The status of the WHIZARD package

Christian Speckner

in collaboration with

W. Kilian, T. Ohl and J. Reuter

Universität Freiburg

June 30th, 2010
TOOLS 2010 Winchester

Outline:

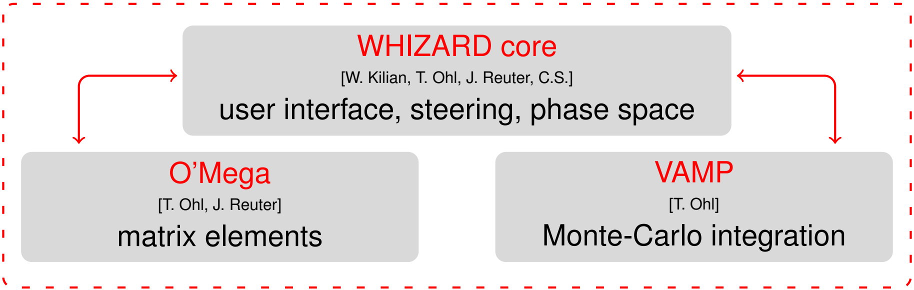
1. An overview over the WHIZARD package
2. Commanding the WHIZARD
3. New spells — adding models via FeynRules
4. Conclusions

What is WHIZARD?

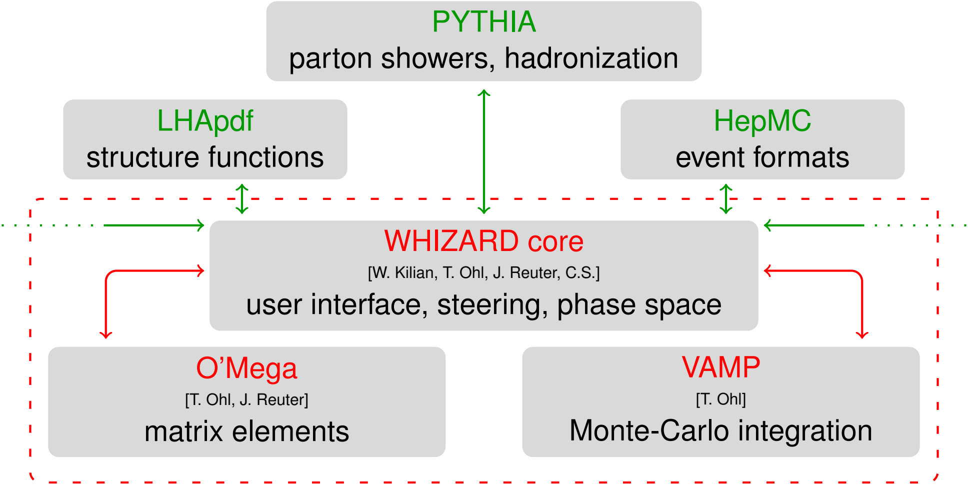
Verbatim from the website:

"WHIZARD is a program system designed for the efficient calculation of multi-particle scattering cross sections and simulated event samples."

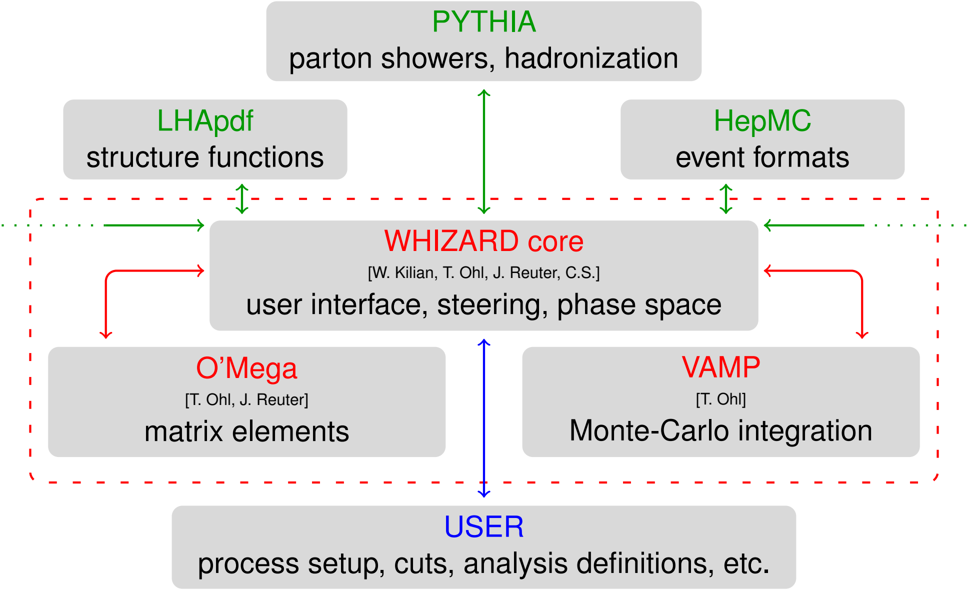
What is WHIZARD?



What is WHIZARD?



What is WHIZARD?

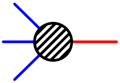


A few words on O'Mega

Algorithm:

- 1-particle off-shell wavefunction (1POW):

$$\langle \text{in} | \, \phi(x) \, | 0 \rangle \quad =$$



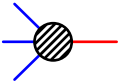
- Number of 1POWs grows exponentially
- Use 1POWs instead Feynman Diagrams $\longleftrightarrow$ exponential complexity (instead of factorial)
- 1POWs satisfy Ward identity $\longrightarrow$ nontrivial gauge cancellations in every step, numerical stability

A few words on O'Mega

Algorithm:

- 1-particle off-shell wavefunction (1POW):

$$\langle \text{in} | \, \phi(x) \, | 0 \rangle \quad =$$



- Number of 1POWs grows exponentially
- Use 1POWs instead Feynman Diagrams $\longleftrightarrow$ exponential complexity (instead of factorial)
- 1POWs satisfy Ward identity $\longrightarrow$ nontrivial gauge cancellations in every step, numerical stability

Implementation:

- Written in O'Caml (impure functional language)
- Graph of 1POWs transformed into FORTRAN 95 code
- Numerical calculation of helicity amplitudes
- No limit on the arity of vertices

The past: WHIZARD 1.x (current: 1.95)

- Matrix elements: O'Mega, (Madgraph, CompHEP)
- Phase space parametrization automatically tailored to matrix element
- Beam modelling:
  - ILC: polarization, ISR, beamstrahlung via CIRCE / CIRCE2
  - LHC / Tevatron: parton distributions via PDFlib / LHApdf
- Fragmentation and hadronization: PYTHIA
- Event generation:
  - Event output in standard formats (e.g. LHA, StdHEP),
  - Integrated analysis facilities; histogram output as postscript / pdf
- Many BSM models: MSSM, NMSSM, Little Higgs, UED, Z', ...

The past: WHIZARD 1.x (current: 1.95)

- Matrix elements: O'Mega, (Madgraph, CompHEP)
- Phase space parametrization automatically tailored to matrix element
- Beam modelling:
  - ILC: polarization, ISR, beamstrahlung via CIRCE / CIRCE2
  - LHC / Tevatron: parton distributions via PDFlib / LHApdf
- Fragmentation and hadronization: PYTHIA
- Event generation:
  - Event output in standard formats (e.g. LHA, StdHEP),
  - Integrated analysis facilities; histogram output as postscript / pdf
- Many BSM models: MSSM, NMSSM, Little Higgs, UED, Z', ...
- Technicalities:
  - FORTRAN 95 for phase space / infrastructure / physics
  - Makefiles and PERL for steering and code generation
  - Custom input files for process definition (compile time), simulation setup, cut and analysis definitions (runtime)

The present: WHIZARD 2.0.x (current: 2.0.2)

Changes and new features w.r.t. 1.9x:

- Major rewrite of the whole package
- Self-contained FORTRAN 2003 code
- Completely dynamic setup:
  - Process library compiled and dynamically loaded on-the-fly at runtime
  - Scripting language SINDARIN — controls all aspects of the simulation
- Decay chains with full spin and color correlations
- New models: Three-Site Higgsless Model, FeynRules interface
- Enhanced analysis capabilities (observables, plots and histograms)

The present: WHIZARD 2.0.x (current: 2.0.2)

Changes and new features w.r.t. 1.9x:

- Major rewrite of the whole package
- Self-contained FORTRAN 2003 code
- Completely dynamic setup:
  - Process library compiled and dynamically loaded on-the-fly at runtime
  - Scripting language SINDARIN — controls all aspects of the simulation
- Decay chains with full spin and color correlations
- New models: Three-Site Higgsless Model, FeynRules interface
- Enhanced analysis capabilities (observables, plots and histograms)
- To be reenabled in 2.0.3 ($\approx$ 1 – 2 weeks): CIRCE / CIRCE2, hadronization / fragmentation via PYTHIA
- PDFlib, Madgraph and CompHEP support dropped

The future — WHIZARD 2.1 / 3.0

In preparation for version 2.1:

- Parton shower matching (D. Wiesler)
- Intrinsic parton shower implementation (S. Schmidt)
- Intrinsic module for multiple interactions (H.-W. Boschmann)
- Interface to NLO amplitudes (BLHA) and automatic dipole substraction (J. Reuter, S. Schmidt, C.S.)
- Generalized lorentz structures (T. Ohl)
- Parallelization (W. Kilian)

The future — WHIZARD 2.1 / 3.0

In preparation for version 2.1:

- Parton shower matching (D. Wiesler)
- Intrinsic parton shower implementation (S. Schmidt)
- Intrinsic module for multiple interactions (H.-W. Boschmann)
- Interface to NLO amplitudes (BLHA) and automatic dipole substraction (J. Reuter, S. Schmidt, C.S.)
- Generalized lorentz structures (T. Ohl)
- Parallelization (W. Kilian)

Plans for 3.0:

- GPU massive parallel computing
- τ decay module / interface
- Dark matter relic computation (“dark WHIZARD”)

Technical prerequisites:

- The O'Caml compiler and runtime system
- A suitable (2003 support!) FORTRAN compiler:
  - gfortran 4.5.0 and NAGFOR 5.2 work great
  - Intel 11.1 works with some issues
  - g95 and Portland are close
- Optional programs:
  - LHApdf for parton distributions
  - StdHEP, HepMC for additional event formats
  - LaTeX with METAPOST for plots and histograms

Technical prerequisites:

- The O'Caml compiler and runtime system
- A suitable (2003 support!) FORTRAN compiler:
  - gfortran 4.5.0 and NAGFOR 5.2 work great
  - Intel 11.1 works with some issues
  - g95 and Portland are close
- Optional programs:
  - LHApdf for parton distributions
  - StdHEP, HepMC for additional event formats
  - LaTeX with METAPOST for plots and histograms

Installation:

- Fully automated configuration via GNU autotools (automake, autoconf and libtool)
- Can be installed either locally or system-wide
- Actual installation procedure:

```
./configure --prefix=my/installation/prefix
make
make check # Optional but recommended
make install
```

Simple example: *W* pair production at the ILC

SINDARIN input:

```
alias j = u:U:d:D:c:C:s:S:b:B:g
process wwprod = "e+", "e-" => "e-", nuebar, j, j
sqrts = 1 TeV

cuts = all Pt > 1 GeV [j:"e-"]
luminosity = 10 / 1 fbarn

simulate (wwprod)
```

What happens:

1. Code for the tree-level matrix elements is generated, compiled and dynamically loaded
2. The phase space parametrization is generated
3. Adaptive integration: $\rightarrow \sigma = 2.154 \pm 0.004 nb$
4. 21544 events are generated and written out

Total running time on my laptop: $\approx$ 8 minutes

Simulating the same process for the LHC:

SINDARIN input:

```
alias pr = u:U:d:D:c:C:s:S:g
alias j = u:U:d:D:c:C:s:S:b:B:g
process wwprod = pr, pr => "e-", nuebar, j, j

sqrts = 14 TeV
beams = p, p => lhapdf
cuts =
    all Pt > 10 GeV [j:"e-"] and
    all M > 50 GeV [collect [j:"e-"]]
luminosity = 10 / 1 fbarn

simulate (wwprod)
```

Changes w.r.t. the ILC example:

- p, p initial state @ 14 TeV
- Convolution with PDFs $\longrightarrow$ LHApdf
- Additional cut on the (visible) total invariant mass: $m_{\text{tot}} \geq 50$ GeV

Running time (on the same laptop):

$\approx$ 2 hours for integration + $\approx$ 3.5 hours per 10000 events

Some advanced features of SINDARIN:

- Loops and plots:

```
plot mh_dependence
scan mH = (100 GeV => 200 GeV /+ 10 GeV) {
    integrate (hprod)
    record mh_dependence (mH, integral (hprod))
}
```

- Variables and conditionals:

```
if (integral ($proc) > 100 fb) then
    printf "Integral of %s larger than 100 fb" ($proc)
else
    printf "Integral of %s smaller than 100 fb" ($proc)
endif
```

- Advanced cut expressions:

```
cuts =
    all Pt > 100 GeV [collect [neutrinos]]
    and any 75 GeV < M < 85 GeV [j, j]
```

- Running α_S with arbitrary scale choices:

```
scale = eval Pt [extract 1 [sort by Pt [j]]]
```

Histogramming the *W* resonance:

Modified ILC example:

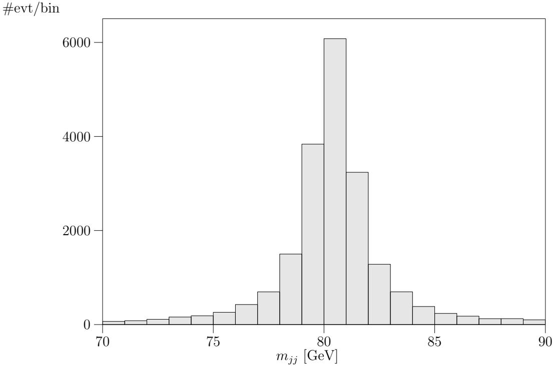
```
alias j = u:U:d:D:c:C:s:S:b:B:g
process wwprod = "e+", "e-" => "e-", nuebar, j, j

sqrts = 1 TeV
cuts = all Pt > 1 GeV [j:"e-"]
luminosity = 10 / 1 fbarn

histogram mjj (70 GeV, 90 GeV, 1 GeV) {
    $title = "jj invariant mass"
    $description = "W resonance in jj invariant mass"
    $xlabel = "$m_jj$ [GeV]"
}

analysis = record mjj (eval M [collect [j]])
simulate (wwprod)

$analysis_filename = "analysis.ps"
write_analysis
```

Sidenote: No new simulation required, WHIZARD will just read in the previously generated Monte Carlo grids and events.

Histogram output:

1 jj invariant mass

W resonance in jj invariant mass



Data within bounds:
$\langle\text{Observable}\rangle = 80.401 \pm 0.017$ [$n_{\text{entries}} = 19771$]

All data:
$\langle\text{Observable}\rangle = 81.62 \pm 0.17$ [$n_{\text{entries}} = 21313$]

Features of the FeynRules interface:

- Supports WHIZARD 2 + legacy versions > 1.92
- Fully validated with: SM + Three-Site Model (N. Christensen), MSSM (B. Fuks)
- Handles all fields supported by FeynRules
- Can do unitarity, Feynman and R_ξ gauges
- Supports (nearly) all dimension 4 operators (+ some higher-order ones)
- Code output is formatted, commented and readable

Features of the FeynRules interface:

- Supports WHIZARD 2 + legacy versions > 1.92
- Fully validated with: SM + Three-Site Model (N. Christensen), MSSM (B. Fuks)
- Handles all fields supported by FeynRules
- Can do unitarity, Feynman and R_ξ gauges
- Supports (nearly) all dimension 4 operators (+ some higher-order ones)
- Code output is formatted, commented and readable

How to get it?

- Will be included in FR 1.5+...
- ... needs to be downloaded separately for FR 1.4.x
- Installation instructions:

`http://projects.hepforge.org/whizard/trac/wiki/WikiStart`

Simple usage example — Standard Model:

1. Minimal Mathematica input:

```
$FeynRulesPath = SetDirectory["."];
«FeynRules`;
LoadModel["SM/SM.fr"];
WriteWOOutput[LSM, WOModelName->"fr_sm", Output->"WO-fr_sm"];
```

2. Run through Mathematica kernel (or type into notebook):

```
math < input.m
```

3. Compile and install the model

```
cd WO-fr_sm
./configure WO_CONFIG=/path/to/whizard/binaries
make install
```

   - Default destination: `${HOME}/.whizard`
   - Other destinations can be selected via `--prefix=...`

4. The model is now ready for use, e.g.

```
model = fr_sm
process eezz = "e+", "e-" => Z, Z
compile
sqrts = 500 GeV
integrate (test)
show (results)
```

Conclusions:

- WHIZARD 2: next-to-full rewrite, major upgrade
- Much improved usability and flexibility (dynamic process libraries, SINDARIN)
- Now supports factorized matrix elements
- Implementing new model significantly simplified via the FeynRules interface
- Many LHC-oriented physics features coming in the near future: matching, parton shower, multiple interactions

`http://projects.hepforge.org/whizard/`

Conclusions:

- WHIZARD 2: next-to-full rewrite, major upgrade
- Much improved usability and flexibility (dynamic process libraries, SINDARIN)
- Now supports factorized matrix elements
- Implementing new model significantly simplified via the FeynRules interface
- Many LHC-oriented physics features coming in the near future: matching, parton shower, multiple interactions

http://projects.hepforge.org/whizard/

Thank you for your attention!