Current state and future plans for
McStas

Peter Willendrup, Kim Lefmann

Emmanuel Farhi

Agenda

- What went wrong?
- What has been done to restore McStas
- Collaboration tools, policies, CVS
- Next release
- Survey results
- Conclusion

What went wrong?

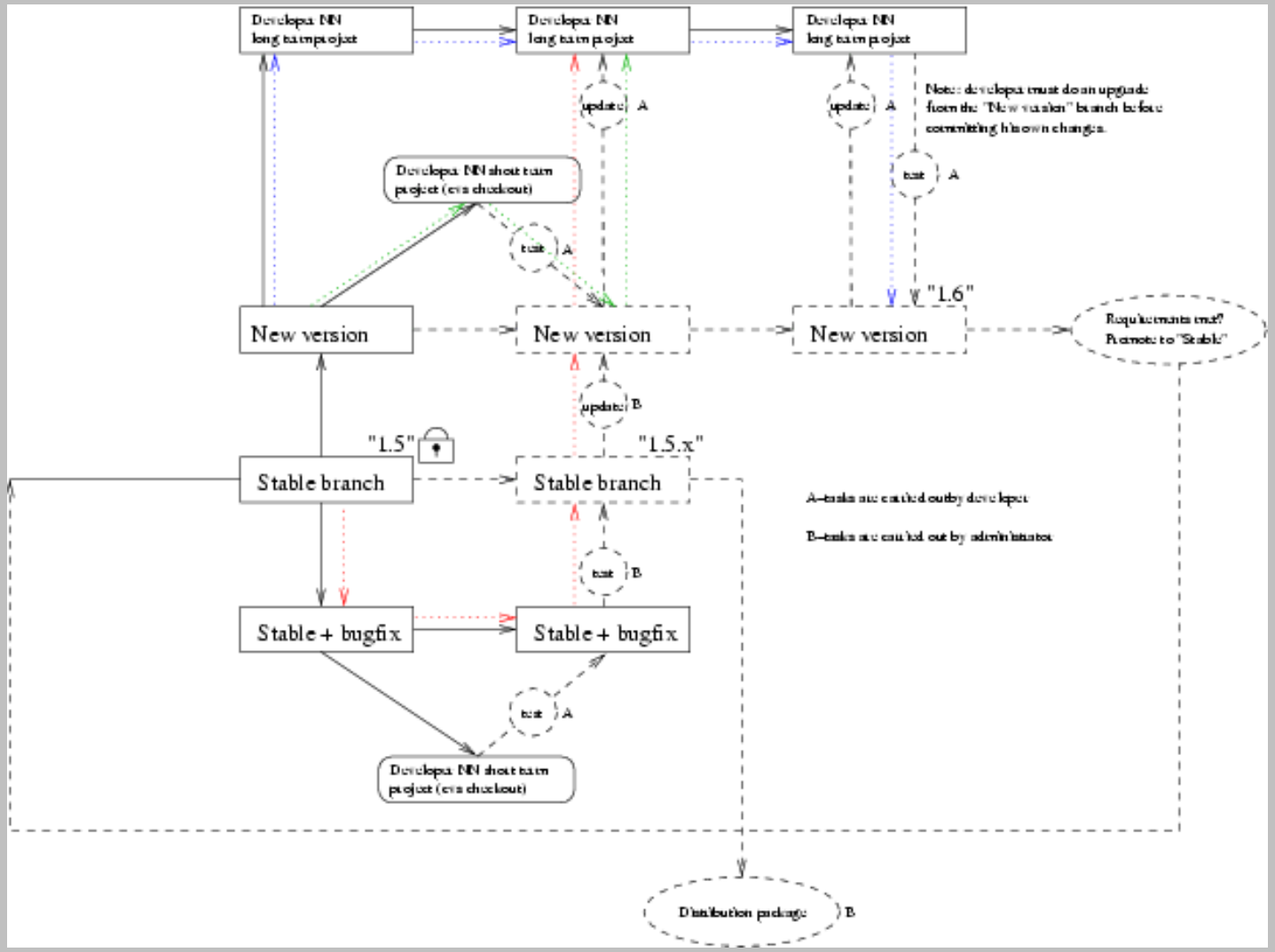
- Disagreement between developers regarding contents of McStas 1.5, eventually closing neutron-mc@risoe.dk mailing list.
- Branching of versions, e.g. 1.5 and 1.6-ill
- Disagreements were eventually sorted out. Per-Olof Åstrand had less time to devote to McStas, and finally got a new position at NTNU
- Hacker attack on McStas web server / mailing list machine.

Result: McStas 'off line' November 2001 - June 2002

What has been done to restore McStas?

- New McStas system responsible as of July 1st 2002 – *Peter Willendrup*
- Updated website running on July 19th 2002
- Mailing list restored on July 31st 2002
- Meeting at *ILL*, august 2002 (*Peter Willendrup and Emmanuel Farhi)*
  - Achievements:
    - TODO list for McStas 1.6 (new release)
    - Policies for future collaboration, CVS
    - First code modifications
    - Release of status page on the McStas website
    - McStas user survey launched on September 2nd
    - McStas user survey ended on September 23rd, results presented here

Future collaboration



Future collaboration

- All current changes to the different 'stable' versions should be merged into a new release, i.e. McStas-1.6
- Once released, two code bases should exist,
  - Development (All developers have access)
  - Stable (Only McStas responsible has access)
- Users will only know of the existence of Stable
- All code check ins should pass a 'test'.
- To upgrade Development to Stable, the package should pass a 'test'
- The test procedure should be available for the users to check their own components etc.

Future collaboration

- Sketch test procedure:
  - Component test – instrument files and output available to test validity of new components (when applicable)
  - Kernel test – a new kernel/runtime should reproduce the results of the previous version
- Test procedure is not set up yet, requires a lot of work

Next release – done/to be done

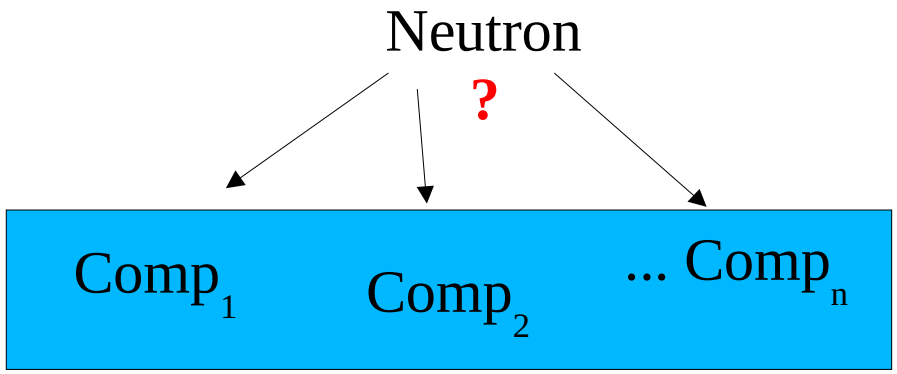
- Kernel/lib/runtime – slides by Emmanuel Farhi
- GUI / plotting facilites – slides by Peter Willendrup

Already done (1.6-ill features already implemented for next release)

Ascending compatibility from *McStas* 1.4.x and 1.5

Features from 1.6-ill

GROUP (make a group of exclusive, equivalent components)



Test each possibility
in the order 1..n
choose the first that 'works'.

A kind of splitting.

EXTEND (add code to existing components, without touching their code)

In an instrument description, we may enrich a component behaviour adding some C code that is executed just as if it came at the end of the original component code/action. *Ex: adding neutron colour*

McStas kernel: done (2)

Already done (either 1.6-ill or new features already implemented for next release)

[Features from 1.6-ill]

SHARE (share common functions for all similar components)

This section is efficient when using many instances of the same component in an instrument (guides, crystals, detectors ...). It enables to only declare once for all (share) some functions, instead of copying the same code for each component instance.

⇒ Code is smaller

New Features

%include (share common functions for all components, from a library)

This section enables to share functions for all components. These functions are described in an external C library (MCSTAS/lib/share), and only imported once, and only if required.

⇒ Code is smaller

McStas kernel: done (3)

Already done (new features already implemented for next release)

[New Features]

Component setting parameters: can be int and char*

Previously, setting parameters could only be of type double. Definition parameters may be of any type, but can not be modified (macros). They still can hold mathematical expressions (C).

SAVE (data saving during and at end of simulation)

This section may be called during the simulation (on signal USR2).
The Finally section can still hold the saving and simulation end operations, but most components will have their save functions moved to SAVE.
Used with the signal USR2, it is possible to save data, analyse it and possibly stop the simulation (signal TERM), e.g when accuracy is good.

McStas library: done

Functions and components used by the instrument

Run-time

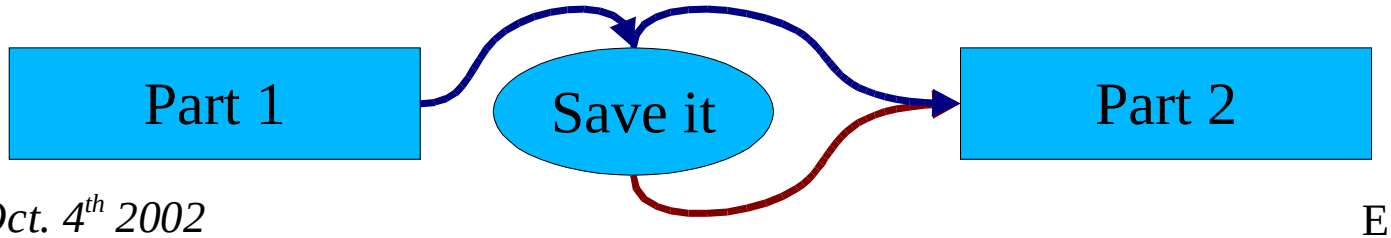
- rand_target_rect (choose a neutron focusing to a rectangular area)
- rand_target_sphere -> rand_target_circle, optimized
- moved occasional functions to lib/share (on request).
- Save data as McStas, *NeXus* XML, HTML, Scilab, IDL ...

Components:

- Monitor_nD can monitor something else that the intensity

ex: <ω> *vs* X,Y

- Monochromators can read a reflectivity table
- New components for load/save of 'virtual' sources



Prepare simulation part so that they are executed faster (just as Vitess modules...)

McStas kernel: will be there

New feature used for building the instrument. Nearly ready

Ability to JUMP from one component to another

Repeat components and jump to others

⇒ can do loops, multiple scattering, duplicate components (guides)...

Extensive help and examples about new features:

- Groups,
- Skip components,
- Extend components,
- Share component code (smaller C files), use libraries
- Optimize a simulation for flux (Source_adapt, source_Optimizer, ...)
- Optimize instrument parameters
- Test instruments to validate package and components

McStas library: will be there

RISØ

New feature used by the instrument. Nearly ready

Run-time

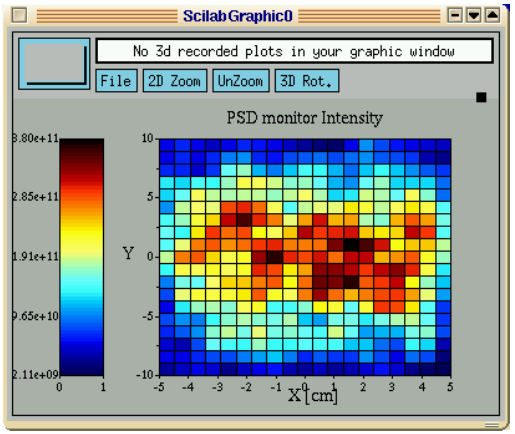
Gravitation for all propagations (at user choice), *for cold neutrons...*
Save data as Vitess, Nexus, raw binary (double or float)
Output files will have internal plot methods.

Components:

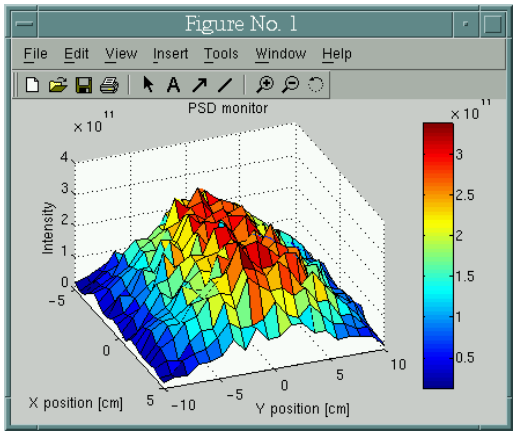
new 'lib/contrib' for unofficial nice components
new 'lib/obsolete' for old/not maintained comps
new 'lib/doc' for documentation
mirror/guides/benders with reflectivity table from files (with pol. ...)
polarising crystals (monochromators., analysers)
general elastic+incoherent powder sample (from Powder_filter)

On the way...

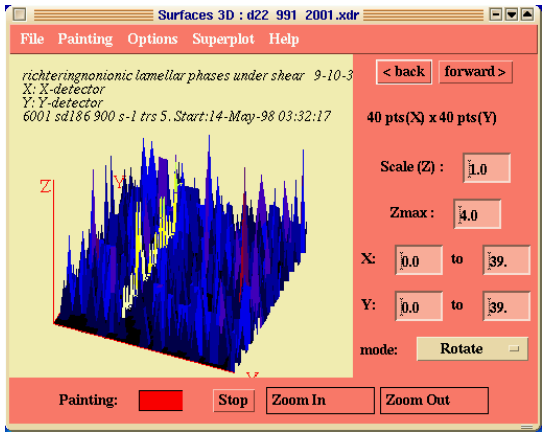
Scilab:
script with plotting



Matlab:
script with plotting



HTML:
Java applet 3D viewer !



Things we are thinking of for the kernel/runtime

- Background estimation (through lost neutrons)
- Restart an interrupted simulation (not so easy...)
- Native parallel computation (MPI/PVM)
- your comments and suggestions !
- Inelastic components: regularly progressing... $\Rightarrow \Rightarrow \Rightarrow$

Inelastic sample (V. Hugouvieux)

Inelastic sample from Molecular Dynamics. With full *S(Q,ω)*

Sample structure and interactions

→ *Molecular Dynamics*

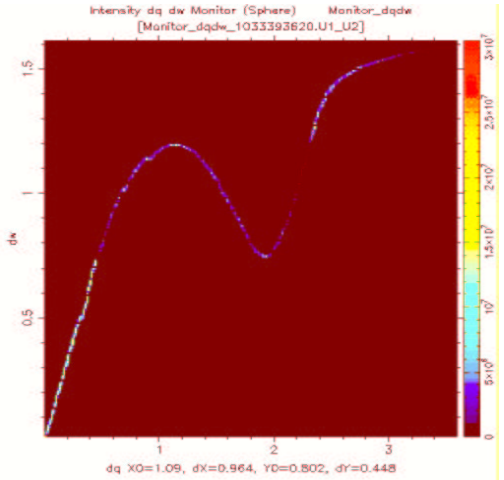
→ Position, velocity of atoms

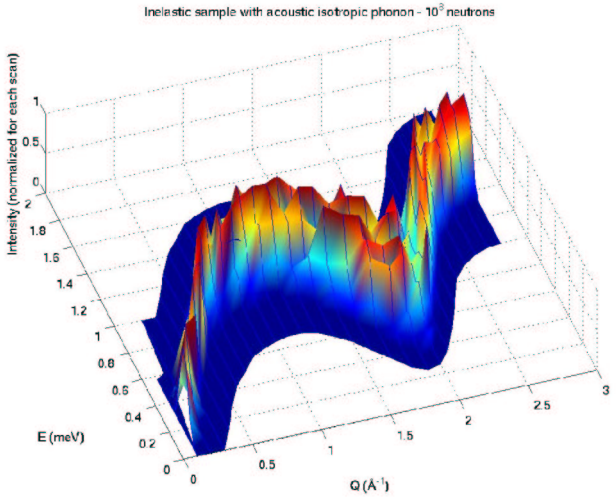
→ *S(Q,ω)* in 2 or 4 dimensional space
(liquids/gases, crystals)
describing both structure and dynamics

→ *McStas* general sample component

→ Instrument simulation

→ Realistic simulated signals for
any neutron instrument/sample !





Next release – GUI / plotting 1

- GUI and plotting tools (mcgui, mcplot...)
  - Currently based on perl+PDL+PGPLOT+perl/tk
  - Causes problems on many systems, since the packages are 'out of sync.'
- Solution
  - A single package should replace the above setup.
  - Candidates (price/advantage/disadvantage)
    - Matlab (Expensive/stable, proven/Expensive)
    - IDL (Expensive/stable, proven/Expensive)
    - SCILAB (Free/matlab like/premature matlab GUI)
    - Octave (Free/matlab like/no GUI)
    - 'Roll your own' - possible, nice, but much work
- SCILAB
  - Matlab GUI too unstable, but also has tk GUI support. Current McStas GUI is partly written in tk, should be easy to rewrite. Work in progress.
  - Has 3D plotting facilities
  - Easy access to scripting + numerical functions for the McStas user

Next release – GUI / plotting 2

- Rewrite in SCILAB will improve
  - Portability of GUI/plotting tools, scilab is available on a large number of platforms: i386/Linux, i386/windows, i386/*BSD, PPC/Linux, PPC/MacOS X, Sparc/Solaris, PA-RISC/HP-UX, Alpha/OSF, MIPS/IRIX (precompiled packages). Should compile on “any” Unix like OS with X.
  - Easier installation procedure, McStas (./configure;make;make install), pick a scilab package. Tk is included, at least on windows.
  - Possibly 3D visualization of neutron rays etc. This requires some changes in McStas kernel code / components .
- The 'old' GUI could stay for completeness

Survey results -1

- Typical user platforms
  - Linux/i386 (21)
  - Windows (9)
  - SGI (3)
  - Linux/alpha (2)
  - HP (1)
  - DEC (1)
  - Macintosh (before Os X) (1)
- Typical Version
  - 1.4 (7)
  - 1.5 (7)
  - 1.6-ill (7)
  - Pre 1.4 (1)
- Most occurring problem
  - No special area (8)
  - Support apps (7)
  - Support (3)
  - Stability (2)
  - Features (1)
  - Installation (1)

Survey results – 2

Many good comments and suggestions:

- Improve methods for grouping components, e.g. large detector array made from linear PSDs. Add some limited capabilities for "reusing" neutrons, e.g. once a neutron gets to the sample can many copies be sent to the detector(s) with some scattering characteristic randomly selected?
- I would like to see an adaptive source algorithm for time-of-flight instruments. I believe that Source_adapt by Kristian Nielsen could be modified to adapt based on energy, emission time, position, angle, etc. I plan to try this when I get back to working with McStas.
- perhaps something like simulation progress (a "." for every 10e6th neutron)
- A bigger collection .instr-files with comments could sometimes be very useful!?
- TOF area detector to simulate Laue-type
  - time-of-flight area detector. Input: Dimensions of detector in mm., ...
- Version 1.4 has enough features for me, assuming there are no major modeling errors in it. I'm reluctant to upgrade because installing the first version, with all related packages, was a bit of a pain on my Debian system. I've upgraded Debian since, and the only part of McStas that stopped working was mcdisplay. I now have a versatile matlab set of routines that makes mcdisplay unneeded.
- See other comments - but one thing is that for McStas to stay trustworthy it must be maintained in a mode where the "official" release has only components that are tested and verified by the McStas maintainers. If most of the useful features of the program depend on user contributed modules it is probably doomed. Also, communication related to verifications much be improved - see next comment.
- My biggest concern is that the results are reliable, not that some bells whistles may be present.

Conclusion

- McStas developer team has been re-united
- The TODO list for the next release is written
- The team is aiming at release before end of 2002
- The next release will be providing
  - One, united version of McStas
  - A cure for the reported GUI problems
  - Meeting up with most of the user wishes from January 2001, reg. Kernel, runtime and tools
- The next release should live up to the results of the survey

TODO highlights 1/2

- Kernel:
  - "Switch/Skip" components (Component group ABSORB GROUP COMPONENT - can be done, example in manual?)
  - "Share" components (done, sharing "shareable" parts - lib style) - Logging event support (Can be done, Monitor_nd - example in manual?)
  - “Split" neutrons (see switch/skip + extend feature - example in manual)
  - Framework for table input ?!? (If about parameter input, should be doable, through defines)
  - Compile runtime+shared functions - through example in manual.
  - %INCLUDE keyword for "inclusion" of components in components... (To think about), done. :) -> component_shares/My_function_name_include
  - Retrieve old monitor data, read from file by Virtual_input -> mcstas_r
- Runtime:
  - Gravity (functions defined, only used in gravity guide - merge propagation routines, easy task)
- Components
  - Prepare all for polarisation - done.
  - ESS moderators - done.
  - Monitor_nD, replace p by other phys. quantity, easy task.
  - Renaming of components (optics/samples) (e.g. Guide_*)
  - Monochromator, reflectivity curve from file. re-normalisation by r0, simple.
  - Rand_target_rect, modification according to formula.
  - Virtual_Output - should be written, macro for calling Monitor_nD? - See %INCLUDE in kernel section.
  - Check fluxes / absolute fluxes - provide experimental data + instrument definition
  - Powder component take input, written, should be tested. 2 lines, n lines.
  - Sample for SANS - existing components (single xtal ex.), other parms, powder to be checked.
  - Bender, to be tested check for absolute fluxes, as compared to a number of linear guides.
  - Sort components in categories, official, contrib, obsolete
  - Log - cvs revision no, etc. for both McStas kernel / lib, automated using cvs?. Origin indicates research institution, McVersion required McStas Version

TODO highlights 2/2

- Instruments
  - Inter comparison instruments H8, IRIS, RITA-II, IN12, TAS1, IN14,...
  - Component test modules
  - Moved to lib/examples/H8,... including instrument files + data + test results from "stable version" incl. test- script.
- Tools
  - Automated optimization -> Optimisation (LM?) - great complexity... Manual example?
  - Improve GUI - scilab.
  - Status monitor support (Tool to be written - script) - access pid, daemon.
  - html / png / page generation. (perhaps scilab based?) - mcplot?
  - mcdisplay date checking for recompilation, through mcrun -n 1
  - export in several formats, ps, gif (scilab), png (Imagemagick when available)
  - mcdisplay show help info when no arguments (l. 517)
  - mcdoc: changed to show instruments also.
- Documentation
  - FAQ list in manual
  - Examples for group, share etc.
  - Component manual info should be included, user auth. for contrib.
  - valid latex2e code. (latex2html, pdftex etc.)
  - Doc directory.

Questions?

Questions welcome!