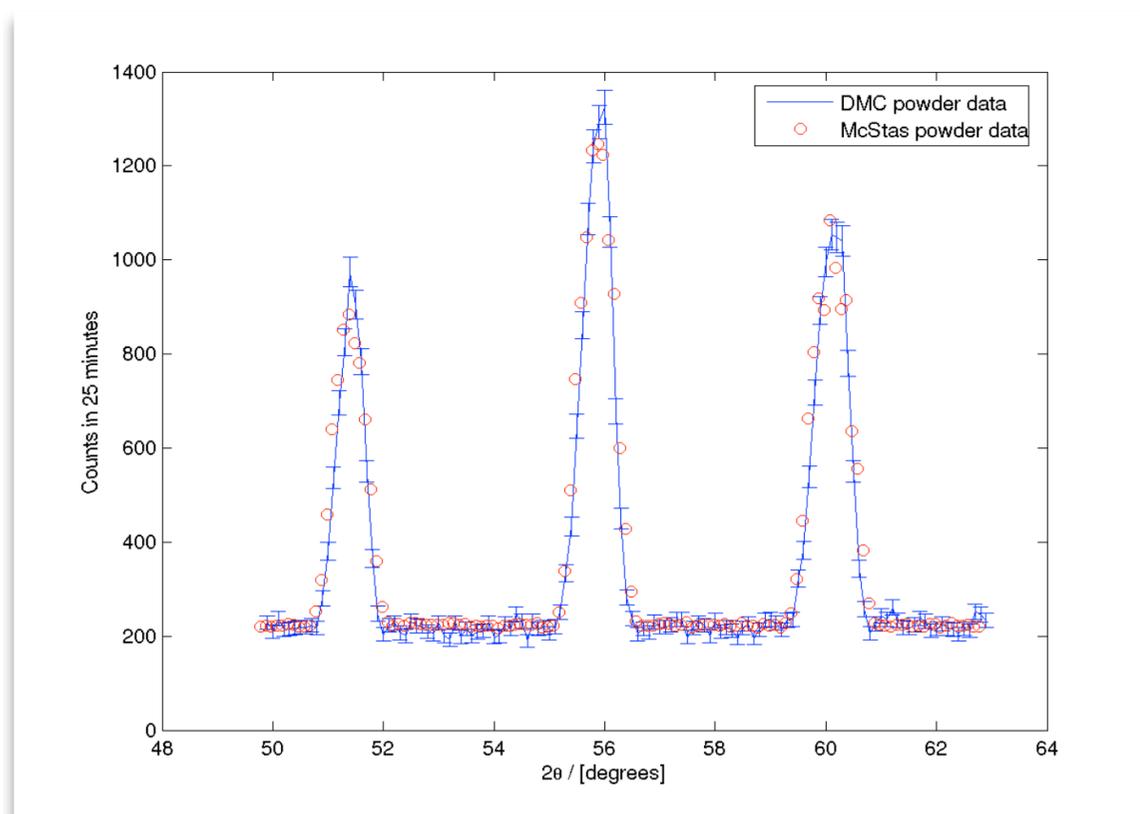


NOBUGS 8, Gatlinburg TN, 2010



Reliability - cross comparisons

- Much effort has gone into this
- Here: simulations vs. exp. at powder diffract. DMC, PSI
- The bottom line is
- McStas agree very well with other packages (NISP, VitESS, IDEAS, RESTRAX, ...)
- Experimental line shapes are within 5%
- Absolute intensities are within 10-30%
- Common understanding: McStas is reliable



P. Willendrup, Risø DTU; Uwe Filges, L. Keller, PSI

McStas overview

- Portable code (Unix/Linux/Mac/Windows)

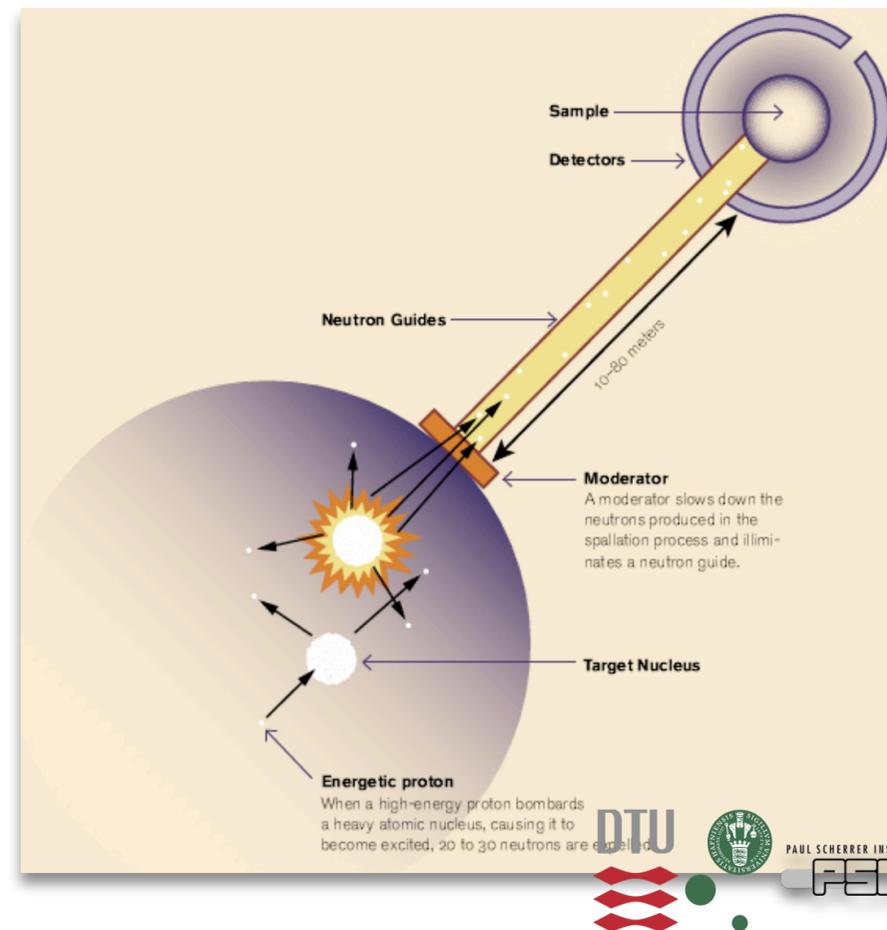


- Ran on everything from iPhone to 1000+ node cluster!

- 'Component' files (~ 100) inserted from library

- Sources
- Optics
- Samples
- Monitors
- If needed, write your own comps

- DSL + ISO-C code gen.



Code/sim infrastructure layers

- GUI + visualisation
- Instrument file (DSL)
- Components (includes user contribs)
- Runtime code / libs etc.
- -> Generated C code

Under-the-hood / inner workings

- Domain-specific-language (DSL) based on compiler technology (LeX+Yacc)

- Simple Instrument language $\xrightarrow{\text{Code generation}}$ ISO C

- Component codes realizing beamline parts (including user contribs)

- Library of common functions for e.g.

- I/O
 - Random numbers
 - Physical constants
 - Propagation
 - Precession in fields
 - ...

Including user contribs

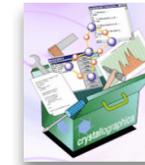
- Well-developed community support
 - 30-40% of existing and new additions are from users
 - No direct refereeing of the code, but these requirements:
 - At least one test-instrument
 - Meaningful documentation headers (in-code docs)
 - Contributions go in dedicated contrib/ section of library
- Natural life-cycle of contrib's
 - Bug-fixes are applied both by contributor and developers
 - If contributor becomes unavailable either:
 - Many users of comp: Promote to official components, e.g. in optics/
 - Few/no users of comp: Move to obsolete/ until next major release



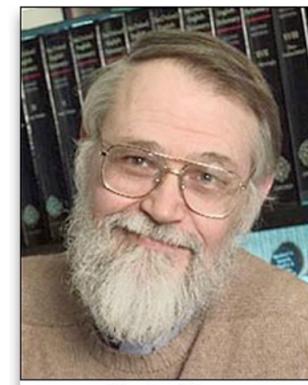
Principles

- Prefer minimum intrusion solutions
 - (has only proven semi-faulty approach in perl-layer)
- If someone does it well, rather wrap/interface than re-implement
 - Material structures from CGRAPH / Lazy
 - Interface with e.g. MCNP

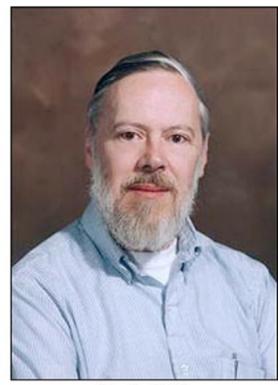
mcnp



- Try to minimize external dependencies (libs etc.)
 - Basic number-crunching dependency for McStas is C! \Rightarrow
 - Automated parameter variation requires perl
 - Gui layer requires perl-Tk
 - Graphics layer requires either
 - PGPLOT, perl-PGPLOT and PDL
 - GNUPLOT



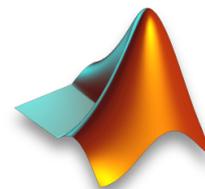
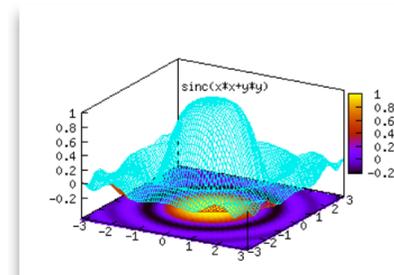
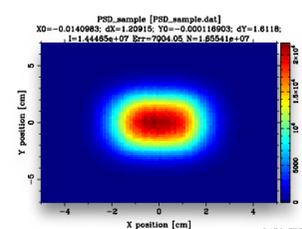
Brian Kernighan



Dennis Ritchie

K & R

• Matlab



Development tools



- Revision control: SVN



- Trac bug-tracking etc.

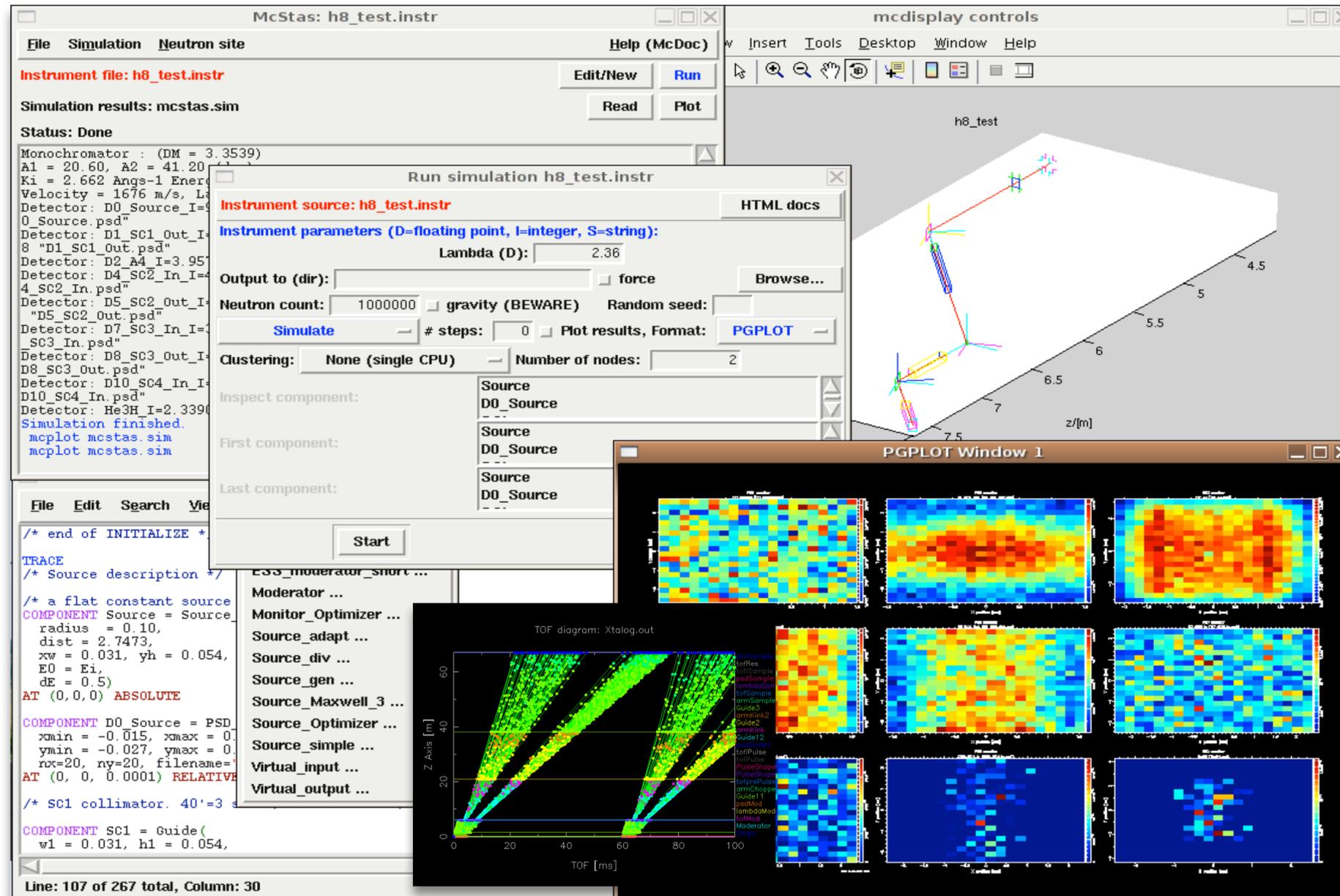


- Auto-generated online-documentation (perl based html from c-codes)
- “Homegrown” website solution:
 - Flat html files in tree generates SSI-based “menu structure” (table)



- Automated build-process for all standard platforms (homegrown)

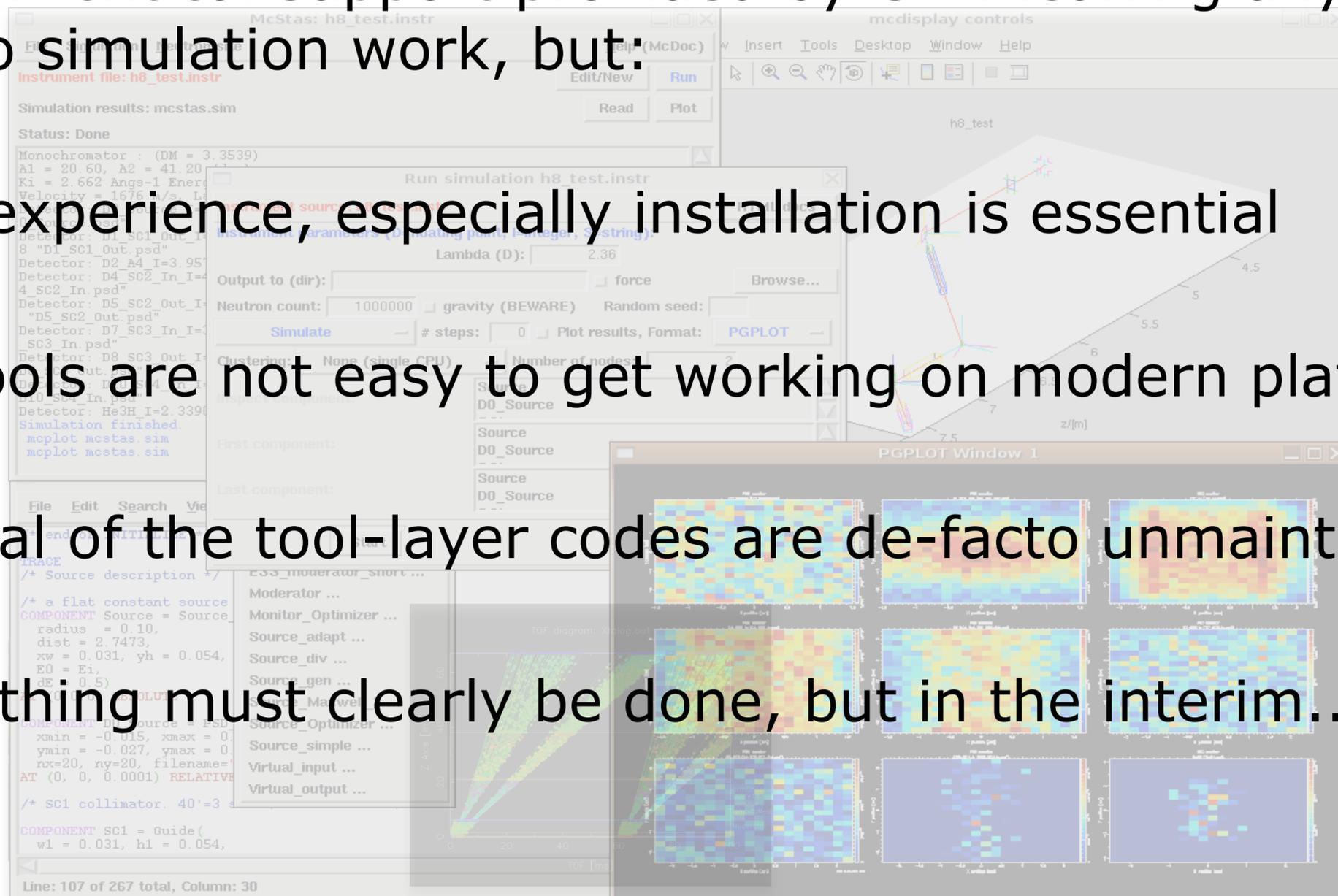
McStas GUI overview



Mildly antiquated.... ?

McStas GUI overview

- Mission-critical support provided by C - meaning anyone can do simulation work, but:
- User experience, especially installation is essential
- Old tools are not easy to get working on modern platforms
- Several of the tool-layer codes are de-facto unmaintained
- Something must clearly be done, but in the interim...



Mildly antiquated.... ?

Provide ready-to-run solutions

- Binary distributions for reference platforms:



ubuntu

- Ubuntu 9.10/10.04 32 and 64 bit deb packages
 - Uses distro-deb packages only for the tools



- Mac OS X Panther -> Snow Leopard
 - Includes binary distributions of support codes
 - Requires X11 and Xcode from OS X media, otherwise standard system



- Windows XP ... 7 32bit (works on 64 also but is 32 bit...)
 - Includes binary distribution of support codes
 - Includes c-compiler, perl distribution, vrml viewer, ...

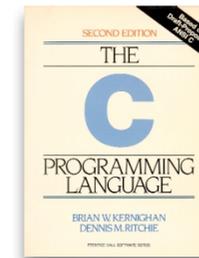


- Complete, installation-free runtime environments:
 - Ubuntu-based vmware and live-dvd solutions

- Supplementary src tar.gz available with pseudo-automated installation of tools - buggy

Future support tool situation

- Basic calc/sim functionality support still provided by C



- Python will provide scripting and glue



- GUI widgets likely wxwidgets or Qt



- Plotting using  (also considering q'n'd GNU PLOT hooks)

- High-level support tools provided using e.g.



Documentation

- As explained, basic use info is available inside comp & instr codes, extracted by perl to html
- 100+ page manuals documenting
 - Metalanguage
 - What is “under the hood”
 - Examples of practical use plus advanced features
 - Assumptions and algorithms applied in the components
- More than 70 example instruments
- Various tutorial and teach yourself solutions are available
- Only missing type of docs: Devel docs (will be improved)

McStas Introduction

McXtrace - since jan 2009 similar in X-rays

- Flexible, general simulation utility for neutron scattering experiments.

• Original

• Develop

• V. 1.0 b

• Current

The screenshot shows the 'Main Page - McXtraceWiki' in a web browser. The page title is 'Main Page' and the URL is 'http://www.mcxtrace.org/index.php?title=Main_Page'. The page content includes the McXtrace logo, a navigation menu with items like 'Main Page', 'Partners', and 'Project People', a search box, and a toolbox. The main text states: 'McXtrace - Monte Carlo Xray ray-tracing is a joint venture by' followed by logos for Risø DTU, DTU, ESRF, and JJ X-RAY. It also mentions funding from NABIIT, DSF, and the above parties, and that the code is based on technology from McStas. A mailing list subscription link is provided at the bottom of the main text area.

- Synergy, knowledge transfer, shared infrastructure

NOBUGS 8, Gatlinburg TN, 2010



Conclusions

- Basic simulation core/language - Properly designed by computer scientist
- GUI and plotting are perl-hacks (effectively not pre-designed)
 - - Start out by a proper design
- Today's problems are essentially in the tool layer, dependencies
 - - Keep as few and simple dependencies as possible
- Choose simple and functional (or at least provide a simple solution also)
- Don't re-implement, rather wrap/link, re-use
- Several layers of code is "infinitely" flexible (so far)
- Documentation and first user impressions are a first priority
- Component developers / advanced users are important - GPL / openness made the project what it is

People

- The success of the project is also about the people:

- Present McStas team members



• K Lefmann



E Farhi



P Willendrup



E Knudsen



U Filges

- Past McStas team members



• K Nielsen



PO Åstrand



K Lieutenant



P Christiansen

Yes it is still working no matter what they say... :)

