Durham University

IP3

From Lagrangians To Experiments

Claude Duhr
in collaboration with N. D. Christensen and B. Fuks

Laboratori Nazionali di Frascati
Frascati, 05 July 2011

BSM @ LHC

- So far, the LHC has found no signal of Physics Beyond the SM.
- As the MSSM parameter space gets more and more constraint, we have to start thinking about new BSM models.
- Already now, the number of proposed BSM scenarios is huge!
- Even more are to come when an excess of the SM will be observed at the LHC.

BSM @ LHC

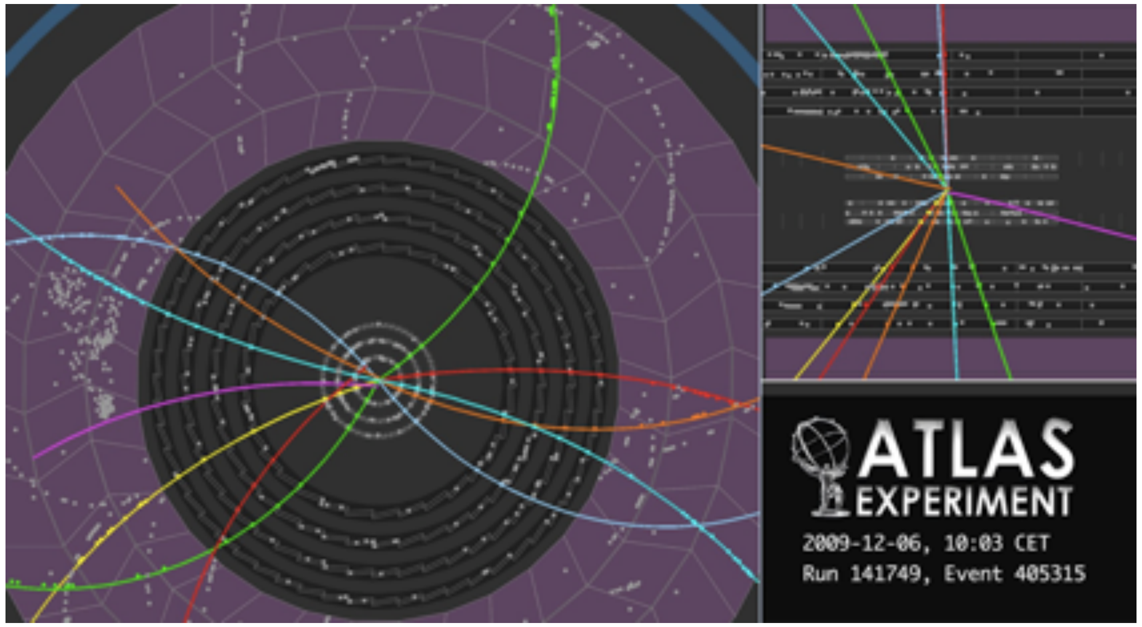
- If an excess is observed, our final goal is twofold:
  - ➡ Identify the New Physics ('Lagrangian')
  - ➡ Measure the parameters inside the LAgrangian ('Benchmark point')
- Two possible approaches:
  - ➡ Top-down: Write down a BSM theory, with new particles, interactions, etc.
  - ➡ Effective theory approach: Add higher-dimensional operators to the SM, induced by some unknown New Physics, to account for a deviation from the SM.
- In any case: we want to find a new Lagrangian.

BSM @ LHC

- If an excess is observed, our final goal is twofold:
  - ➡ Identify the New Physics (‘Lagrangian’)
  - ➡ Measure the parameters in (‘Benchmark point’)
- However, we cannot compare Lagrangians to events at the LHC

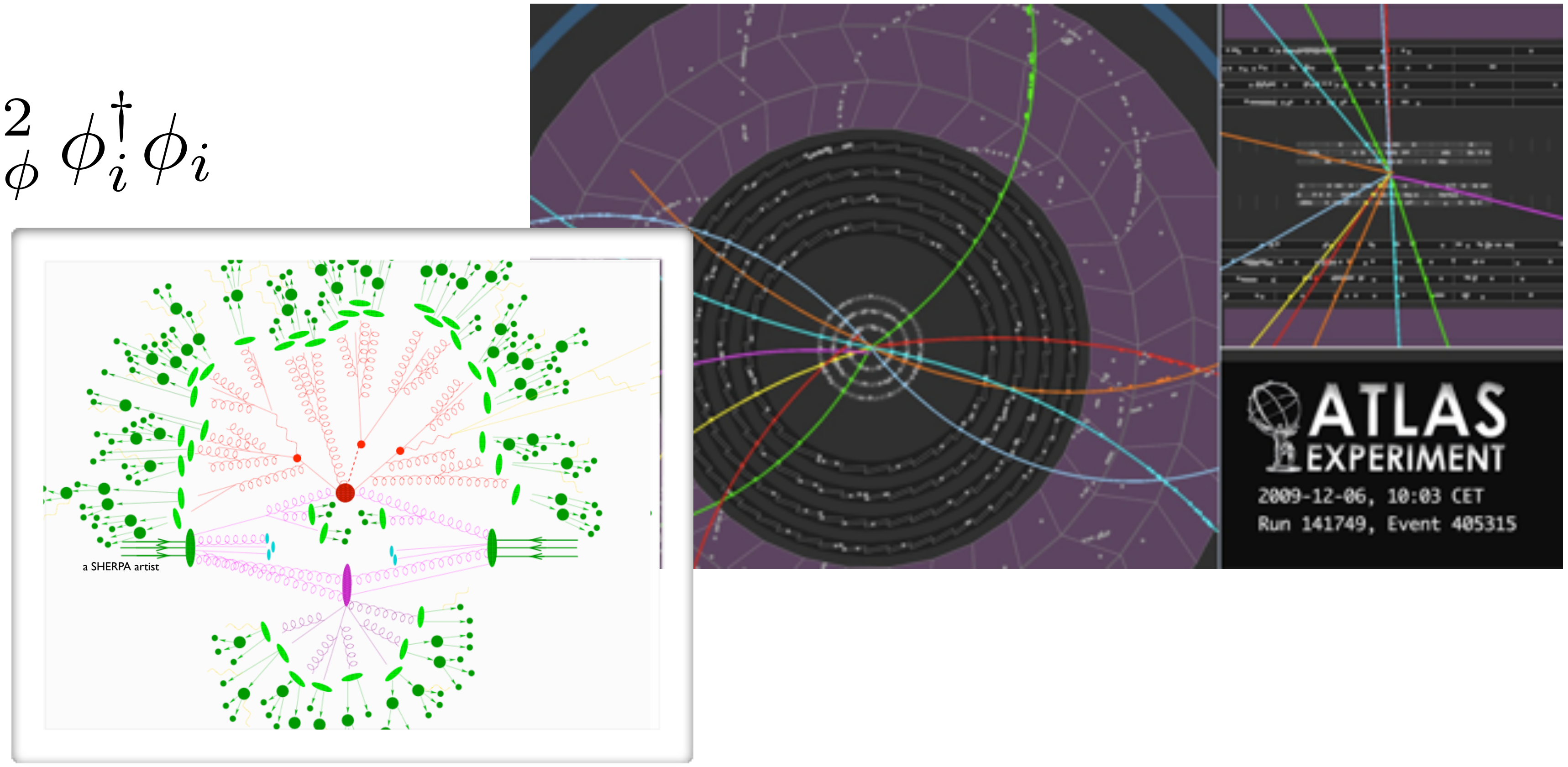
$$\begin{aligned}\mathcal{L} = {} & D_\mu \phi_i^\dagger D^\mu \phi_i - m_\phi^2\, \phi_i^\dagger \phi_i \\ & + \lambda\, \epsilon_{ijk}\, \phi_i\, \bar{t}_j^c P_R b_k \\ & - \lambda\, \epsilon_{ijk}\, \phi_i^\dagger\, \bar{t}_j P_L b_k^c\end{aligned}$$

vs.



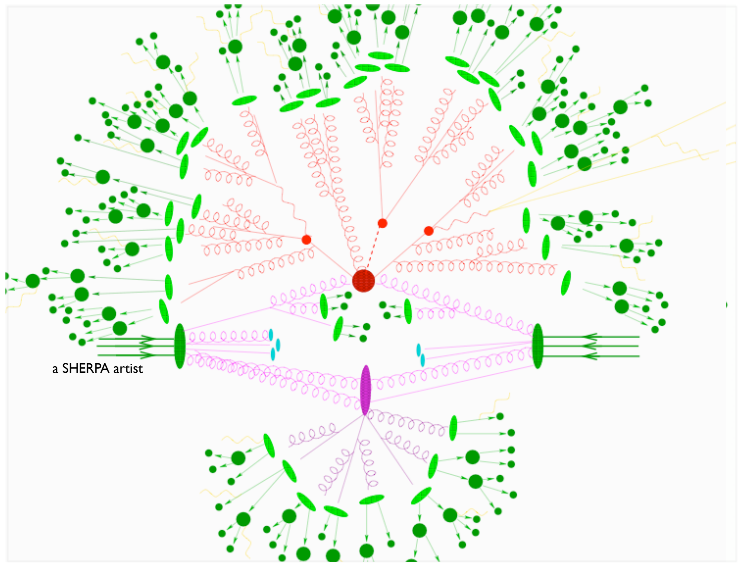
From Lagrangians to Experiments?

$$
\begin{aligned}
\mathcal{L} = {} & D_\mu \phi_i^\dagger D^\mu \phi_i - m_\phi^2\, \phi_i^\dagger \phi_i \\
& + \lambda\, \epsilon_{ijk}\, \phi_i\, \bar{t}_j^c P_R b_k \\
& - \lambda\, \epsilon_{ijk}\, \phi_i^\dagger\, \bar{t}_j P_L b_k^c
\end{aligned}
$$



- We need an additional layer that transforms the BSM model into a simulation of proton collisions, as accurately as possible.
  - ➡ Monte Carlo event generators

From Lagrangians to Experiments?



- Parton Shower Monte Carlo Codes
  - ➡ Herwig
  - ➡ Pythia
  - ➡ Sherpa
- Multi-purpose LO matrix element generators (parton level)
  - ➡ CalcHep / CompHep
  - ➡ MadGraph / MadEvent
  - ➡ Sherpa (AMEGIC++, Comix)
  - ➡ Whizard / Omega

From Lagrangians to Experiments?

- Most of these codes have only a very limited amount of models implemented by default (~ SM and MSSM).
- However, still these codes do not work at the level of Lagrangians, but need explicit vertices.
- The process of implementing Feynman rules can be particularly tedious and painstaking:
  - ➡ Each code has its own conventions (signs, factors of *i*, ...).
  - ➡ Vertices need to be implemented one at the time.
- Most codes can only handle a limited amount of color and / or Lorentz structures (~ SM and MSSM)

From Lagrangians to Experiments?

- Example 1: SUSY model

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

- Very easy ‘theory description’
  - ➡ Choose a gauge group (+ additional internal symmetries).
  - ➡ Choose the matter content (= chiral superfields in some representation).
  - ➡ Write down the most general superpotential.
  - ➡ Write down the soft-SUSY breaking terms.
  - ➡ (+ check validity of the model)

From Lagrangians to Experiments?

- Example 1: SUSY model

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

- ‘Monte Carlo description’
  - ➡ Express superfields in terms of component fields.
  - ➡ Express everything in terms of 4-component fermions (beware of the Majoranas!).
  - ➡ Express everything in terms of mass eigenstates.
  - ➡ Integrate out D and F terms.
  - ➡ Implement vertices one-by-one (beware of factors of *i*, *etc*!)

From Lagrangians to Experiments?

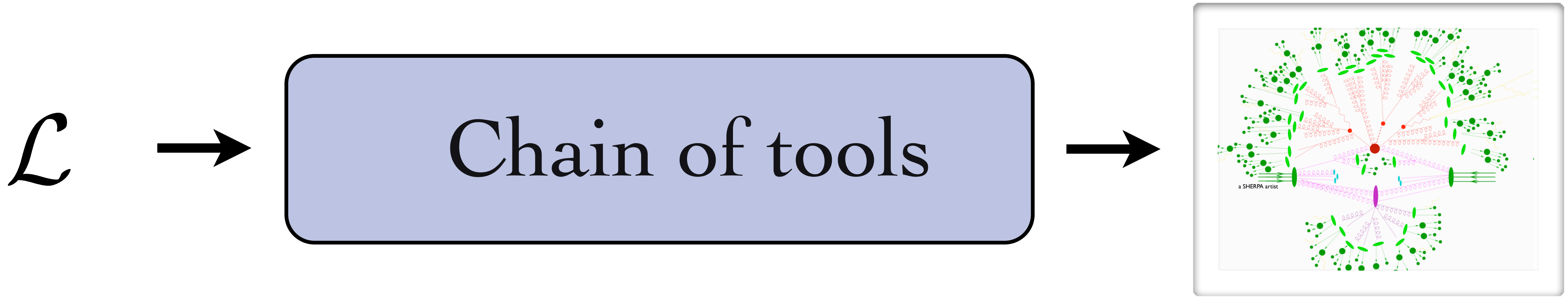
- Example 2: Effective theories

$$\mathcal{L} = g_1 \, \Phi \, \bar{Q}_L \sigma^{\mu\nu} T^a t_R \, G^a_{\mu\nu}$$

- This is a correction to ttH production.
- However, no matrix element generator natively supports this operator.
  - ➡ Implementation might be possible, but very hard to do for non-specialists.
- Similar restrictions occur with non-standard color structures.

From Lagrangians to Experiments?

- The aim of this talk is to present a new framework where the aforementioned problems disappear.
- New workflow:

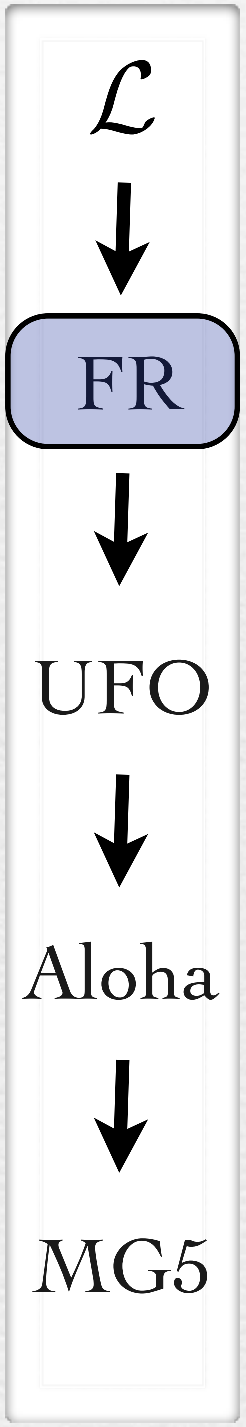


$\mathcal{L}$ → Chain of tools →

- The chain of tools:

FeynRules → UFO → ALOHA → MadGraph 5

FeynRules (CD, B. Fuks, N. D. Christensen)



$\mathcal{L}$
↓
FR
↓
UFO
↓
Aloha
↓
MG5

- FeynRules is a Mathematica package that allows to derive Feynman rules from a Lagrangian.

- 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

FeynRules

$\mathcal{L}$

↓

FR

↓

UFO

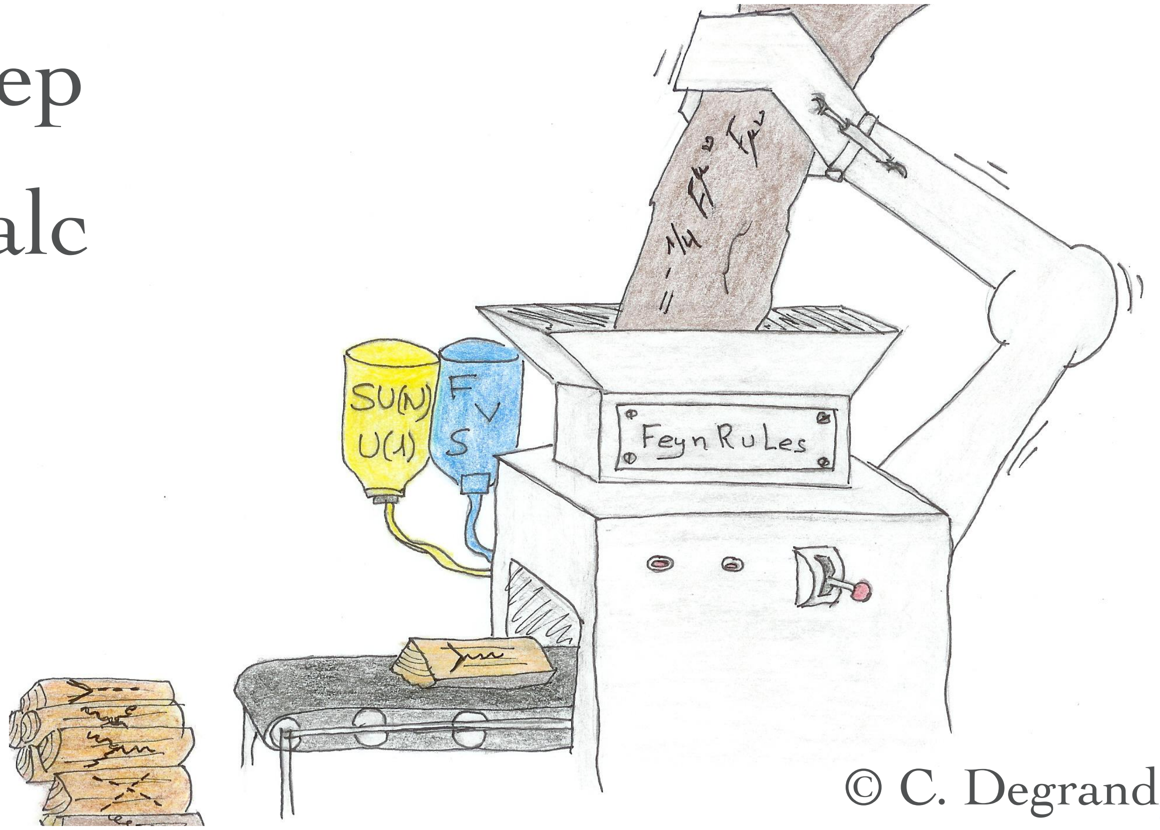
↓

Aloha

↓

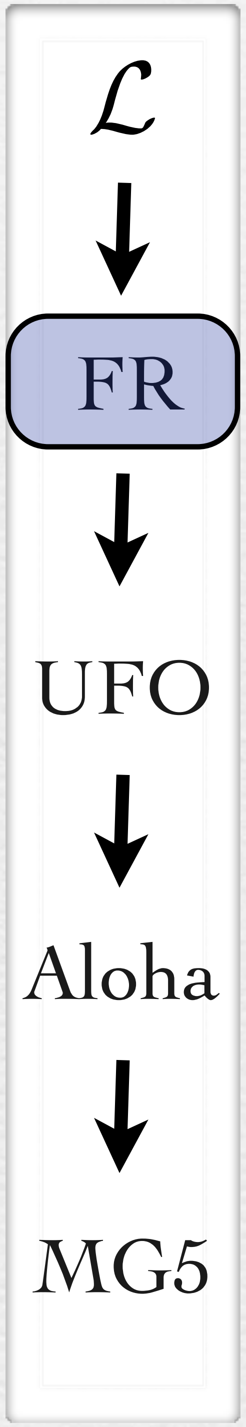
MG5

- 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
  - ➡ MadGraph 4 & 5
  - ➡ Sherpa
  - ➡ Whizard / Omega



© C. Degrande

FeynRules



$\mathcal{L}$

FR

UFO

Aloha

MG5

- 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
  - ➡ MadGraph 4 & 5
  - ➡ Sherpa
  - ➡ Whizard / Omega

© C. Degrande

How to use FeynRules

- 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:

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

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

How to use FeynRules

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

`FeynmanRules[ L ]`

- Equivalently, we can export the Feynman rules to a matrix element generator, e.g., for MadGraph 4,

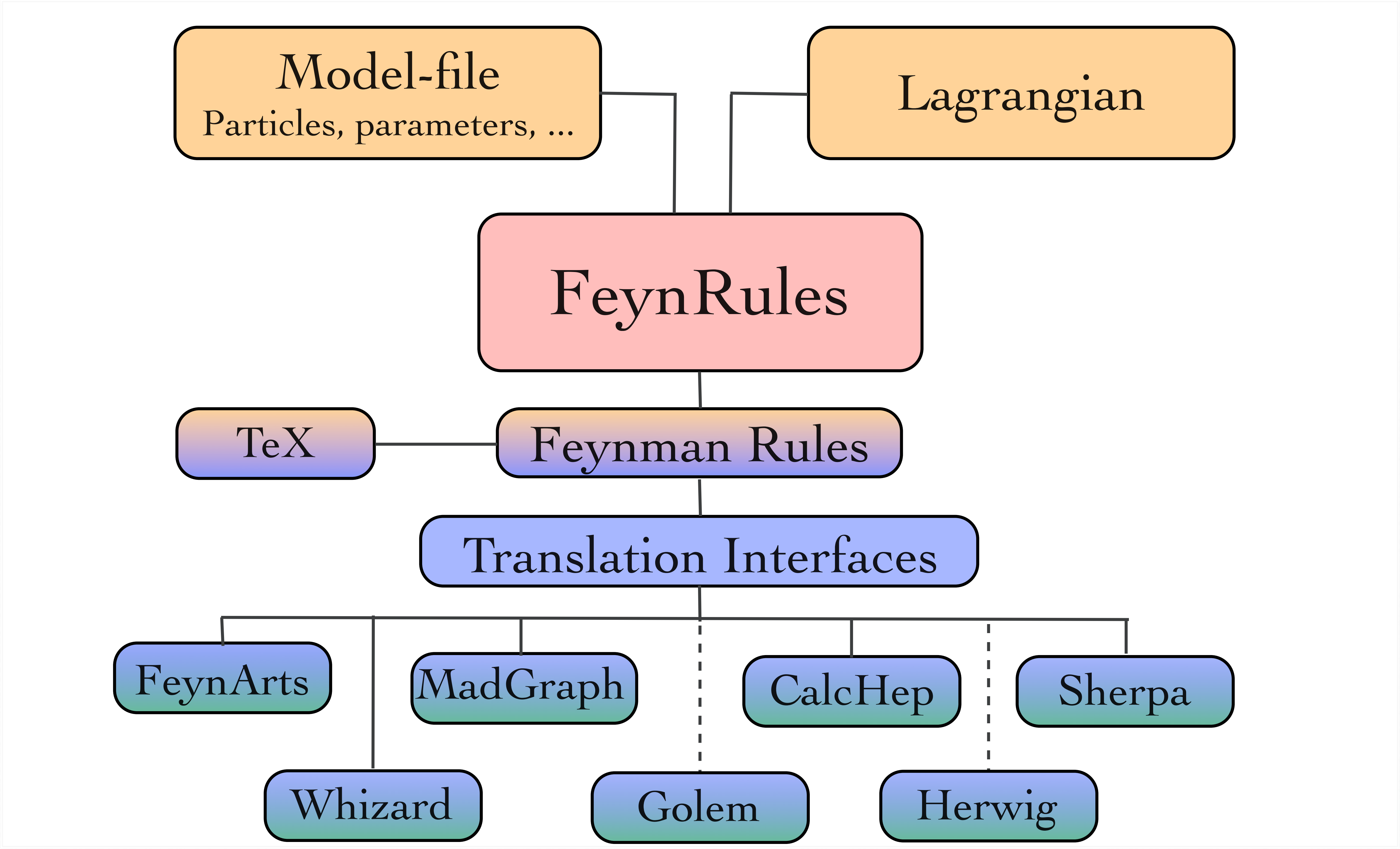
`WriteMGOutput[ L ]`

- This produces a set of files that can be directly used in the matrix element generator ("plug 'n' play").

SUSY in FeynRules

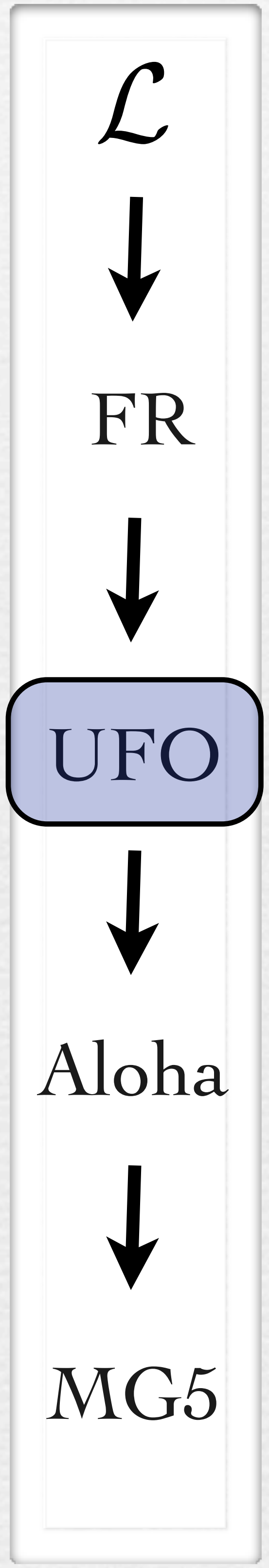
- Recently, we extended FeynRules to work with superfields.
- The user defines his superfields (+ their component fields), and the superpotential.
- FeynRules then
  - ➡ Expands the superfields in Grassmann parameters.
  - ➡ Integrates out the Grassmann parameters.
  - ➡ Integrates out the D and F terms.
  - ➡ Transforms Weyl fermions into Dirac and Majorana fermions.
  - ➡ Computes the Feynman rules.

FeynRules


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

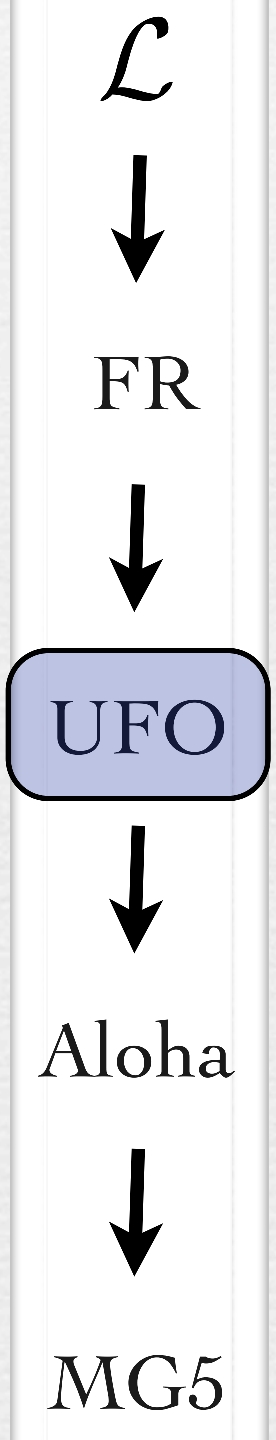
The UFO

(C. Degrande, CD, B. Fuks,
D. Grellscheid, O. Mattelaer, T. Reiter)



- UFO = Universal FeynRules Output
- Idea: Create Python modules that can be linked to other codes and contain all the information
- The UFO is a self-contained Python code, and not tied to a specific matrix element generator.
- Golem, MadGraph 5 and Herwig++ will use the UFO.
- In particular, the UFO is the default model format for MadGraph 5.

The UFO


$\mathcal{L}$

FR

UFO

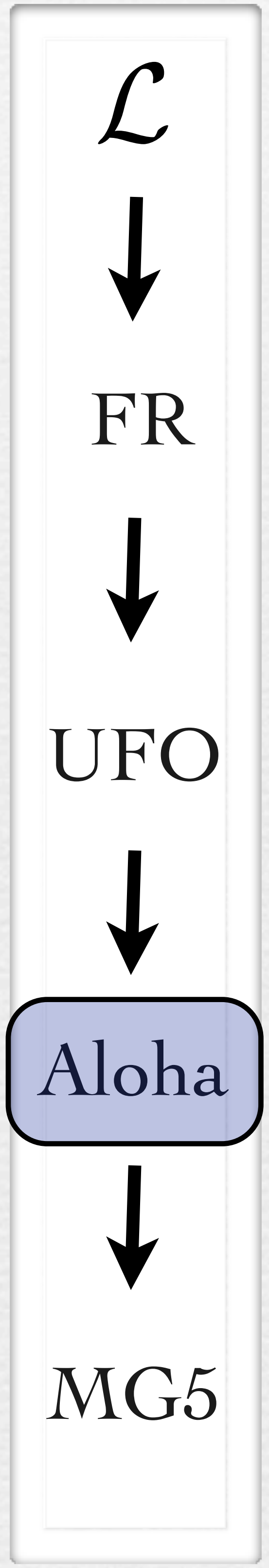
Aloha

MG5

- 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.
- Allowed color structures: 1, 3, 6, 8
- No limitation on the Lorentz structures, or number of particles per vertex!
- The UFO format allows to represent *any* model in terms of a Python module that can be linked to other codes.

ALOHA

(P. de Aquino, F. Maltoni,
O. Mattelaer, T. Stelzer)


$\mathcal{L}$

FR

UFO

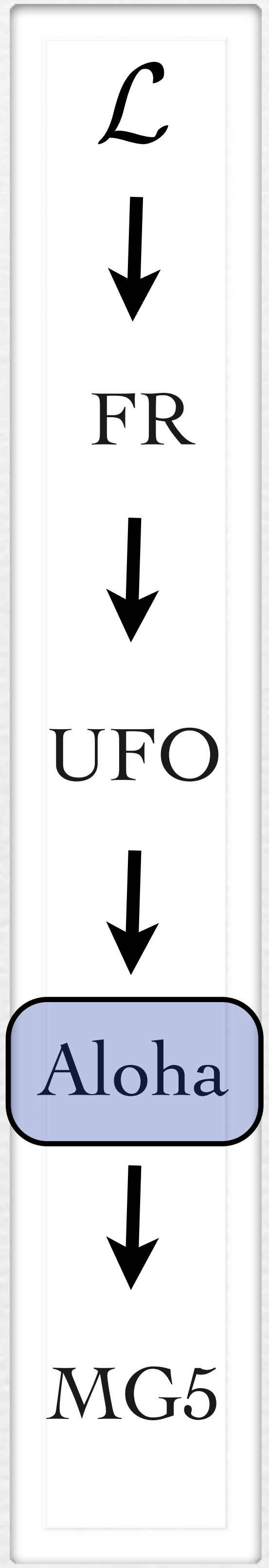
Aloha

MG5

- Many matrix element generators have the information on the Lorentz structures hardcoded.
- In MadGraph, this library is the Helas library.
- In general, these libraries are hardcoded and only contain renormalizable interactions.
- Hence, even if an UFO contains all the information, some codes cannot use it, because they rely on hardcoded libraries

ALOHA

(P. de Aquino, F. Maltoni, O. Mattelaer, T. Stelzer)



$\mathcal{L}$

↓

FR

↓

UFO

↓

Aloha

↓

MG5

- ALOHA is a Python code that reads UFO's, and produces Helas routines for MadGraph 5.
- ALOHA is fully embedded in and distributed with MadGraph 5.
- In other words, unlike v4, MadGraph v5 does not rely on a static Helas library, but the Helas routines are created on the file via ALOHA.
- As a consequence, MadGraph 5 can handle any kind of Lorentz structures!

MadGraph 5

(J. Alwall, M. Herquet,
F. Maltoni, O. Mattelaer,
T. Stelzer)

$\mathcal{L}$
↓
FR
↓
UFO
↓
Aloha
↓
MG5

- MadGraph 5 a complete rewrite of MadGraph 4.
- The code is entirely written in Python.
- By design, MG5 uses UFO's as the default model input, and produces Helas routines on the fly via ALOHA.
- Supported color structures: 1, 3, 6, 8
- Putting everything together, we have a tool chain that allows to simulate events for virtually any model, just by starting from the Lagrangian!

Running the full chain

$\mathcal{L}$

↓

FR

↓

UFO

↓

Aloha

↓

MG5

- Implement the new particle

	Spin	SU(3)	SU(2)	U(1)
ϕ	0	3	1	-1/3

- Implement the new parameter (Yukawa coupling)
- Enter Lagrangian into Mathematica

$$\mathcal{L} = D_\mu \phi_i^\dagger D^\mu \phi_i - m_\phi^2\, \phi_i^\dagger \phi_i$$

$$+\, \lambda\, \epsilon_{ijk}\, \phi_i\, \bar{t}_j^c P_R b_k - \lambda\, \epsilon_{ijk}\, \phi_i^\dagger\, \bar{t}_j P_L b_k^c$$

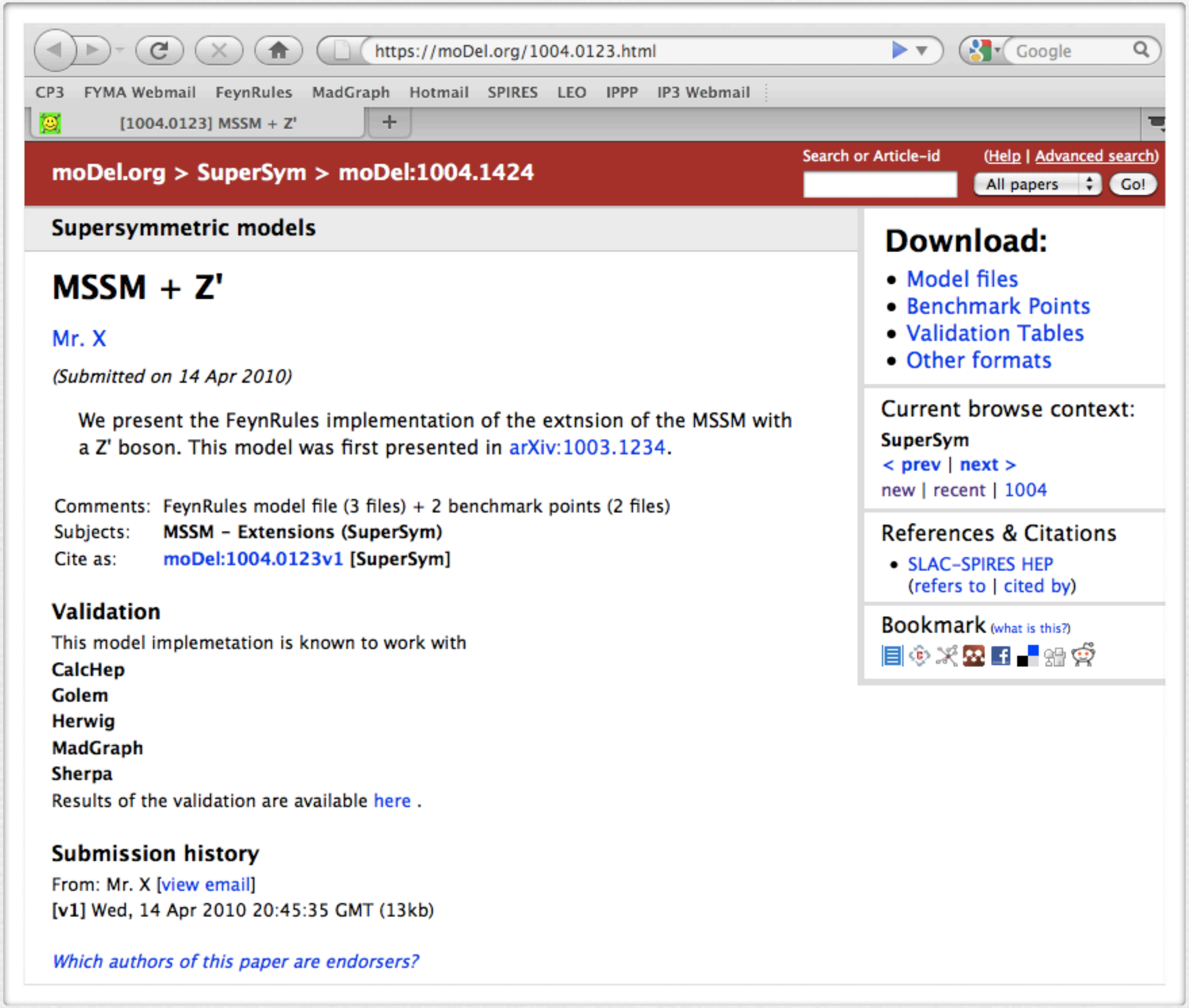
Conclusion

- The chain

  FR → UFO → ALOHA → MG5

  is about to open a completely new era of HE phenomenology!
- This chain allows to simulate events for any BSM model, by just starting form the Lagrangian.
- N.B.: MadGraph 2011 meeting, 19 - 24 September @ the Academia Belgicae in Rome!

Towards a database of models...



moDel.org > SuperSym > moDel:1004.1424

Supersymmetric models

MSSM + Z'

Mr. X

(Submitted on 14 Apr 2010)

We present the FeynRules implementation of the extnsion of the MSSM with a Z' boson. This model was first presented in arXiv:1003.1234.

Comments:	FeynRules model file (3 files) + 2 benchmark points (2 files)
Subjects:	**MSSM - Extensions (SuperSym)**
Cite as:	**moDel:1004.0123v1 [SuperSym]**

Validation

This model implemetation is known to work with

CalcHep
Golem
Herwig
MadGraph
Sherpa

Results of the validation are available here .

Submission history

From: Mr. X [view email]
[v1] Wed, 14 Apr 2010 20:45:35 GMT (13kb)

Which authors of this paper are endorsers?

Download:

- Model files
- Benchmark Points
- Validation Tables
- Other formats

Current browse context:

SuperSym
< prev | **next >**
new | recent | 1004

References & Citations

- SLAC-SPIRES HEP
  (refers to | cited by)

Bookmark (what is this?)