Do you have a flair for FLUKA?

Vasilis.Vlachoudis@cern.ch

FLUKA Users Meeting, 9/Nov/2006

About

/fleə(r)/ n [U,C] natural or instinctive ability (to do something well, to select or recognize what is best, more useful, etc.

[*Oxford Advanced Dictionary of Current English*]

What makes a good UI?

General:

- Simple
- Intuitive
- Respects the commonly accepted conventions
- Visually organized
- Native look
- Easily install and setup
- Extensible / Programmable

Especially for FLUKA:

- Do not hide the inner functionality
- Provide a platform for working/analyzing results

Motivation

- During the life-span of a complicated program like FLUKA is expected that if no major re-writing takes place (a very-very time consuming task) there will be a need of features not initially foreseen that cannot fit in the original format, thus breaking all syntax rules and/or backward compatibility.
- FLUKA is full of exceptions which follow sometimes complicated logic, few examples are the following cards:
  TITLE, GLOBAL, BEAM, Geometry definition, PLOTGEOM, COMPOUND, USRYIELD, USRBDX, RADDECAY, EMF-BIAS ...
- Not sufficient error checking and obscure error messages
- Various post processing programs and complicated plotting procedures
- All the above makes it working with FLUKA a hard task, especially for the beginners
- Flair is trying to address the above points, by providing an All-in-one Graphical Interface with a coherent interface.

What is flair

flair = FLUKA advanced user interface

Front-End interface:

- Input Editing
  - mini-dialogs for each card, allows easy and almost error free editing
  - Card grouping in categories and filtering
  - Error checking and validation of the input file during editing;
- Geometry Debugging
- Compiling of Executable
- **Running** and **monitoring** of the status of a/many run(s)

Back-End interface:

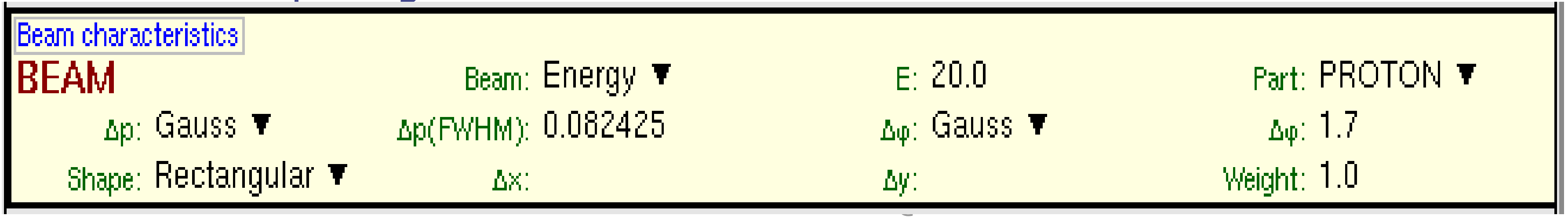
- Inspection and post-processing of the output files
- Plot generation through an interface with gnuplot or 3D photo-realistic images with PovRay (ToDo)

Other Goodies:

- **Nuclear wallet cards**
- **library** of **materials** and **geometrical objects** (ToDo)
- Programming python **API**

Concepts

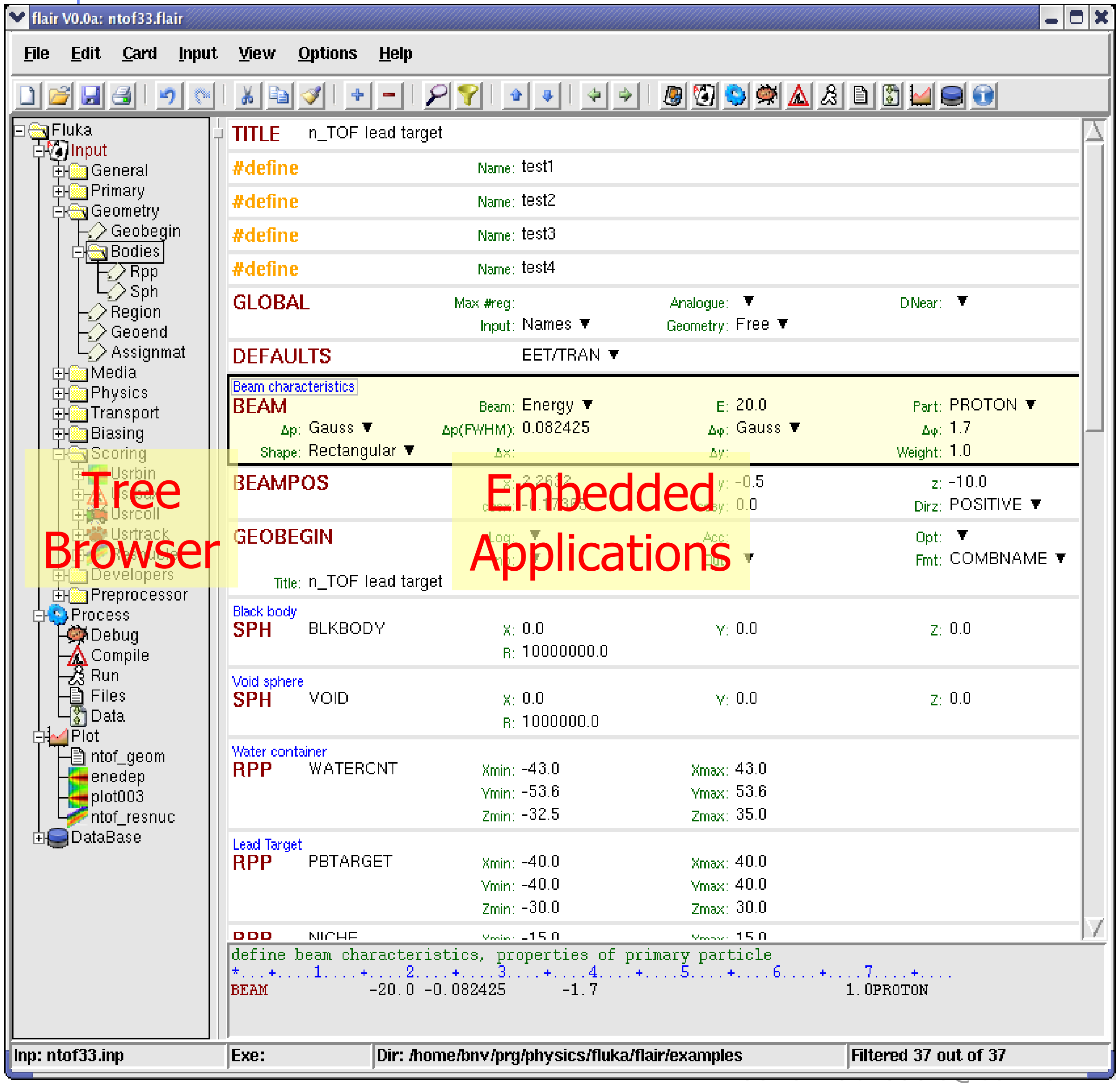
- **FLUKA Project**
  - Store in a single file all necessary information + procedures, from the input file, running of the code, data merging rules and plot generation
  - Flair is an editor for the FLUKA project files.
  - Uses the names format for the input, free with names for the geometry
- **Extended Cards**
  - Flair is treating the input file as a list of extended cards
  - Each extended card contains:
    - Comment (All commented lines preceding the card)
    - Tag and Multiple number of whats (0=sdum, 1-6 first line, 7-12 continuation line...) and one field of extra information (multi line string)
    - State (Enable/Disable)
  - For each extended card flair has a mini dialog (currently 4 columns), interpreting all information stored in the card



Beam characteristics

BEAM	Beam: Energy ▼	E: 20.0	Part: PROTON ▼
Δp: Gauss ▼	Δp(FWHM): 0.082425	Δφ: Gauss ▼	Δφ: 1.7
Shape: Rectangular ▼	Δx:	Δy:	Weight: 1.0

Program Interface



```
TITLE       n_TOF lead target
#define                 Name: test1
#define                 Name: test2
#define                 Name: test3
#define                 Name: test4
GLOBAL         Max #reg:              Analogue: ▼            DNear: ▼
               Input: Names ▼         Geometry: Free ▼
DEFAULTS                 EET/TRAN ▼
Beam characteristics
BEAM           Beam: Energy ▼         E: 20.0                Part: PROTON ▼
   Δp: Gauss ▼  Δp(FWHM): 0.082425    Δφ: Gauss ▼            Δφ: 1.7
   Shape: Rectangular ▼  Δx:          Δy:                    Weight: 1.0
BEAMPOS        x:                     y: -0.5                z: -10.0
               cosx:                  cosy: 0.0              Dirz: POSITIVE ▼
GEOBEGIN       Log: ▼                 Acc:                   Opt: ▼
                                      Inp: ▼                 Fmt: COMBNAME ▼
   Title: n_TOF lead target
Black body
SPH     BLKBODY        x: 0.0         y: 0.0                 z: 0.0
                       R: 10000000.0
Void sphere
SPH     VOID           x: 0.0         y: 0.0                 z: 0.0
                       R: 1000000.0
Water container
RPP     WATERCNT       Xmin: -43.0    Xmax: 43.0
                       Ymin: -53.6    Ymax: 53.6
                       Zmin: -32.5    Zmax: 35.0
Lead Target
RPP     PBTARGET       Xmin: -40.0    Xmax: 40.0
                       Ymin: -40.0    Ymax: 40.0
                       Zmin: -30.0    Zmax: 30.0
RPP     NICHE          Xmin: -15.0    Xmax: 15.0

define beam characteristics, properties of primary particle
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
BEAM           -20.0  -0.082425      -1.7                         1.0PROTON

Inp: ntof33.inp | Exe: | Dir: /home/bnv/prg/physics/fluka/flair/examples | Filtered 37 out of 37
```

- Wrapper of standalone applications
- Tree browser to select application
- Allow different ways of viewing the same object
- Input:
  - Filtering Cards
  - Show card links
  - Units: i.e. 20 GeV/c (ToDo)
  - Data validation
  - Import/Export on various formats
- Process:
  - Debugging
  - Compilation
  - Run monitoring
  - Merging
- Plotting:
  - Interface to plot packages
  - Table of Isotopes
- Python Libraries:
  - Input file manipulation
  - Processing
  - Plotting

Download & Requirements

- Flair web site to download code and documentation

  http://www.fluka.org/flair

  Until the official release, is preferable to use the CVS repository with the instruction in the download section of the web site
- Installation
  - Unpack the code in a directory of your choice i.e. /usr/local/flair
  - Create an alias to the flair executable in your login script

    alias flair=/usr/local/flair/flair
- Besides the latest FLUKA version flair requires:
  - Python interpreter (≥2.3) (http://www.python.org). Present on almost all linux and unix systems.
  - Tkinter, usually is included in the python distribution. Lately some linux versions decided to distribute it as a separate package.
  - Gnuplot (≥4.0) (http://www.gnuplot.info) (and gplevbin substitute of pawlevbin)
  - PovRay (≥3.6) (http://www.povray.org)

How to Contribute

- Python programming
  - Interactive help file
  - Parsing and processing output files
  - Web based database for sharing resources with other users
  - …
- Input Editor
  - Cards Layout, in other formats (from 3 up to 8 columns)
  - Labels have to be intuitive, if something is not comprehensible please propose an alternative
- Manual, Online documentation, Tips database
- Icons for tool bars and cards
- Gnuplot scripts or ideas for better presentation
- Comments & Ideas, on new features that one wants to see
- Testing, Bug reporting
- …

Features to be added

- Interface
  - Interactive Help page of both FLUKA and flair
  - Working on multiple project
  - Configuration dialog
  - Exportation of processing scripts and formats (MCNP...)
- Input Editor
  - Drag 'n Drop
  - Geometry manipulation (Transformations, CSG optimization etc)
- Post Processing
  - Re-binning or USRBINs
  - Maximum trace
- Plotting:
  - Information of Input File
  - Single and double differential quantities (USRBDX, USRYIELD...)
  - 3D Ray Tracing
  - Particle tracks