Beam Dynamics Simulation Codes: Required Interfaces, Benchmarks and Extensions

Nick Walker

CARE (ELAN) Workshop

DESY 2.11.2004

Beam Dynamics Simulation Codes: Required Interfaces, Benchmarks and Extensions

Nick Walker

CARE (ELAN) Workshop

DESY 2.11.2004

Interfaces: A statement of the problem

- Many codes in use around the world by different people
  - MAD
    - And its many variants
  - DIMAD
  - BMAD
  - LIAR
  - PLACET
  - MERLIN
  - ELEGANT
  - LUCRETIA (coming soon ☺)
  - GUINEA-PIG
  - More I’ve not thought of…

Interfaces: A statement of the problem

- Many codes in use around the world by different people
  - MAD
    - And its many variants
  - DIMAD
  - BMAD
  - LIAR
  - PLACET
  - MERLIN
  - ELEGANT
  - LUCRETIA (coming soon ☺)
  - GUINEA-PIG
  - More I've not thought of…

Each have their particular strengths and weaknesses

Some specific to particular problems (e.g. linac dynamics) , others are more general.

Some are good for design work (MAD), others are pure simulation tools (most of the list!)

Interfaces: A statement of the problem

- Many codes in use around the world by different people
  - MAD
    - And its many variants
  - DIMAD
  - BMAD
  - LIAR
  - PLACET
  - MERLIN
  - BDSIM
  - ELEGANT
  - LUCRETIA (coming soon ☺)
  - GUINEA-PIG
  - Many more…

We do not want to insist people use a ‘single’ code (quite the opposite)

But transferring lattice models and other relevant information between these codes is a non-trivial and time consuming problem.

We need some ‘common exchange format’

An Example

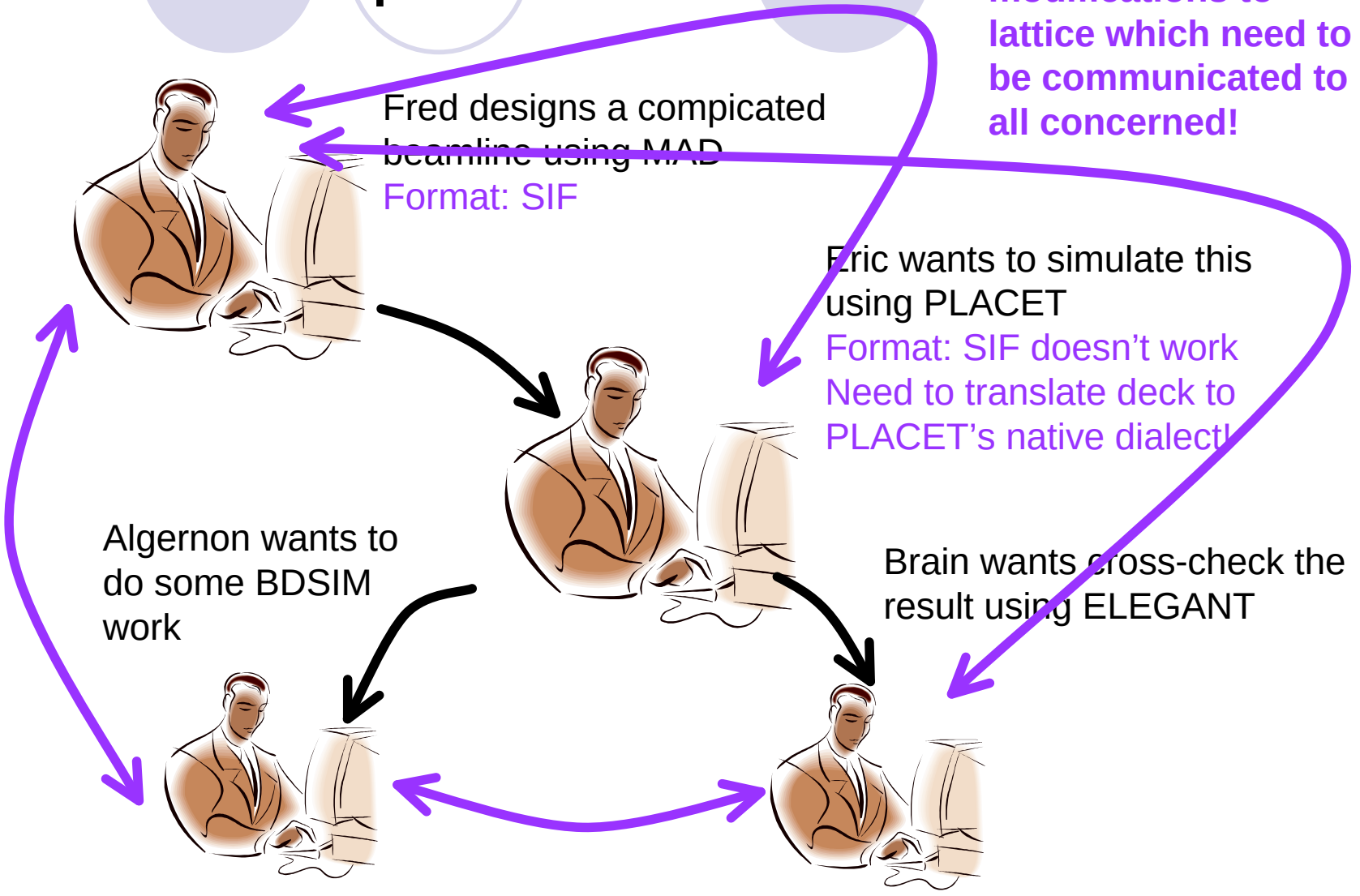
Fred designs a complicated beamline using MAD
Format: SIF

Eric wants to simulate this using PLACET
Format: SIF doesn't work
Need to translate deck to PLACET's native dialect!

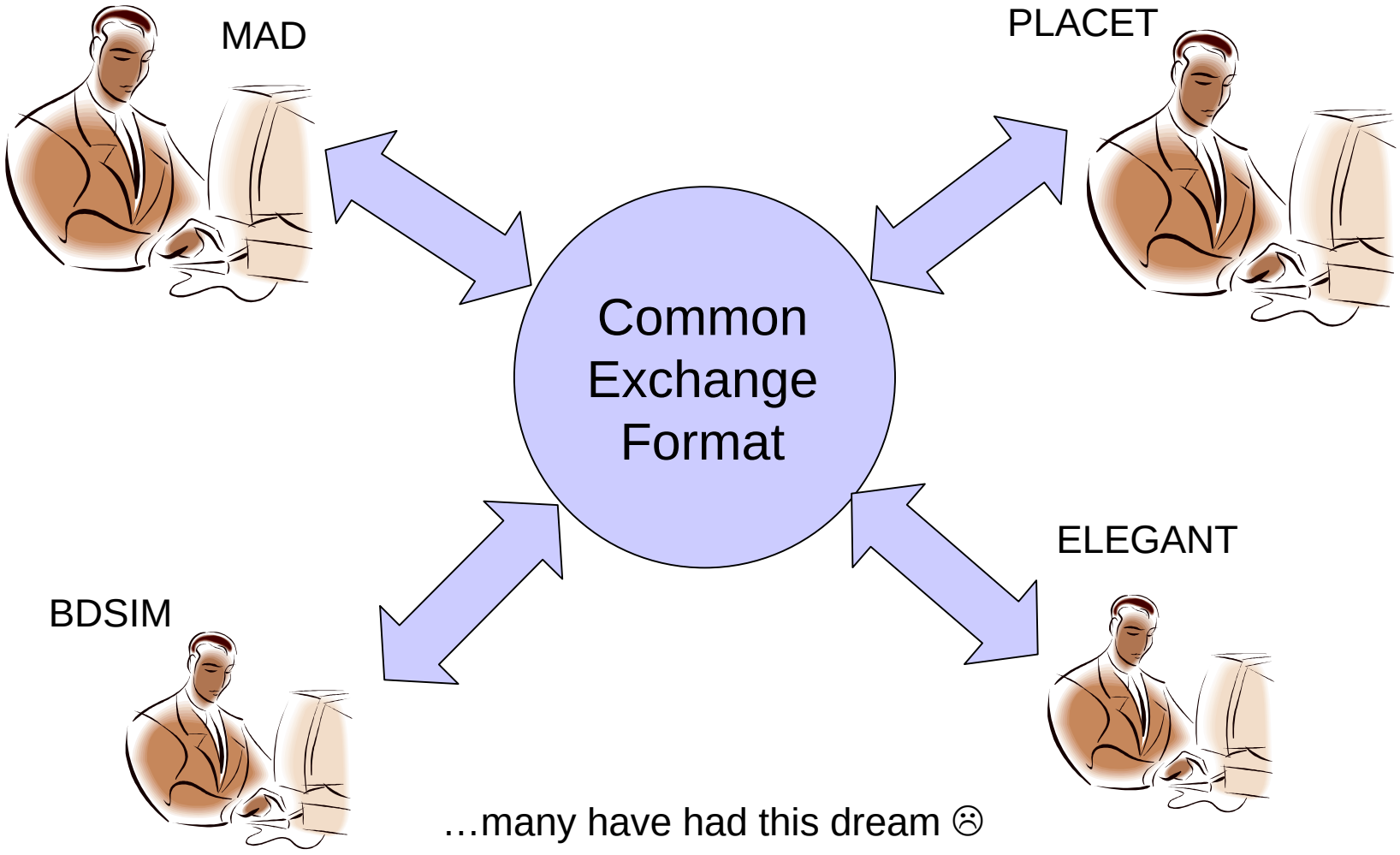
Algernon wants to do some BDSIM work

Brian wants cross-check the result using ELEGANT

An Example

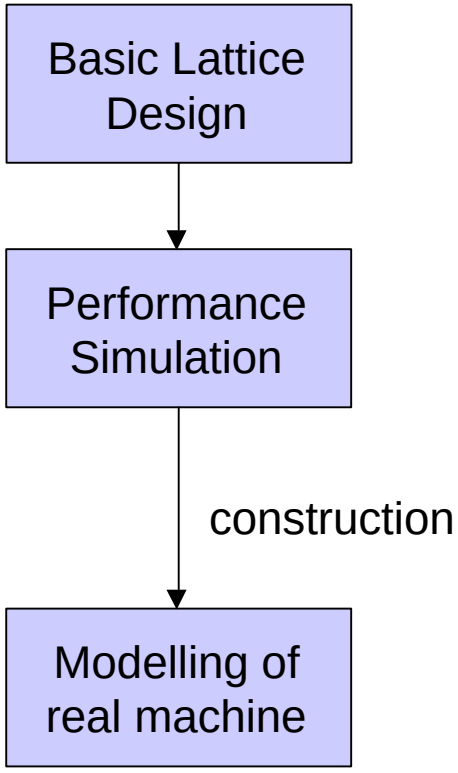

Everybody makes modifications to lattice which need to be communicated to all concerned!
Fred designs a compicated beamline using MAD
Format: SIF
Eric wants to simulate this using PLACET
Format: SIF doesn't work
Need to translate deck to PLACET's native dialect!
Algernon wants to do some BDSIM work
Brain wants cross-check the result using ELEGANT

A Better Way

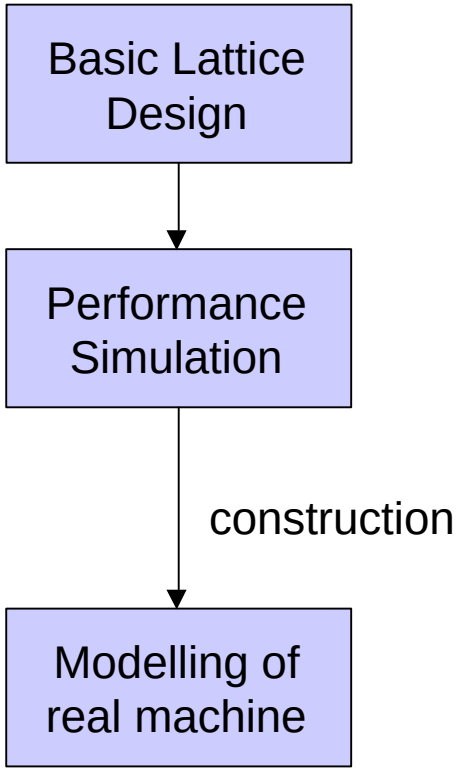


…many have had this dream ☹

(Typical) Life Cycle of an Accelerator Project



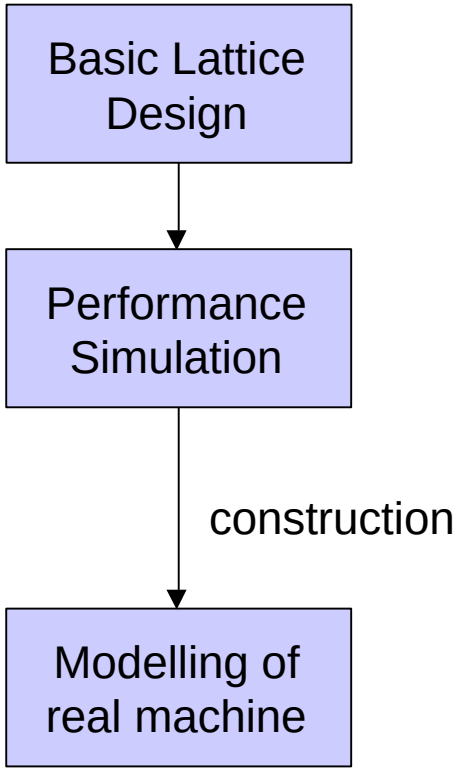
(Typical) Life Cycle of an Accelerator Project



- Simple models (sequences of elements)
- Tend to work with smaller modules
- Fitting, constraints etc. Lattice matching
- ‘Generic’ magnet families
- Definition of basic parameters
  - (may have more than one possible optics)

Quadrupole
L
K1 *value or range of* K1 *values*

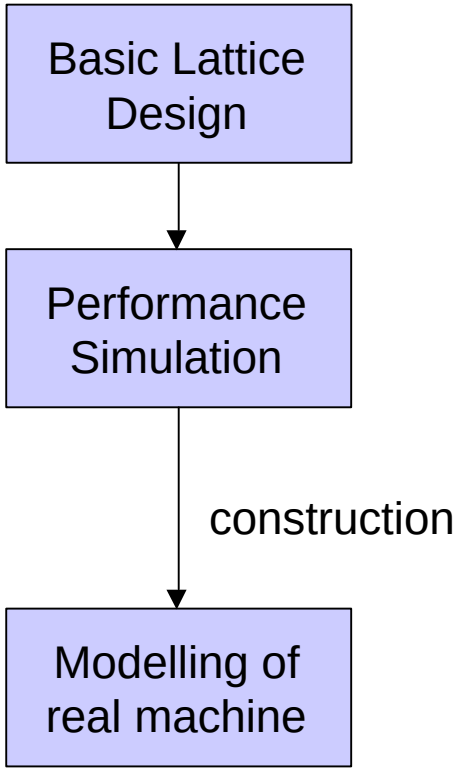
(Typical) Life Cycle of an Accelerator Project



- More complex (complete) models
- Tolerance studies
  - Simulation of a range of “errors”
- Refinement of parameter specifications
  - Definition of prototype component
- Tuning algorithms, diagnostics specs.
- Power supplies (circuits), Klystrons etc.

Quadrupole
L
Pole tip radius
max / min pole-tip field
Tolerances (used to generate random errors)

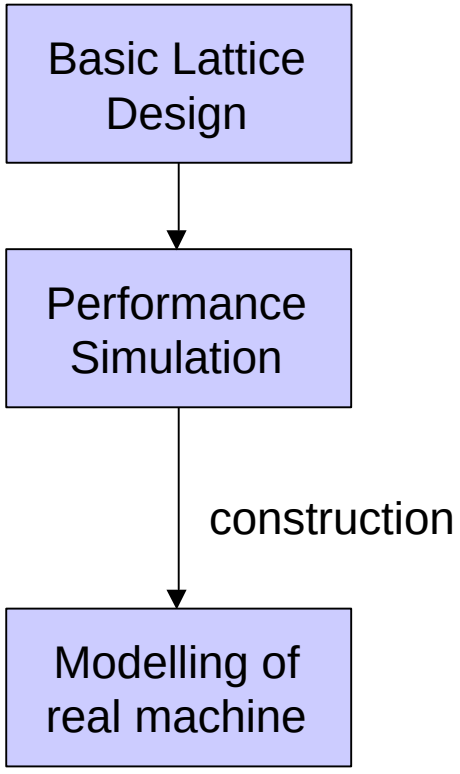
(Typical) Life Cycle of an Accelerator Project



- Engineering design of accelerator components

Quadrupole prototype (family)
L
Pole tip radius
max / min pole-tip field
Tolerances
Documentation (drawings, cad files etc)

(Typical) Life Cycle of an Accelerator Project



- Complete model of *real* machine
- Used for
  - Online modelling
  - Continued performance studies
  - Tuning studies etc.

Quadrupole QF10 (now unique)
L
Pole tip radius
max / min pole-tip field
Measured field map
Other unique physical attributes

My proposal

- To define and develop an ‘accelerator description’ that supports the entire process
- Should be flexible so that
  - Simple syntax can be used at the initial phases
  - Complex (rich) enough to correctly model the real machine
    - Including vacuum system, power supplies, klystrons, diagnostics
- Close the link between acc. phys. and engineering.

Wish list

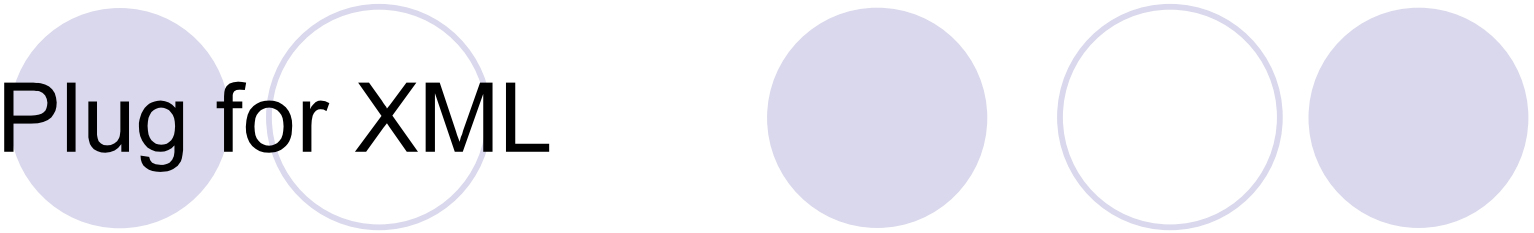
- Basic accelerator components
  - Different degrees of complexity reflecting the different phases of the project
- Complex geometries
  - IR geometries often difficult
  - Girders and magnet mover systems
  - Beamline junctions
- Power supplies, klystrons
  - Circuits
- Vacuum system
  - Beampipe apertures / cross sections
  - Pump locations / speeds

Wish List

- Should also support
  - Errors (for exchange between two codes/people)
  - Optics configurations
    - Including associated lattice parameters
  - Can also define some standard formats for
    - Particle information or other simulation output.

Note: Need to clearly define ‘boundaries’ of modelling language (don’t want to include unnecessary mechanical design details not relevant to our goals)

A Plug for XML



- Industry standard
  - Type XML into Google and stand well back!
- Many parsers available in public domain
- Easily readable by man and machine
- Certainly has the structure to support all my wish list
- Define an xml schema which *enforces* conformance (reduce errors)
- Put your model directly on the web ☺

Some Results of discussions held at MERLIN Dev. Meeting at LLBL in October

- David Sagan (Cornell) has tabled a suggestion for a ‘universal’ parser
  - http://www.lns.cornell.edu/~dcs/universal_parser
  - Mailing list set up:

    Peter Tennenbaum
    Nick Walker
    Andy Wolski
    Mark Woodley
    Yunhai Cai
    Frank Schmidt
    Yiton Yan
    Dick Talman
    Nikolay Malitsky
    David Sagan

  - All interested welcome to join!
  - More than just a Standard Exchange Format
    - A single translator which will convert from one format to the other without loss of formatting
    - ambitious

First pass a requirements for UXL

- Support all ‘Standard Components’
  - Fields can be specified as K values, B fields or a range of allowed B field $<B_{min}, B_{max}>$
  - Geometry separate from ‘magnetic fields’
  - Generalised coordinate transformations (‘patch’)
  - Wakefield should be included
- Concept of ‘prototype’ definitions
  - Other components can ‘inherit’ parameters from

First pass a requirements for UXL

- Definition of Beamlines
  - Standard line-like definitions (MAD: Beamline)
  - Ability to arbitrarily place component as specific location in lattice (MAD: Sequence)
  - Important: Accelerator Geometry is always defined by the elements!

First pass a requirements for UXL

- Additional Stuff
  - Definition of girders, supports, magnet movers
  - Power supplies, klystrons etc. with ‘circuits’
  - ‘Multi-knob’ definitions (parametrically driven component settings)
  - Multi-pass definition (Complex geometries such as re-circulating linacs)
  - Much more…

See
http://awolski.lbl.gov/MerlinMeeting2004/Agenda_files/ParserRequirements.pdf

Summary

- Everyone agrees on the need of a common exchange format
  - People have been talking about this for years
- Everybody has different ideas how it should look like
  - How to avoid a ‘standard’ built by committee
- Manpower a problem: who will do the real work
  - There is some hope here with our ‘proto-collaboration’