McStas 25 year celebration

2023/11/16

PETER WILLENDRUP, ESS DMSC & DTU PHYSICS
McStas Introduction

• Flexible, general simulation utility for neutron scattering experiments.

• Original design for Monte carlo Simulation of triple axis spectrometers

• Developed at DTU Physics, ILL, PSI, Uni CPH, ESS DMSC

• V. 1.0 by K Nielsen & K Lefmann (1998) RISØ

• Currently ~6 people on joint McStas-McXtrace team but only 2 full time, based at ESS DMSC / DTU

Project website at http://www.mcstas.org mcstas-users@mcstas.org mailinglist
McXtrace - since jan 2009 similar for X-rays

Main Page

McXtrace

McXtrace - Monte Carlo X-ray ray-tracing is a joint venture by

- Risø DTU
- ESRF
- X-RAY

Funding from NABIIT, DSF, and the above parties.

Our code will be based on technology from McStas.

For information on our progress, please subscribe to our user mailinglist.

mailto:webmaster@mcxtrace.org

• Synergy, knowledge transfer, shared infrastructure, repo etc.
Fundina sources

During its lifetime, McStas efforts have been supported through several European Union RTD and JRA programmes, plus several instrument development projects for facilities.

<table>
<thead>
<tr>
<th>Project</th>
<th>Program type</th>
<th>Funding period</th>
</tr>
</thead>
<tbody>
<tr>
<td>XENNI</td>
<td>RTD (EU FP4)</td>
<td>1996 - 2000</td>
</tr>
<tr>
<td>Cool Neutrons</td>
<td>RTD (EU FP4)</td>
<td>1998 - 2001</td>
</tr>
<tr>
<td>SCANS</td>
<td>RTD (EU FP5)</td>
<td>2000 - 2004</td>
</tr>
<tr>
<td>MCNSI</td>
<td>JRA in NMI3 (EU FP6)</td>
<td>2004 - 2006</td>
</tr>
<tr>
<td>MCNSI7</td>
<td>JRA in NMI3/FP7 (EU FP7)</td>
<td>2006 - 2008</td>
</tr>
<tr>
<td>NMI3-II/FP7 outreach project</td>
<td>JRA in NMI3/FP7 (EU FP7)</td>
<td>2012 - 2016</td>
</tr>
<tr>
<td>ISIS TS2 EU project</td>
<td>Infrastructure project in (EU FP6)</td>
<td>2006-2009</td>
</tr>
<tr>
<td>Instrument simulations for the ESS design update</td>
<td>Danish in-kind project toward the ESS</td>
<td>2009-2012</td>
</tr>
<tr>
<td>Secondment of P Willendrup from DTU for supporting instrument simulations for the ESS</td>
<td>33% part of the ESS DMSC</td>
<td>2014-2022</td>
</tr>
<tr>
<td>SINE2020</td>
<td>Part of WP3 and WP8 (EU H2020)</td>
<td>2016 - 2019</td>
</tr>
<tr>
<td>HighNESS</td>
<td>Contributions to WPs 6,8,9 (EU H2020)</td>
<td>2020 - 2023</td>
</tr>
<tr>
<td>Secondment of P Willendrup from DTU for supporting instrument simulations for the ESS</td>
<td>100% part of the ESS DMSC</td>
<td>2023-2027</td>
</tr>
</tbody>
</table>
McStas: simulation toolkit for neutron scattering instruments, virtual experiments

Starts with a source of neutrons, be it a reactor- or spallation source

Neutron moderators is where McStas starts

Sample-environments - Union subsystem

Detectors

Scientific model-samples

Neutron optics

2020: McStas 3.0 with support for MPI and multiple GPUs
McStas 3.x acc. on NVIDIA GPUs
- 2 orders of magnitude speedup. (1x Tesla V100 vs 1 modern Intel Xeon core)

McStas 25 year celebration
Peter Willendrup, DTU Physics and ESS DMSC
Used in many places
Most were designed / simulated with the help of McStas :-)

**Instrument suite at ESS**

15 instruments
- 5 Large-scale structure instruments
- 5 Diffractometers
- 5 Spectrometers

**Areas of research**
- life sciences
- soft condensed matter
- chemistry of materials
- energy research
- magnetism & superconductivity
- engineering & geo-sciences
- archeology & heritage conservation
- particle physics
McStas and McXtrace are Monte Carlo ray-tracers

1. Particles emitted with random starting conditions via MC

2. Particles are "ray-traced" through space

3. Will eventually meet other objects e.g. a studied experimental sample and get scattered via MC again

4. At various points in the instrument the particle states are measured in so-called monitors or detectors

• For the neutrons, gravity kicks in... A cold neutron falls ~10cm over 150m!

• Classical Newtonian mechanics, i.e.

  (independent, particles though...)

• McStas and McXtrace are Monte Carlo ray-tracers
McStas and McXtrace
Monte Carlo ray-tracers

The "tool layer" consists of programs manipulated by the McStas user:
- mcgui, graphical user interface
- mcplot, visualize histogram output
- mcdisplay, visualize instrument

mcgui is used to assemble an instrument file, which is taken over by the McStas system

```
DEFINE INSTRUMENT Example(Param1=1, string Param2="two", ...)
COMPONENT A = Source(Parameters...) AT (0, 0, 0) ABSOLUTE
COMPONENT B = Guide(Parameters...) AT (0, 0, 1) RELATIVE A
COMPONENT C = DiskChopper(Parameters...) AT (0, 0, 1) RELATIVE B
COMPONENT D = TOF_monitor(Parameters, filename="Tof.dat") AT (0, 0, Param1) RELATIVE PREVIOUS
```

"Instrument file"

The McStas system generates an "ISO C file" and an executable from instrument file and c-codes

The simulation executable produces data output which can be visualized using the mcplot and mcdisplay tools
McStas tech overview

- Portable code (Unix/Linux/Mac/Windoze)
  - On the CPU-side, ran on everything from iPhone to 1000+ node cluster, intel, Alpha, PA-RISC etc.

- ‘Component' files (>300) inserted from library
  - Sources
  - Optics
  - Samples
  - Monitors
  - If needed, write your own comps - many are USER developments ~200-line “physicist” codes

- DSL + ISO-C code-gen. (compiler technology / LeX+Yacc)
  - Simple Instrument language → ISO C
  - Component codes realizing beamline parts (including user contribs)

- Library of common functions
  - I/O
  - Random numbers
  - Physical constants
  - Propagation
  - Precession in fields
  - ...

User experience:
- Write instrument
- Launch simulation (generates binary and runs simulation)
- Look at output data
McStas: simulation toolkit for neutron scattering instruments, V.E.

... in .py / Jupyter notebooks using McStasscript
McStas is by design a “linear chain” of components

• But:
  • We have syntaxes/logic to e.g. GROUP components. (Think: XOR and similar logic)

{SPLIT} COMPONENT name = comp(parameters) {WHEN condition}
AT (...) [RELATIVE [reference] | PREVIOUS | ABSOLUTE]
{ROTATED {RELATIVE [reference] | PREVIOUS | ABSOLUTE} }
{GROUP group_name}
{EXTEND C_code}
{JUMP [reference] | PREVIOUS | MYSELF | NEXT} [ITERATE number_of_times | WHEN condition] }
McStas is by design a “linear chain” of components

- But:
  - We can also use syntaxes/logics to e.g.
    - GROUP components. (Think: XOR and similar logic)

```
{SPLIT} COMPONENT name = comp(parameters) {WHEN condition}
AT (...) [RELATIVE [reference|PREVIOUS] | ABSOLUTE]
{ROTATED {RELATIVE [reference|PREVIOUS] | ABSOLUTE} }
{GROUP group_name}
{EXTEND C_code}
{JUMP [reference|PREVIOUS\|MYSELF\|NEXT] [ITERATE number_of_times | WHEN condition] }
```

- Material-assemblies may be arranged in “concentric” onion-shells (Single_crystal, PowderN, …)
McStas is by design a “linear chain” of components

- Material-assemblies may be arranged in “concentric” onion-shells (Single_crystal, PowderN, ...)

- The Union subsystem (Mads Bertelsen) has been added, defining region(s) of the instrument where geometry and materials are decoupled and we completely deviate from the linear approximation

- NCrystal may be used to describe materials, also within Union. cfg="materials_galore.ncmat"
McStas is by design a “linear chain” of components

(Other recent grammar additions…. )

• But:
  • “CPU” to label non-GPU execution of a component in mixed CPU/GPU setups
  • “SHELL” for cogen/compile-time system-hacks (create an instr snippet using e.g. bash)

SEARCH “/the/path/to/add/” COMPONENT …
SEARCH SHELL "the_executable --and --some --options" COMPONENT …

• Material-assemblies may be arranged in “concentric” onion-shells (Single_crystal, PowderN, …)

COMPONENT Origin = Progress_bar()
  AT (0,0,0) ABSOLUTE
METADATA JSON eniius_data %{
  ...{JSON string}...
%
METADATA Python extra_function %{
  def the_function(a, b, c):
    return a + b / c
%

• NCrystal may be used to describe materials, also within Union. cfg="materials_galore.ncmat"

• The Union subsystem (Mads Bertelsen) has been added, defining region(s) of the instrument where geometry and materials are completely decoupled and we completely deviate from the linear approximation

• NCrystal may be used to describe materials, also within Union. cfg="materials_galore.ncmat"
Writing new comps or understanding existing is not so complex...

• Many of them are quite simple and short... Statistics:

- 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
Component lib development

More than 350 comps, more than 250 instr examples
Stats and information on the codebase

In a Nutshell, McCode...

... has had 24,640 commits made by 60 contributors representing 2,177,413 lines of code.

... is mostly written in Postscript with a low number of source code comments.

... has a well established, mature codebase maintained by a large development team with increasing Y-O-Y commits.

... took an estimated 636 years of effort (COCOMO model) starting with its first commit in June, 1997 ending with its most recent commit about 1 month ago.

McStas 25 year celebration
Peter Willendrup, DTU Physics and ESS DMSC

Lines of Code

nope, that's our DSL and grammar. :-) Which is close to "English".

https://www.openhub.net/p/mccode
Stats and information on the codebase

In a Nutshell, McCode...

... has had 24,640 commits made by 60 contributors representing 2,177,413 lines of code.

... is mostly written in Postscript with a low number of source code comments.

... has a well established, mature codebase maintained by a large development team with increasing Y-O-Y commits.

... took an estimated 636 years of effort (COCOMO model) starting with its first commit in June, 1997 ending with its most recent commit about 1 month ago.

McStas 25 year celebration
Peter Willendrup, DTU Physics and ESS DMSC

Lines of Code

nope, that's our DSL and grammar. :-) Which is close to "English".
McStas timeline at a glance

When did what functionality arrive

- **v1.0**, October 15, 1998
  - Windows supp.
  - GPL license,
  - new tools

- **v1.12c**, 2008-2011
  - McXtrace project start,
  - ESS-oriented simulation
  - work, workshop efforts take off
  - v1.12c is last 1.x release

- **v2.3-2.4 x**, 2015-2017
  - ESS_butterfly,
  - MCPL
  - Union subsystem,
  - Python tools fully default

- **v3.0-3.1**, 2020-2021
  - Official GPU support arrives

- **v3.2, 2022**
  - Mature GPU support

- **v2.7.2 is last 2.x release**

- **v3.3-3.4, 2023**
  - Embedded NeXus,
  - mcstas-pygen,
  - McStasscript
  - embedded

Steady in-flow of “smaller” developments, bugfixes, user contributions…
McStas collaboration

Thanks and acknowledgements to lots of people…

Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik)
architect behind “internals” and LeX-Yacc gram.
Releases 1.0-1.4

Kim Lefmann
Author of many components, orig. “power user”,
 supervised many KU stud. and secured funding.
Part of McStas all along the way.

Kurt Clausen:
Came up with original idea of “framework”,
secured first EU funding.

Emmanuel Farhi
First non-DK “McStas system developer”,
PW mentor for the deeper tech.
Grammar ext.
advanced comps. + lots more
Now works mainly on McXtrace.

Henrik Rønnow
Drafted v 1.0 grammar w. Kim & Kristian
McStas collaboration

Thanks and acknowledgements to lots of people…

Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.

Releases 1.0-1.4

Kim Lefmann
Author of many components, orig. “power user”, supervised many KU stud. and secured funding. Part of McStas all along.

Peter Christiansen
1st pol. solution

Kurt Clausen
Came up with original idea of “framework”, secured first EU funding.

Emmanuel Farhi
First non-DK “McStas system developer”, PW mentor for the deeper tech.

Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.

Henrik Rønnow
Drafted v 1.0 grammar w. Kim & Kristian

PW:
“Mr. McStas” 2002- ?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.

Kurt Clausen
Came up with original idea of “framework”, secured first EU funding.

Peter Christiansen
1st pol. solution

Emmanuel Farhi
First non-DK “McStas system developer”, PW mentor for the deeper tech.

Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.

Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.

Releases 1.0-1.4

Kim Lefmann
Author of many components, orig. “power user”, supervised many KU stud. and secured funding. Part of McStas all along.

Peter Christiansen
1st pol. solution

Kurt Clausen
Came up with original idea of “framework”, secured first EU funding.

Emmanuel Farhi
First non-DK “McStas system developer”, PW mentor for the deeper tech.

Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.

Henrik Rønnow
Drafted v 1.0 grammar w. Kim & Kristian

PW:
“Mr. McStas” 2002- ?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.
McStas collaboration

Thanks and acknowledgments to lots of people...

Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.
Releases 1.0-1.4

Kim Lefmann
Author of many components, orig. “power user”, supervised many KU stud. and secured funding.
Part of McStas all along the way.

Kim Lefmann
Author of many components, orig. “power user”, supervised many KU stud. and secured funding.
Part of McStas all along the way.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Erik Rønnow
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Emmanuel Farhi
First non-DK “McStas system developer”, PW mentor for the deeper tech.
Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.

PW:
“Mr. McStas” 2002-?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Erik Rønnow
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Erik Rønnow
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Erik Rønnow
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Erik Rønnow
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Erik Rønnow
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Erik Rønnow
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Mads Bertelsen
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.
McStas collaboration

Thanks and acknowledgements

Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.
Releases 1.0-1.4

Kim Lefmann
Author of many components, orig. “power user”, supervised many KU students, secured funding.
Part of McStas all along the way.

Part of McStas all along the way.

Kurt Clausen
Came up with original idea of “framework”, secured first fund.

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace.
+ lots more

Peter Christiansen:
1st pol. solution

PW:
“Mr. McStas” 2002-?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.

Mads Bertelsen
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas:
guide_bot and Union systems.

Johan Brinch
Transfer to CMake
First round of .py tools, GPU deeper tech.
Grammar ext. advanced comps.
+ lots more
Now works mainly on McXtrace.

Emmanuel Farhi
First non-DK “McStas system developer”, PW mentor for the deeper tech.
Grammar ext. advanced comps.
+ lots more
Now works mainly on McXtrace.

Esben Klinkby
Mcnp-hooks, scatter-logger

Johan Brinch
Transfer to CMake
First round of .py tools, GPU deeper tech.
Grammar ext. advanced comps.
+ lots more
Now works mainly on McXtrace.

Erik Rønnow
Drafted v 1.0 grammar w. Kim & Kristian

Kurt Clausen:
Came up with original idea of “framework”, secured first fund.

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace.
+ lots more

Peter Christiansen:
1st pol. solution

PW:
“Mr. McStas” 2002-?
Custodian, community caretaker, sustained workforce.
Support for newcomers and power users alike.

Mads Bertelsen
“Next generation”, Phd with KL, contributed adv. systems “around” and “in” McStas:
guide_bot and Union systems.

Johan Brinch
Transfer to CMake
First round of .py tools, GPU deeper tech.
Grammar ext. advanced comps.
+ lots more
Now works mainly on McXtrace.

Emmanuel Farhi
First non-DK “McStas system developer”, PW mentor for the deeper tech.
Grammar ext. advanced comps.
+ lots more
Now works mainly on McXtrace.

Esben Klinkby
Mcnp-hooks, scatter-logger
Thanks and acknowledgements to lots of people…

**Kristian Nielsen**
Got input from RISØ physicists, (Incl. Kim & Henrik) architect behind “internals” and LeX-Yacc gram.

Releases 1.0-1.4

**Kim Lefmann**
Author of many components, orig. “power user”, supervised many KU stud. and secured funding.

Part of McStas all along the way.

**Henrik Rønnow**
Drafted v 1.0 grammar w. Kim & Kristian

**Kurt Clausen**
Came up with original idea of “framework”, secured first EU funding.

**Mads Bertelsen**
“Next generation”, PhD with KL, contributed adv. systems “around” and “in” McStas: guide_bot and Union systems.

**Erik Rønnow**
“Mr. McStas” 2002- ?
Custodian, community caretaker, sustained workforce.

Support for newcomers and power users alike.

**Peter Christiansen**
1st pol. solution

**Erik Knudsen**
Developer of prod. pol.env., main historical workforce behind McXtrace. + lots more

Left for CPH Atomics in 2022.

**Esben Klinkby**
Mcnp-hooks, scatter-logger

**Jakob Garde**
2nd round of .py tools, GPU deeper tech.

**Thomas Kittelmann**
Main developer of MCPL particle list Format and NCrystal structure/dynamics lib for MC

**Gregory Tucker**
ESS event-formation hookup, recent syntax / code-generator additions

**Johan Brinch**
Transfer to CMake 1st round of .py tools

**Torben R. Nielsen**
Contributed solution for connection with “Mantid” data reduction, work on SASmodel integration
McStas collaboration

Thanks and acknowledgements to lots of people…

Kristian Nielsen
Got input from RISØ physicists, (Incl. Kim & Henrik)
architect behind “internals” and LeX-Yacc gram.

Releases 1.0-1.4

Kim Lefmann
Author of many components, orig. “power user”,
supervised many KU stud. and secured funding.
Part of McStas all along the way.

Henrik Rønnow
Drafted v 1.0 grammar w. Kim & Kristian

Kurt Clausen:
Came up with original idea of “framework”,
secured first EU funding.

Emmanuel Farhi
First non-DK “McStas system developer”,
PW mentor for the deeper tech.
Grammar ext. advanced comps. + lots more
Now works mainly on McXtrace.

Mads Bertelsen
“Next generation”, Phd with KL, contributed adv. sys. “around” and “in” McStas:
guide_bot and Union systems.

PW:
“Mr. McStas” 2002-?
Custodian, community caretaker, sustained workforce.
Support for newcomers

Esben Klinkby
Mcnp-hooks, scatter-logger

Johan Brinch
Transfer to CMake
1st round of .py tools

Torben R. Nielsen:
Contributed solution for connection with “Mantid” data reduction, work on SASmodel integration

Gregory Tucker
ESS event-formation hook up, recent syntax / code-generator additions

Thomas Kittelmann
Main developer of MCPL particle list Format and NCrystal structure/dynamics lib for MC

Peter Christiansen
1st pol. solution

Erik Knudsen:
Developer of prod. pol.env., main historical workforce behind McXtrace.

Left for CPH Atomics in 2022.

Johan Brinch Transfer to CMake
1st round of .py tools

Plus MANY others among the user community at neutron-facilities, students etc.
Demo time?? :-)

McStas

v. 1.0 1998
v. 3.4 2023

25th ANNIVERSARY