Motivation to contribute (same as from the last meeting)

- The motivation to contribute to CASCADE (and RAPGAP) **so far** comes from the Data Preservation for HEP perspective. It is important to assure the MC event generators relevant for (mostly) ep physics have interfaces to modern tools and the produced events can be **easily** and robustly passed to the detector simulation.
- The focus of contribution is **HepMC(3) interface** and **portability**.
- Some other improvements come on top.

Updates since the last meeting

What was done:

- Cleanup of unneeded files, request for missing files – **merged**.
- Implementation of cmake buid system – **merged**.
- Update of HepMC2 interface – **under review, cmake included**, see
  `https://gitlab.cern.ch/jung/cascade/-/merge_requests/4`
- Implementation of HepMC3 interface – **under review, cmake included**, see
  `https://gitlab.cern.ch/jung/cascade/-/merge_requests/4`

Nearest future

What can be done:

- cmake makes possible to do tests with multiple available compilers: NVidia (ex-PGI)@Linux, Intel@Linux, Sun(Oracle)@Linux, IBM XL@Linux. **In Fortran codes this is the way to find bugs!**
- Fix warnings from the compilers above.
- Check for floating point exceptions and fix them.
- Add support for more compillers, e.g. NAG Fortran.

How that looks in practice: use e.g. Intel compiler

```
[user@host mastercascade]$ source /opt/intel/oneapi/setvars.sh

:: initializing oneAPI environment ...
   bash: BASH_VERSION = 4.4.20(1)-release
.................................SKIP.....................................
```

How that looks in practice: configuration with cmake

```
[user@host mastercascade]$ cmake -S. -B mydirforintelbuild  -DCMAKE_Fortran_COMPILER=
     ifort -DCMAKE_CXX_COMPILER=icpc
-- The C compiler identification is GNU 8.5.0
-- The CXX compiler identification is Intel 2021.4.0.20210910
-- The Fortran compiler identification is Intel 2021.4.0.20210910
.................................SKIP.....................................
-- CASCADE: CASCADE_CREATE_VERSION_FILES  OFF
-- CASCADE: CASCADE_PYTHIA8               ON
-- CASCADE: CASCADE_HEPMC2                ON
-- CASCADE: CASCADE_TMDLIB                ON
-- CASCADE: CASCADE_TMDLIB_EW             ON
-- CASCADE: Fortran_COMPILER_NAME=ifort  CMAKE_Fortran_COMPILER_ID=Intel
-- CASCADE: CXX_COMPILER_NAME=icpc  CMAKE_CXX_COMPILER_ID=Intel
-- CASCADE: C_COMPILER_NAME=cc  CMAKE_C_COMPILER_ID=GNU
-- Found ZLIB: /usr/lib64/libz.so (found version "1.2.11")
-- CASCADE: ZLIB_VERSION_STRING=1.2.11 ZLIB_LIBRARIES=/usr/lib64/libz.so
     ZLIB_INCLUDE_DIRS=/usr/include
-- Found PkgConfig: /usr/bin/pkg-config (found version "1.4.2")
-- Found GSL: /usr/include (found version "2.5")
-- CASCADE: GSL_VERSION=2.5 GSL_LIBRARIES=/usr/lib64/libgsl.so;/usr/lib64/libgslcblas
     .so GSL_INCLUDE_DIRS=/usr/include
.................................SKIP.....................................
-- Generating done
-- Build files have been written to: /home/user/Projects/mastercascade/
     mydirforintelbuild
[user@host mastercascade]$
```

A more complex configuration requirs more flags, e.g.
-DLHAPDF_DIR=/install/prefix/of/lhapdf

How that looks in practice: build

```
[user@host mastercascade]$ cmake --build mydirforintelbuild/
.....................................SKIP.....................................
[ 93%] Building Fortran object CMakeFiles/cascade3.dir/src/meoffchi.F.o
/opt/intel/oneapi/compiler/2021.4.0/linux/bin/intel64/ifort -Dcascade3_EXPORTS -I/
      home/user/Projects/mastercascade/include -save -extend-source 132 -fPIC -c /
      home/user/Projects/mastercascade/src/meoffchi.F -o CMakeFiles/cascade3.dir/src/
      meoffchi.F.o
/home/user/Projects/mastercascade/src/meoffchi.F(764): warning #6178: The return
      value of this FUNCTION has not been defined.   [EPS2]
      DOUBLE PRECISION FUNCTION EPS2(q1,q2,q3,q4)
--------------------------------^
.....................................SKIP.....................................
[ 99%] Building Fortran object CMakeFiles/cascade.dir/src/pydata.F.o
/opt/intel/oneapi/compiler/2021.4.0/linux/bin/intel64/ifort  -I/home/user/Projects/
      mastercascade/include -I/usr/include -save -extend-source 132 -c /home/user/
      Projects/mastercascade/src/pydata.F -o CMakeFiles/cascade.dir/src/pydata.F.o
/home/user/Projects/mastercascade/src/pydata.F(329): remark #7784: Symbol in BLOCK
      DATA program unit is not in a COMMON block.   [PYK]
      INTEGER PYK,PYCHGE,PYCOMP
--------------^
.....................................SKIP.....................................
```

The warnings can be considered and fixed.

Medium-term future

What can be done:

- HepMC3 interfaces provides a way to store extra data in addition to HepMC2 standard content. This can be used to add e.g. information from TMDlib to the event record.
- A code for Rivet interface can be submitted as a MR.
- Add HZTOOL support.
- Use CASCADE (and RAPGAP) to debug some MC validation routines in Rivet.
- Syncronize the CASCADE and RAPGAP codes and build systems.

Technical details on HepMC3 interfaces (merge request 4)

- The difference of older-style "glue code" vs. the current HepMC(3) interface is that the later **is** an interface, i.e. the individual HepMC(3) functions can be called from Fortran and the HepMC(3)::GenEvent object can be build/modified from different Fortran routines.
- HepMC3 is not a file format but a **library**. It allows an output in different formats, e.g. HepMC2-native IO_GenEvent, HepMC3-native Asciiv3, ROOT trees or **custom format**. HepMC3::GenEvent can hold the information specific to CASCADE.

Other notes

The elephant in the room in the CASCADE (and RAPGAP) codes the fact that according to the official GNU docs[1]

"Automake currently provides limited support for creating programs and shared libraries that are a mixture of Fortran 77 and C and/or C++."

Means the projects with mixed of C++/Fortran codes and libraries will not build easily in anything but GNU environment with GNU compilers. And the chances to build a typical physics software on something more exotic (e.g. Windows) with autotools is 0. **Big problem for software development and preservation.**

[1] `https://www.gnu.org/software/automake/manual/html_node/Mixing-Fortran-77-With-C-and-C_002b_002b.html`