# Agenda of MAD-X Day

## frs.home.cern.ch/frs/Xdoc/mad-X_day.html

| Time | Speaker | Subject |
|---|---|---|
| 8:30 - 8:50 | F. Schmidt | MAD-X An Overview |
| 9:00 - 9:20 | H. Grote | MAD-X Design Criteria |
| 9:30 - 9:50 | F. Schmidt | PTC Integration into MAD-X |
| 10:00 - 10:20 | W. Herr | Closed Orbit Techniques in MAD-X |
| | Coffee Break | |
| 11:00 - 11:10 | J. Jowett | Windows Version of MAD-X |
| 11:10 - 11:20 | J. Jowett | Higher Level Interaction with MAD |
| 11:30 - 11:50 | H. Burkhardt | MAD-X for Transfer Lines |
| | Lunch Break | |
| 13:00 - 13:20 | F. Zimmermann | MAD-X for Linear Colliders (CLIC) |
| 13:30 - 13:50 | C. Milardi (Frascati) | MAD-X for Small Machines |
| 14:00 - 14:20 | N. Malitsky (BNL) | MAD-X & UAL linked by SXF |
| | Pause | |
| 15:00 - 16:00 | All | Open Discussion Session |
| | Coffee Break | |
| 16:30 - 18:00 | All | Treating Examples on Computers |

# F.Schmidt. An Overview
# Why MAD-X

- **MAD8: well debugged but:**
  - Program Structure has grown much too complex over the years
  - Zebra Memory Management outdated and maintenance questionable
  - Modules interleaved and not independent
  - New modules hard to integrate
  - Upgrade to TPSA, Normal Form etc practically impossible
- **MAD9: modern, C++, object-oriented but:**
  - Very complex & slow
  - Our group invested a lot of time to get to run it for the LHC, but to no avail…
  - Hardly any expertise in C++ in our group

➔ **New Project based on MAD8 called MAD-X to quickly have a tool ready for LHC Design work**

# Project Style

## Organized Team work rather than a single expert

1. One "custodian" to look after the program core and to oversee module development
2. Team of module keepers
3. Disadvantages: more work for users (Code, Example & Documentation!!!), more disciplined and structured work (CVS etc), less homogeneous code, some modules may be delayed, succession issue
4. Advantages: better sharing of work load, potentially better code since user defined, user steer progress of code, users relate better to code, I.e. improved user satisfaction

# F.Schmidt. An Overview
## Module Keepers

1. Alexander Bolshakov (ITEP) – various additions
2. André Verdier – survey, thintrack
3. Daniel Brandt– ibs
4. Etienne Forest (KEK) – PTC proper
5. Frank Schmidt – Custodian, c6t, twiss, PTC modules
6. Frank Zimmermann – dynap
7. Fulvia Pilat (BNL) – SXF
8. Helmut Burkhardt – makethin, MAC system 10
9. John Jowett – Windows Version
10. Oliver Brüning – match
11. Ralph Aßmann – emit
12. Thys Risselada – threader & MMK secretary
13. Tommaso d’Amico – plot
14. Werner Herr – error, cororbit

# F.Schmidt. An Overview
# Status and Outlook

- MAD-X is ready and used in the LHC design
- It is well debugged and documented
- There is a host of new features
- The code is well prepared for long-term maintenance
- New modules can be added with ease
- There is an excellent extension path using PTC (see talk):
  - Most left out modules can be replaced by much better ones
  - Better physics models even for extreme cases
  - Complete TPSA packages

# F. Schmidt . An Overview
# Purpose of MAD-X Day

- **Presentation of MAD-X Code to the World Accelerator Community**
- **Assessment of the present MAD-X Code including Services**
- **Collecting missing features of MAD-X**
  1. Covered by PTC
  2. Requiring New or Upgraded Modules
  3. Windows Version
- **Distributing work load to Module Keepers**

# I Part of MAD-X Day
## Core Software

- **Hans Grote:** Design Criteria
- **Frank Schmidt:** PTC Integration
- **Werner Herr:** Closed Orbit Techniques

# H.Grote. MAD-X Design Criteria
## Program Organization

- **Right from the start the following principles were adopted:**
  - the program will consist of one core part and attached modules
  - the core will be managed by the "custodian", the modules by "module keepers"
  - it must be very easy to add modules, and they must be independent
  - the data structure for the input and output of the modules, and the interface to the modules will be provided in the core program
  - the input language will follow to some extent the more structured format of MAD-9
- **For the core (which had to be written from scratch) C was chosen as programming language, because:**
  - it offers structures and object-orientation at least for the data
  - its memory management is truly dynamic
  - it is known by many people in the AP group
  - it can easily be mixed with Fortran
- **The modules can be in any languages that can be linked with C**

# H.Grote. MAD-X Design Criteria
## Program Organization (continued)

- **When adapting the MAD-8 modules, the following principles were applied:**
  - elimination of Zebra
  - the module output is normally stored in TFS format tables (this format has been developed originally for LEP online data)
  - many low-level checks in the code have been moved up to the core
- **The input language was extended to include the following new features:**
  - truly instantaneous and deferred definition of variables
  - command-line access to variables in tables for further usage
  - vectors of numbers, or strings
  - IF…ELSEIF…ELSE, and WHILE constructs
  - macros: group of commands with the possibility to replace formal parameters by real ones
- **Last but not least, certain execution errors are sent to termination routines that allow sensible error messages instead of "segmentation violation" followed by 100 Mbyte memory dump.**

# H.Grote. MAD-X Design Criteria
## Program Organization (continued)

- **Although intended right from the start, it is only recently that the program became really "stand-alone":**
  - no special libraries are needed (e.g. CERN, XLib, etc.) for the new version without online graphics
  - the plot program writes its own PostScript file
  - the regular expression match was finally written from scratch in C, because it did not exist under Windows
  - the expression decoding I took from a converter program I wrote some time ago in C++
  - the minimization routines were taken from MAD-8 (double precision versions of CERN Library routines)

# H.Grote. MAD-X Design Criteria
## WHILE + table access

### WHILE + table access

```
n=0; q1x=1.25; q2x=1.26;
deltaf=1.e-3; deltad=-1.e-3; eps=1.e-6;
kqf0=kqf; kqd0=kqd;
k11=deltaf*kqf0; k21=deltad*kqd0;
k12=deltaf*kqf0; k22=-deltad*kqd0;
twiss;
q10=table(summ,q1); q20=table(summ,q2);
while(abs(q1x-q10)+abs(q2x-q20) > eps) {
n=n+1;
kqf=kqf0+k11; kqd=kqd0+k21;
twiss;
q11=table(summ,q1)-q10; q21=table(summ,q2)-q20;
kqf=kqf0+k12; kqd=kqd0+k22;
twiss;
q12=table(summ,q1)-q10; q22=table(summ,q2)-q20;
nom=q11*q22-q12*q21;
dkqf=((k11*q22-k12*q21)*(q1x-q10)
     +(k12*q11-k11*q12)*(q2x-q20))/nom;
dkqd=((k21*q22-k22*q21)*(q1x-q10)
     +(k22*q11-k21*q12)*(q2x-q20))/nom;
kqf=kqf0+dkqf; kqd=kqd0+dkqd;
kqf0=kqf; kqd0=kqd;
twiss;
q10=table(summ,q1); q20=table(summ,q2);
}
```

# Some Facts about PTC

- Medium sized Package:

  PTC 68'000 Lines

  MAD-X 53'000 Lines
- Part I of PTC: The Truncated Power Series Analysis (TPSA) called FPP, comprises 20 Years of Development in Power Series and Normal Form Analysis. Takes full advantage of object oriented programming and polymorphism.
- Part II of PTC proper: (5 last years) makes use of FPP to track through magnets in a truly symplectic way.

# F.Schmidt. PTC Integration into MAD-X
## Magnet Treatment in PTC

- **Same Elements as in MAD-X, including all special features $\varepsilon_1$, $\varepsilon_2$, $H_1$, $H_2$ etc.**
- **Rectangular and Sector bends in all generality**
- **Magnets are preferably treated as thick elements including multipole errors, which will be split for symplectification**
- **Elaborate Symplectification Techniques**
  - Various Split Combinations: Drift_Kick, Matrix_Kick, Delta_Matrix_Kick etc
  - Different Split Method to several Orders
  - Number of Splits
  - Exact Option: on/off
- **Normal Form: Order & Phase Space variables**

# F.Schmidt. PTC Integration into MAD-X

## How will we use PTC in MAD-X?

- Linear coupled Lattice Functions TWISS3 (presently available in rudimentary form)
- Replace Modules that involve Normal Form Analysis DYNAMIC & STATIC (presently available in rudimentary form)
- Parameter Dependence of Variables of Interest on other Variables ➔ Nonlinear Matching
- Extend the Physics Models in particular for small Machines
- Proper treatment of Geometry of Magnet in the Tunnel
- Survey with CAD like presentation including particle tracks

# W.Herr. Closed Orbit Techniques in MAD-X
## Orbit correction in MAD8

- **What was there in MAD-8 ?**
  - correct closed orbit with MICADO
- **What was missing?**
  - trajectory correction
  - additional algorithms
  - treatment of missing monitors
  - realistic simulation features
  - correction to (non-zero) traget orbits
- **What has changed?**
  - everything

# Additional algorithms

- Best kick: MICADO (for use with small number of correctors)
- Pseudo inversion (least squares, LSQ) of the problem (use of all available correctors)
- Singular Value Decomposition (SVD) (uses all correctors, but minimizes strength)
- Selected by MODE=MICADO/LSQ/SVD

# Orbit simulation

- Simulation of measurement errors:
  - Monitor misalignment (specified or random)
  - Missing monitors:
    * Specified monitors
    * Random according to given percentage
  - Monitor scaling errors (specified or random)
- Done at level of orbit correction module
- Allows to accumulate large statistics
- Useful to simulate correction procedures

# Treatment of missing elements

- **Missing correctors: in general no problem (but worse correction)**
- **Missing monitors: singular matrix/ill-conditioned problem**
- **MICADO correction with MAD8 will crash, MAD-X will stop with diagnostics or:**
  - **Conditioning with SVD possible**
  - **New command COND=0/1**
  - **Corresponding (redundant) correctors disabled**
- **Correction with SVD: strength distributed between redundant correctors**

# Simulation of optical errors

- **MAD8: orbit (to be corrected) = model (for correction) = Twiss table**
- **MAD-X: can be taken from previously saved (named) Twiss table with: ORBIT=*name*, MODEL=*name*, TARGET=*name* any can be omitted → default = last Twiss table**
- **Allows to test correction with imperfect (or ideal) model**

# Part II of MAD-X Day
## Windows Version and Mathematica Interface

- **John Jowett:** Windows Version
- **John Jowett:** Higher Level Interface

# J.Jowett. Windows Version of MAD-X
## Access and Future Plans

Download "Beta-version" from MAD-X for Windows Web page, accessible from http://cern.ch/mad/

Follow simple installation instructions

- No registry entries required.
- Recommended to set up file associations by analogy with Windows MAD-8.

Current version: MAD-X 1.12, of 13 June 2003

Next version:

- Bring up to date with Linux version
- Expand installation instructions to level of MAD-8
- Expect to restore built-in graphics (system-independent Postscript file creation only).
- Include PTC, etc ?

# J.Jowett. Higher Level Interaction with MAD
## What happened elsewhere

If starting now a IT project manager's advice would be:

- Don't even think about inventing a new language for such a specialised purpose: get one off the shelf.
- *Especially* if your resources are limited.
- Use XML to describe your data (like everybody else).

Developers of SAD program adopted the MAD language

- but gradually modified it to adopt the *syntax* of the Mathematica language.

BeamOptics package

- Optics and other calculations in Mathematica

MAD is not a world unto itself

- Typically, results computed by MAD are part of some calculation done in a wider environment.

# J.Jowett. Higher Level Interaction with MAD
## Madtomma

Since 1994, we have been developing a suite of Mathematica packages called (inappropriately) Madtomma

Contributors have included:

- Yuri Alexahin, Rachel Burgess, Kryzstof Goral, Moira Gresham, Simon Tredwell, JMJ

Related developments in other places, e.g.,

- H. Owen, J. Jones et al at ASTeC, UK

(For the moment) we do not try to change MAD but to imbed it in a larger environment

- Could go further by implementing the MathLink protocol in MAD itself: a project

# J.Jowett. Higher Level Interaction with MAD
## Some things that have been implemented

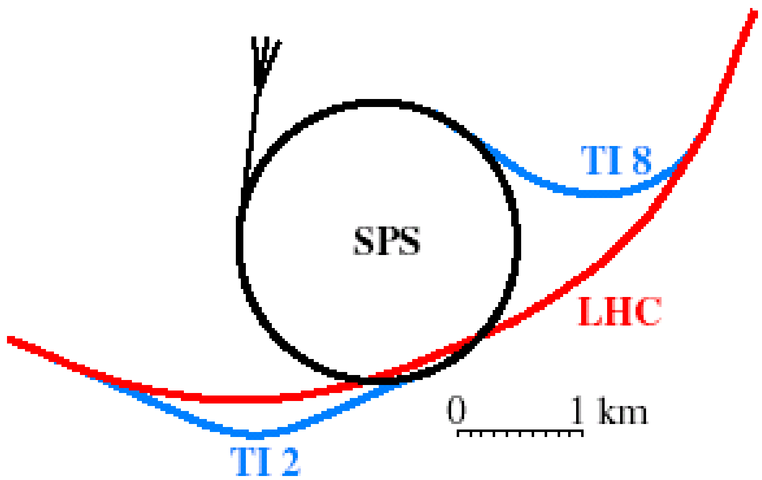
- Plots and tables galore!
- Resonance driving terms
- Automated dynamic aperture scans in 3D and 4D phase space
  - Much used for LEP
- Systematic Monte-Carlo of errors in imperfect machines
  - Databases of machine properties, correlations
- Interactive tracking
  - Point and click to launch particles
  - Interactive transformations to normalised coordinates, action-angle
- The LHC Optics Web
  - LHC home page/Optics and Layouts
- Detailed comparison between LHC versions
- Complex sequence-editing
- Orbit correction by SVD
- LHC Aperture Model (new)
  - 3D Graphics, Java
- MAD input from Web (soon!)
- Localisation of losses on LHC beam screen inside magnets (new)
- Beam envelope visualisation in real space
  - Local and global coordinates (example later)

# Part III of MAD-X Day
## Applications

- **Helmut Burkhardt:** Transfer Lines
- **Frank Zimmermann:** Linear Colliders (CLIC)
- **Catia Milardi (Frascati):** Small Machines
- **Nikolay Malitsky (BNL):** MADX-SXF-UAL

# Transfer Lines TI 2, 8

New Injection lines from the SPS into the LHC ring

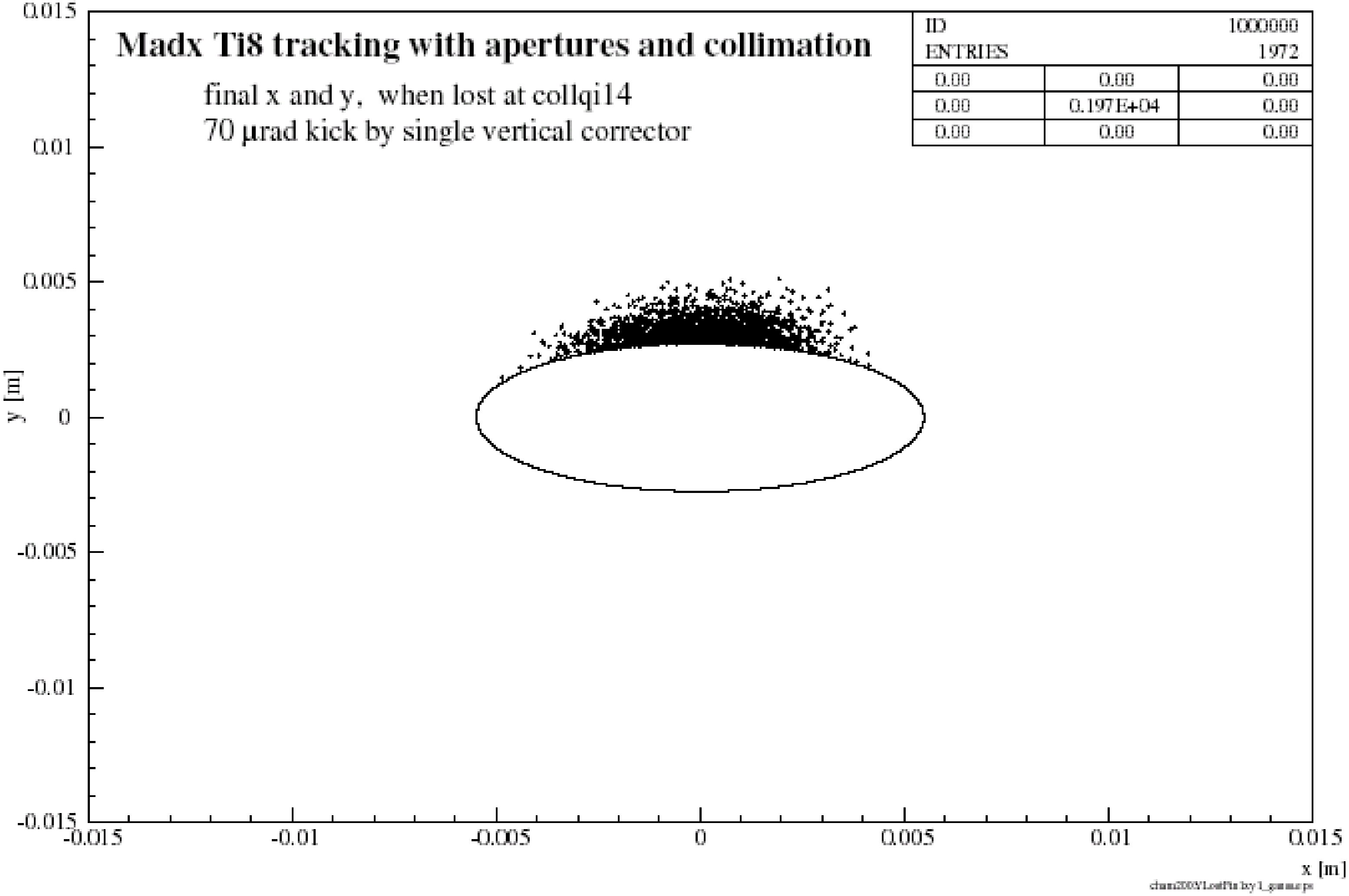



TI 2 : injection into LHC ring 1 (clockwise), 3190 m long, 48 ° horizontal deflection, 61 and 35 mrad vertical deflection to avoid underground water flow.

TI 8 : injection into LHC ring 2 (anti - clockwise), 2700 m long, 103° horizontal deflection and 38 and 35 mrad vertical bend at beginning and end to adapt to 70 m SPS / LHC height difference

# H.Burkhardt. MAD-X for Transfer Lines

## MAD-X Ti8 tracking with aperture and collimation




N. Malitsky. Report from MAD-X day, September 12, 2003

# H.Burkhardt. MAD-X for Transfer Lines
## Linking TI 8 and LHC MAD-X optics

- define a common link point (names and even some elements incl. MKI different, inj. beam off-centre in Q5 to get kick ..., for the moment link at Q5 in front of MKI
- modify TI8 to stop at that point "ti8line2.seq"
- LHC ring2 optics at injection:

```
install,element=Ti8Ref,class=marker,at= l.mqy/2,from=MQY.A5R8.B2; ! set marker before 2nd Q5 (as seen from Ti8)
flatten; ! include all sequences in the sequence being edited
cycle,start=LHCB2$END;   ! start at end for reflect
reflect; ! go backwards, as coming from Ti8
  save    and check (plot)
cycle,start=Ti8Ref;
   save as   lhcb2_backw_fromInj.seq
```

- edit ti8line2.seq   flip signs of quadrupoles   to get   bv=-1 line as lhc ring2
- Link the two as "ti8Andlhcb2.seq"

```
real const  RefPos = 2673.18448; // pos at beginning of second Q5; number from end of ti8line2.seq
edit   lhcb2_backw_fromInj.seq   by hand:     "at = "  -> "at = RefPos + "
ti8Andlhcb2: sequence, l = 26658.8832 + RefPos;
ti8line2,at=0; //hbu
followed by  lhcb2_backw_fromInj.seq
```

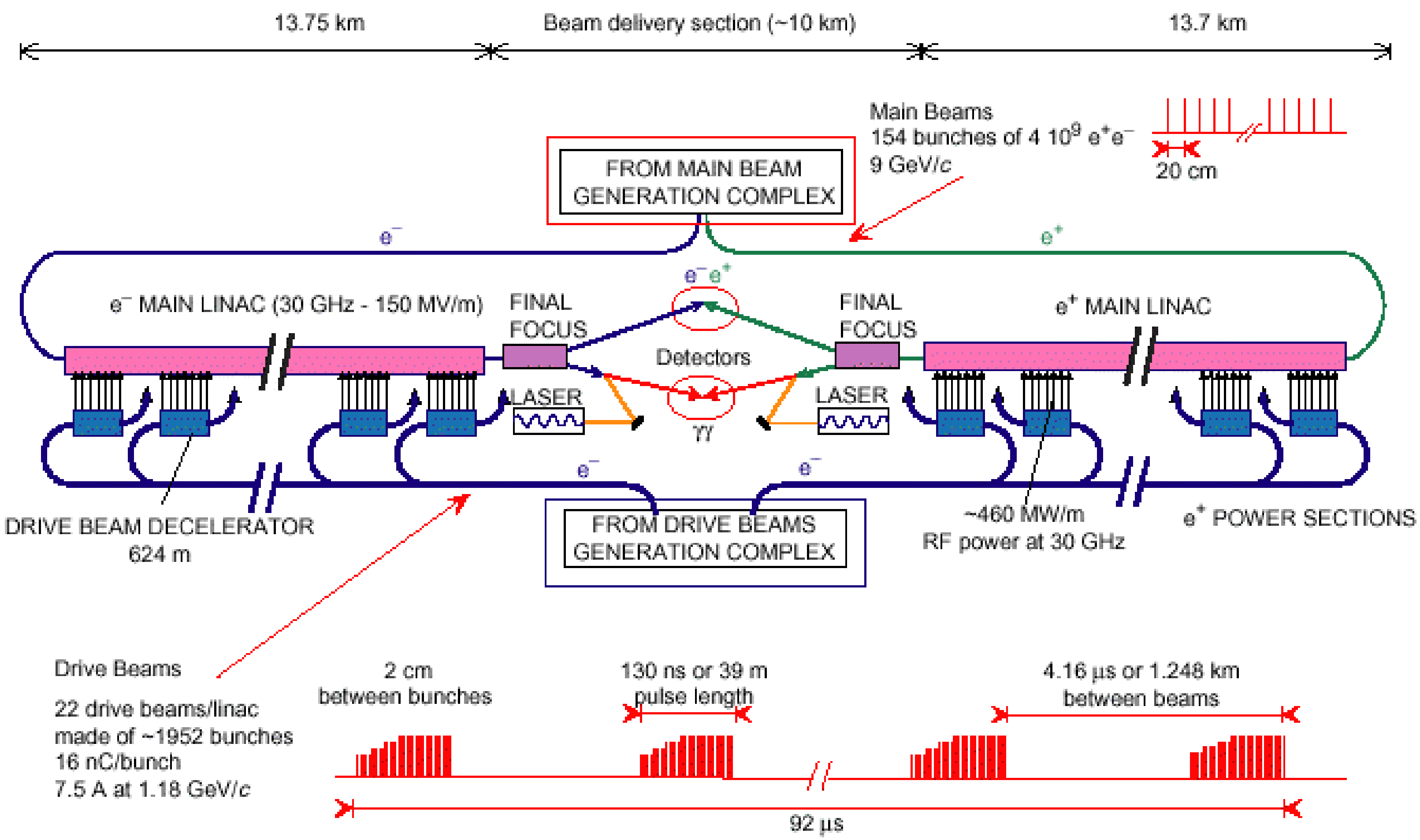
# H.Burkhardt. MAD-X for Transfer Lines
## General Comments

- I see MAD-X as public open software project like Geant4, just on a smaller scale with all source code openly available, running with standard gnu compilers
- There is no problem to include C++ and the MAD-X structures and memory management could profit a lot from the C++ standard library data structure and algorithms (vector, list, …)
- Use of Fortran: was probably necessary to have a quick start from MAD 8 and to not exclude people which still did not learn C/C++ from contributing to MAD 8 should however be avoided as mush as possible for further developments

H. Braun, R. Corsini, T.d'Amico, A. Faus-Golfe, M. Korostelev, S. Redaelli, T. Risselada, D. Schulte, F. Tecker, and F. Zimmermann for the CLIC Team

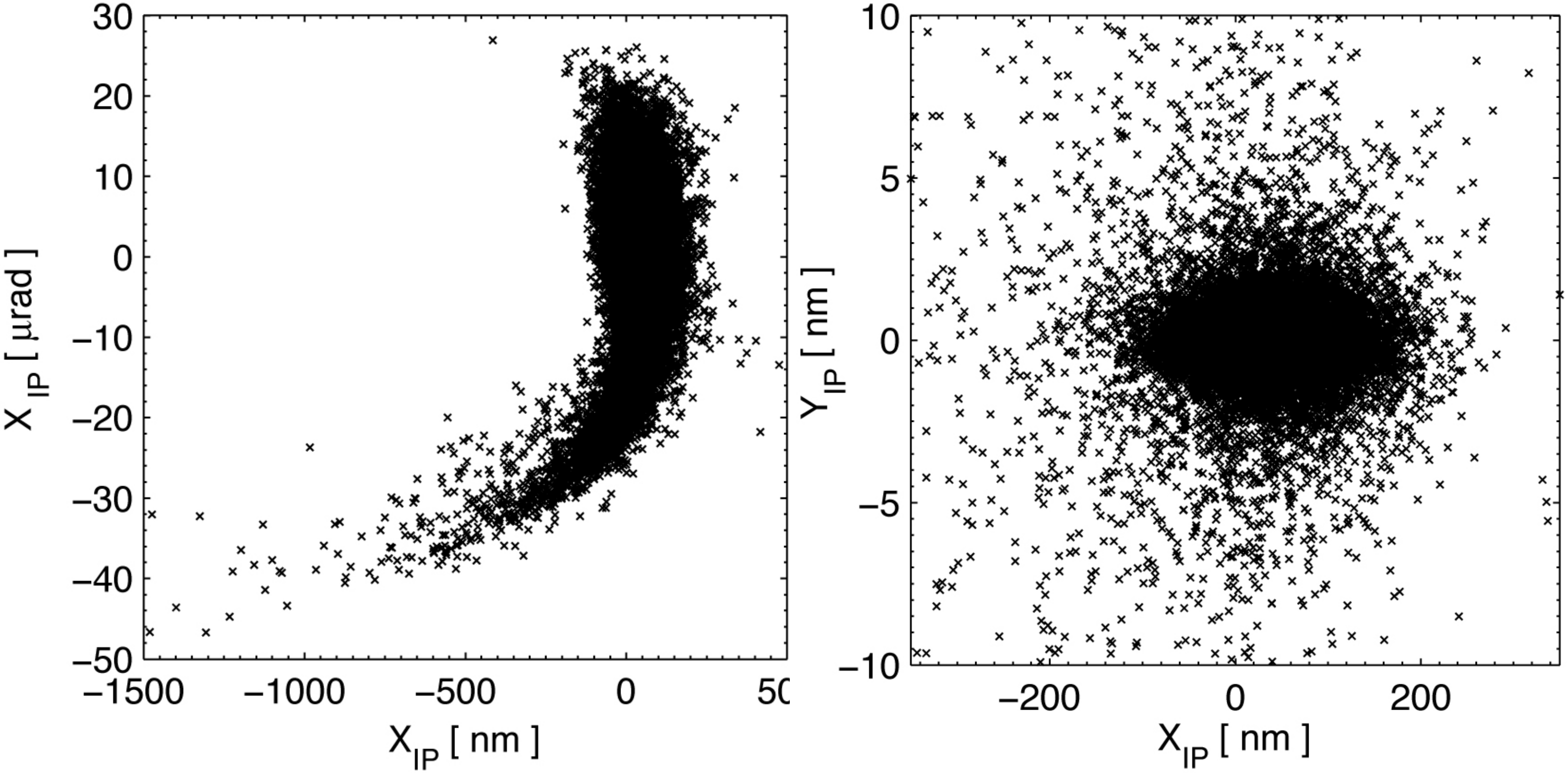
- **applications of MAD8 to CLIC & CTF3**
- **minimum requirements for MAD-X**
- **desirable extensions**

# F.Zimmermann. MAD-X for CLIC
## 3 TeV CLIC



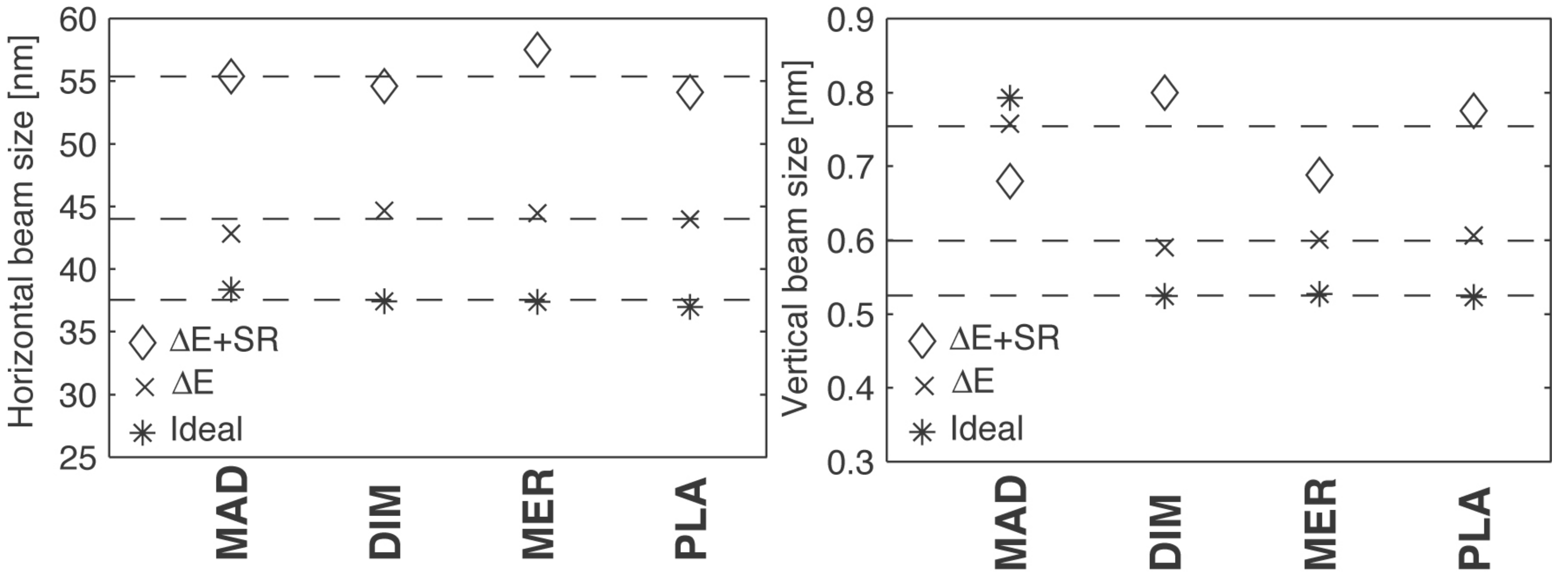
# F.Zimmermann. MAD-X for CLIC

## CLIC IP Beam (S. Redaelli)




**Tracking** through CLIC Beam Delivery at 3 TeV.
MAD8 simulations include finite **energy spread & synchrotron radiation**.

# F.Zimmermann. MAD-X for CLIC
## CERN-DESY-SLAC Code Comparison (S. Redaelli)




Without SR, **MAD8 Y spot size** is much larger than in MERLIN, PLACET and DIMAD. May **PTC** help!?

# F.Zimmermann. MAD-X for CLIC
## MAD is important for CLIC

**In the past, the design and on-line models of all components for CTF-3 and for the 3-TeV CLIC design have relied heavily on MAD8.**

**We would like to have at least the MAD8 functionality maintained in MAD-X.**

**This is not yet completely the case. Certain extensions would be welcome and are encouraged!**

# F.Zimmermann. MAD-X for CLIC
## Top Ten MAD modules

**TWISS**

**MATCHING**

**SURVEY**

**TRACK**

**EMIT**

**DYNAP**

**IBS**

**EFIELD**

**EALIGN**

**CORRECT**

## F.Zimmermann. MAD-X for CLIC
## More Urgent MAD-X Problems

- **RMAT & TMAT** not working?
  **[needed for any transfer line matching]**
  **re-establish late MAD8 functionality!**
- **GLOBAL** not working.
  **[needed for any transfer line matching]**
  **re-establish late MAD8 functionality!**
- **L-Cavity does not work with Twiss**
  **[needed for optics matching of all linacs including CTF-3 linac]**
  **re-establish late MAD8 functionality!**

# F.Zimmermann. MAD-X for CLIC
## MADX' Positive Features

- IBS is working fine
- Closed orbit correction considerably improved
- PTC will hopefully facilitate tracking and might aid in nonlinear optimization for DR and BDS

## CLIC Special Requirements

- Many problems require **specific tracking features,** e.g.:
  - coherent synchrotron radiation, collimator wake fields, linac wakes, gas scattering, or collimator scattering.
- Another important group of beam-dynamics studies need **integrated simulations** including various beam-line segments, e.g.:
  - simulations from the Damping Ring to the IP
- Suggested extension: **Data-exchange interface to allow fast data piping to and from user programs,** e.g.:
  - enabling user-defined coordinate transformations by external program for selected element classes.

# F.Zimmermann. MAD-X for CLIC
## Conclusions

- MAD8 is a valuable tool for CLIC studies.
- We expect MAD-X to follow this tradition and not to lose the contact to the lepton machines. We hope that a few minor requirements (**LCAV, dipole end faces, RMAT, GLOBAL)** can be resolved quickly.
- **A communication interface for tracking will greatly enhance the reach of MAD-X simulations for CLIC.**

## C. Milardi. MAD-X for Small Machines

## Modeling requirements for some Lepton Low Energy Accelerators

- DAΦNE
- DAΦNE II
- CTF3

In terms of:

- magnetic element description
- linear & non-linear optics
- longitudinal dynamics

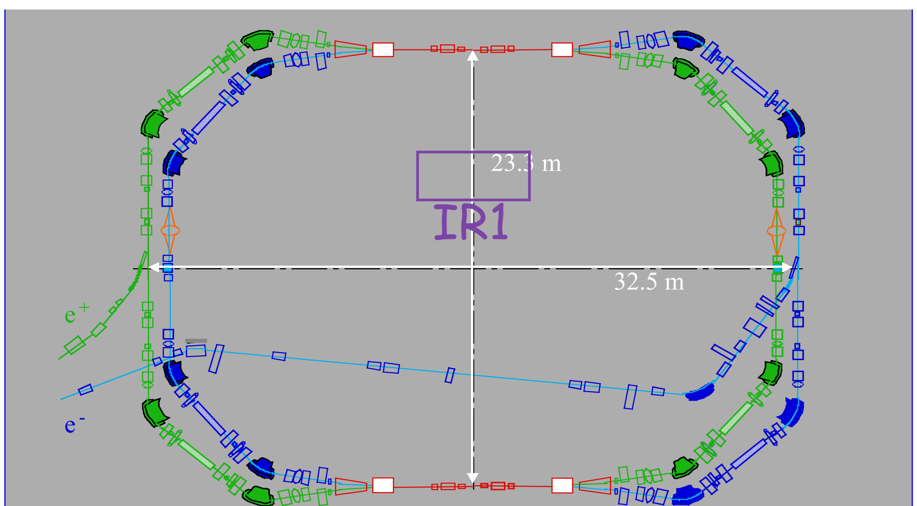
Keeping MAD 8 as a reference

# 2 Rings sharing 2 Interaction Regions

## DAΦNE the Frascati $e^+ e^-$ collider

## Each Ring

- no periodicity
- 4 different kind of short dipoles
- 4 wigglers ($l_w$ = 2 m  $B_w$ = 1.9 T)
- strong focusing by short QUADs
- huge solenoidal field $B_{IR1}$= 2.4 T m  $B_{IR2}$ = 2.2 T m




## Parameters:

- E = .51 GeV
- C = 97.68 m
- $f_{RF}$ = 368.263 MHz
- $B\rho$ = 1.7  Tm
- Crossing angle @ IP

# C. Milardi. MAD-X for Small Machines

**MAD 8 has been successfully used for:**

- DAFNE design
- optimizing first and II order optics
- tuning Betatron function at the IPs, even during data-tacking
- tuning $\alpha_c$
- Dynamic aperture calculation & optimization

**The DAFNE model has been adapted for MAD-X**

**Real improvements in the modeling capabilities are expected from PTC**

## Improving Wiggler Model

If Wigglers are represented as dipole arrays to step further the MAD 8 capabilities it's necessary to introduce, at least, dipole splitting to deal with:

- multipole insertion
- matching @ the element centre

then two *FINT* values must be specified

Better analytic wiggler field model & symplectic integration for long term tracking to:

- optimize sextupole configuration
- enlarge dynamic aperture
- increase the Touschek lifetime

# C. Milardi. MAD-X for Small Machines

## About the 'Match' module

- Make the **if else structure** working within a matching procedure (MAD-X)
- Allow user defined function minimization

## Longitudinal Dynamics

Include the BMPM module to compute:

- Touschek lifetime
- bunch length
- ..

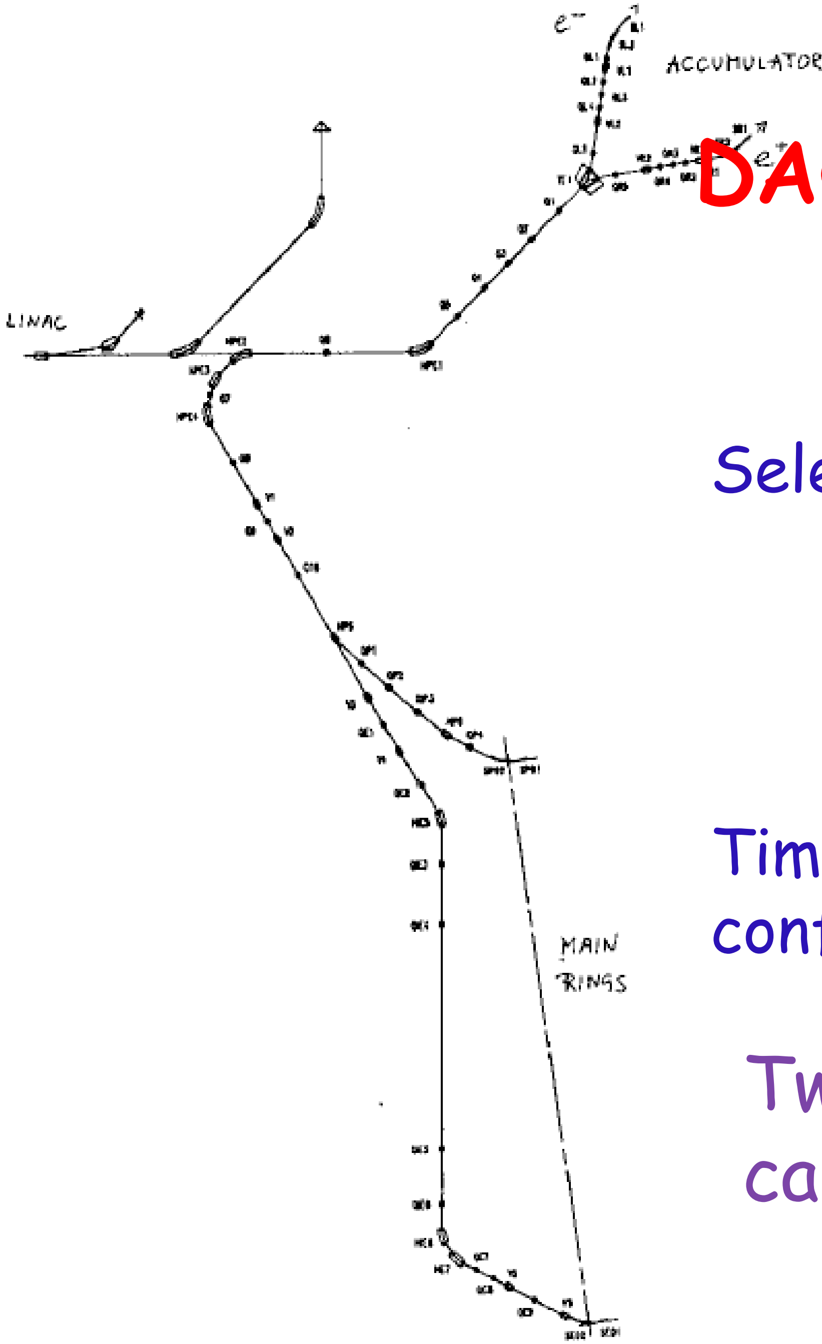
Include wake field

- to deal with anomalous bunch lengthening

For RF cavity add the capability to

- define $f_{RF}$ frequency (MAD-X)
- deal with more than one cavity

# DAΦNE Transfer Line




Selected TL area transports:

- $e^+$ LINAC -> Acc
- Acc -> $MR^+$
- $e^-$ LINAC -> Acc
- Acc -> $MR^-$

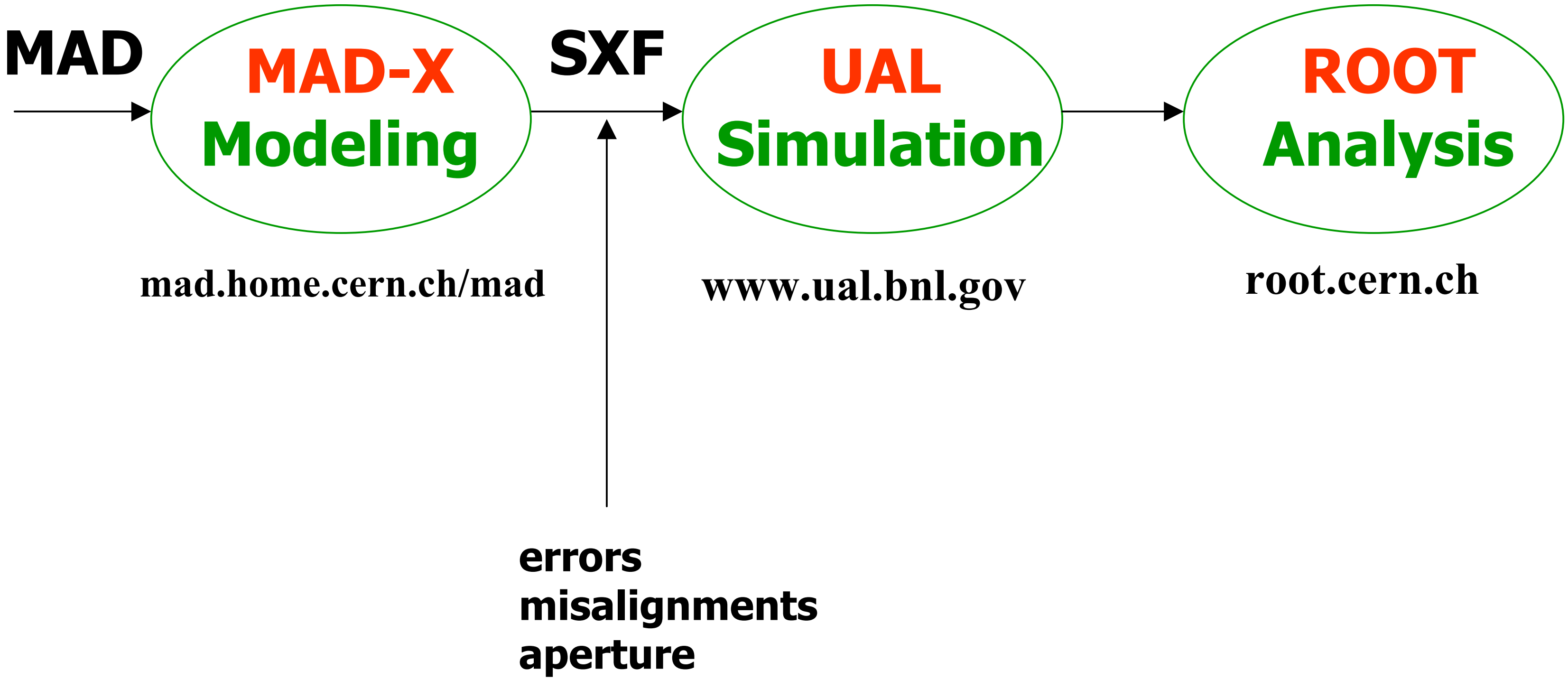
Time consuming switch of QUADs configuration affects $\tau_{inj}$ -> $L_{int}$

Two lines matching capability (MAD-X)

# N.Malitsky MAD-X & UAL linked by SXF
## Standard eXchange Format (SXF)

| | |
|---|---|
| **Description** | The SXF represents a flat, complete, and independent description of the current accelerator state. At this time, it includes only the full-instantiated view of the accelerator, however it provides the mechanism for its integration with the present site-specific design hierarchical models. This approach stems from the desire to make this format neutral to different conceptual models and adaptable to arbitrary data stores. The SXF can be considered as the additional independent layer to existing design data structures, particular to the Standard Input Format (SIF) *design* components, *element* and *beam line*. The relationship between SIF and SXF structures is provided by the references embedded in the SXF objects. The SXF format is based on the integration of the MAD *sequence* and UAL/SMF *element buckets*, orthogonal collection of element attributes. It preserves all SIF element types and provides the mechanism for introducing new ones, such as CESR superimposed solenoid and quadrupole elements, LHC and CESR parasitic beam-beam effects, Muon Collider Ionization Cooling, and others. |
| **Authors** | H.Grote, J. Holt, N. Malitsky, F. Pilat, R. Talman, C.G. Trahern, W.Fischer |
| **References** | • H. Grote, J. Holt, N. Malitsky, F. Pilat, R. Talman, C.G. Trahern. "SXF (Standard eXchange Format): definition, syntax, examples", RHIC/AP/155, August 1998.<br>• W. Fischer. "An SXF Extension for Alignment", RHIC/AP/165, November 1998. |
| **Parser Developer** | Nikolay Malitsky |
| **Specification** | • C++ API<br>• sxf.ll - FLEX lexical patterns.<br>• sxf.yy - YACC grammar .<br>• element.gperf - GPERF keyfile for the NodeRegistry. |

# N.Malitsky MAD-X & UAL linked by SXF
# MADX-UAL-ROOT Integrated Approach



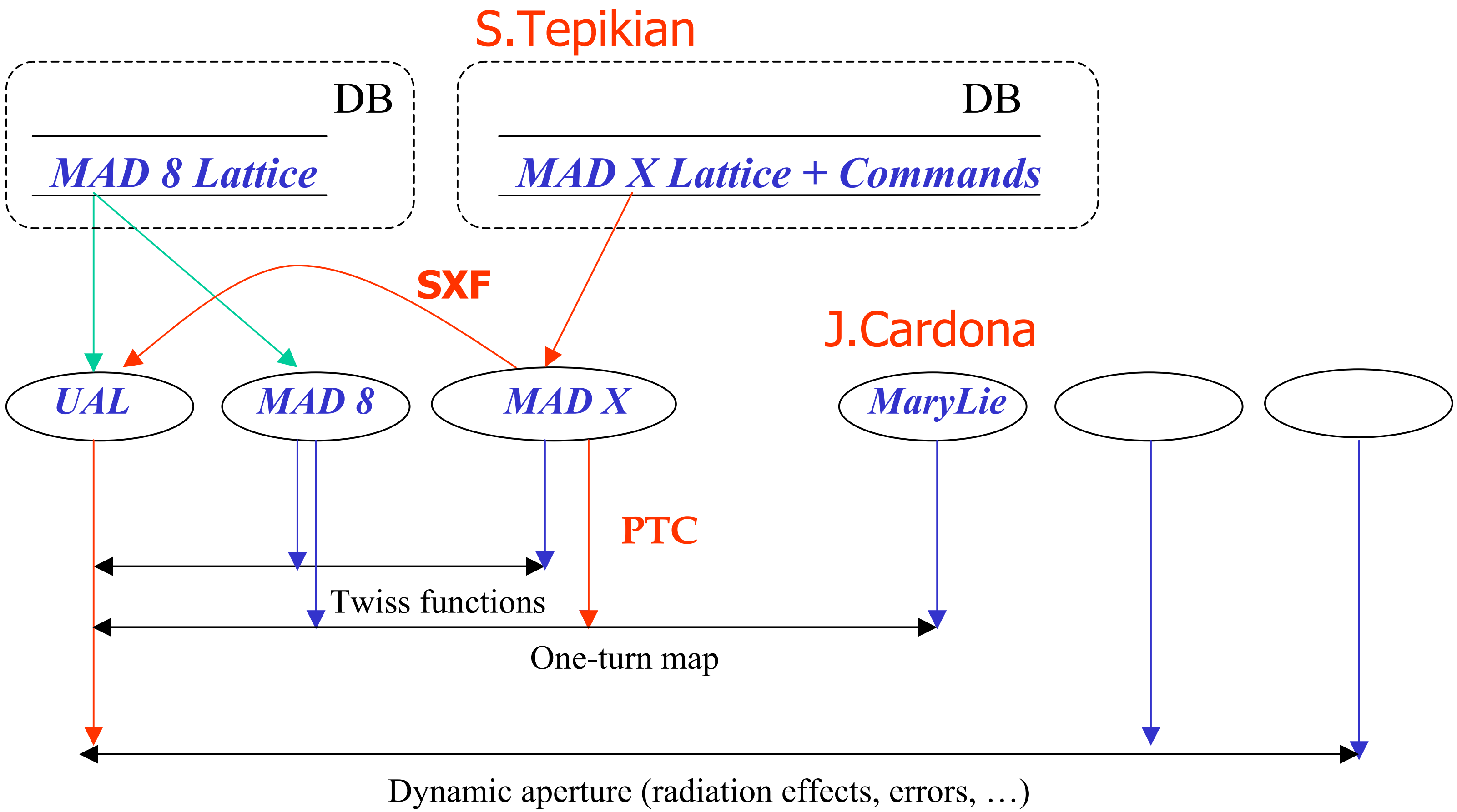
# N.Malitsky MAD-X & UAL linked by SXF
## MADX-UAL-ROOT Integrated Approach (continued)

- **MADX vs MAD8**
  - eRHIC project: IR design, electron ring (S. Tepikian)
- **SXF**
  - coordination (F.Pilat)
  - maintenance and upgrade of RHIC filters (V.Ptitsyn)
  - parser upgrade with aperture attributes (R.Fliller III)
  - porting to the Windows platform (Y.Malitsky)
- **UAL**
  - APDF-based Framework (N.Malitsky, T.Satogata, R.Talman)
  - SIMBAD (A.Luccio, N.D'Imperio)
  - Accelerator Instrumentation Model (P.Cameron, M.Blaskiewicz, R.Calaga, N.Malitsky)

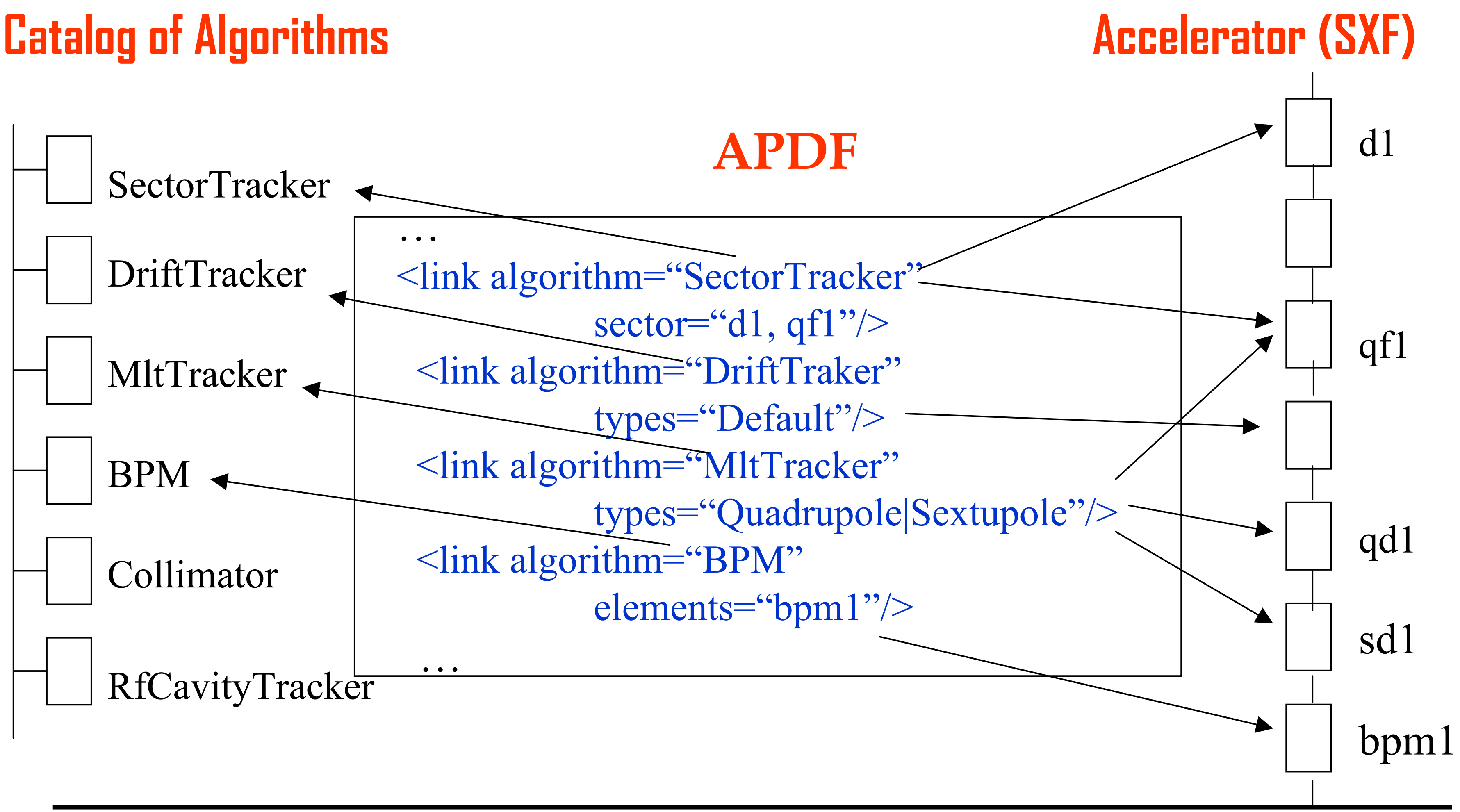
# N.Malitsky MAD-X & UAL linked by SXF
## MADX-UAL-ROOT Integrated Approach (continued)

- **ROOT**
  - UAL/ROOT interface (R.Fliller III)
  - CORAANT: COmprehansive Root-based Accelerator ANalysis Toolkit (R.Fliller III)

# N.Malitsky MAD-X & UAL linked by SXF
## eRHIC Electron Ring Application

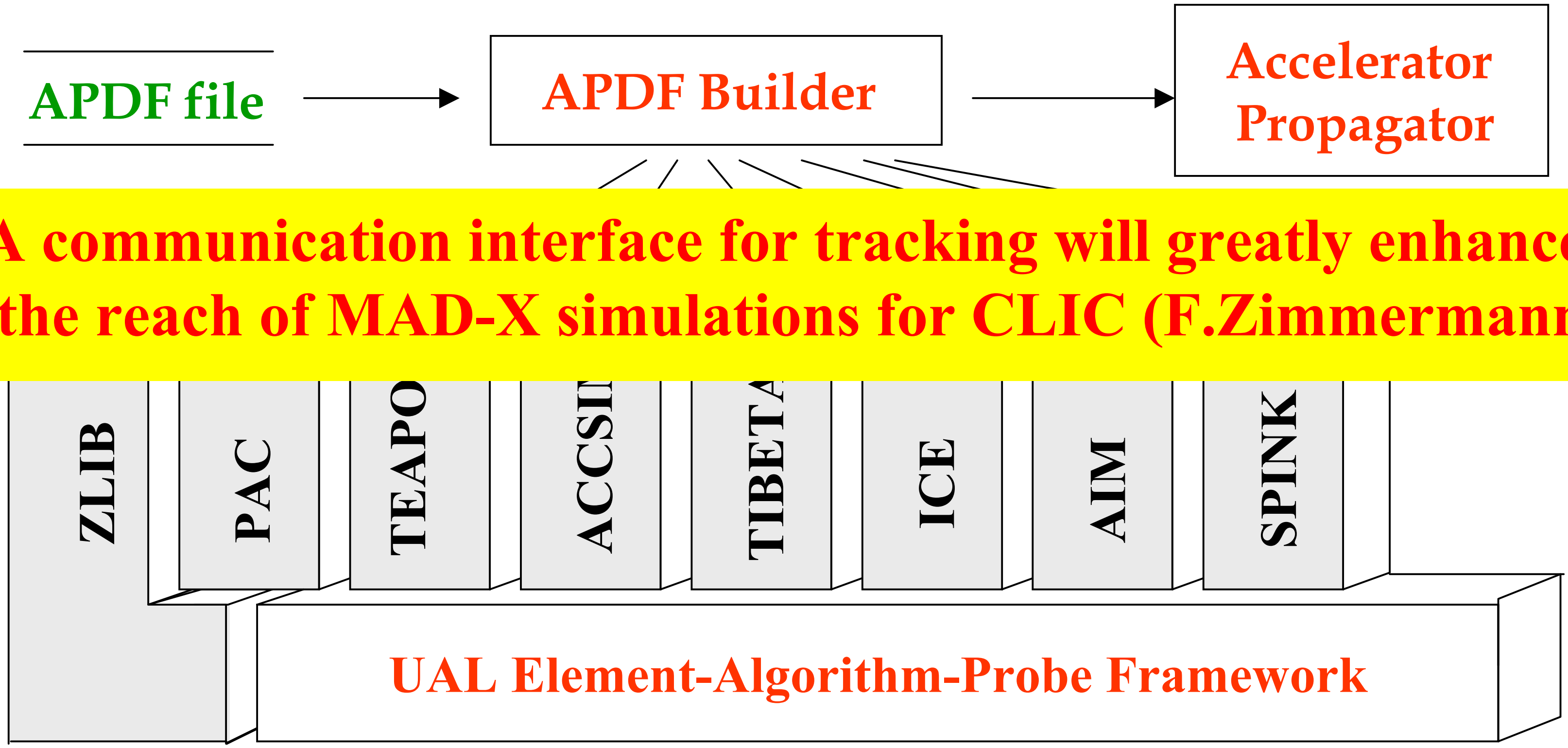


# N.Malitsky MAD-X & UAL linked by SXF
## Next Steps: Accelerator Propagator Description Format



# N.Malitsky MAD-X & UAL linked by SXF

## Next Steps: APDF (Continued)



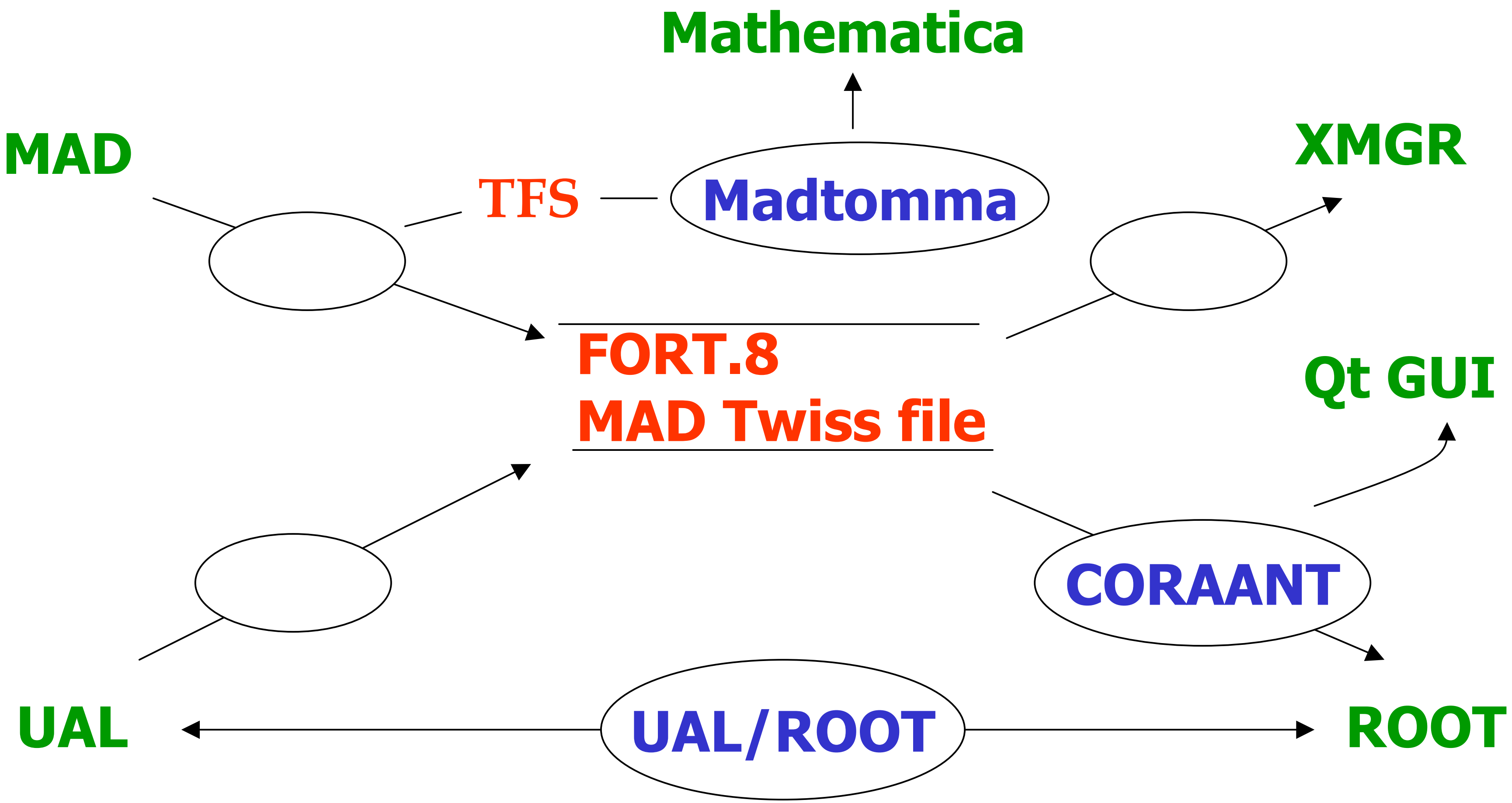

**A communication interface for tracking will greatly enhance the reach of MAD-X simulations for CLIC (F.Zimmermann)**

# N.Malitsky MAD-X & UAL linked by SXF

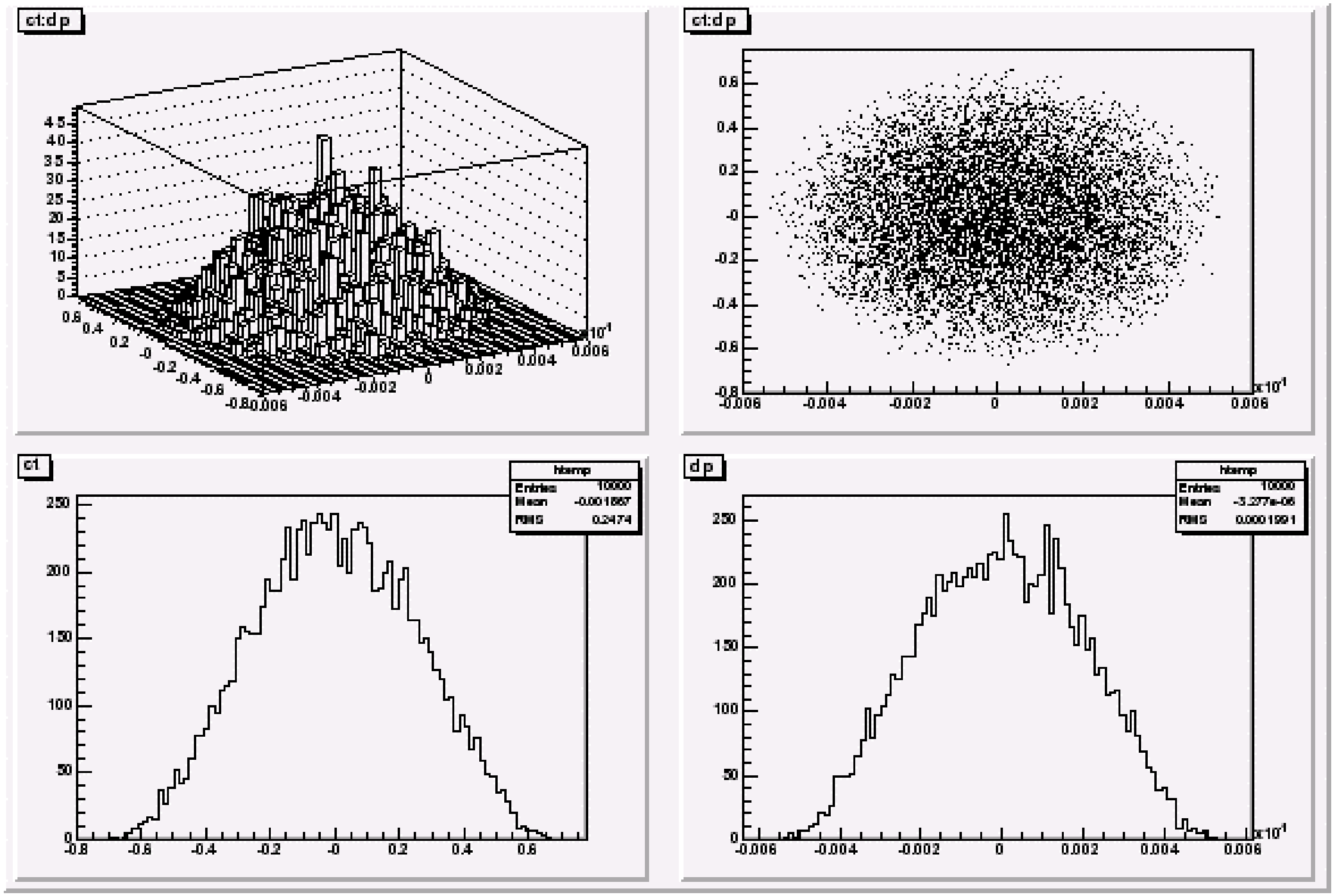
## APDF-based applications

- Longitudinal Dynamics: sector 2D matrices + RF +
- Dynamic Aperture: element-by-element tracker with type-based associations
- Online model: sector maps + trackers of selected elements (e.g. Fast Teapot)
- Instrumentation modeling (e.g. MIA, BTF) : conventional tracker + diagnostics devices
- Space charge studies: conventional tracker + space charge insertions (e.g. SIMBAD)
- Spin studies: Spin tracker as a wrapper of conventional tracker (e.g. SPINK )
- eRHIC: conventional tracker with synchrotron radiation effects

# N.Malitsky MAD-X & UAL linked by SXF
## Next Steps: Post processing environment



# N.Malitsky MAD-X & UAL linked by SXF
## ROOT graphics (R.Fliller III)



# IV Part of the MAD-X Day
# Open Discussion Session: Wish List

- Pool Dump 4-6 weeks (low)
- Flushing buffers FS
- Kill Memory Leaks FS
- Delta/p ➔ PTC
- Matching with PTC FS, (ITEP)
- RMAT, TMAT Matching OB
- User defined Penalty Functions OB
- Matching with Strength Limits OB&FS (low)
- Synchrotron Radiation Alexander Bolshakov (ITEP)
- LCAV ➔ PTC FS
- Space Charge, Wake fields ➔ GSI module keeper
- Translation of coor without the change of change of reference ➔ PTC FS
- Fint different at both ends CM
- Nested Elements of different lengths and types EF&FS
- Communication with other Modules FS
- Time dependent effects VK
- Interactive Plotting on Windows HG&JJ
- DYNAMIC/STATIC FS&ITEP
- Ptc_Twiss FS
- Equilibrium emittance improve (EMIT) FZ (low)
- Toucheck FZ&CM

# BNL CAD initiatives for adapting and contributing to the MAD-X software

- **Maintenance and applications of the MADX-SXF package** (F. Pilat, N.Malitsky, V.Ptitsyn, S.Tepikian)
  - eRHIC, RHIC, …
- **PTC vs UAL (**D. Trbojevic, E. Forest, N.Malitsky, …**)**
  - "On a more serious topic, actually an extremely serious topic, it involves … flat file …A beam line is NOT a collection of magnets"….