A Novel Online (Accelerator) Simulator for High-Level Applications Requiring a Model Reference

Christopher Allen
Los Alamos National Laboratory

ICALEPCS 2003, Kyongju, Korea

XAL Collaborators

- ORNL
  - Tom Pelaia
  - Paul Chu
  - Wolf-Dieter Klotz
  - Andre Shislo
  - John Galambos
- BNL
  - Dan Attavio
- LANL
  - Craig McChesney
  - Bob Dalesio
  - Chris Allen
- Cosylab
  - Igor Kriznar
  - Igor Verstovsek
  - Ales Pucelj
  - Mark Plesko

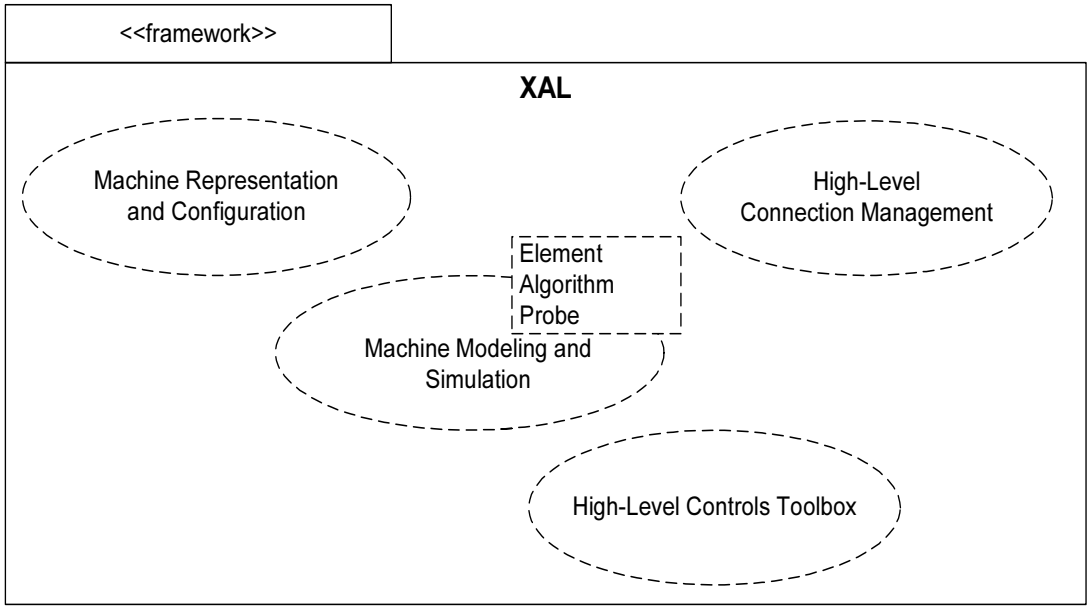
Outline

1. XAL Overview
2. Simulator Overview
   - Status
   - Features
   - Architecture
3. Verification and Validation
4. Summary

1. XAL Overview

RAD programming environment for high-level control applications

- Machine representation
- Machine connection
- Machine modeling
- High-level tools

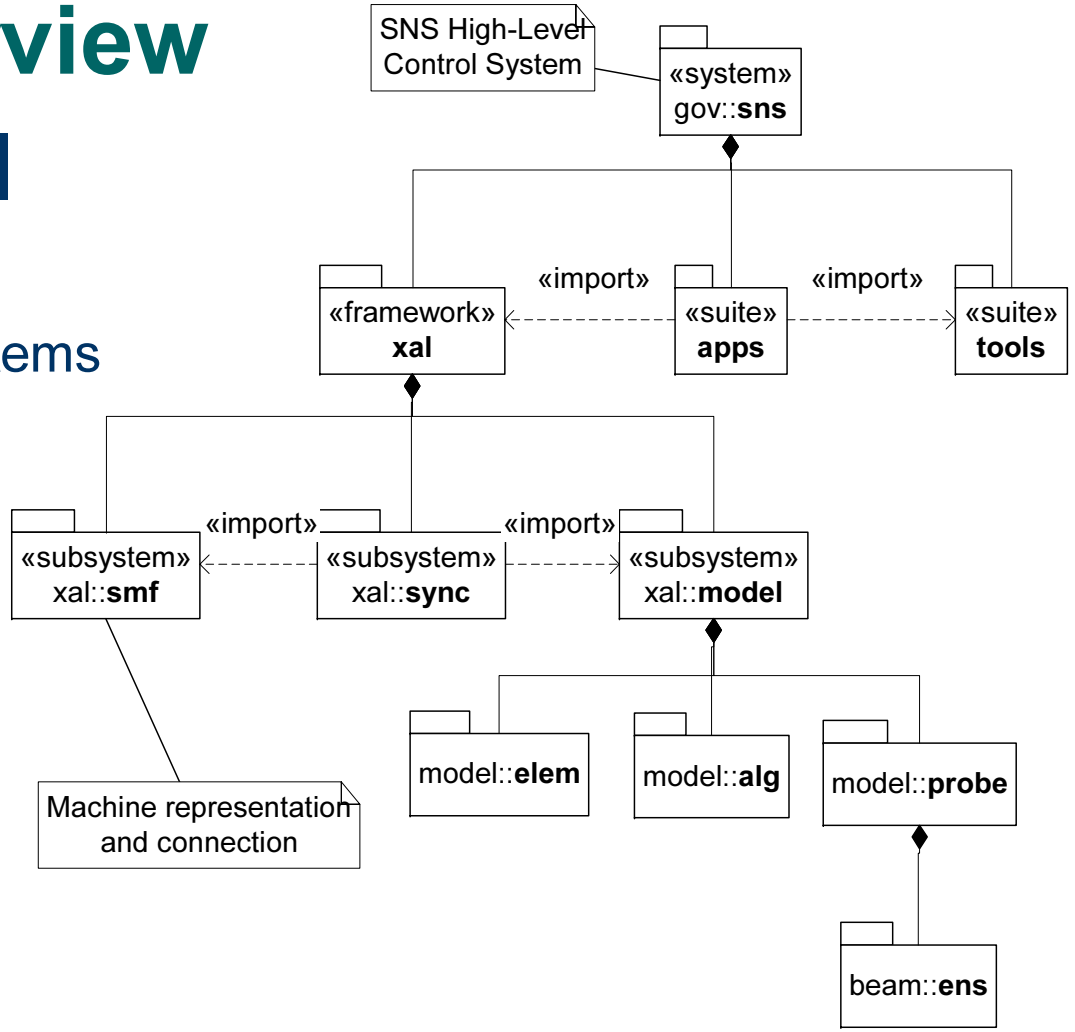


See J.D. Galambos "XAL Application Programming Framework" WE204

Similar design strategy used **Cosylab,** see I. Verstovsek, "CosyFramework: Application Development Framework for Java" WE203

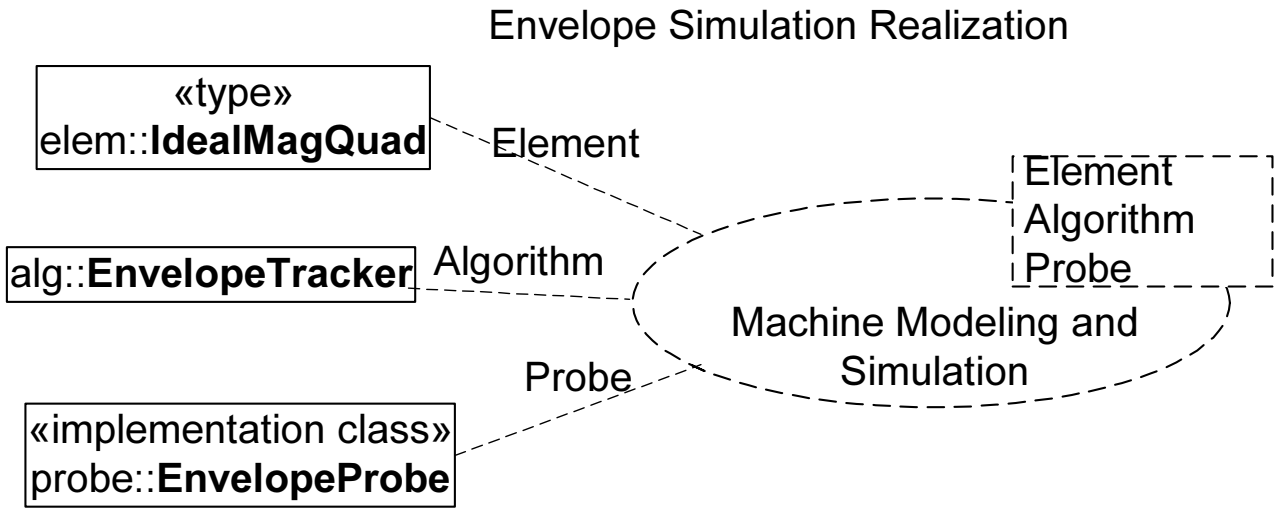
2. Model Overview

- XAL composed of tools, frameworks, and subsystems
- Simulator is “modeling” subsystem of XAL.
  (on-line or stand-alone)



Model Architecture

- Element/Algorithm/Probe design pattern
  Developed by N. Malitsky at BNL
- Separates the following
  - Machine representation
  - Beam representation
  - Dynamics calculations



Current Status

- Single-particle simulation
- Envelope simulation
- Response matrix calculation
- Dynamic model synchronization
- Used by working applications at SNS

Features & Benefits

- Open architecture - Extensible
  - Easily upgraded
  - Easily maintained
- Same engine supports multiple simulation strategies
  - Envelope
  - Multi-particle
  - Response matrix
- Dynamic machine synchronization
- Straightforward API (Elem/Alg/Probe)

⇒Rapid development of applications requiring model

3. Verification and Validation

Verification – does code compute what it is supposed to?

Validation – does code compute the answer?

- Verification
  - Particle simulation verified against Trace3D
  - Envelope simulation verified against Trace3D
- Validation
  - Envelope simulation validated against SNS commissioning data

Trace3D Approximation

Space charge envelope simulation requires evaluation of $R_D(x,y,z)$

- XAL numerically evaluates R_D

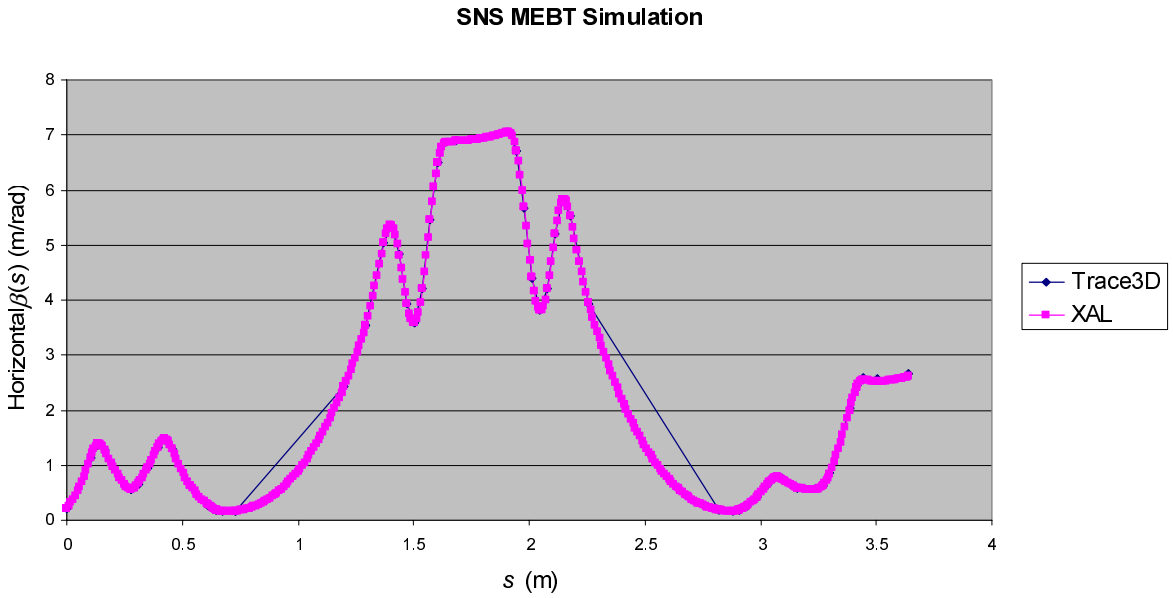
$$R_D(x,y,z) \equiv \frac{3}{2}\int_0^\infty \frac{dt}{(x+t)^{1/2}(y+t)^{1/2}(z+t)^{3/2}}$$

- Trace3D approximates R_D (via lookup table on ξ)

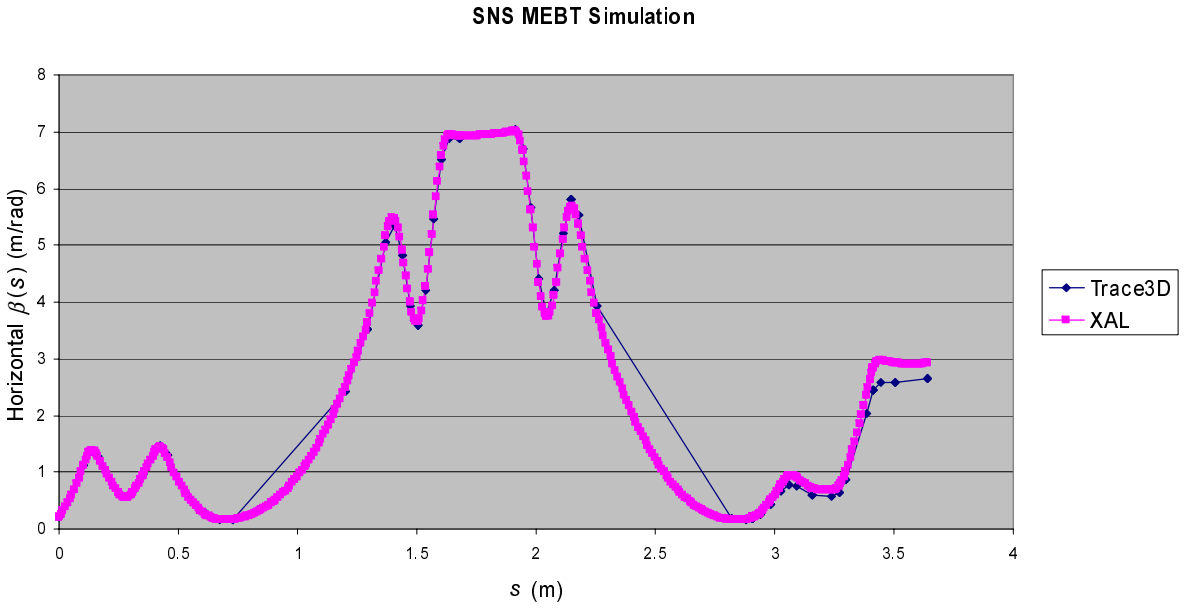
$$R_D\left(Z^2,Y^2,X^2\right) \approx \frac{3}{XZ}\frac{1}{X+Y}\left[1-\xi\left(\frac{Z}{\sqrt{XY}}\right)\right] + O(X-Y)$$

Trace3D Verification - Envelope Simulation

- Implementing Trace3D R_D approximation

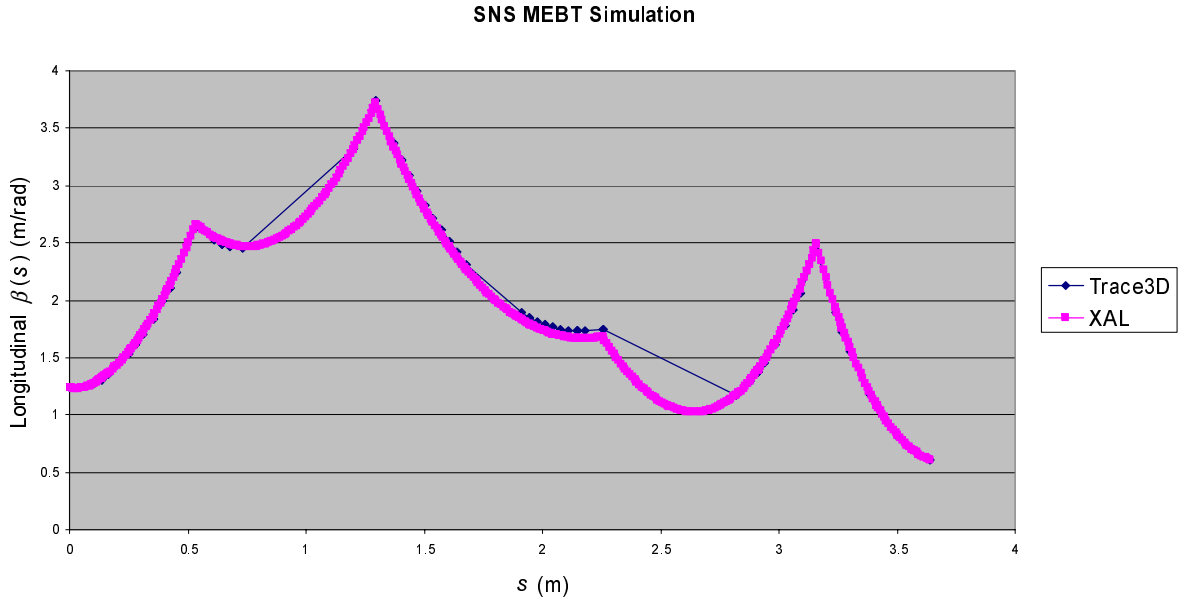


- Numerically evaluating R_D (*a la* Carlson)

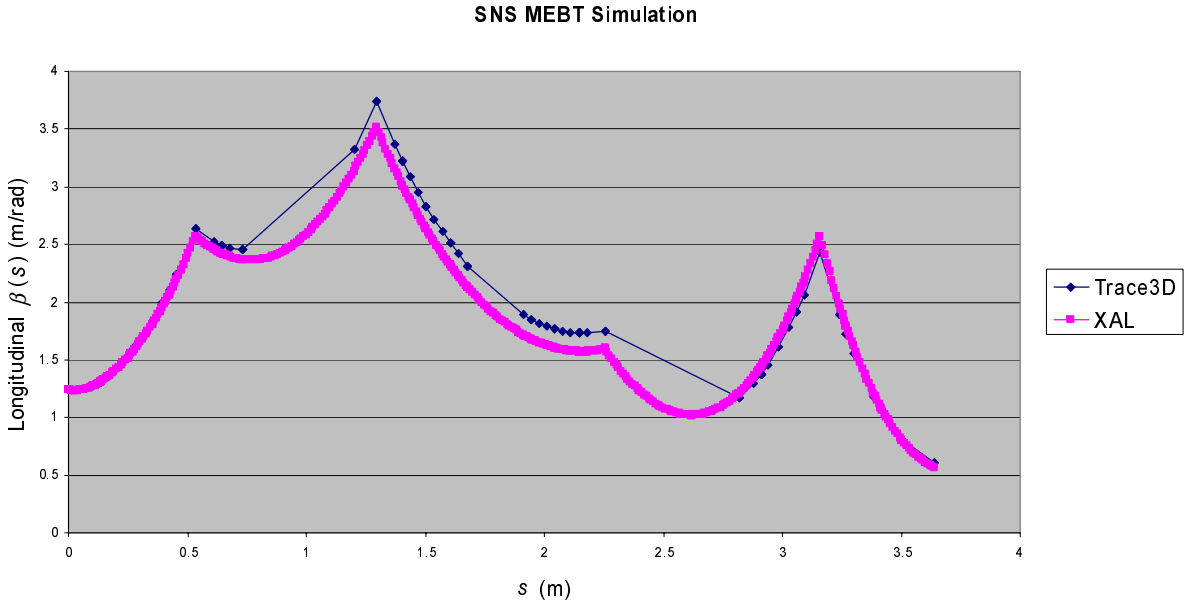


Trace3D Verification - Envelope Simulation

- Implementing Trace3D R_D approximation



- Numerically evaluating R_D (*a la* Carlson)



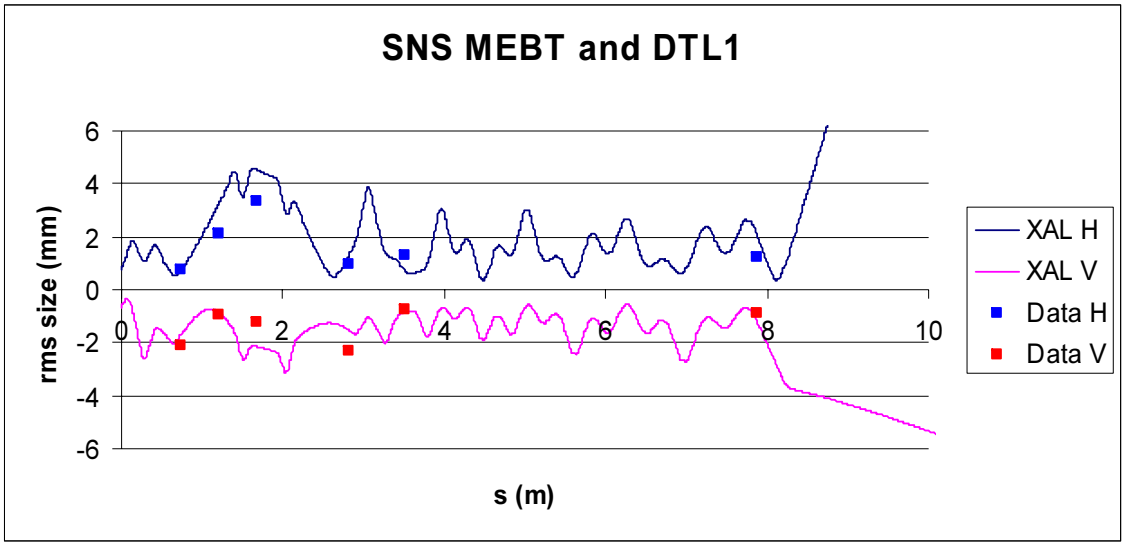
3. Model Validation

John Galambos, ORNL

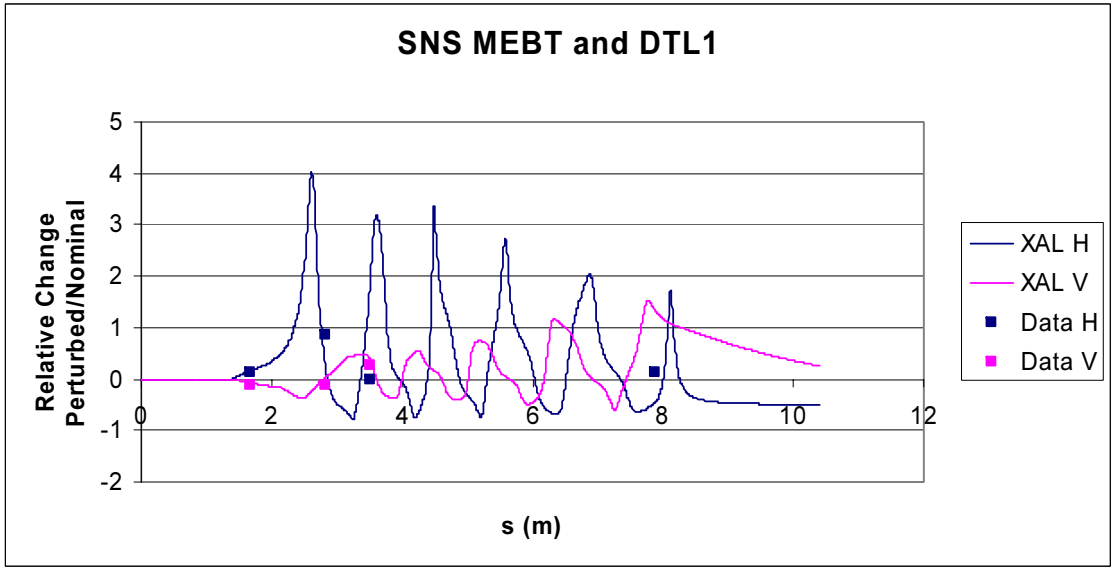
- Compared model prediction to live data
  - SNS Accelerator at Oak Ridge
  - Wire scanner data at six (6) beamline locations
- Two measurements
  - Direct comparison
  - Perturbed quadrupole 5 setting

SNS Verification - Envelope Simulation

- First measurement
  - Wire scanner data



- Second Measurement
  - Perturbed quadrupole 5 setting
  - Measure percent difference



Courtesy John Galambos, ORNL

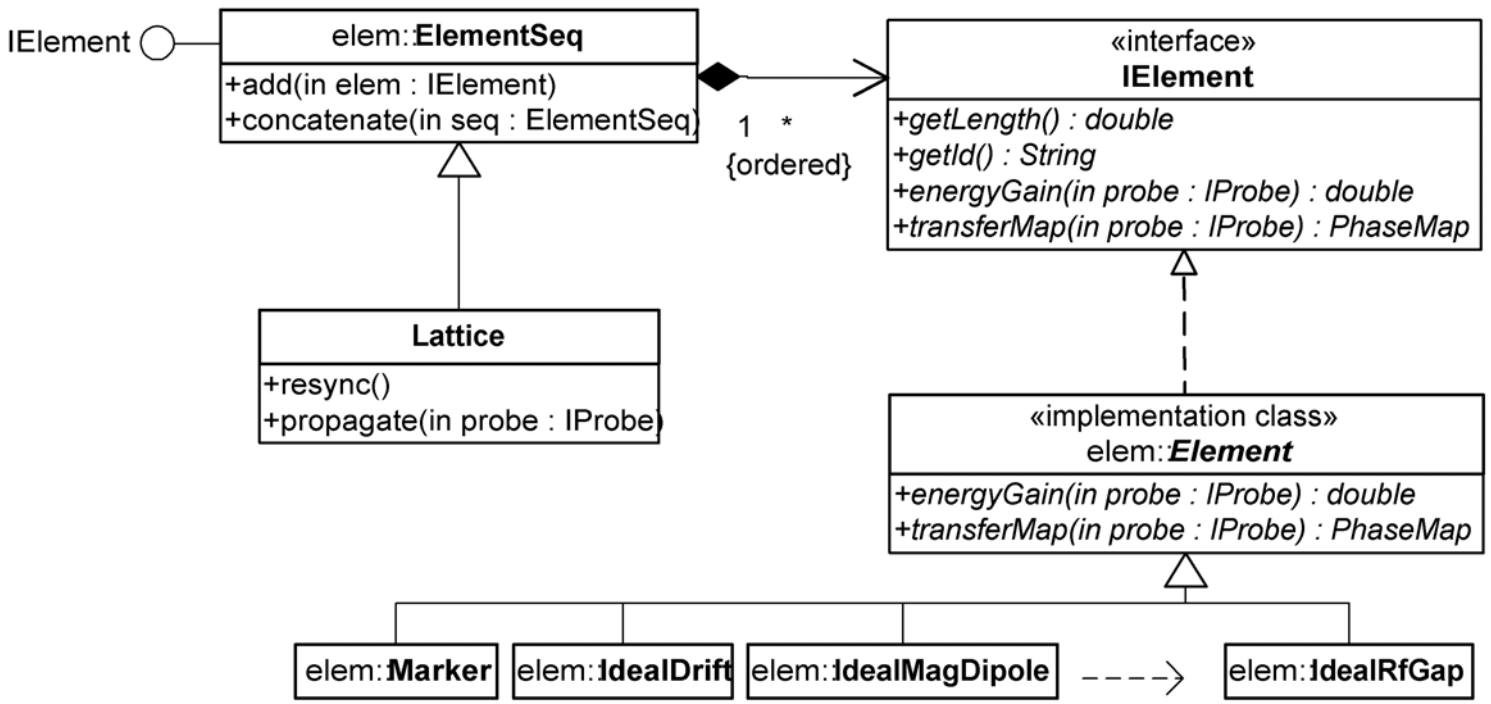
4. Summary

- Modern architecture
  - Extensible, maintainable
  - Supports multiple simulation strategies
  - Straightforward API (Elem/Alg/Probe)
- Dynamic configuration/synchronization
- XAL Model verified and validated

Future Work

- Ring modeling
- Envelope perturbations
- Multi-particle simulation
- Optimizers, solvers, etc.

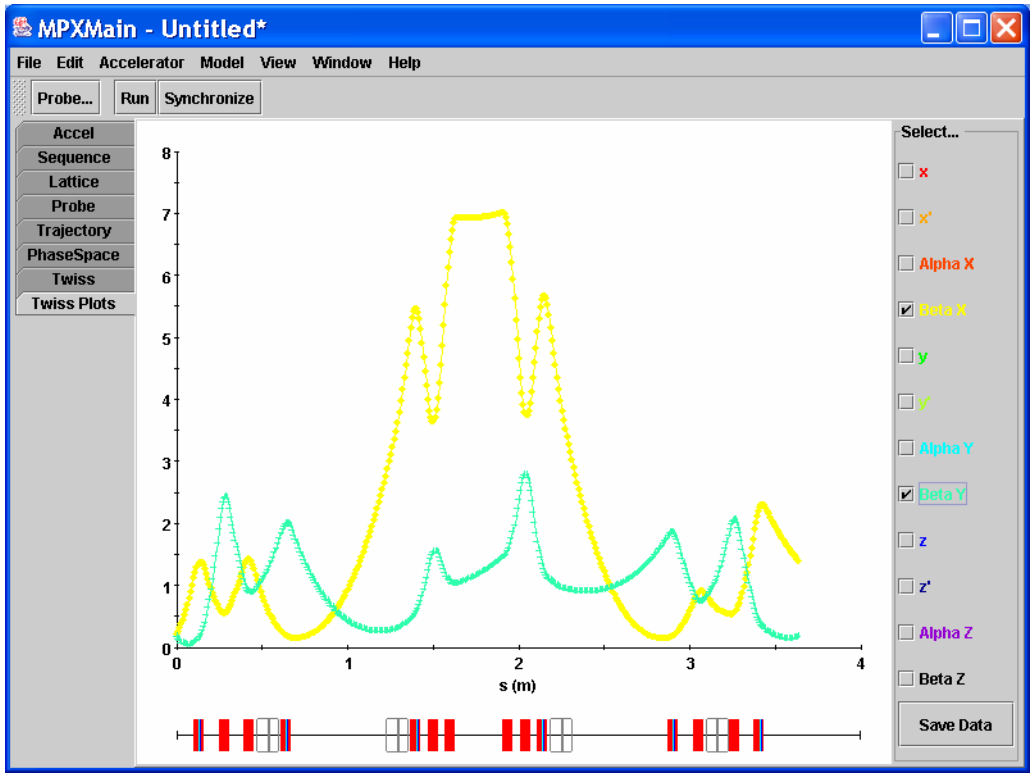
Lattice Representation



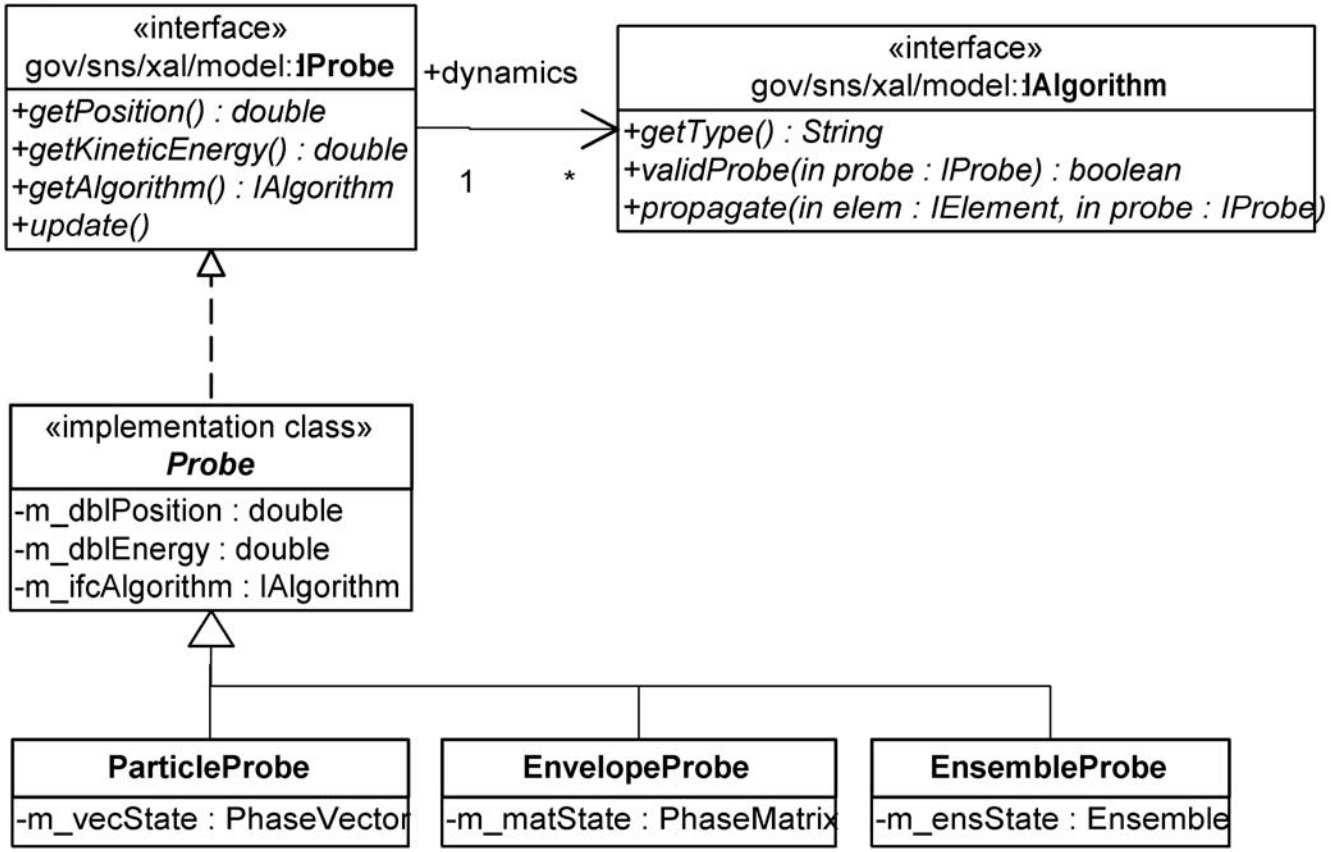
Machine Simulation Application

Wolf-Dieter Klotz, ORNL, ESRF

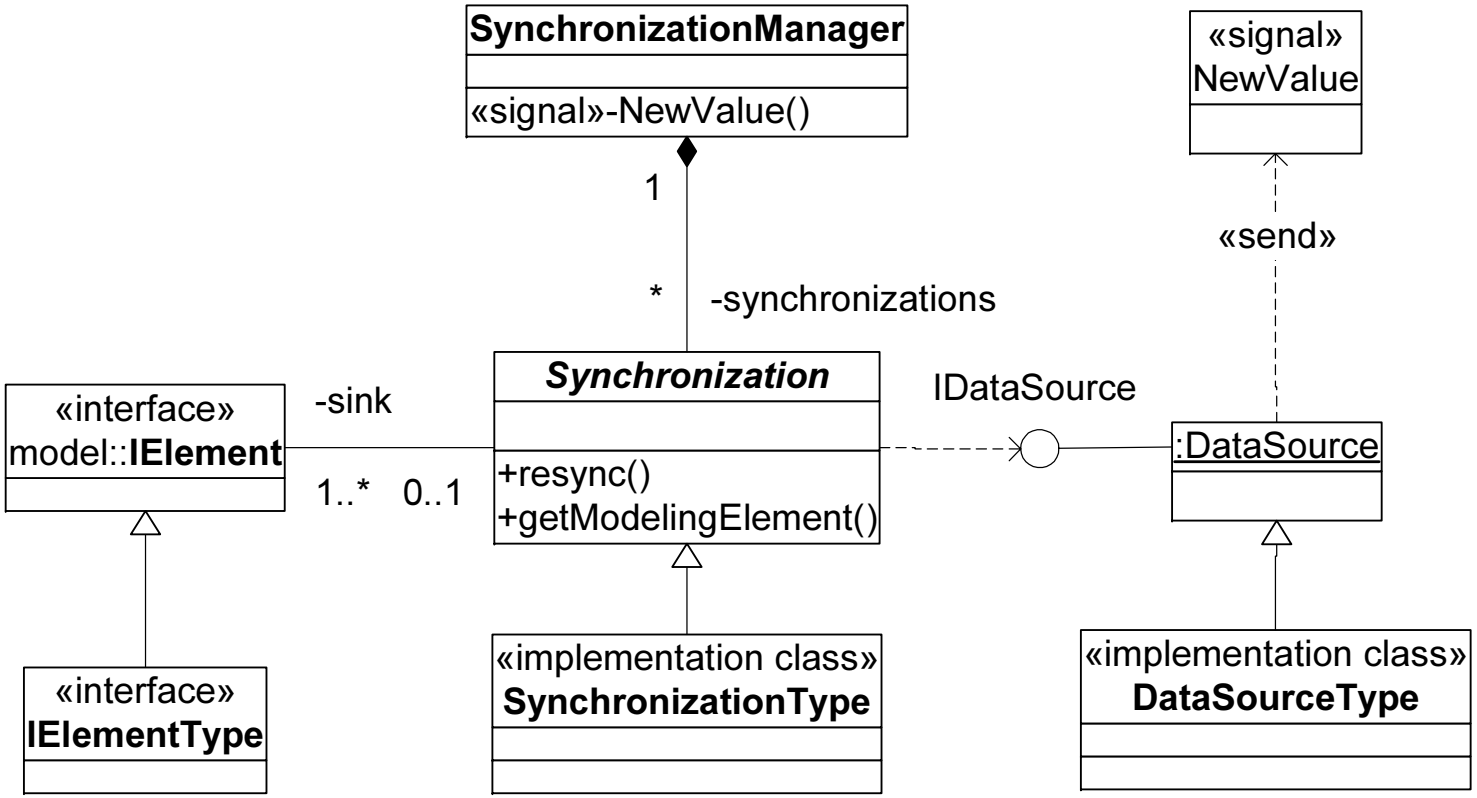
- Simulation using
  - Design values
  - Machine values
  - User specified values



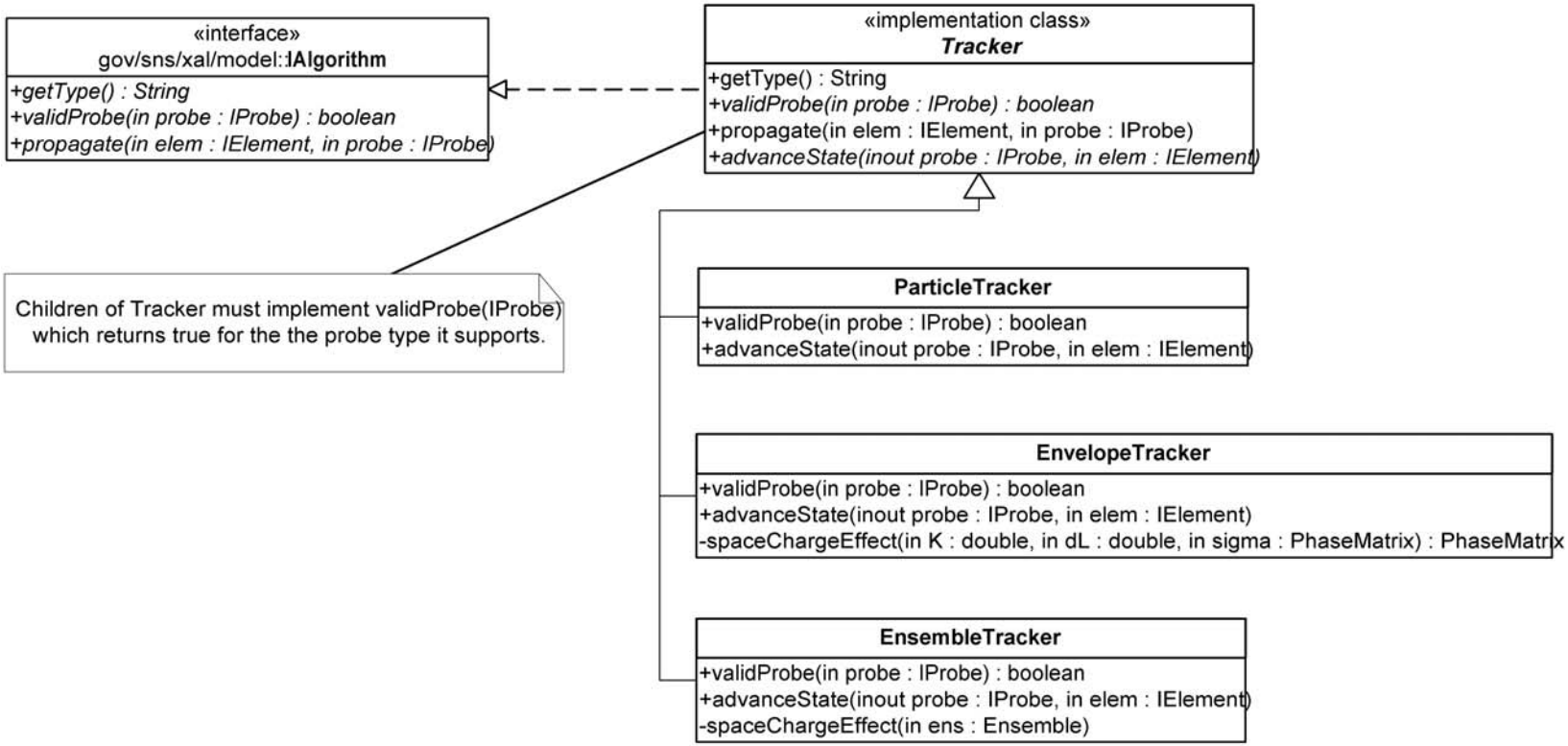
Beam Representation



Synchronization Mechanism



Dynamics Representation



Trace3D Validation - Envelope Simulation
Both transverse planes