Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

FeynRules
Status and Plans

Claude Duhr

FeynRules 2012, St. Odile 26/03/2012

Outline

- FeynRules in a nutshell.
- The status, or what has happened since FeynRules 2010
- Plans for the future.

Outline

- FeynRules in a nutshell.
- The status, or what has happened since FeynRules 2010
- Plans for the future.
- Disclaimer:

  This whole meeting is supposed to be very informal, with lots of discussions... so feel free to interrupt at any time!

FeynRules
in a nutshell

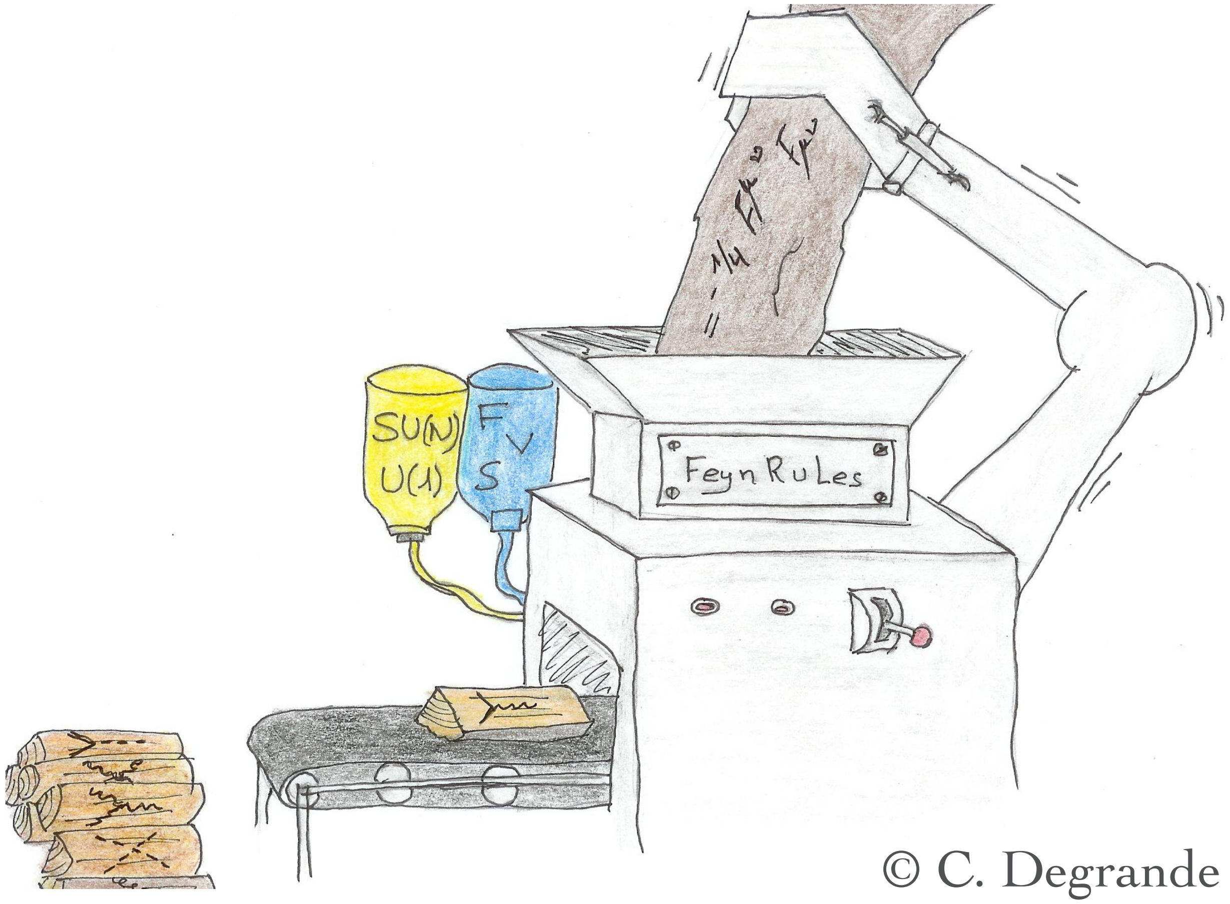
FeynRules in a nutshell

- FeynRules is a Mathematica package that allows to derive Feynman rules from a Lagrangian.
- Current public version: 1.6.x, available from

  http://feynrules.phys.ucl.ac.be
- The only requirements on the Lagrangian are:
  - ➡ All indices need to be contracted (Lorentz and gauge invariance)
  - ➡ Locality
  - ➡ Supported field types: spin 0, 1/2, 1, 2, ghosts & superfields

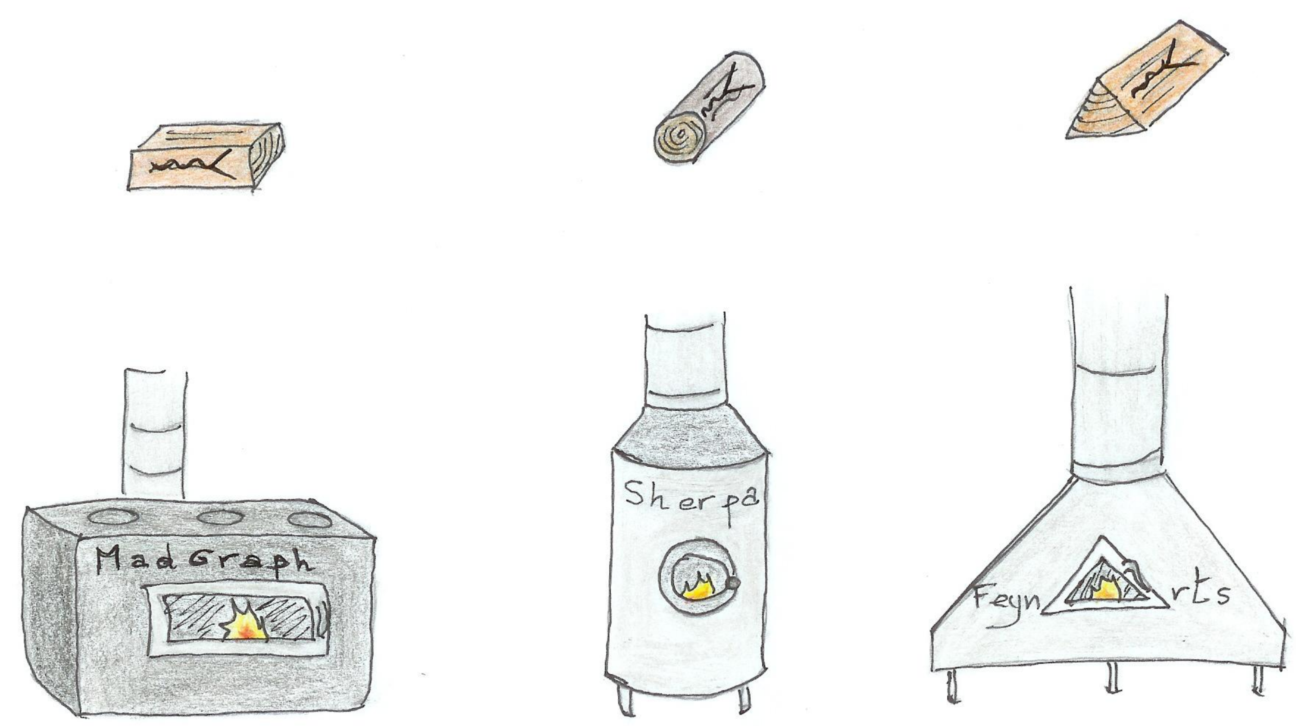
FeynRules in a nutshell

- FeynRules comes with a set of interfaces, that allow to export the Feynman rules to various matrix element generators.
- Interfaces coming with current public version
    - ➡ CalcHep / CompHep
    - ➡ FeynArts / FormCalc
    - ➡ GoSam
    - ➡ MadGraph (4 & 5)
    - ➡ Sherpa
    - ➡ Whizard / Omega



FeynRules in a nutshell

- FeynRules comes with a set of interfaces, that allow to export the Feynman rules to various matrix element generators.
- Interfaces coming with current public version
  - ➡ CalcHep / CompHep
  - ➡ FeynArts / FormCalc
  - ➡ GoSam
  - ➡ MadGraph (4 & 5)
  - ➡ Sherpa
  - ➡ Whizard / Omega



FeynRules in a nutshell

- The input requested form the user is twofold.

- The Model File:
  Definitions of particles and parameters (e.g., a quark)

```
F[1]  ==
  {ClassName      ->   q,
   SelfConjugate ->  False,
   Indices          ->  {Index[Colour]},
   Mass             ->  {MQ,  200},
   Width            ->  {WQ, 5}   }
```

- The Lagrangian:

$$\mathcal{L} = -\frac{1}{4} G^a_{\mu\nu}\, G_a^{\mu\nu} + i\bar{q}\,\gamma^\mu\, D_\mu q - M_q\,\bar{q}\,q$$

```
L =
-1/4 FS[G,mu,nu,a] FS[G,mu,nu,a]
+ I qbar.Ga[mu].DC[q,mu]
- MQ qbar.q
```

FeynRules in a nutshell

- Once this information has been provided, FeynRules can be used to compute the Feynman rules for the model:

```
FeynmanRules[ L ]
```

FeynRules in a nutshell

- Once this information has been provided, FeynRules can be used to compute the Feynman rules for the model:

FeynmanRules[L]

Vertex 1

Particle 1 : Vector , G

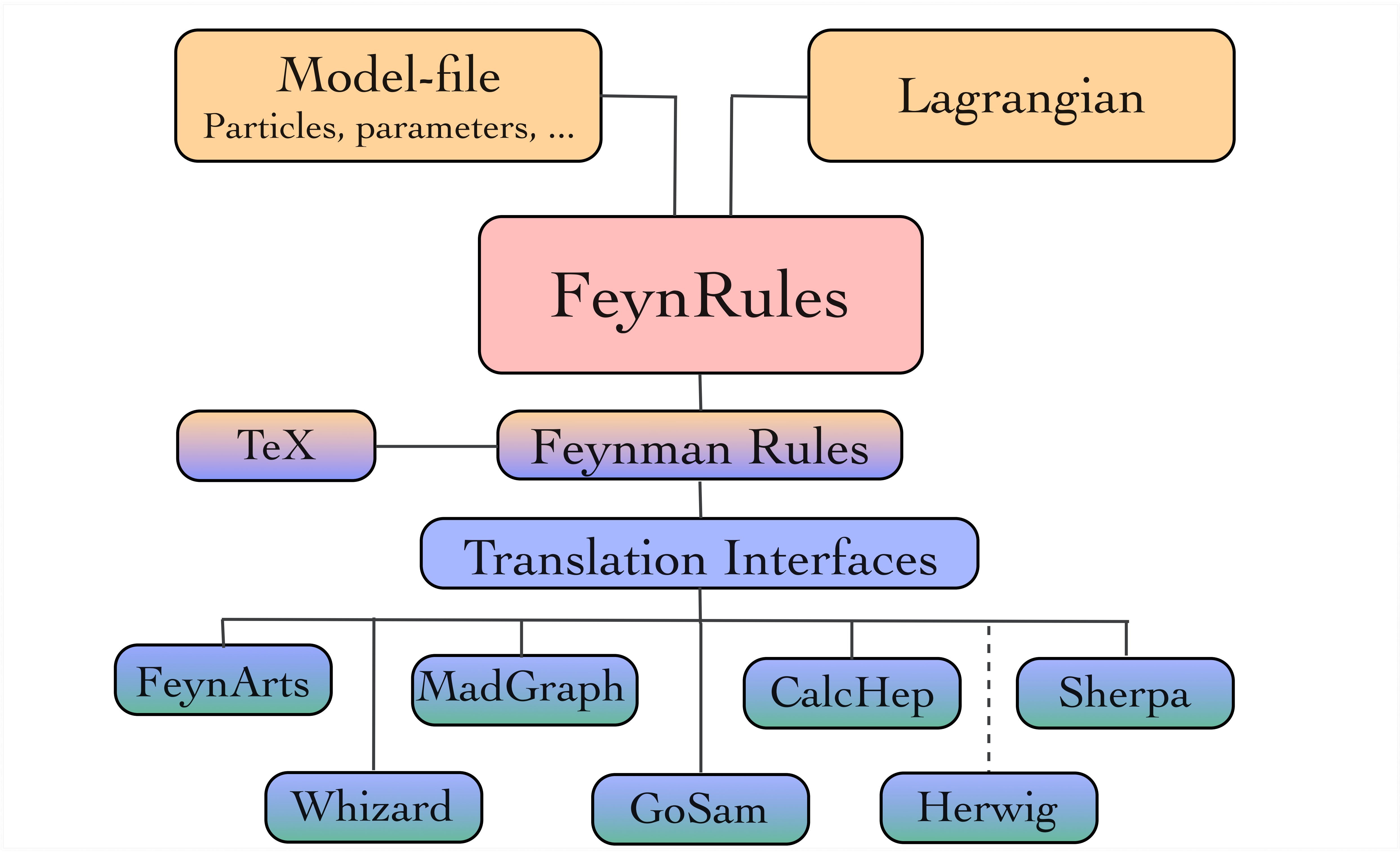
Particle 2 : Dirac , $q^\dagger$

Particle 3 : Dirac , q

Vertex:

$$i\,\mathrm{gs}\,\gamma^{\mu_1}{}_{\mathrm{s}_2,\mathrm{s}_3}\,\delta_{\mathrm{f}_2,\mathrm{f}_3}\,T^{\mathrm{a}_1}{}_{\mathrm{i}_2,\mathrm{i}_3}$$

FeynRules



Model-file
Particles, parameters, ...
Lagrangian
FeynRules
TeX
Feynman Rules
Translation Interfaces
FeynArts
MadGraph
CalcHep
Sherpa
Whizard
GoSam
Herwig

FeynRules 2010

FeynRules 2010

- A big fraction of the projects that brought FeynRules to the state it is in now were initiated during the last workshop:
  - ➡ Superfields in FeynRules.
  - ➡ More support for higher dimensional operators:
    - higher dimensional operators in FeynArts.
    - UFO - ALOHA.
  - ➡ Automated web validation platform.

Superfields

[CD, Fuks]

- FeynRules 1.4.x required the Lagrangian to be written in component fields.

Superfields

[CD, Fuks]

- FeynRules 1.4.x required the Lagrangian to be written in component fields.

$$\mathcal{L} = \Phi^\dagger e^{-2gV}\Phi_{|_{\theta^2\bar{\theta}^2}} + \frac{1}{16g^2\tau_{\mathcal{R}}}\mathrm{Tr}(W^\alpha W_\alpha)_{|_{\theta^2}} + \frac{1}{16g^2\tau_{\mathcal{R}}}\mathrm{Tr}(\bar{W}_{\dot{\alpha}}\bar{W}^{\dot{\alpha}})_{|_{\bar{\theta}^2}}$$
$$+ W(\Phi)_{|_{\theta^2}} + W^\star(\Phi^\dagger)_{|_{\bar{\theta}^2}} + \mathcal{L}_{\mathrm{soft}}$$

Superfields

[CD, Fuks]

- FeynRules 1.4.x required the Lagrangian to be written in component fields.

$$\mathcal{L} = \Phi^\dagger e^{-2gV}\Phi|_{\theta^2\bar{\theta}^2} + \frac{1}{16g^2\tau_{\mathcal{R}}}\mathrm{Tr}(W^\alpha W_\alpha)|_{\theta^2} + \frac{1}{16g^2\tau_{\mathcal{R}}}\mathrm{Tr}(\bar{W}_{\dot{\alpha}}\bar{W}^{\dot{\alpha}})|_{\bar{\theta}^2}$$
$$+ W(\Phi)|_{\theta^2} + W^\star(\Phi^\dagger)|_{\bar{\theta}^2} + \mathcal{L}_{\mathrm{soft}}$$

- 'Monte Carlo description':
  - ➡ Express superfields in terms of component fields.
  - ➡ Express everything in terms of 4-component fermions (beware of the Majoranas!).
  - ➡ Integrate out D and F terms.

Superfields

- FeynRules 1.6.x allows to define superfields directly:

```
CSF[1] == { ClassName  -> ER,
            Chirality     -> Left,
            Weyl           -> ERw,
            Scalar         -> ERs,
            QuantumNumbers -> {Y-> 1},
            Indices       -> {Index[GEN]}
          }
```

- The F term does not need to be defined, but is added automatically.
- Once the superfields (and their component fields) have been defined, FeynRules takes care of the rest.

Higher-dimensional operators

- Even though FeynRules 1.4.x could already compute the Feynman rules for higher-dimensional operators, they were ‘useless’, in the sense that they could be exported to almost no Monte Carlo code.
- Reason: Most Monte Carlo codes have internal limitations for the vertices:
  - ➡ hardcoded library of color and/or Lorentz structures.
  - ➡ Upper limit on the number of particles enter in a vertex (usually 4).
- To overcome this problem, a joint effort between the FeynRules team and the MC developers was needed!

The UFO

[Degrande, CD, Fuks, Grellscheid, Mattelaer, Reiter]

UFO = Universal FeynRules Output

- Idea: Create Python modules that can be linked to other codes and contain all the information on a given model.
- The UFO is a self-contained Python code, and not tied to a specific matrix element generator.
- The content of the FR model files, together with the vertices, is translated into a library of Python objects, that can be linked to other codes.
- By design, the UFO does not make any assumptions on Lorentz/color structures, or the number of particles.
- GoSam and MadGraph 5 use the UFO as the default model format for BSM, Herwig++ will use it in the future.

The UFO & ALOHA

- The development of the UFO goes hand in hand with the development of ALOHA.
- Idea: ALOHA uses the information contained in the UFO to create the (previously-hardcoded) library of Lorentz structures for MadGraph 5 on the fly.
  - ➡ See Olivier Mattelaer's talk.

The UFO & ALOHA

- The development of the UFO goes hand in hand with the development of ALOHA.
- Idea: ALOHA uses the information contained in the UFO to create the (previously-hardcoded) library of Lorentz structures for MadGraph 5 on the fly.
  - ➡ See Olivier Mattelaer's talk.

```
FFV1 = Lorentz(name = 'FFV1',
               spins = [ 2, 2, 3 ],
               structure = 'Gamma(3,2,1)')
```

The UFO & ALOHA

- A neat application...

The UFO & ALOHA

- A neat application... in supergravity!

The UFO & ALOHA

- A neat application... in supergravity!

$$\mathcal{L} = -\frac{1}{4} F^a_{\mu\nu} F_a^{\mu\nu} + \lambda f^{abc} \mathrm{Tr}(F^a_{\mu\nu} F_b^{\nu\rho} F^c_{\rho\mu})$$

- Broedel and Dixon have a CSW construction for the color-ordered helicity amplitudes, but had no way to check the validity of the construction.

The UFO & ALOHA

- A neat application... in supergravity!

$$\mathcal{L} = -\frac{1}{4} F^a_{\mu\nu} F_a^{\mu\nu} + \lambda f^{abc} \mathrm{Tr}(F^a_{\mu\nu} F_b^{\nu\rho} F^c_{\rho\mu})$$

- Broedel and Dixon have a CSW construction for the color-ordered helicity amplitudes, but had no way to check the validity of the construction.
- Solution:
  - ➡ Put it into FeynRules, and let it run for a long time...
  - ➡ Get the UFO, and put it into MadGraph 5.
  - ➡ Hack matrix.f to read out the color-ordered helicity amplitudes for individual phase space points.

The UFO & ALOHA

```
V_4 = Vertex(name = 'V_4',
             particles = [ P.G, P.G, P.G, P.G, P.G, P.G ],
             color = [ 'f(-2,-3,-1)*f(-1,1,2)*f(3,4,-2)*f(5,6,-3)', 'f(-2,-3,-1)*f(-1,1,2)*f(3,4,-3)*f(5,6,-2)', 'f(-2,-3,-1)*f(-1,1,2)*f(3,5,-2)*f(4,6,-3)', 'f
(-2,-3,-1)*f(-1,1,2)*f(3,5,-3)*f(4,6,-2)', 'f(-2,-3,-1)*f(-1,1,2)*f(3,6,-2)*f(4,5,-3)', 'f(-2,-3,-1)*f(-1,1,2)*f(3,6,-3)*f(4,5,-2)', 'f(-2,-3,-1)*f(-1,1,3)*f(2,4,-2)*f
(5,6,-3)', 'f(-2,-3,-1)*f(-1,1,3)*f(2,4,-3)*f(5,6,-2)', 'f(-2,-3,-1)*f(-1,1,3)*f(2,5,-2)*f(4,6,-3)', 'f(-2,-3,-1)*f(-1,1,3)*f(2,5,-3)*f(4,6,-2)', 'f(-2,-3,-1)*f(-1,1,3)
*f(2,6,-2)*f(4,5,-3)', 'f(-2,-3,-1)*f(-1,1,3)*f(2,6,-3)*f(4,5,-2)', 'f(-2,-3,-1)*f(-1,1,4)*f(2,3,-2)*f(5,6,-3)', 'f(-2,-3,-1)*f(-1,1,4)*f(2,3,-3)*f(5,6,-2)', 'f
(-2,-3,-1)*f(-1,1,4)*f(2,5,-2)*f(3,6,-3)', 'f(-2,-3,-1)*f(-1,1,4)*f(2,5,-3)*f(3,6,-2)', 'f(-2,-3,-1)*f(-1,1,4)*f(2,6,-2)*f(3,5,-3)', 'f(-2,-3,-1)*f(-1,1,4)*f(2,6,-3)*f
(3,5,-2)', 'f(-2,-3,-1)*f(-1,1,5)*f(2,3,-2)*f(4,6,-3)', 'f(-2,-3,-1)*f(-1,1,5)*f(2,3,-3)*f(4,6,-2)', 'f(-2,-3,-1)*f(-1,1,5)*f(2,4,-2)*f(3,6,-3)', 'f(-2,-3,-1)*f(-1,1,5)
*f(2,4,-3)*f(3,6,-2)', 'f(-2,-3,-1)*f(-1,1,5)*f(2,6,-2)*f(3,4,-3)', 'f(-2,-3,-1)*f(-1,1,5)*f(2,6,-3)*f(3,4,-2)', 'f(-2,-3,-1)*f(-1,1,6)*f(2,3,-2)*f(4,5,-3)', 'f
(-2,-3,-1)*f(-1,1,6)*f(2,3,-3)*f(4,5,-2)', 'f(-2,-3,-1)*f(-1,1,6)*f(2,4,-2)*f(3,5,-3)', 'f(-2,-3,-1)*f(-1,1,6)*f(2,4,-3)*f(3,5,-2)', 'f(-2,-3,-1)*f(-1,1,6)*f(2,5,-2)*f
(3,4,-3)', 'f(-2,-3,-1)*f(-1,1,6)*f(2,5,-3)*f(3,4,-2)', 'f(-2,-3,-1)*f(-1,2,3)*f(1,4,-2)*f(5,6,-3)', 'f(-2,-3,-1)*f(-1,2,3)*f(1,4,-3)*f(5,6,-2)', 'f(-2,-3,-1)*f(-1,2,3)
*f(1,5,-2)*f(4,6,-3)', 'f(-2,-3,-1)*f(-1,2,3)*f(1,5,-3)*f(4,6,-2)', 'f(-2,-3,-1)*f(-1,2,3)*f(1,6,-2)*f(4,5,-3)', 'f(-2,-3,-1)*f(-1,2,3)*f(1,6,-3)*f(4,5,-2)', 'f
(-2,-3,-1)*f(-1,2,4)*f(1,3,-2)*f(5,6,-3)', 'f(-2,-3,-1)*f(-1,2,4)*f(1,3,-3)*f(5,6,-2)', 'f(-2,-3,-1)*f(-1,2,4)*f(1,5,-2)*f(3,6,-3)', 'f(-2,-3,-1)*f(-1,2,4)*f(1,5,-3)*f
(3,6,-2)', 'f(-2,-3,-1)*f(-1,2,4)*f(1,6,-2)*f(3,5,-3)', 'f(-2,-3,-1)*f(-1,2,4)*f(1,6,-3)*f(3,5,-2)', 'f(-2,-3,-1)*f(-1,2,5)*f(1,3,-2)*f(4,6,-3)', 'f(-2,-3,-1)*f(-1,2,5)
*f(1,3,-3)*f(4,6,-2)', 'f(-2,-3,-1)*f(-1,2,5)*f(1,4,-2)*f(3,6,-3)', 'f(-2,-3,-1)*f(-1,2,5)*f(1,4,-3)*f(3,6,-2)', 'f(-2,-3,-1)*f(-1,2,5)*f(1,6,-2)*f(3,4,-3)', 'f
(-2,-3,-1)*f(-1,2,5)*f(1,6,-3)*f(3,4,-2)', 'f(-2,-3,-1)*f(-1,2,6)*f(1,3,-2)*f(4,5,-3)', 'f(-2,-3,-1)*f(-1,2,6)*f(1,3,-3)*f(4,5,-2)', 'f(-2,-3,-1)*f(-1,2,6)*f(1,4,-2)*f
(3,5,-3)', 'f(-2,-3,-1)*f(-1,2,6)*f(1,4,-3)*f(3,5,-2)', 'f(-2,-3,-1)*f(-1,2,6)*f(1,5,-2)*f(3,4,-3)', 'f(-2,-3,-1)*f(-1,2,6)*f(1,5,-3)*f(3,4,-2)', 'f(-2,-3,-1)*f(-1,3,4)
*f(1,2,-2)*f(5,6,-3)', 'f(-2,-3,-1)*f(-1,3,4)*f(1,2,-3)*f(5,6,-2)', 'f(-2,-3,-1)*f(-1,3,4)*f(1,5,-2)*f(2,6,-3)', 'f(-2,-3,-1)*f(-1,3,4)*f(1,5,-3)*f(2,6,-2)', 'f
(-2,-3,-1)*f(-1,3,4)*f(1,6,-2)*f(2,5,-3)', 'f(-2,-3,-1)*f(-1,3,4)*f(1,6,-3)*f(2,5,-2)', 'f(-2,-3,-1)*f(-1,3,5)*f(1,2,-2)*f(4,6,-3)', 'f(-2,-3,-1)*f(-1,3,5)*f(1,2,-3)*f
(4,6,-2)', 'f(-2,-3,-1)*f(-1,3,5)*f(1,4,-2)*f(2,6,-3)', 'f(-2,-3,-1)*f(-1,3,5)*f(1,4,-3)*f(2,6,-2)', 'f(-2,-3,-1)*f(-1,3,5)*f(1,6,-2)*f(2,4,-3)', 'f(-2,-3,-1)*f(-1,3,5)
*f(1,6,-3)*f(2,4,-2)', 'f(-2,-3,-1)*f(-1,3,6)*f(1,2,-2)*f(4,5,-3)', 'f(-2,-3,-1)*f(-1,3,6)*f(1,2,-3)*f(4,5,-2)', 'f(-2,-3,-1)*f(-1,3,6)*f(1,4,-2)*f(2,5,-3)', 'f
(-2,-3,-1)*f(-1,3,6)*f(1,4,-3)*f(2,5,-2)', 'f(-2,-3,-1)*f(-1,3,6)*f(1,5,-2)*f(2,4,-3)', 'f(-2,-3,-1)*f(-1,3,6)*f(1,5,-3)*f(2,4,-2)', 'f(-2,-3,-1)*f(-1,4,5)*f(1,2,-2)*f
(3,6,-3)', 'f(-2,-3,-1)*f(-1,4,5)*f(1,2,-3)*f(3,6,-2)', 'f(-2,-3,-1)*f(-1,4,5)*f(1,3,-2)*f(2,6,-3)', 'f(-2,-3,-1)*f(-1,4,5)*f(1,3,-3)*f(2,6,-2)', 'f(-2,-3,-1)*f(-1,4,5)
*f(1,6,-2)*f(2,3,-3)', 'f(-2,-3,-1)*f(-1,4,5)*f(1,6,-3)*f(2,3,-2)', 'f(-2,-3,-1)*f(-1,4,6)*f(1,2,-2)*f(3,5,-3)', 'f(-2,-3,-1)*f(-1,4,6)*f(1,2,-3)*f(3,5,-2)', 'f
(-2,-3,-1)*f(-1,4,6)*f(1,3,-2)*f(2,5,-3)', 'f(-2,-3,-1)*f(-1,4,6)*f(1,3,-3)*f(2,5,-2)', 'f(-2,-3,-1)*f(-1,4,6)*f(1,5,-2)*f(2,3,-3)', 'f(-2,-3,-1)*f(-1,4,6)*f(1,5,-3)*f
(2,3,-2)', 'f(-2,-3,-1)*f(-1,5,6)*f(1,2,-2)*f(3,4,-3)', 'f(-2,-3,-1)*f(-1,5,6)*f(1,2,-3)*f(3,4,-2)', 'f(-2,-3,-1)*f(-1,5,6)*f(1,3,-2)*f(2,4,-3)', 'f(-2,-3,-1)*f(-1,5,6)
*f(1,3,-3)*f(2,4,-2)', 'f(-2,-3,-1)*f(-1,5,6)*f(1,4,-2)*f(2,3,-3)', 'f(-2,-3,-1)*f(-1,5,6)*f(1,4,-3)*f(2,3,-2)' ],
             lorentz = [ L.VVVVVV16, L.VVVVVV17, L.VVVVVV18, L.VVVVVV19, L.VVVVVV20, L.VVVVVV21, L.VVVVVV22, L.VVVVVV23, L.VVVVVV24, L.VVVVVV25, L.VVVVVV26,
L.VVVVVV27, L.VVVVVV28, L.VVVVVV29, L.VVVVVV30 ],
             couplings = {(5,5):C.GC_7,(4,5):C.GC_8,(3,3):C.GC_8,(2,3):C.GC_7,(11,9):C.GC_8,(10,9):C.GC_7,(7,1):C.GC_8,(6,1):C.GC_7,(17,12):C.GC_8,(16,12):C.GC_7,
(13,2):C.GC_8,(12,2):C.GC_7,(21,10):C.GC_7,(20,10):C.GC_8,(19,11):C.GC_7,(18,11):C.GC_8,(33,11):C.GC_8,(32,11):C.GC_7,(31,2):C.GC_7,(30,2):C.GC_8,(39,10):C.GC_8,
(38,10):C.GC_7,(37,1):C.GC_7,(36,1):C.GC_8,(51,12):C.GC_7,(50,12):C.GC_8,(49,9):C.GC_7,(48,9):C.GC_8,(63,12):C.GC_8,(62,12):C.GC_7,(61,3):C.GC_7,(60,3):C.GC_8,
(71,10):C.GC_7,(70,10):C.GC_8,(67,5):C.GC_8,(66,5):C.GC_7,(75,9):C.GC_8,(74,9):C.GC_7,(73,5):C.GC_7,(72,5):C.GC_8,(83,11):C.GC_7,(82,11):C.GC_8,(79,3):C.GC_8,
(78,3):C.GC_7,(89,2):C.GC_8,(88,2):C.GC_7,(87,1):C.GC_8,(86,1):C.GC_7,(9,7):C.GC_8,(8,7):C.GC_7,(15,13):C.GC_8,(14,13):C.GC_7,(27,8):C.GC_7,(26,8):C.GC_8,
(25,14):C.GC_7,(24,14):C.GC_8,(35,14):C.GC_8,(34,14):C.GC_7,(41,8):C.GC_8,(40,8):C.GC_7,(45,13):C.GC_7,(44,13):C.GC_8,(43,7):C.GC_7,(42,7):C.GC_8,(65,8):C.GC_7,
(64,8):C.GC_8,(69,13):C.GC_8,(68,13):C.GC_7,(77,14):C.GC_7,(76,14):C.GC_8,(81,7):C.GC_8,(80,7):C.GC_7,(1,0):C.GC_8,(0,0):C.GC_7,(23,4):C.GC_8,(22,4):C.GC_7,
(53,4):C.GC_7,(52,4):C.GC_8,(57,4):C.GC_8,(56,4):C.GC_7,(55,0):C.GC_7,(54,0):C.GC_8,(85,0):C.GC_8,(84,0):C.GC_7,(29,6):C.GC_7,(28,6):C.GC_8,(47,6):C.GC_8,(46,6):C.GC_7,
(59,6):C.GC_7,(58,6):C.GC_8})
```

The UFO & ALOHA

```
VVVV42 = Lorentz(name = 'VVVVV42',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,5)*Metric(1,3)*Metric(2,5) - P(1,5)*Metric(2,5)*Metric(3,4) - P(4,5)*Metric(1,2)*Metric(3,5) + P(1,5)*Metric(2,4)*Metric(3,5)')

VVVV43 = Lorentz(name = 'VVVVV43',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(5,1)*Metric(1,4)*Metric(2,3) - P(3,1)*Metric(1,4)*Metric(2,5) - P(5,1)*Metric(1,2)*Metric(3,4) + P(3,1)*Metric(1,2)*Metric(4,5)')

VVVV44 = Lorentz(name = 'VVVVV44',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,1)*Metric(1,5)*Metric(2,3) - P(3,1)*Metric(1,5)*Metric(2,4) - P(4,1)*Metric(1,2)*Metric(3,5) + P(3,1)*Metric(1,2)*Metric(4,5)')

VVVV45 = Lorentz(name = 'VVVVV45',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(5,2)*Metric(1,3)*Metric(2,4) - P(3,2)*Metric(1,5)*Metric(2,4) - P(5,2)*Metric(1,2)*Metric(3,4) + P(3,2)*Metric(1,2)*Metric(4,5)')

VVVV46 = Lorentz(name = 'VVVVV46',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,2)*Metric(1,3)*Metric(2,5) - P(3,2)*Metric(1,4)*Metric(2,5) - P(4,2)*Metric(1,2)*Metric(3,5) + P(3,2)*Metric(1,2)*Metric(4,5)')

VVVV47 = Lorentz(name = 'VVVVV47',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,1)*Metric(1,5)*Metric(2,3) - P(4,1)*Metric(1,3)*Metric(2,5) - P(2,1)*Metric(1,5)*Metric(3,4) + P(2,1)*Metric(1,3)*Metric(4,5)')

VVVV48 = Lorentz(name = 'VVVVV48',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(5,1)*Metric(1,4)*Metric(2,3) - P(5,1)*Metric(1,3)*Metric(2,4) - P(2,1)*Metric(1,4)*Metric(3,5) + P(2,1)*Metric(1,3)*Metric(4,5)')

VVVV49 = Lorentz(name = 'VVVVV49',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(5,3)*Metric(1,3)*Metric(2,4) - P(5,3)*Metric(1,2)*Metric(3,4) + P(2,3)*Metric(1,5)*Metric(3,4) - P(2,3)*Metric(1,3)*Metric(4,5)')

VVVV50 = Lorentz(name = 'VVVVV50',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,3)*Metric(1,3)*Metric(2,5) - P(4,3)*Metric(1,2)*Metric(3,5) + P(2,3)*Metric(1,4)*Metric(3,5) - P(2,3)*Metric(1,3)*Metric(4,5)')

VVVV51 = Lorentz(name = 'VVVVV51',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(3,4)*Metric(1,4)*Metric(2,5) - P(2,4)*Metric(1,4)*Metric(3,5) - P(3,4)*Metric(1,2)*Metric(4,5) + P(2,4)*Metric(1,3)*Metric(4,5)')
```

The UFO & ALOHA

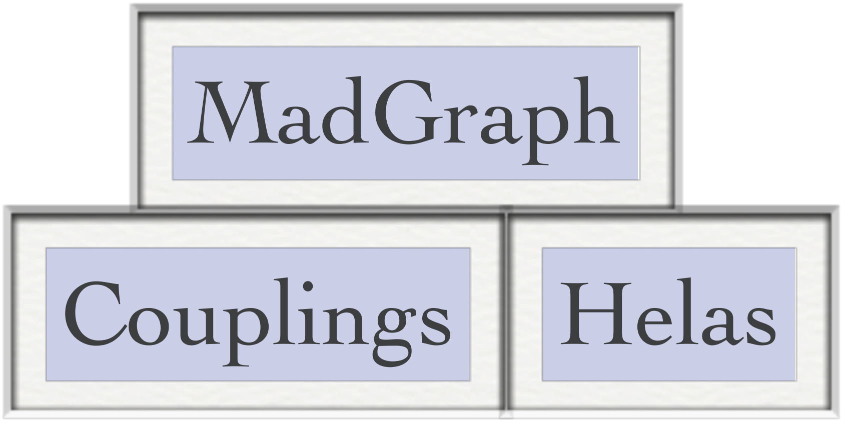
```
VVVV42 = Lorentz(name = 'VVVV42',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,5)*Metric(1,3)*Metric(2,5) - P(1,5)*Metric(2,5)*Metric(3,4) - P(4,5)*Metric(1,2)*Metric(3,5) + P(1,5)*Metric(2,4)*Metric(3,5)')

VVVV43 = Lorentz(name = 'VVVV43',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(5,1)*Metric(1,4)*Metric(2,3) - P(3,1)*Metric(1,4)*Metric(2,5) - P(5,1)*Metric(1,2)*Metric(3,4) + P(3,1)*Metric(1,2)*Metric(4,5)')

VVVV44 = Lorentz(name = 'VVVV44',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,1)*Metric(1,5)*Metric(2,3) - P(3,1)*Metric(1,5)*Metric(2,4) - P(4,1)*Metric(1,2)*Metric(3,5) + P(3,1)*Metric(1,2)*Metric(4,5)')

VVVV45 = Lorentz(name = 'VVVV45',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(5,2)*Metric(1,3)*Metric(2,4) - P(3,2)*Metric(1,5)*Metric(2,4) - P(5,2)*Metric(1,2)*Metric(3,4) + P(3,2)*Metric(1,2)*Metric(4,5)')

VVVV46 = Lorentz(name = 'VVVV46',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,[illegible])*Metric(1,3)*Metric(2,5) - P(3,2)*Met[illegible]c(1,4)*Metric([illegible],5) - P(4,2)*Metric(1,2)*Met[illegible]c(3,5) + P(3,2)*Metric(1,2)*Metric(4,5)')

VVVV47 = Lorentz(name = 'VVVV47',
                 spins = [ 3, 3, [illegible] ],
                 structure = 'P(4,[illegible])[illegible]ric(1,5)*Metric(2,3) - P(4,1)*Metric(1,3)*Metric(2,5) - P(2,1)*Metric(1,5)*Metr[illegible]c(3,4) + P(2,1)*Metric(1,3)*Metric(4,5)')

VVVV48 = Lorentz(name = 'VVVV48',
                 spins = [ 3, 3, [illegible]3, 3, 3 ],
                 structure = 'P(5,1)*Metric(1,4)*Metric(2,3) [illegible] - P(2,1)*Metric(1,4)*Metric(3,5) + P(2,1)*Metric(1,3)*Metric(4,5)')

VVVV49 = Lorentz(name = 'VVVV49',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(5,3)*Metric(1,3)*Metric(2,4) - P(5,3)*Metric(1,2)*Metric(3,4) + P(2,3)*Metric(1,5)*Metric(3,4) - P(2,3)*Metric(1,3)*Metric(4,5)')

VVVV50 = Lorentz(name = 'VVVV50',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(4,3)*Metric(1,3)*Metric(2,5) - P(4,3)*Metric(1,2)*Metric(3,5) + P(2,3)*Metric(1,4)*Metric(3,5) - P(2,3)*Metric(1,3)*Metric(4,5)')

VVVV51 = Lorentz(name = 'VVVV51',
                 spins = [ 3, 3, 3, 3, 3 ],
                 structure = 'P(3,4)*Metric(1,4)*Metric(2,5) - P(2,4)*Metric(1,4)*Metric(3,5) - P(3,4)*Metric(1,2)*Metric(4,5) + P(2,4)*Metric(1,3)*Metric(4,5)')
```

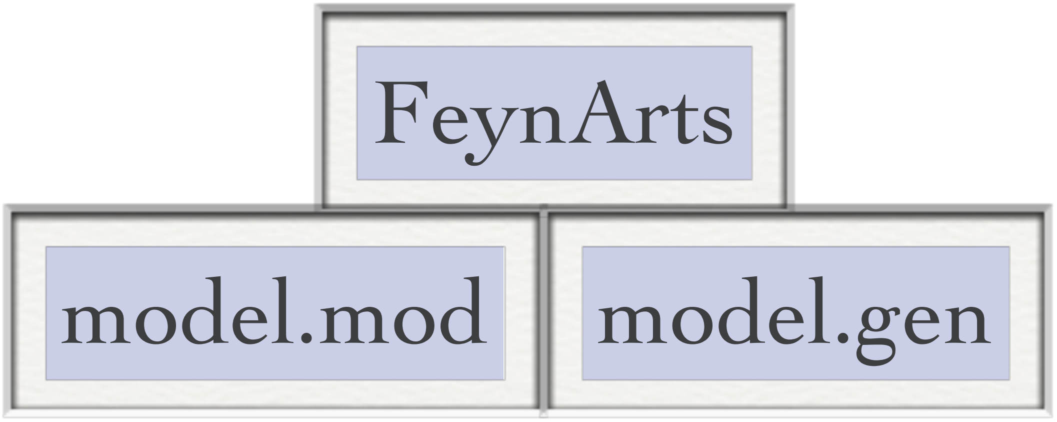
All Helicity amplitudes agreed out of the box with the CSW construction!

Higher-dim Operators in FeynArts

[Degrande]

- Similar improvements were made to the FeynArts interface.
- FeynArts has a similar structure to MadGraph/Helas:

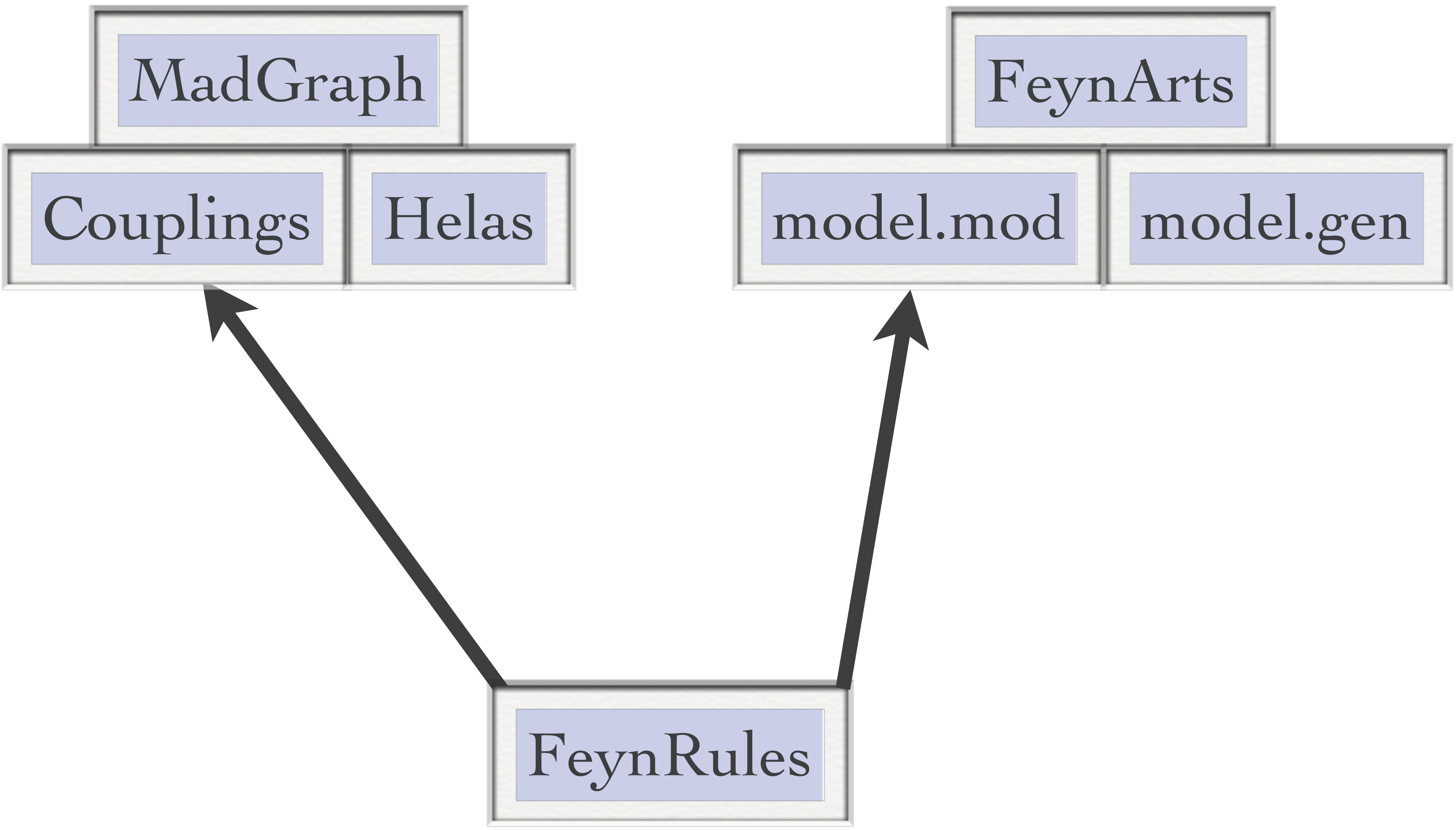




Higher-dim Operators in FeynArts

[Degrande]

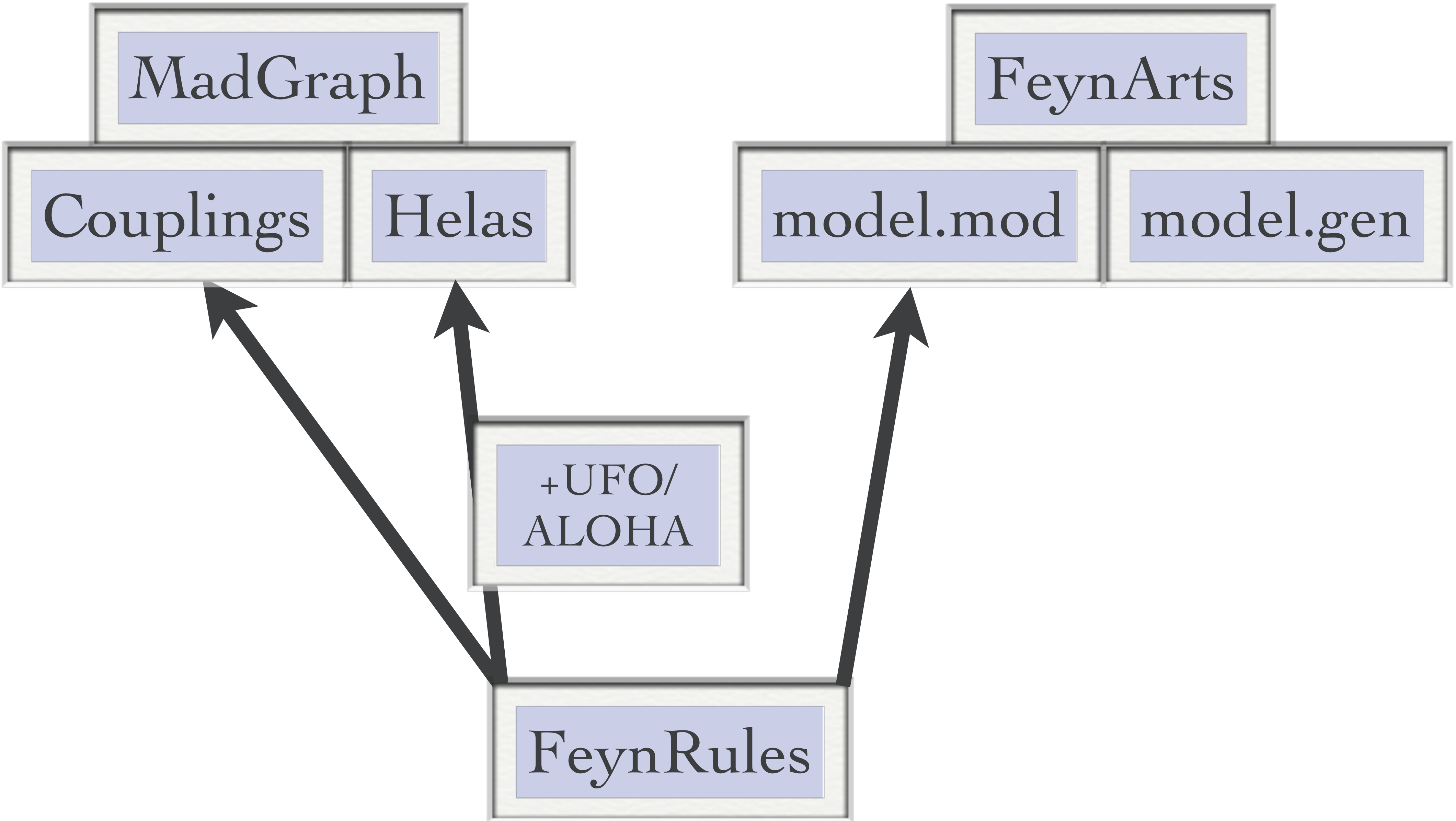
- Similar improvements were made to the FeynArts interface.
- FeynArts has a similar structure to MadGraph/Helas:



Higher-dim Operators in FeynArts

[Degrande]

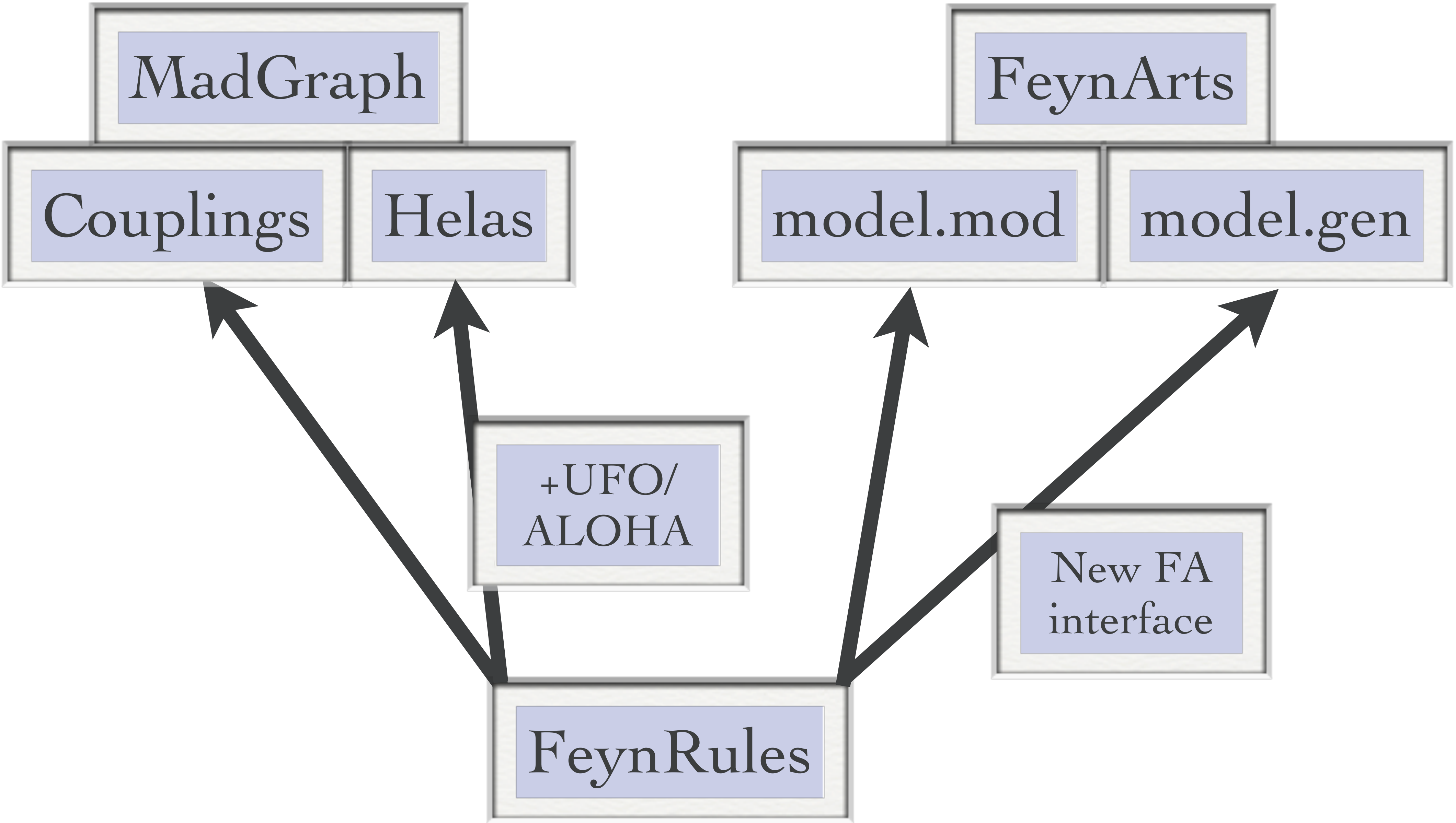
- Similar improvements were made to the FeynArts interface.
- FeynArts has a similar structure to MadGraph/Helas:



Higher-dim Operators in FeynArts

[Degrande]

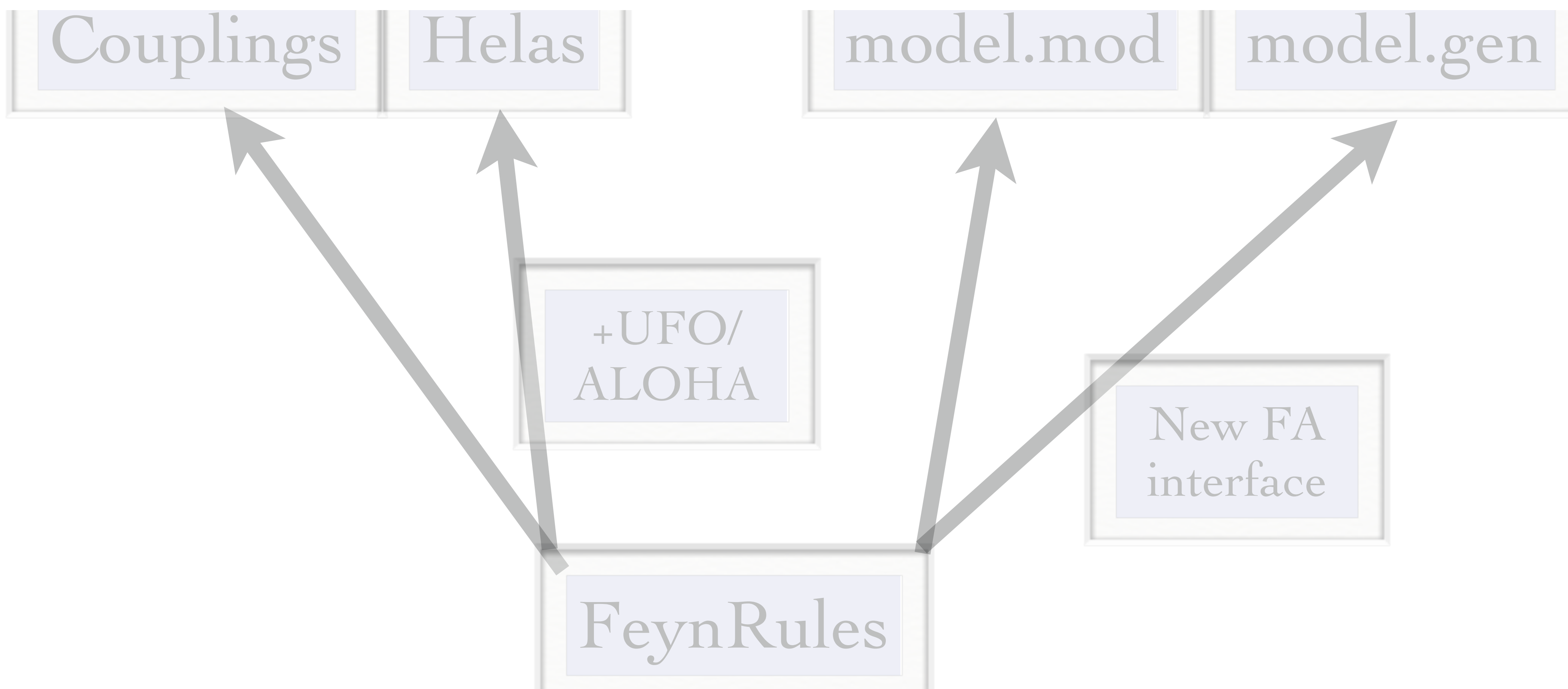
- Similar improvements were made to the FeynArts interface.
- FeynArts has a similar structure to MadGraph/Helas:



Higher-dim Operators in FeynArts

[Degrande]

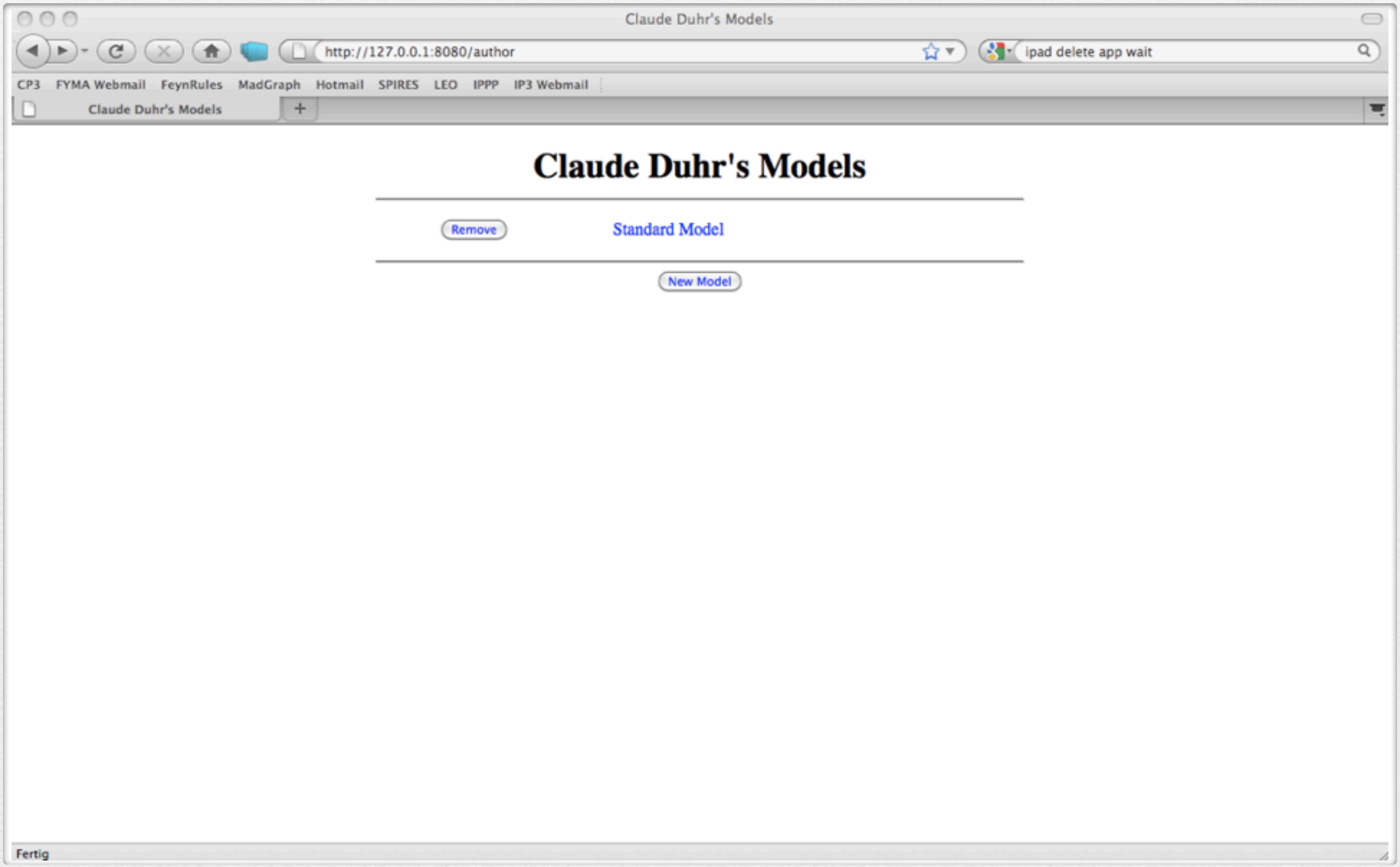
- Similar improvements were made to the FeynArts interface.
- FeynArts has a similar structure to MadGraph/Helas:
- The new FeynArts interface brings the interface to the same level as the UFO interface!

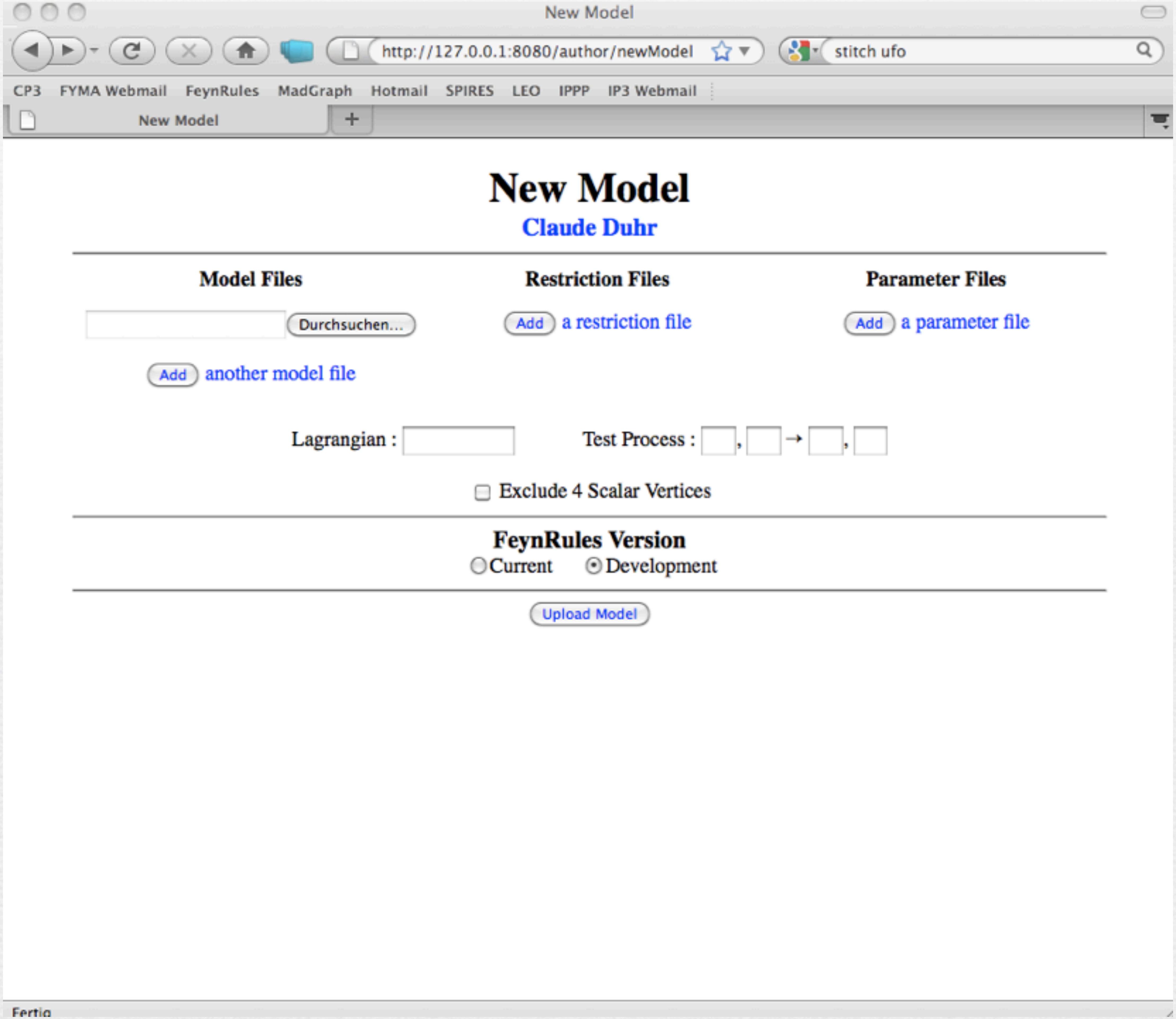


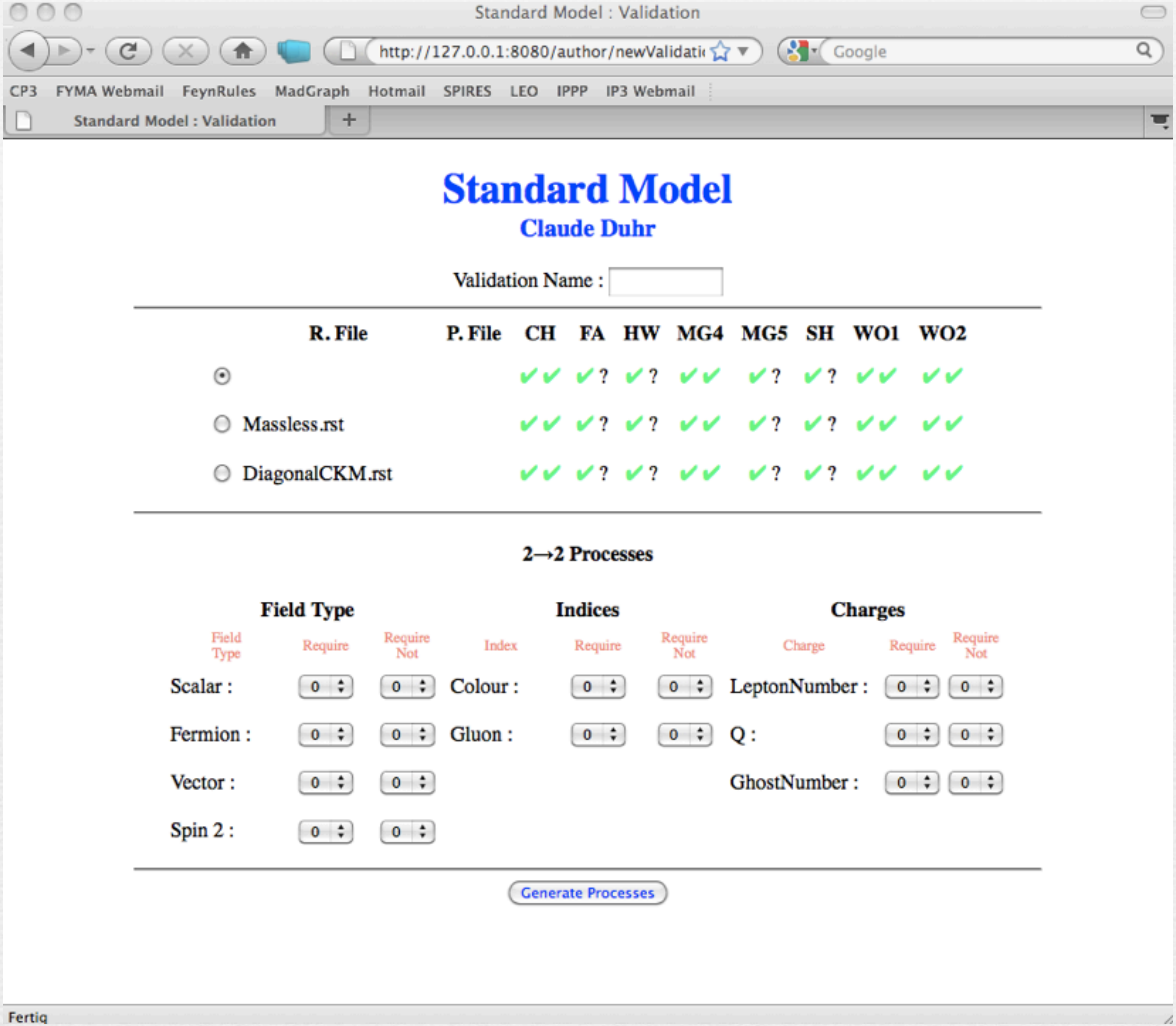
Validation of new models

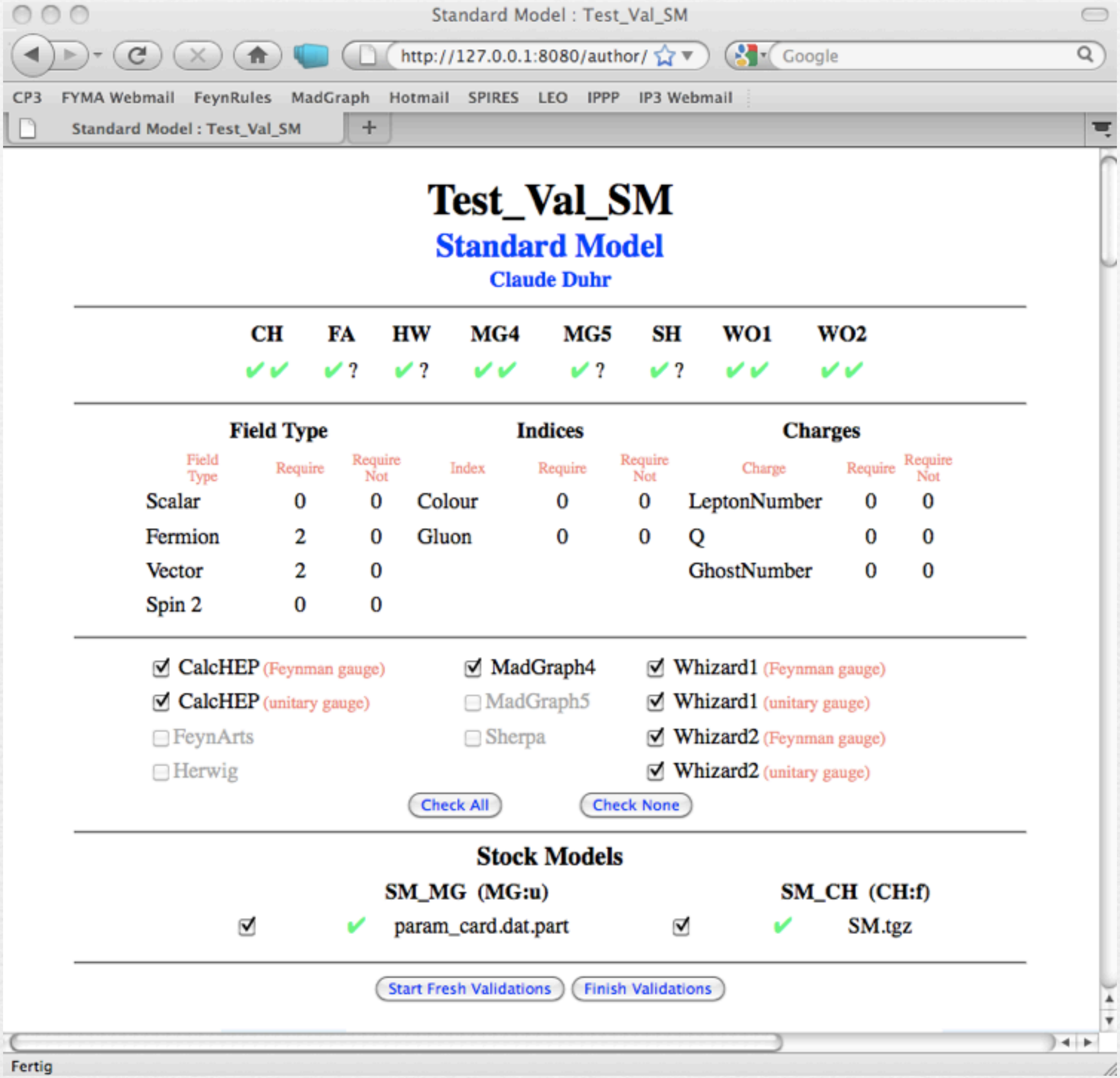
- FeynRules does not only provide the power to develop and validate new models, but also to validate them to an unprecedented level!
- A given model can be output to more than one matrix element generator, and their results can be compared
  - ➡ Different conventions
  - ➡ Different gauges
  - ➡ Different ways of handling large cancellations.
- This procedure can easily be automatized!

Web validation




New Model
http://127.0.0.1:8080/author/newModel
stitch ufo
CP3 FYMA Webmail FeynRules MadGraph Hotmail SPIRES LEO IPPP IP3 Webmail
New Model
New Model
Claude Duhr
Model Files
Restriction Files
Parameter Files
Durchsuchen...
Add a restriction file
Add a parameter file
Add another model file
Lagrangian :
Test Process : , → ,
Exclude 4 Scalar Vertices
FeynRules Version
Current
Development
Upload Model
Fertig


Standard Model : Validation
http://127.0.0.1:8080/author/newValidati
Google
CP3 FYMA Webmail FeynRules MadGraph Hotmail SPIRES LEO IPPP IP3 Webmail
Standard Model : Validation
Standard Model
Claude Duhr
Validation Name :
R. File P. File CH FA HW MG4 MG5 SH WO1 WO2
Massless.rst
DiagonalCKM.rst
2→2 Processes
Field Type
Indices
Charges
Field Type Require Require Not
Index Require Require Not
Charge Require Require Not
Scalar :
Fermion :
Vector :
Spin 2 :
Colour :
Gluon :
LeptonNumber :
Q :
GhostNumber :
Generate Processes
Fertig


Standard Model : Test_Val_SM
http://127.0.0.1:8080/author/
Google
CP3 FYMA Webmail FeynRules MadGraph Hotmail SPIRES LEO IPPP IP3 Webmail
Standard Model : Test_Val_SM
Test_Val_SM
Standard Model
Claude Duhr
CH FA HW MG4 MG5 SH WO1 WO2
Field Type
Field Type Require Require Not
Scalar 0 0
Fermion 2 0
Vector 2 0
Spin 2 0 0
Indices
Index Require Require Not
Colour 0 0
Gluon 0 0
Charges
Charge Require Require Not
LeptonNumber 0 0
Q 0 0
GhostNumber 0 0
CalcHEP (Feynman gauge)
CalcHEP (unitary gauge)
FeynArts
Herwig
MadGraph4
MadGraph5
Sherpa
Whizard1 (Feynman gauge)
Whizard1 (unitary gauge)
Whizard2 (Feynman gauge)
Whizard2 (unitary gauge)
Check All
Check None
Stock Models
SM_MG (MG:u)
param_card.dat.part
SM_CH (CH:f)
SM.tgz
Start Fresh Validations
Finish Validations
Fertig

Standard Model : Test_Val_SM

http://127.0.0.1:8080/author/validation?v

CP3 FYMA Webmail FeynRules MadGraph Hotmail SPIRES LEO IPPP IP3 Webmail

Standard Model : Test_Val_SM

ve , ve~ → Z , Z	730.0	182.5	0.49452	0.49452	0.49384	0.494604	0.494622	0.494547	0.494668	0.49351	0.4945	●	0.17%
ve , ve~ → W+ , W-	639.0	159.75	1.0603	1.0603	1.0604	1.06053	1.0604	1.06035	1.06073	1.0665	1.0603	●	0.51%
ve , ve~ → G , G	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vm~ → A , A	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vm~ → A , Z	365.0	91.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vm~ → Z , Z	730.0	182.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vm~ → W+ , W-	639.0	159.75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vm~ → G , G	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vt~ → A , A	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vt~ → A , Z	365.0	91.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vt~ → Z , Z	730.0	182.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vt~ → W+ , W-	639.0	159.75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , vt~ → G , G	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , e+ → A , W+	319.0	79.75	2.2219	1.9846	1.9809	1.98496	1.98478	1.98454	1.98491	1.9756	1.9845	✘	10.56%
ve , e+ → Z , W+	684.0	171.0	0.71578	0.54663	0.54717	0.54657	0.546756	0.54641	0.546869	0.54864	0.54661	✘	26.53%
ve , m+ → A , W+	320.0	80.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , m+ → Z , W+	684.0	171.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , tt+ → A , W+	326.0	81.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
ve , tt+ → Z , W+	691.0	172.75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
vm , vm~ → A , A	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
vm , vm~ → A , Z	365.0	91.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
vm , vm~ → Z , Z	730.0	182.5	0.49452	0.49452	0.49384	0.494505	0.494545	0.494559	0.494447	0.49351	0.4945	●	0.17%
vm , vm~ → W+ , W-	639.0	159.75	1.0603	1.0603	1.0604	1.0604	1.06033	1.06027	1.06006	1.0665	1.0603	●	0.52%
vm , vm~ → G , G	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
vm , vt~ → A , A	200.0	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
vm , vt~ → A , Z	365.0	91.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
vm , vt~ → Z , Z	730.0	182.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%
vm , vt~ → W+ , W-	639.0	159.75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	✔	0%

Fertig

A look into the future...

Future plans

- We have now the possibility to
  - ➡ easily implement SUSY models.
  - ➡ deal with higher-dimensional operators in a successful way.

Future plans

- We have now the possibility to
  - ➡ easily implement SUSY models.
  - ➡ deal with higher-dimensional operators in a successful way.
- The tree level story is basically closed!

Future plans

- We have now the possibility to
  - ➡ easily implement SUSY models.
  - ➡ deal with higher-dimensional operators in a successful way.
- The tree level story is basically closed!
- In other words, many things have already been achieved...

Future plans

- We have now the possibility to
  - ➡ easily implement SUSY models.
  - ➡ deal with higher-dimensional operators in a successful way.
- The tree level story is basically closed!
- In other words, many things have already been achieved...
- ... but there are many more left to do!

Future plans

- Where can we improve...?

Future plans

- Where can we improve...?

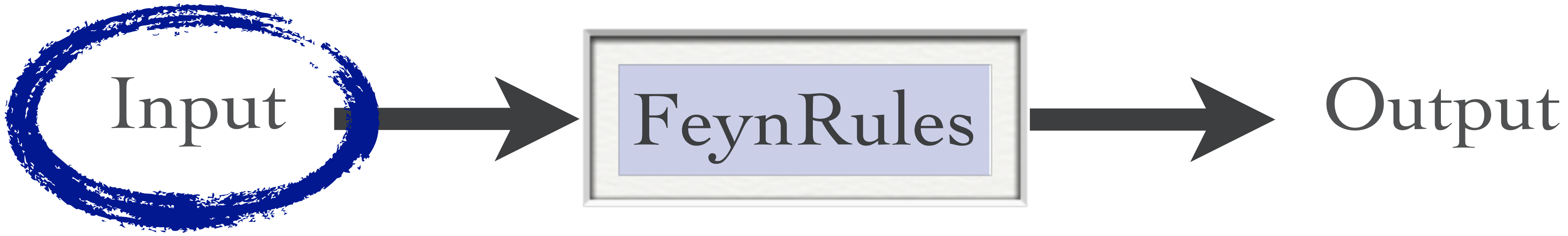
Future plans

- Where can we improve...?

- Your ideas are welcome!

Future plans

- Where can we improve...?



- Your ideas are welcome!

Model file GUI

- A GUI is being developed that allows to generate a FeynRules model file automatically

    - ➡ Select your fields.
    - ➡ Select your symmetry groups (gauged or global).
    - ➡ Most general Lagrangian (up to dimension 6) is generated automatically.

Model file GUI

- A GUI is being developed that allows to generate a FeynRules model file automatically

  - ➡ Select your fields.
  - ➡ Select your symmetry groups (gauged or global).
  - ➡ Most general Lagrangian (up to dimension 6) is generated automatically.

- Works for both SUSY and non-SUSY theories.

Model file GUI

- A GUI is being developed that allows to generate a FeynRules model file automatically
  - ➡ Select your fields.
  - ➡ Select your symmetry groups (gauged or global).
  - ➡ Most general Lagrangian (up to dimension 6) is generated automatically.
- Works for both SUSY and non-SUSY theories.
- A sneak preview will be given by Neil on Thursday.

Mass diagonalization

- Even after the Lagrangian has been obtained, FeynRules cannot diagonalize the mass matrix automatically.

- FeynRules should be able to
  - ➡ Get the mass matrix.
  - ➡ Diagonalize it numerically.
  - ➡ Put the numerical values back into FeynRules / an updated parameter card.
  - ➡ See Adam Alloul's talk.

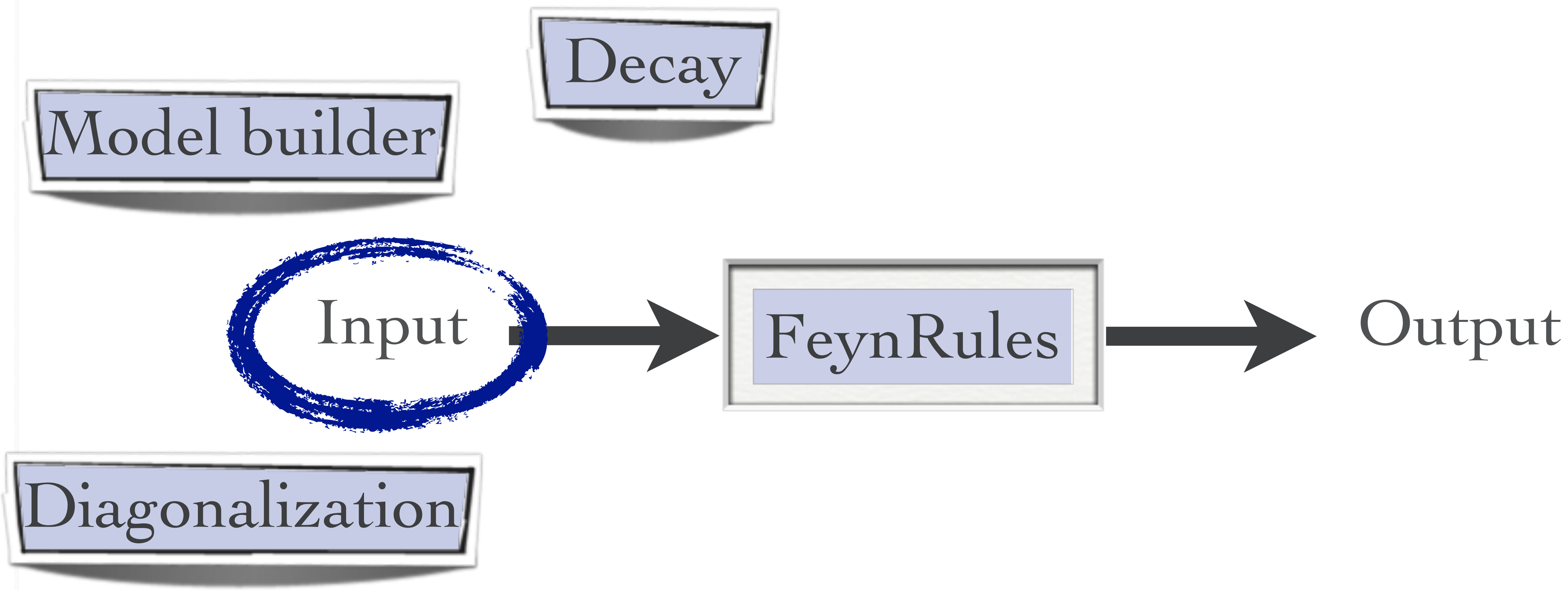
- This was on the to-do list of FeynRules 2010 already.

Decay rates and branching ratios

- Another missing piece for a complete model file are the widths of the particles.
- Idea: Compute decays (analytically) in Mathematica:
  - ➡ Use FeynRules to generate vertices.
  - ➡ Compute (1-to-2) decays analytically.
  - ➡ Put the numerical values back into FeynRules / an updated parameter card.
- The branching ratios can be output in a LHA style decay table.
- Interfacing to Python / UFO to compute branching ratios numerically on the fly in the UFO?

Future plans

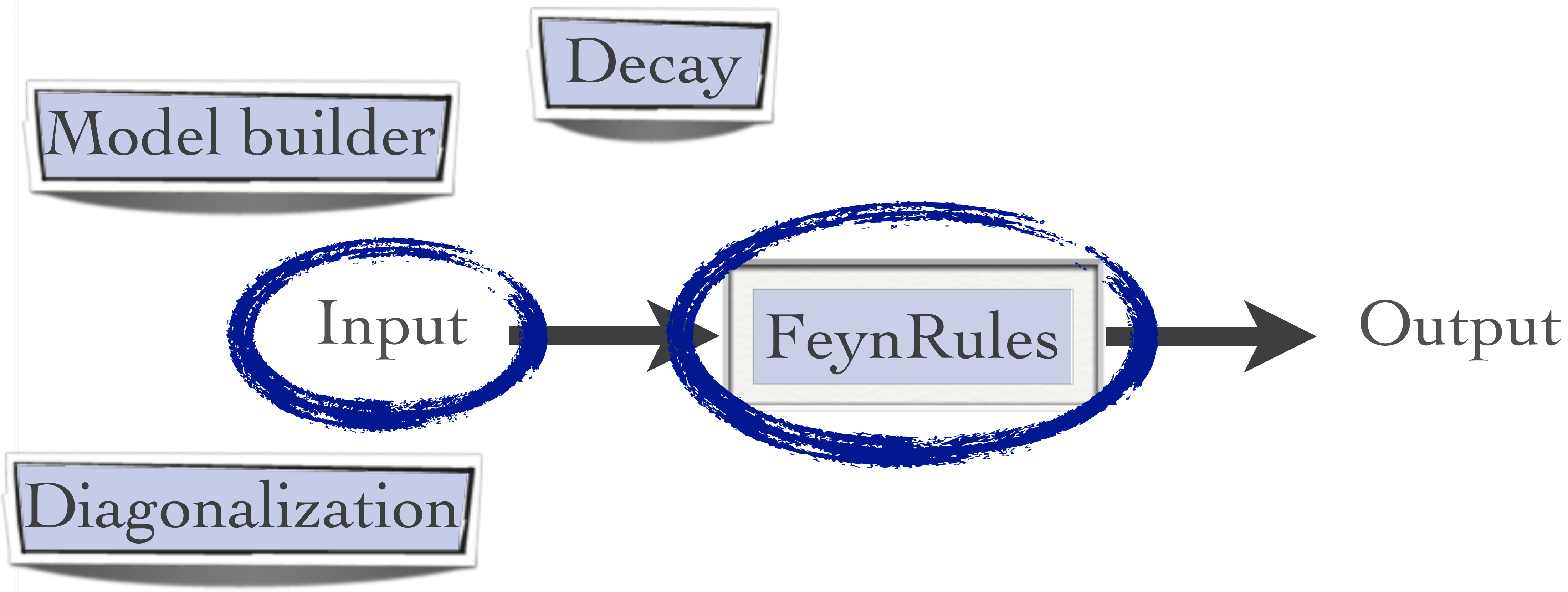
- Where can we improve...?



- Your ideas are welcome!

Future plans

- Where can we improve...?



- Your ideas are welcome!

Spin 3/2 fields

- The development version of FeynRules allows to implement models including spin 3/2 particles.
- Implementation basically ready:
  - ➡ Feynman rules can be computed.
  - ➡ Interfaces to CalcHep and MadGraph 5 (UFO) have been updated.
- Currently under testing against independent MadGraph 4 implementation. [Hagiwara, Mawatari, Takaesu]
- What about spin 3/2 in FeynArts..?

SUSY RGE's

- The development version of FeynRules allows to extract the one-loop renormalization group equations for generic SUSY models.

SUSY RGE's

- The development version of FeynRules allows to extract the one-loop renormalization group equations for generic SUSY models.
- Starting from the superspace action, the RGE's are simply obtained via

```
RGE[LSoft, SuperW]
```

SUSY RGE's

- The development version of FeynRules allows to extract the one-loop renormalization group equations for generic SUSY models.
- Starting from the superspace action, the RGE's are simply obtained via

```
RGE[LSoft, SuperW]
```

$$\frac{\mathrm{d}\mu}{\mathrm{d}t} = \mu\Big[-\frac{3g'^2}{80\pi^2} - \frac{3g_w^2}{16\pi^2} + \frac{3}{16\pi^2}\mathrm{Tr}\big[\mathbf{y^d}^\dagger\mathbf{y^d}\big] + \frac{3}{16\pi^2}\mathrm{Tr}\big[\mathbf{y^u}^\dagger\mathbf{y^u}\big] + \frac{1}{16\pi^2}\mathrm{Tr}\big[\mathbf{y^e}^\dagger\mathbf{y^e}\big]\Big]$$

$$\frac{\mathrm{d}b}{\mathrm{d}t} = b\Big[-\frac{3g'^2}{80\pi^2} - \frac{3g_w^2}{16\pi^2} + \frac{3}{16\pi^2}\mathrm{Tr}\big[\mathbf{y^d}^\dagger\mathbf{y^d}\big] + \frac{3}{16\pi^2}\mathrm{Tr}\big[\mathbf{y^u}^\dagger\mathbf{y^u}\big] + \frac{1}{16\pi^2}\mathrm{Tr}\big[\mathbf{y^e}^\dagger\mathbf{y^e}\big]\Big]$$
$$+ \mu\Big[\frac{3g'^2M_1}{40\pi^2} + \frac{3g_w^2M_2}{8\pi^2} + \frac{3}{8\pi^2}\mathrm{Tr}\big[\mathbf{y^d}^\dagger\mathbf{T^d}\big] + \frac{3}{8\pi^2}\mathrm{Tr}\big[\mathbf{y^u}^\dagger\mathbf{T^u}\big] + \frac{1}{8\pi^2}\mathrm{Tr}\big[\mathbf{y^e}^\dagger\mathbf{T^e}\big]\Big]$$

SUSY RGE's

- The development version of FeynRules allows to extract the one-loop renormalization group equations for generic SUSY models.

  RGE[LSoft, SuperW]

  ➡ See Adam Alloul's talk.

- In parallel, an interface to SuSpect 3 is being developed that allows to input the RGE's obtained by FeynRules into SuSpect to solve them numerically.

  WriteSuSpectOutput[LSoft, SuperW]

Towards NLO

- We are slowly getting to the point that we have automated tools for NLO computations:
  - ➡ Blackhat
  - ➡ GoSam
  - ➡ Helac-NLO
  - ➡ MadLoops
  - ➡ Rocket
- Most of these codes only do SM processes so far.
- Reason: Beyond LO, we do not only need tree-level Feynman rules, but also counterterms, etc.
- Future releases of FeynRules will allow to compute also these quantities!

Extraction of counterterms

- The (not public) development version of FeynRules already allows to extract counterterm Feynman rules.

```
ExtractCounterterms[l[s,f],{aS,1}]
```

$$\blacktriangleright l_{sf} \to l_{sf} + \frac{\alpha_s}{4\pi}\left[(\delta Z_{ll}^{L(1)})_{ff'}\ (P_L)_{ss'} + (\delta Z_{ll}^{R(1)})_{ff'}\ (P_R)_{ss'}\right]\ l_{s'f'}$$

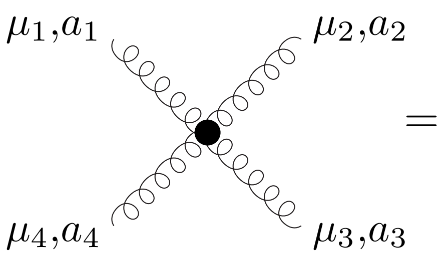
```
ExtractCounterterms[ydo,{{aS,2},{aEW,1}}]
```

$$\blacktriangleright y_d \to y_d + \frac{\alpha_s}{2\pi}\delta y_d^{(1,0)} + \frac{\alpha}{2\pi}\delta y_d^{(0,1)} + \frac{\alpha_s^2}{4\pi^2}\delta y_d^{(2,0)} + \frac{\alpha_s\alpha}{4\pi^2}\delta y_d^{(1,1)} + \frac{\alpha_s^2\alpha}{8\pi^3}\delta y_d^{(2,1)}$$

- At the moment, the values of the counterterms for the independent parameters and the fields must still be given by hand.
- In addition, this should also provide the counterterms for FeynArts.
- Still to do: How to get the values for the ‘independent’ counterterms.

R2 terms

- All the automatized NLO codes are based, in one way or another, on some unitary-based approach.
- Unitarity, however, does not provide everything, but misses the rational pieces (without cuts).
- Some can be obtained, others (R2) need a different approach.
- R2 terms can be obtained via effective tree-level Feynman rules.



$$= -\frac{ig^4 N_{col}}{96\pi^2} \sum_{P(234)} \Bigg\{ \Bigg[\frac{\delta_{a_1a_2}\delta_{a_3a_4} + \delta_{a_1a_3}\delta_{a_4a_2} + \delta_{a_1a_4}\delta_{a_2a_3}}{N_{col}}$$

$$+ 4\, Tr(t^{a_1}t^{a_3}t^{a_2}t^{a_4} + t^{a_1}t^{a_4}t^{a_2}t^{a_3})\,(3 + \lambda_{HV})$$

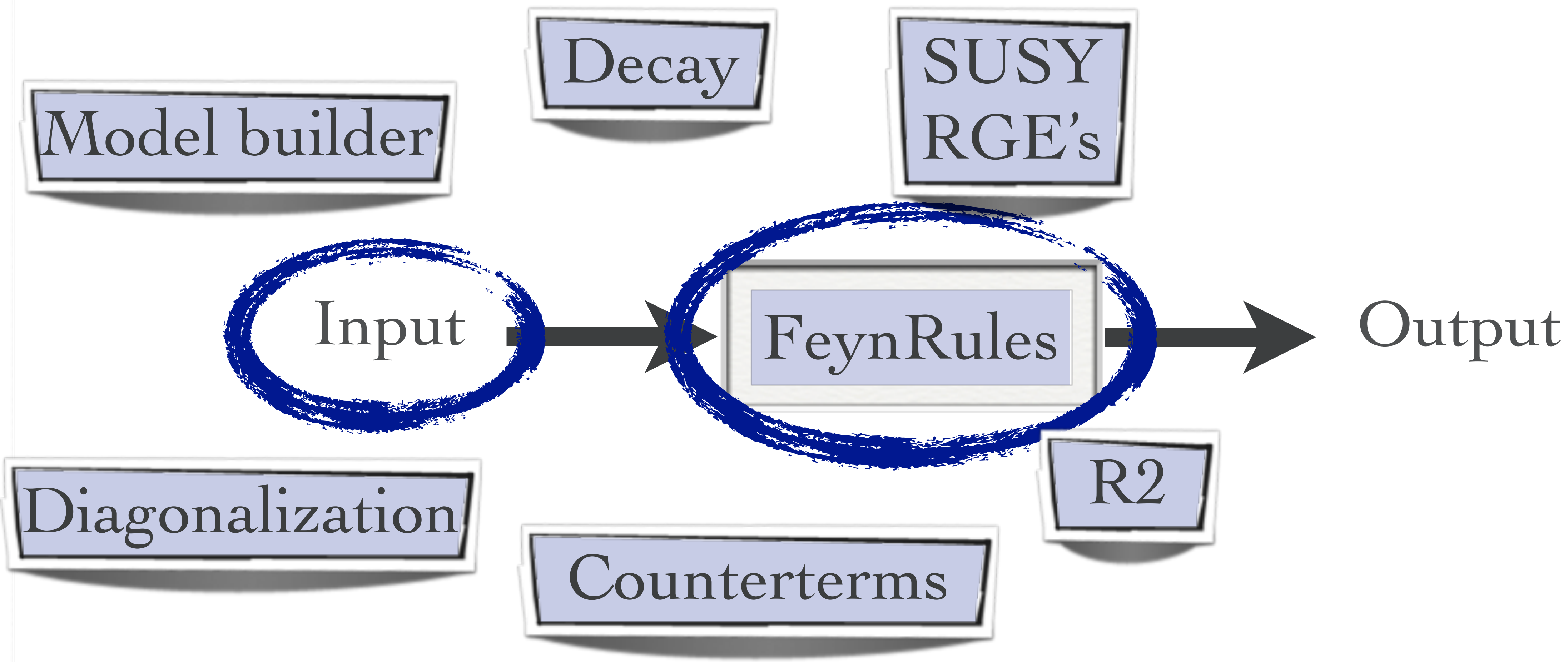
$$- Tr(\{t^{a_1}t^{a_2}\}\{t^{a_3}t^{a_4}\})\,(5 + 2\lambda_{HV}) \Bigg] g_{\mu_1\mu_2} g_{\mu_3\mu_4}$$

$$+ 12 \frac{N_f}{N_{col}} Tr(t^{a_1}t^{a_2}t^{a_3}t^{a_4}) \left(\frac{5}{3} g_{\mu_1\mu_3} g_{\mu_2\mu_4} - g_{\mu_1\mu_2} g_{\mu_3\mu_4} - g_{\mu_2\mu_3} g_{\mu_1\mu_4} \right) \Bigg\}$$

[Draggiotis, Garzelli, Papadopoulos, Pittau;
Garzelli, Malamos, Pittau]

Future plans

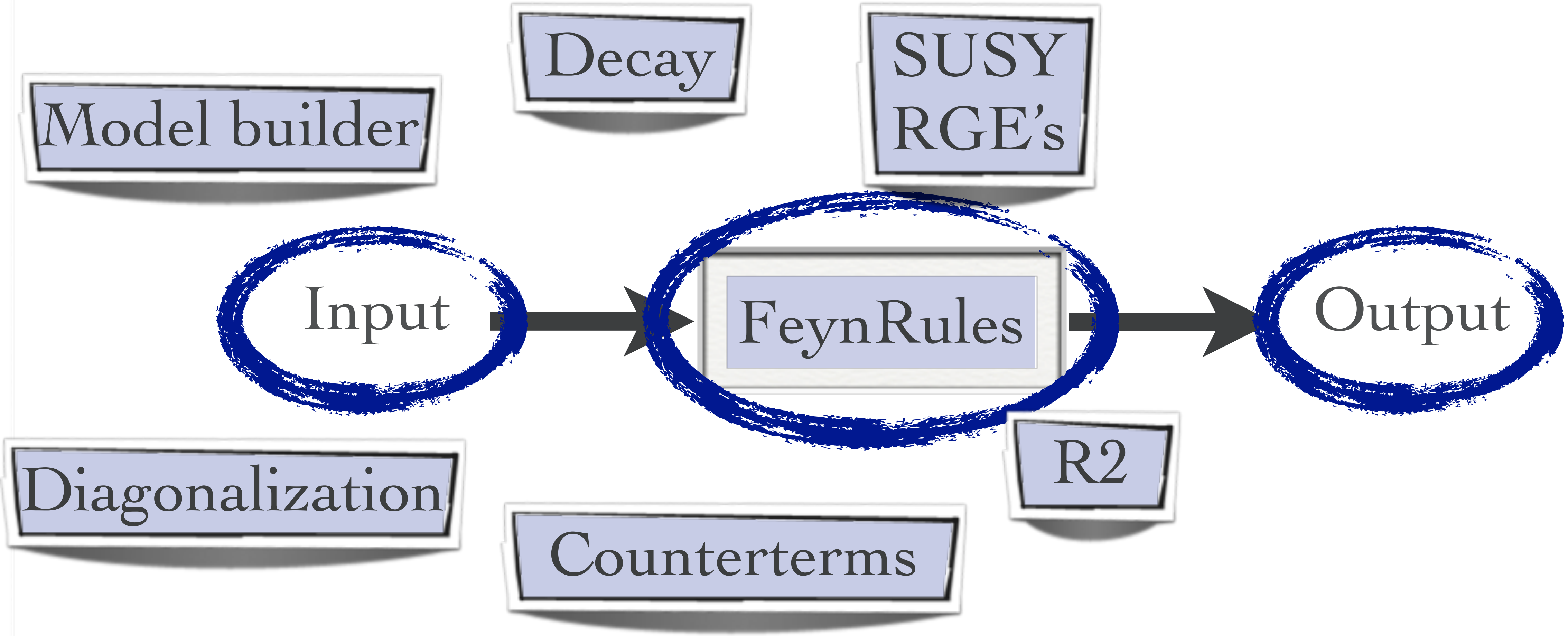
- Where can we improve...?



- Your ideas are welcome!

Future plans

- Where can we improve...?



- Your ideas are welcome!

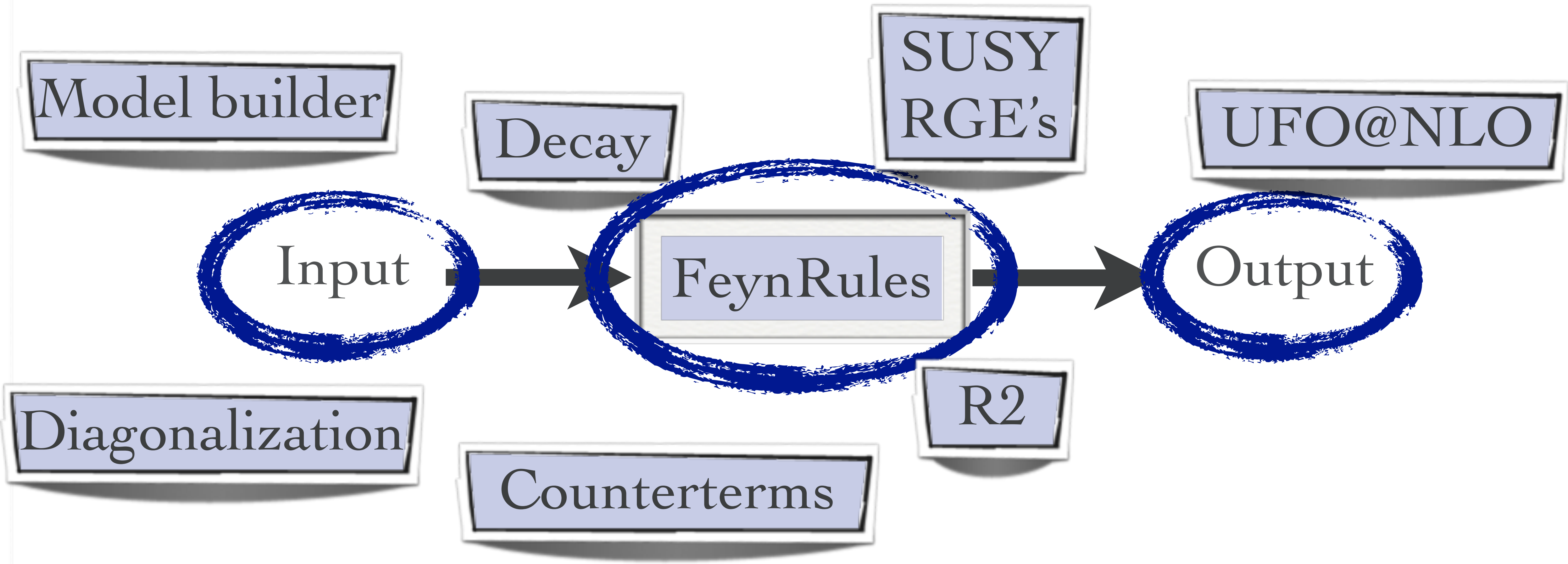
UFO@NLO

- Both GoSam and MadLoops use the UFO as the BSM model format.
  - ➡ Need to extend the existing UFO format to NLO.
- Some steps were already taken

```
V_R24G = CTVertex(name = 'V_R24G',
          particles = [ P.G, P.G, P.G, P.G ],
          color = [ 'Tr(1,2)*Tr(3,4)' , 'Tr(1,3)*Tr(2,4)' , 'Tr(1,4)*Tr(2,3)', \
                    'd(-1,1,2)*d(-1,3,4)' , 'd(-1,1,3)*d(-1,2,4)' , 'd(-1,1,4)*d(-1,2,3)'],
          lorentz = [  L.R2_4G_1234, L.R2_4G_1324, L.R2_4G_1423 ],
          loop_particles = [ [[P.G]], [[P.u],[P.d],[P.c],[P.s]] ],
          couplings = {(0,0,0):C.GC_4GR2_Gluon_delta5,(0,1,0):C.GC_4GR2_Gluon_delta7,(0,2,0):C.GC_4GR2_Gluon_delta7, \
                       (1,0,0):C.GC_4GR2_Gluon_delta7,(1,1,0):C.GC_4GR2_Gluon_delta5,(1,2,0):C.GC_4GR2_Gluon_delta7, \
                       (2,0,0):C.GC_4GR2_Gluon_delta7,(2,1,0):C.GC_4GR2_Gluon_delta7,(2,2,0):C.GC_4GR2_Gluon_delta5, \
                       (3,0,0):C.GC_4GR2_4Struct,(3,1,0):C.GC_4GR2_2Struct,(3,2,0):C.GC_4GR2_2Struct, \
                       (4,0,0):C.GC_4GR2_2Struct,(4,1,0):C.GC_4GR2_4Struct,(4,2,0):C.GC_4GR2_2Struct, \
                       (5,0,0):C.GC_4GR2_2Struct,(5,1,0):C.GC_4GR2_2Struct,(5,2,0):C.GC_4GR2_4Struct , \
                       (0,0,1):C.GC_4GR2_Fermion_delta11,(0,1,1):C.GC_4GR2_Fermion_delta5,(0,2,1):C.GC_4GR2_Fermion_delta5, \
                       (1,0,1):C.GC_4GR2_Fermion_delta5,(1,1,1):C.GC_4GR2_Fermion_delta11,(1,2,1):C.GC_4GR2_Fermion_delta5, \
                       (2,0,1):C.GC_4GR2_Fermion_delta5,(2,1,1):C.GC_4GR2_Fermion_delta5,(2,2,1):C.GC_4GR2_Fermion_delta11, \
                       (3,0,1):C.GC_4GR2_11Struct,(3,1,1):C.GC_4GR2_5Struct,(3,2,1):C.GC_4GR2_5Struct, \
                       (4,0,1):C.GC_4GR2_5Struct,(4,1,1):C.GC_4GR2_11Struct,(4,2,1):C.GC_4GR2_5Struct, \
                       (5,0,1):C.GC_4GR2_5Struct,(5,1,1):C.GC_4GR2_5Struct,(5,2,1):C.GC_4GR2_11Struct },
          type = 'R2')
```

Future plans

- Where can we improve...?



- Your ideas are welcome!

Conclusion

- Many milestones have been achieved since the last FeynRules workshop in 2010:
  - ➡ Superfields
  - ➡ UFO & ALOHA
  - ➡ Support of color sextets.
  - ➡ Web validation:
- New developments that are in the pipeline:
  - ➡ Spin 3/2
  - ➡ Susy RGE's
  - ➡ Interface to SuSpect
  - ➡ Web validation platform.
  - ➡ Moving towards NLO

Enjoy the workshop!