XAL Framework and Applications

Thomas Pelaia II

SNS Accelerator Physics R&D Staff

June 18-20, 2003

Active Developers

- Chris Allen (LANL)
- Chungming P. Chu (ORNL)
- John Galambos (ORNL)
- Wolf-Dieter Klotz (ESRF)
- Craig McChesney (LANL)
- Dan Ottavio (BNL)
- Thomas Pelaia (ORNL)
- Andrei Shishlo (ORNL)

Acknowledgements

- Nicholay Malitsky (BNL)
- Peregrine McGehee (LANL)
- Nick Pattengale (formerly at LANL)
- Cosylab Group
- EPICS Community

What is XAL?

- XAL is a Java framework for rapidly developing accelerator based applications.
- Had its beginnings in UAL but is now independent.
- Provides common tools for the entire accelerator.
- Primarily built to accommodate SNS but is quite general.

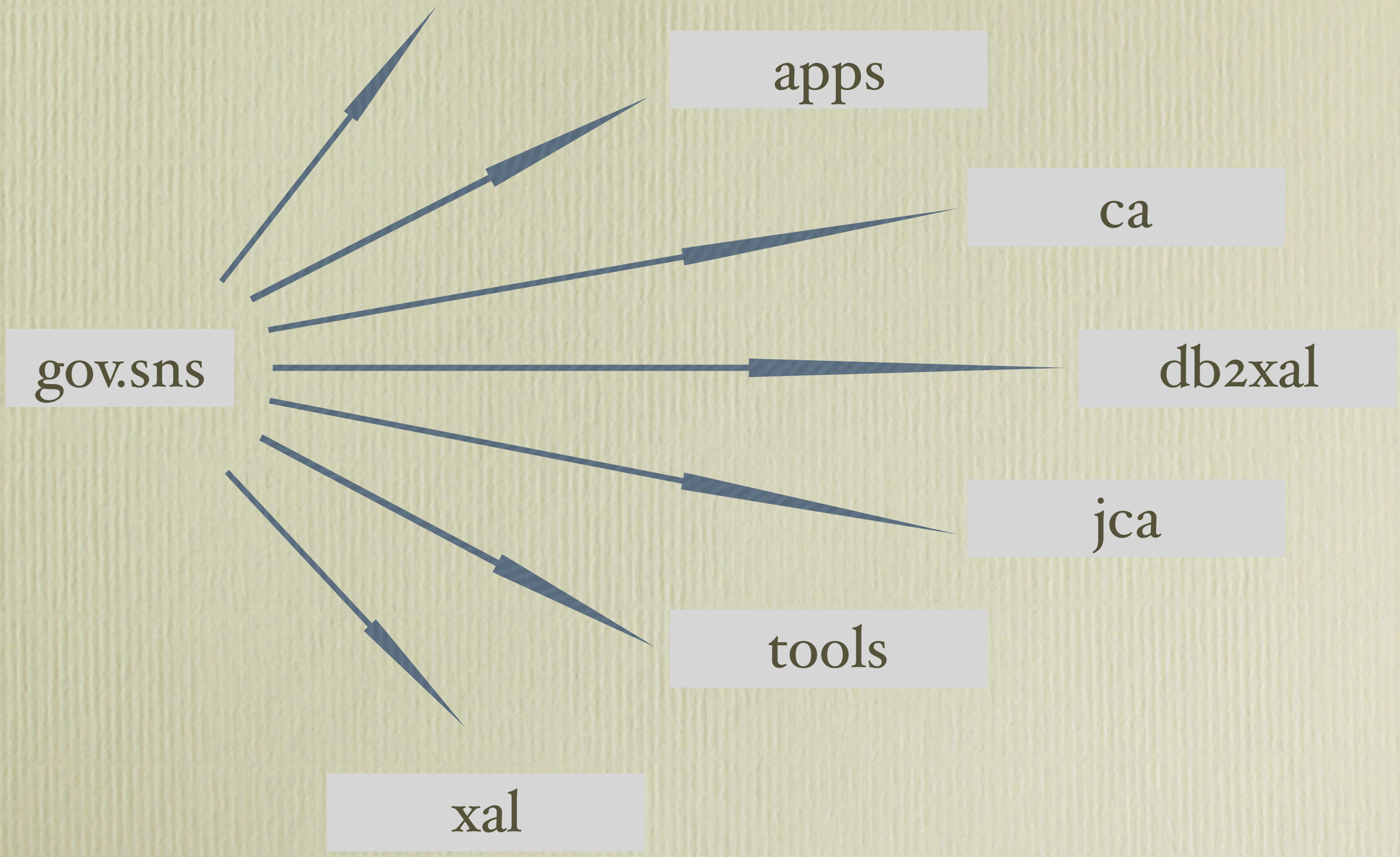
Goals

- Allow rapid development of robust applications
- Develop components once and share them across applications
- Have a common look and feel for all applications
- Develop intuitive, rich human interfaces
- Maintain good performance
- Support online modeling

Sample Subset of XAL Applications

Application	Description
Orbit Correction	Monitor and correct the orbit
MPS Post Mortem	Monitor MPS for concurrent faults and order sequentially
Orbit Difference	Changes magnet field and compares orbit differences of machine against the model
Xio	Browse devices by type and display live data in tabular and graphical forms
Scope	A virtual scope that plots live waveform PVs temporally aligned
XYZ PV Correlator	Allows the user to select two or three PVs and plot their correlated paths

Top Level Packages



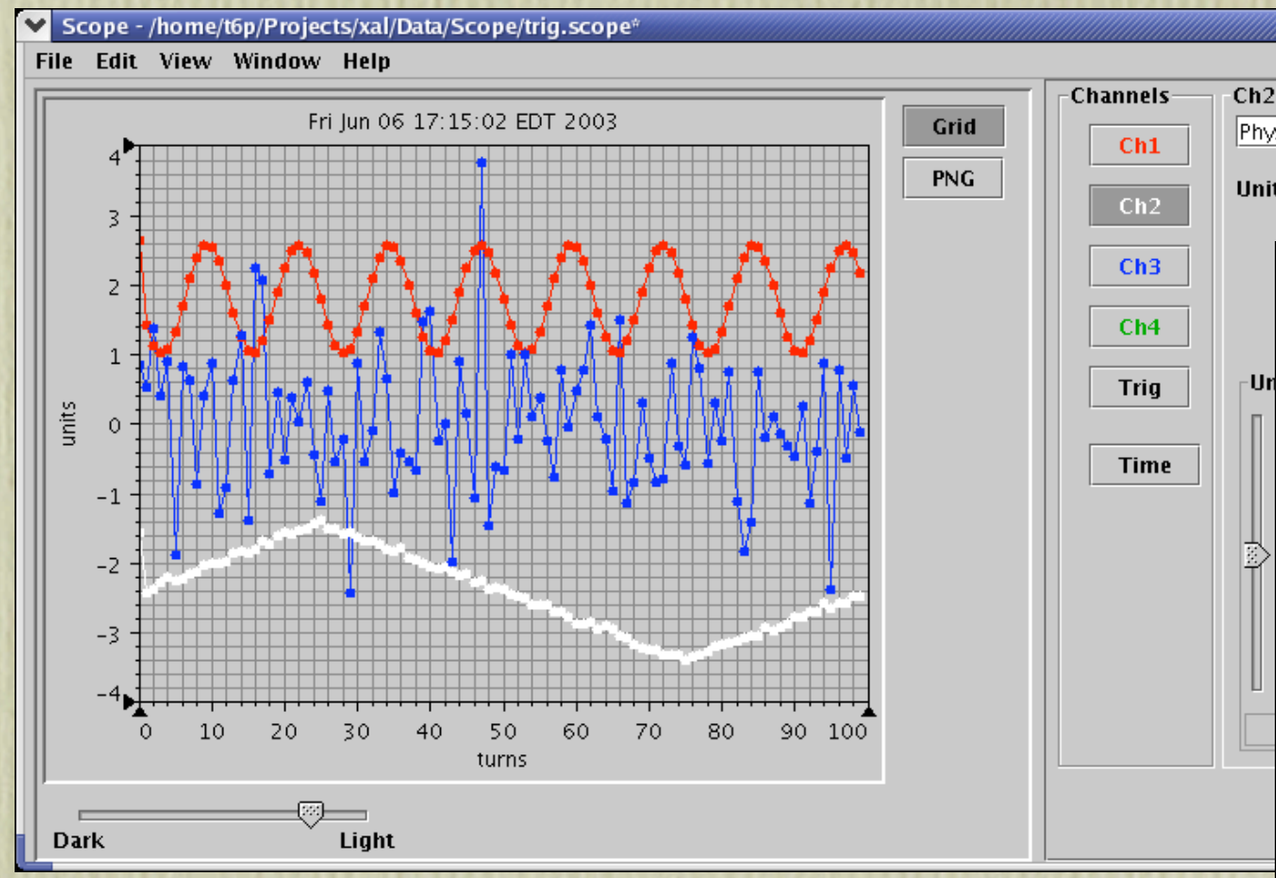
Application Framework

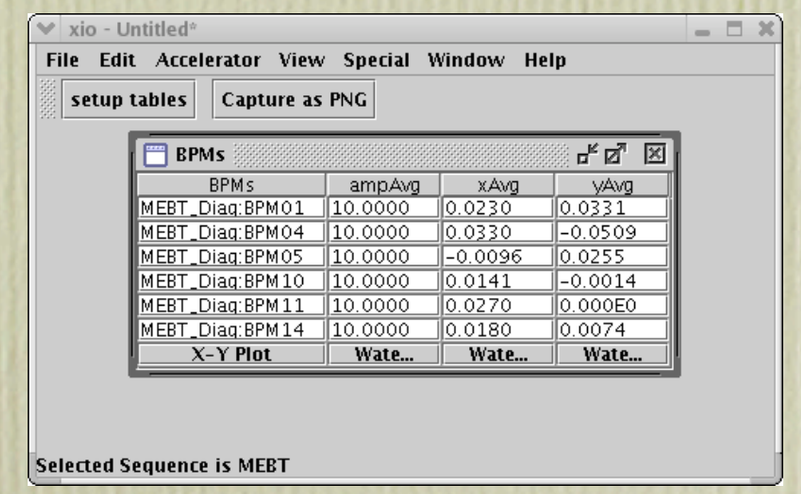
- Provides a common look and feel to all of our applications
- Generic and one accelerator theme
- Conforms to familiar user interface guidelines
- Document based
- Facilitates rapid application development
- Minimal boundaries on the developer
- Easily extensible

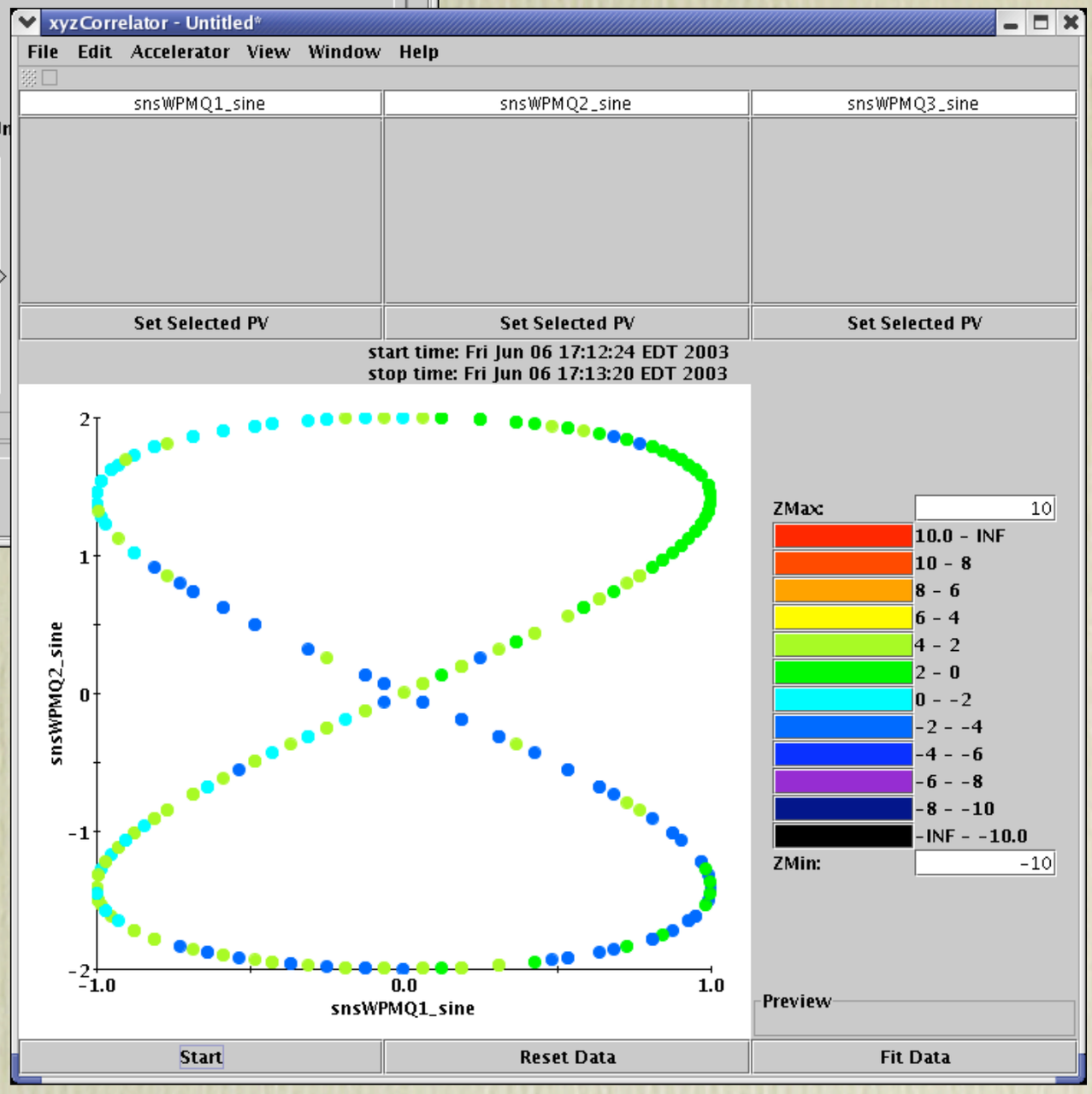
Application Classes to Subclass

- ApplicationAdaptor
  - Hooks to handle application events
  - Defines application attributes
- XalDocument
  - Handle document events and behaviors
- XalWindow
  - Main window for a document

Some Application Screenshots



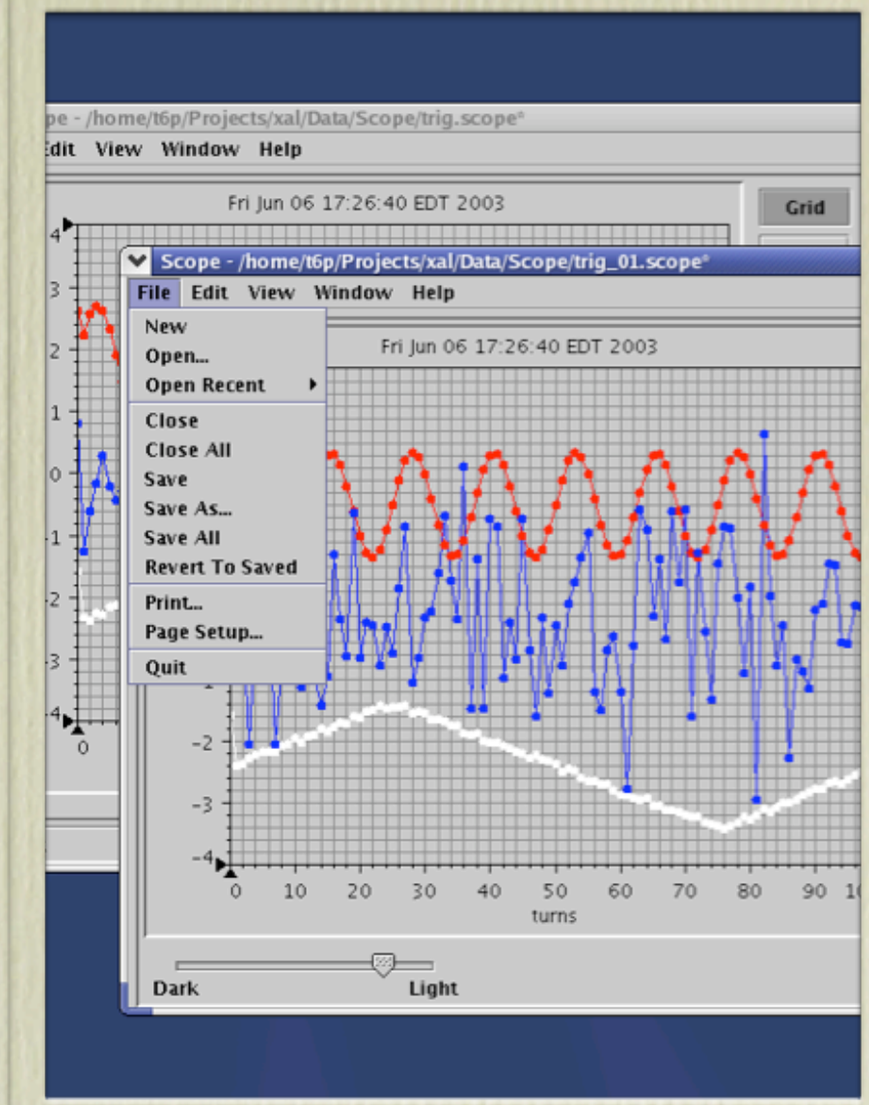




Familiar Look and Feel for Free

- Familiar Menus and commands
- Convenience actions
- Multi-document paradigm
- Standard output and error console
- Human interface cues
- Inherit several free application features

Scope Application



Tools

- Facilitate rapid development
- Some provide a common look and feel
- A few examples among dozens of current tools:
  - GUI components and utilities
  - data management
  - charting
  - math
  - correlator

Channel Access

- Two packages: gov.sns.ca and gov.sns.jca
- gov.sns.ca provides a convenient, abstract layer for generic channel access
  - Our applications and the XAL framework uses this package alone for channel access
- gov.sns.jca is an adaptor that bridges gov.sns.ca with a slightly modified version of jca

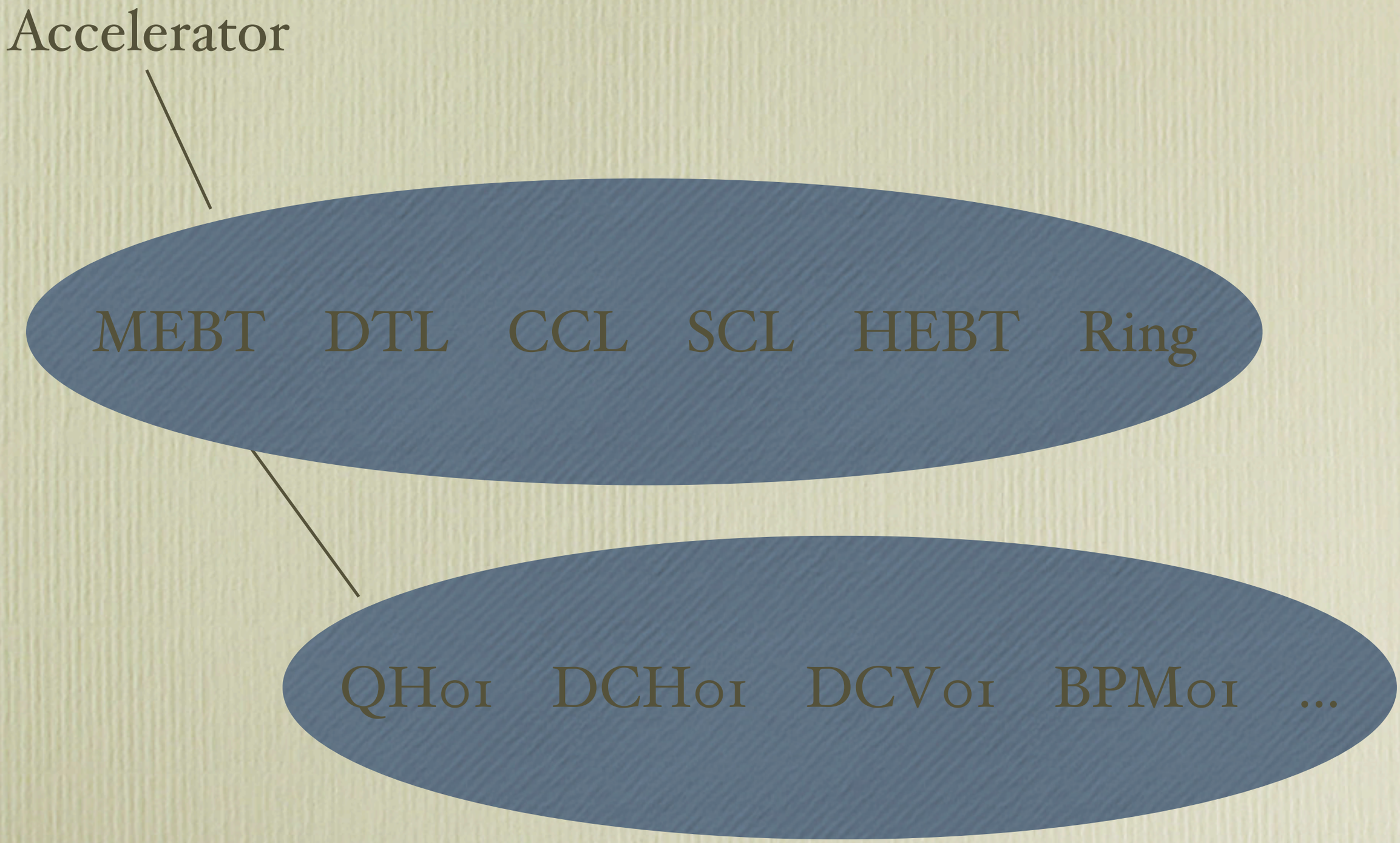
Accelerator Input

- Oracle database holds the accelerator definition which consists of accelerator sequences, devices and PVs
- The accelerator definition is copied into an XML file
- XAL reads the XML file and generates an object graph view of the accelerator
  - Accommodates overriding and extension
- Online model generates a model specific view of the accelerator

SMF Object Graph

- Accelerator contains accelerator sequences
- Accelerator sequence
  - corresponds to a physical section of the machine
    - contains accelerator nodes and sequences
- Accelerator node
  - One accelerator node per physical device
  - No drifts

Accelerator Hierarchy



Common Accelerator Nodes

- One class per device type
- Magnets
  - Bends, Correctors, Quadrupoles
- Diagnostics
  - BPM, BCM
- RF devices
- Generic
  - Completely data driven (no specific class)

Online Model

- Based on UAL Element/Algorithm/Probe architecture
- Calculates twiss parameters and transfer matrices
- Data synchronization for live analysis
- Model view is generated from SMF view but has drifts and element slices which are appropriate for model calculations

Present Activities

- Move to JCA 2.0 when it is released
- Collaborate with other JCA stakeholders
- Complete the online model development
- Begin design and development of an agent-based architecture
- Write several applications

Conclusion

- XAL has provided a rapid development environment for developing accelerator physics applications in Java
- XAL has proven to be flexible enough to meet new challenges and demands
- Our experience with XAL has been positive
- For more information and resources please visit: **http://www.sns.gov/APGroup/appProg/xal/xal.htm**