A **Collaborative** platform for matrix-element related development

takes various input

- **MODEL**
- **PROCESS**

and produces various output

- Standalone (C++/Fortran)
- Pythia 8
- MadEvent
- More in development
MadGraph 5
MadGraph 5

MG5 is moving very fast

1.4.0
MG5 is moving very fast

NOT a Small project

=> Need some guidelines

1.4.0
Languages

- Python
  - Easy to Learn/Read
  - Object Oriented
  - Lot of Library
  - Very Flexible
  - Not Compiled
  - Possibility of easy test suite

- Fortran
  - Very fast
  - Very basic => Good for heavy numerical computation.
Languages

- Python
  - Easy to Learn/Read
  - Object Oriented
  - Lot of Library
  - No Perl / Bash / csh / ...
  - Not Compiled
  - Possibility of easy test suite

- Fortran
  - Very fast
  - Very basic => Good for heavy numerical computation.
Structure of the code

- madgraph
- models
- aloha
- Template
- bin
  - input
  - apidoc
  - vendor
  - tests
Structure of the code

- **madgraph**: Python source code for MadGraph 5
- **models**: UFO interface UFO and v4 models
- **aloha**: Source code for output of ALOHA helicity amplitudes from UFO models
- **Template**: Fortran source code and control files for event generation using MadEvent
- **bin**:
  - **input**:
  - **apidoc**:
  - **vendor**:
  - **tests**:

jeudi 10 mai 2012
Structure of the code

- **madgraph**
  - Python source code for MadGraph 5

- **models**
  - UFO interface
  - UFO and v4 models

- **aloha**
  - Source code for output of ALOHA
  - Helicity amplitudes from UFO models

- **Template**
  - Fortran source code and control files for event generation using MadEvent

- **core**
  - Basic objects
  - Color algebra
  - Color amplitudes
  - Diagram drawing
  - Diagram generation
  - HelAmp generation

- **iolibs**
  - In/Output modules:
    - ME output (C++, Fortran, Python)
    - EPS file drawing
    - File writing tools
    - MG4 model import

- **interface**
  - Command line interface modules

- **various**
  - Helper modules

jeudi 10 mai 2012
# Structure of the code

## madgraph
- Python source code for MadGraph 5

## models
- UFO interface
- UFO and v4 models

## aloha
- Source code for output of ALOHA helicity amplitudes from UFO models

## Template
- Fortran source code and control files for event generation using MadEvent

## core
- Basic objects
- Color algebra
- Color amplitudes
- Diagram drawing
- Diagram generation
- HelAmp generation

## iolibs
- In/Output modules:
  - ME output (C++, Fortran, Python)
  - EPS file drawing
  - File writing tools
  - MG4 model import

## interface
- Command line interface modules

## various
- Helper modules

### Pick and choose functionality/plug in new modules easy!
Structure of the code

- Template
  - aloha
    - models
    - madgraph
- bin
  - Run scripts
- input
  - Config files
- apidoc
  - Automatically generated code documentation
- vendor
  - External packages
- tests
  - Test suite
Structure of the code

- **Template**
  - aloha
  - models
- **bin**
  - Run scripts
- **input**
  - Config files
- **apidoc**
  - Automatically generated code documentation
- **vendor**
  - External packages
- **tests**
  - Test suite

- **madgraph**
  - unit tests
  - acceptance tests
  - parallel tests

jeudi 10 mai 2012
Structure of the code

- Template
  - aloha
  - models
  - madgraph
- bin
  - Run scripts
- input
  - Config files
- apidoc
  - Automatically generated code documentation
- vendor
  - External packages
- tests
  - Test suite

- unit tests
- acceptance tests
- parallel tests

Tests for all routines and functionality in the code, including high-level tests like file output.
Structure of the code

- **Template**
  - madgraph
  - models
  - aloha

- **bin**
  - Run scripts

- **input**
  - Config files

- **apidoc**
  - Automatically generated code documentation

- **vendor**
  - External packages

- **tests**
  - Test suite

**Unit tests**
Tests for all routines and functionality in the code, including high-level tests like file output.

**Acceptance tests**
Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.

**Parallel tests**
Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.
Structure of the code

- **Template**
  - aloha
  - models
  - madgraph

- **bin**
  - Run scripts

- **input**
  - Config files

- **apidoc**
  - Automatically generated code documentation

- **vendor**
  - External packages

- **tests**
  - Test suite

**Folder Structure**

- **unit tests**
  - Tests for all routines and functionality in the code, including high-level tests like file output

- **acceptance tests**
  - Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.

- **parallel tests**
  - Tests against another matrix element generator such as MadGraph 4, to ensure that the results are identical. Over 6000 processes compared in SM, MSSM, HEFT and RS models.
Structure of the code

- **Template**
  - **madgraph**
  - **aloha**
  - **models**

- **bin**
  - Run scripts

- **input**
  - Config files

- **apidoc**
  - Automatically generated code documentation

- **vendor**
  - External packages

- **tests**
  - Test suite

- **unit tests**
  - Tests for all routines and functionality in the code, including high-level tests like file output

- **acceptance tests**
  - Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.

- **parallel tests**
  - Tests against another matrix element generator such as MadGraph 4, to ensure that the results are identical. Over 6000 processes in SM, MSSM, RS models.

**TEST ARE BORING TO IMPLEMENT**
Structure of the code

Template
- madgraph
- models
- aloha

bin
- Run scripts

input
- Config files

apidoc
- Automatically generated code documentation

vendor
- External packages

tests
- Test suite

unit tests
- Tests for all routines and functionality in the code, including high-level tests like file output

acceptance tests
- Tests the behavior of the code under different user scenarios, such as generating processes, outputting files, compilation, and running.

parallel tests
- Tests against another matrix element generator such as MadGraph 4, to ensure that the results are identical. Over 6000 processes in SM, MSSM, RS models.

Test are boring to implement
But Are CRUCIAL!!
Unit-test: Should probe each line of the code.

- should test each function (for each input)

- should be very fast.

acceptance-test: Checking “as a user”
Unit-test: Should probe each line of the code.

should test each function (for each input)

should be very fast.

acceptance-test: Checking “as a user”

If the test passes the code is supposed to be working!!!
- Modular Code highly documented
- Fast release of small implementation
- Code Review by an independant developer
  - Small reviews are better than big ones.
- The functionality FIRST
- Think as user

- Test are part of the code (should be 50%)
Launchpad

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Last Modified</th>
<th>Last Commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>lp:madgraph5</td>
<td>Mature</td>
<td>2012-05-05</td>
<td>211. adding model EWdim6 done by C. Degran...</td>
</tr>
<tr>
<td><strong>Series: trunk</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lp:madgraph5/2.0</td>
<td>Experimental</td>
<td>2012-03-12</td>
<td>181. merge with 1.4.3</td>
</tr>
<tr>
<td><strong>Series: 2.0</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/FKSS</td>
<td>Development</td>
<td>10 hours ago</td>
<td>196. complex mass scheme is passed to the ...</td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/1.4.6</td>
<td>Development</td>
<td>19 hours ago</td>
<td>225. Fixed problem with decay chains that ...</td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/spin32</td>
<td>Development</td>
<td>21 hours ago</td>
<td>222. merge with 1.4.5</td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/ML5_faster</td>
<td>Development</td>
<td>2012-05-07</td>
<td>207. 1. Output restructured so that: a)...</td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/madweight</td>
<td>Development</td>
<td>2012-05-07</td>
<td>207. tt~ bar semileptonic was broken becau...</td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/FKSS_new_born</td>
<td>Development</td>
<td>2012-05-07</td>
<td>253. working on an imporing of the madfks...</td>
</tr>
<tr>
<td>born_ko</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/WWW</td>
<td>Development</td>
<td>2012-05-05</td>
<td>11. find a way to have it working on both...</td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/faster_aloha</td>
<td>Development</td>
<td>2012-05-04</td>
<td>241. progress on the structure of the writ...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lp:-maddevelopers/madgraph5/FKSS_new_born</td>
<td>Development</td>
<td>2012-05-04</td>
<td>252. added reweightNLO.inc to the linked f...</td>
</tr>
</tbody>
</table>
- We use bazaar to save progress and be able to merge easily our work
  - merge often (less conflicts and easier to solve)

- We use Launchpad to organize our work
  - serve as communication tools
  - include review
  - Please subscribe to the group maddevelopers
- PEP 08
- do auto-completion/help right away!
- We have a command “ask” which also allow auto-completion
- usefull functions in
  - various/misc.py
  - iolibs/files.py
- Do not repeat your-self (or the work of someone else)
2.0 class map
2.0 class map
Conclusion

- Make Tests
- Use BZR/ Launchpad
- Have FUN with Madgraph 5!!!!
- We are glad to have many developers and many projects moving forward!