flair for FLUKA

Vasilis.Vlachoudis@cern.ch

FLUKA Coordination Committee 4/Jul/2007

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*]

Why is UI design important

- User Interfaces are what allows end users to interact with an application.
- A good UI will make an application intuitive and easy to use
- Excellent applications without good UI will be less popular than inferior ones with a good UI

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

What is flair [1/2]

- **FLUKA Advanced Interface** [http://www.fluka.org/flair]
- **All-in-one** Graphical Interface
- With minimum requirements on additional software
- Working in an intermediate level:
  Not hiding the inner functionality of FLUKA

Front-End interface:

- Fully featured **Input file Editor**
  - mini-dialogs for each card, allows easy and error free editing
  - Uniform treatment of all FLUKA cards
  - Card grouping in categories and card filtering
  - Error checking and validation of the input file during editing
  - Templates with basic input files
- **Geometry:** transformation, optimizations and debugging
- **Compilation** of the FLUKA Executable
- **Running** and **monitoring** of the status of a/many run(s)

What is flair [2/2]

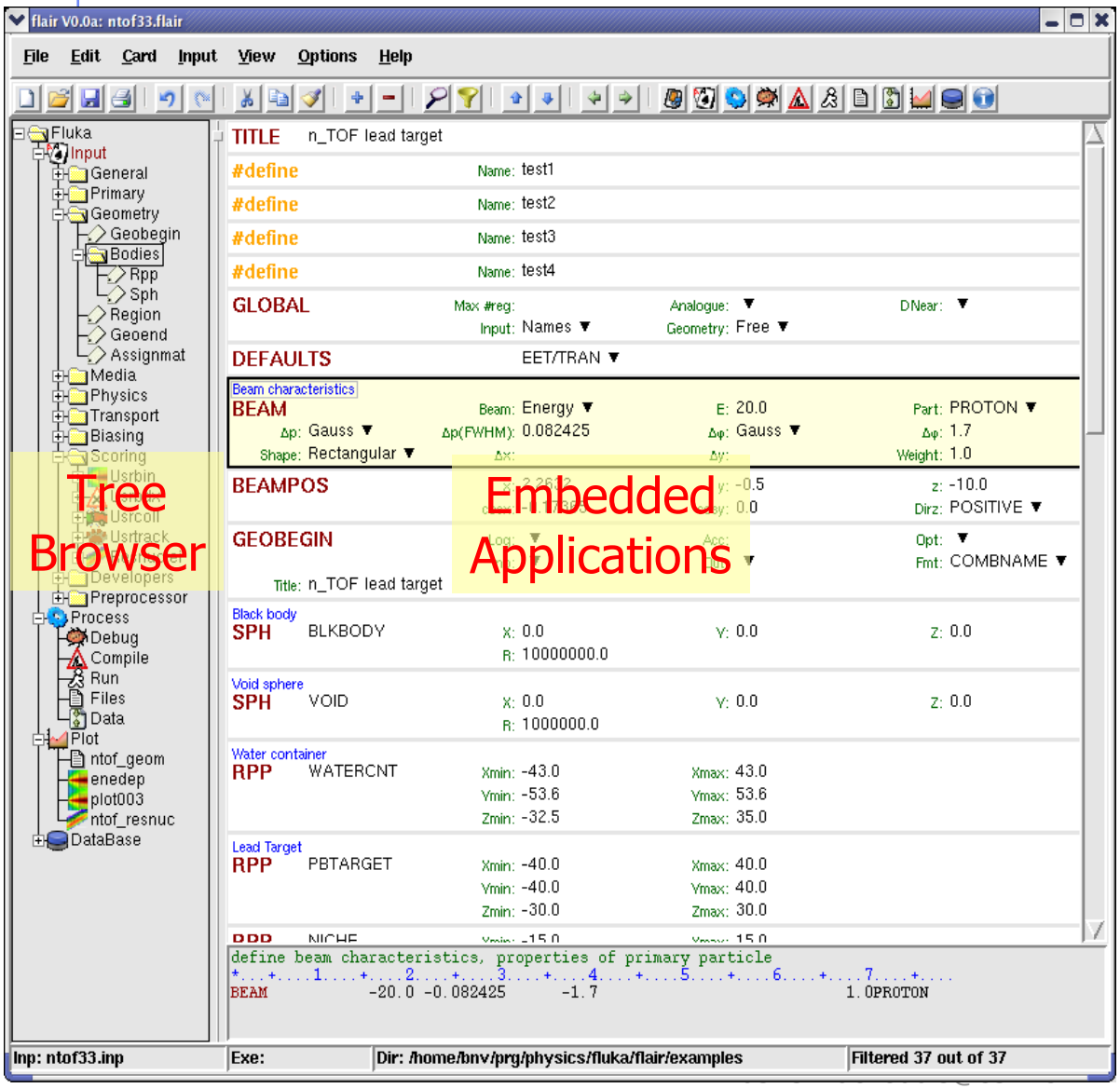
Back-End interface:

- Inspection of the output files
- Post processing (merging) the output data files
- Plot generation through an interface with gnuplot
  Input information, USRxxx, RESNUCLEI and geometry
- 3D photo-realistic images with PovRay (ToDo)

Other Goodies:

- Access to FLUKA manual as hyper text
- Checking for release updates of FLUKA and flair
- Nuclear wallet cards
- Library of materials
- Database of geometrical objects (ToDo)
- Programming python **API**

Program Interface

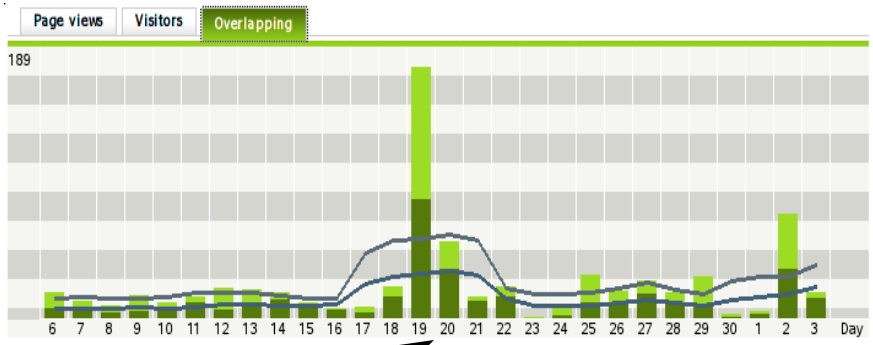


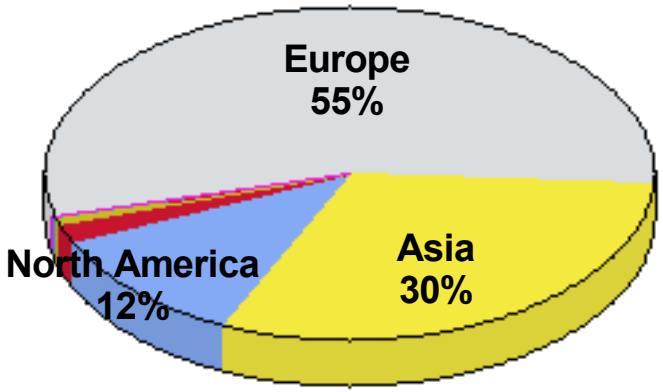
- Wrapper of standalone applications
- Tree browser to select application
- Allow different ways of viewing the same object
- Input:
  - Filtering Cards
  - Show card links
  - Units: i.e. 20 GeV/c (ToDo)
  - Data validation
  - Import/Export on various formats
- Process:
  - Debugging
  - Compilation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

Flair History

Jan 2005	During the FLUKA course at Houston, first idea about a possible graphical interface
Mar 2006	Pavia FLUKA course confirmed the need of such an interface
Jun 2006	Start working on the conceptual design
Nov 2006	Announcement at the CERN FLUKA users meeting
Dec 2006	Announcement at the FLUKA collaboration meeting
May 2007	Introduction and use with success at the FLUKA course at Houston. The program is quite evolved and counting ~50'000 lines of code.
Jun 2007	First public announcement at the FLUKA users list. 250 downloads during the first 24h!

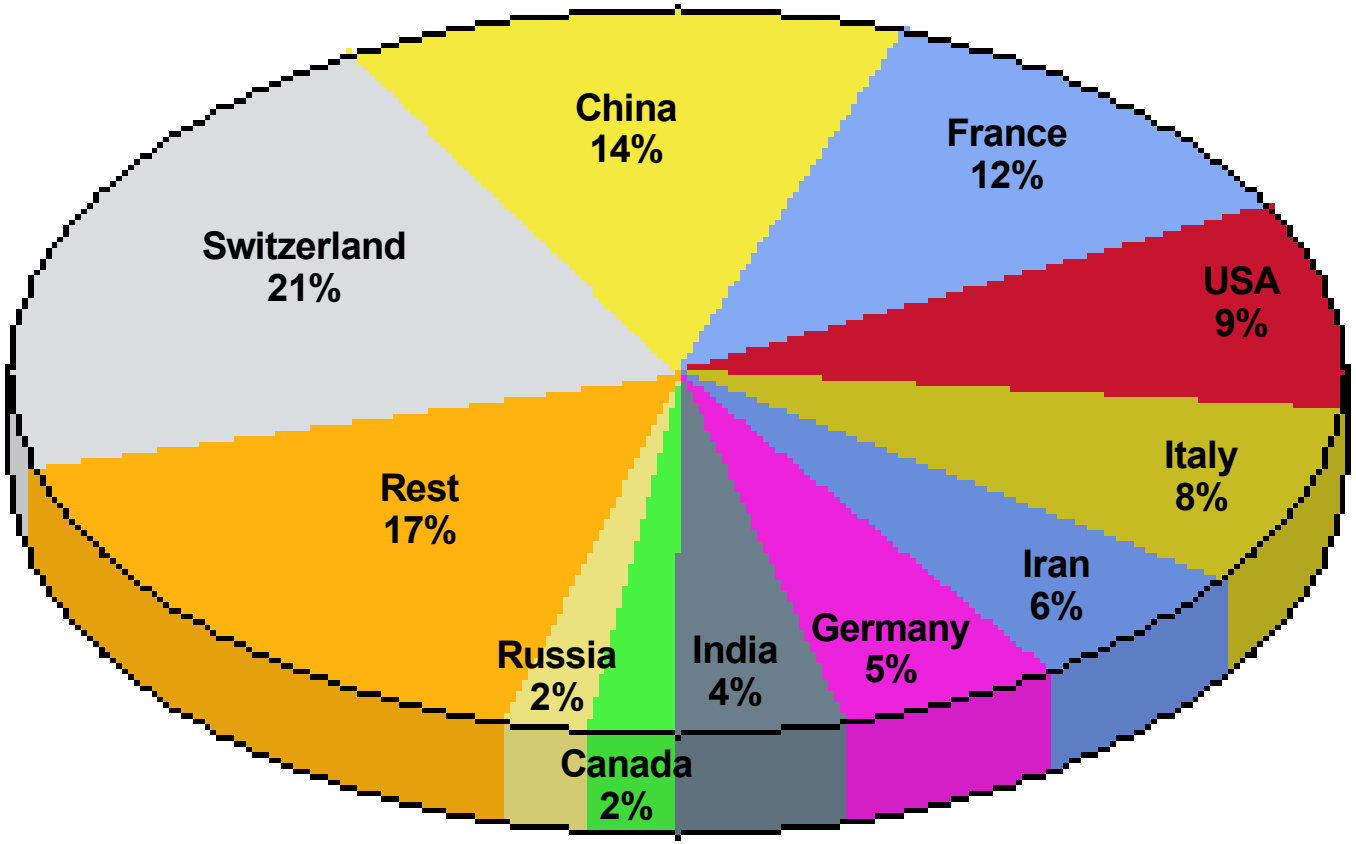
Website statistics





flair 0.5 announced in FLUKA mailing list

Website – Country of origin



Software choices [1/2]

Requirements:

- Open source software
- Multiplatform with easy installation
- Minimum requirements on other package
- Large community of users and years of development

Python [http://www.python.org]
is a scripting language which is:

- interpreted
- interactive
- object-oriented
- like pseudo code
- dynamically typed
- available for many platforms
- extensible with C-API

Software choices [2/2]

- **Tkinter** [http://wiki.python.org/moin/TkInter]
  default GUI toolkit for Python.
  Good for simple UIs.
  Portable, wrapper around tk/tcl

- **Gnuplot** [http://www.gnuplot.info]
  is a command-line driven, interactive function plotting program specially suited for scientific data representation. Gnuplot can be used to plot functions and data points in both two and three dimensions and in many different formats.

- **Povray** [http://www.povray.org]
  POV-Ray™ is short for the Persistence of Vision™ Raytracer, a tool for producing high-quality computer graphics. POV-Ray™ is copyrighted freeware. POV-Ray is the worlds most popular raytracer.

Flair Project

- Store in a single file all relevant information:
  - Project notes
  - Links to needed files: input file, source routines, output files …
  - Multiple runs from the same input file, as well running status
  - Procedures on how to run the code
  - Rules on how to perform data merging
  - Information on how to post process and create plots of the results
- You can consider flair as an editor for the project files.
- Can handle any FLUKA input format (reading & writing), but internally it works using the names format for the input, free with names for the geometry (Recommended way of working)
- The format is plain ASCII file with extension: .flair

Card Categories

For easier access, cards are groups in the following categories:

- **General** General purpose (TITLE, DEFAULTS, GLOBAL…)
- **Primary** Definition of the primary starting particles
- **Geometry** Cards related to the definition of the geometry bodies/regions/lattices plotting and rotations/translations
  - **… Bodies** Subcategory containing only the bodies definition
- **Media** Definition and assignment of materials
- **Physics** Setting physics properties of the simulation
- **Transport** Modify the way particles are transported in FLUKA
- **Biasing** Cards for importance biasing definition
- **Scoring** Cards related to scoring
- **Developers** *…reserved for FLUKA developers…*
- **Preprocessor** definitions for creating conditional input files

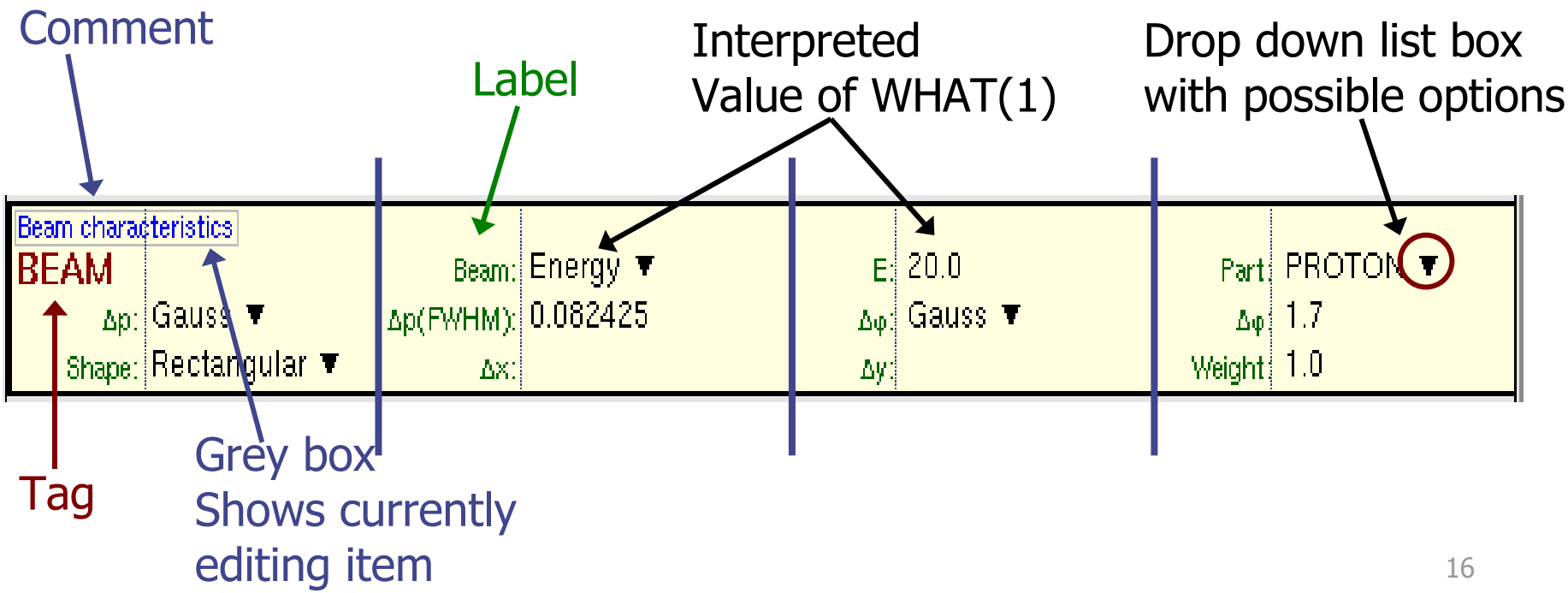
Extended Cards

- Flair is treating the input file as a list of extended cards
- Each extended card contains:
  - Comment: All commented lines preceding the card(s) as well the inline comments
  - 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)
- Special cards to homogenise the FLUKA input file representation:
  - REGION, referring to a region declaration in geometry
  - COMPOUND, all compound cards related to one material are joined in one card
  - END card for bodies/regions is no longer required
- Flair will try to find the best floating point representation of each number, to ensure the maximum accuracy; number of digits that fits in the specific width (10 for the fixed format, 22 for the free format)

Anatomy of a card mini-dialog [1/2]

- For each extended card flair has a mini dialog (currently in 4 columns), interpreting all information stored in the card

```
* Beam characteristics
BEAM            -20.0 -0.082425      -1.7                          1.0PROTON
```



Anatomy of a card mini-dialog [2/2]

```
* Energy deposition in 3D binning
USRBIN         10.0     ENERGY      -50.0       45.0       54.0       36.0EneDep
USRBIN        -45.0      -54.0      -33.0      100.0      100.0      100.0&
```

USRBIN				
Type: X-Y-Z	Xmin: -45.0	Unit: 50 BIN	Name: EneDep
Part: ENERGY	Ymin: -54.0	Xmax: 45.0	NX: 100.0
	Zmin: -33.0	Ymax: 54.0	NY: 100.0
		Zmax: 36.0	NZ: 100.0

```
* Polypyromellitimide Polyimide, Kapton
* Chemical       O = C   H-C   C = O
* Formula          / \  // \ / \      H-C - C-H        H-C - C-H
*                 /   \C    C   \    //     \\       //     \\
*            ---- N     |    ||   N - C       C - O - C       C ----
*                 \   /C    C   /    \     /         \     /
*  C  H  N O       \ /  \\ / \ /      H-C = C-H        H-C = C-H
*   22 10 2 5    O = C   H-C    C = O
MATERIAL                                1.43                            Polyimid
COMPOUND        10.0  HYDROGEN      22.0    CARBON       2.0  NITROGENPolyimid
COMPOUND         5.0    OXYGEN                                          Polyimid
```

MATERIAL			
Z:	Name: Polyimid	#	ρ: 1.43
	Am:	A:	dE/dx:

COMPOUND			
f1: 10.0	Name: Polyimid	Mix: Atom	Elements: 6
f3: 2.0	M1: HYDROGEN	f2: 22.0	M2: CARBON
f5:	M3: NITROGEN	f4: 5.0	M4: OXYGEN
	M5:	f6:	M6:

Validating input and Error correction

- flair validates the input while loading and each card during editing.
- Errors are highlighted with **red**.
- Popup-menu option "Show errors" displays a short message on what is expected as correct value.
- Menu item "Input / Filter Invalid" shows only the invalid cards from the last filtered view

Material Database

- flair contains an internal database of ~500 predefined materials / compounds.
- Some (~300) with the Sternheimer parameters

- **Please use this data as Reference only**
- Validate always the correctness of the data
- If errors found please contact the author

- The database can be edited, and populated with your own materials. In this case a local copy of the database will be made in ~/.flair directory

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)

Features to be added

- Interface
  - Working on multiple project
  - Exportation of processing scripts and formats (MCNP...)
- Input Editor
  - Full Undo/Redo
  - Show hidden cards
  - Geometry manipulation (Transformations, CSG optimization etc)
  - Error checking on correlated information
- Post Processing
  - Re-binning or USRBINs
  - Maximum trace
- Plotting:
  - Information of Input File
  - Double differential quantities for USRBDX
  - 3D Ray Tracing