XAL

eXtensible Accelerator Language
eXtensible Accelerator Library
eXtreme Application Layer

A Framework for Portable High-Level Control Applications

Christopher K. Allen
Los Alamos National Laboratory
ckallen@lanl.gov

Developers

- Tom Pelaia, ORNL
- Paul Chu, ORNL
- John Galambos, ORNL
- Chris Allen, LANL
- Igor Verstovsek, Cosylab
- Igor Kriznar, Cosylab
- Ales Pucelj, Cosylab
- Gasper Pajor, Cosylab
- Mark Plesko, Cosylab

Presentation Outline

1. Description
2. Example applications using XAL
3. Specifications
4. Status
5. Future Directions

1. XAL Overview

- High-Level Control Application Framework
  - Java library
  - High-level view of machine (device-oriented)
  - Connection, modeling, and simulation
  - Controls toolbox
- Born out of UAL (Unified Accel. Lib.)
  - N. Malitsky et. al. at BNL

For more information see http://www.sns.gov/APGroup/appProg/xal

1. XAL Features

- Portable applications
  - Application written over XAL works on any machine
  - Represent arbitrary machines (configurable)
  - Machine introspection
- Global control
  - Control applications applied to machines at remote sites
- “Matlab-like” environment for HLA development
  - Modeling and simulation on-line
  - Controls “toolbox” for utilizing modern control theory

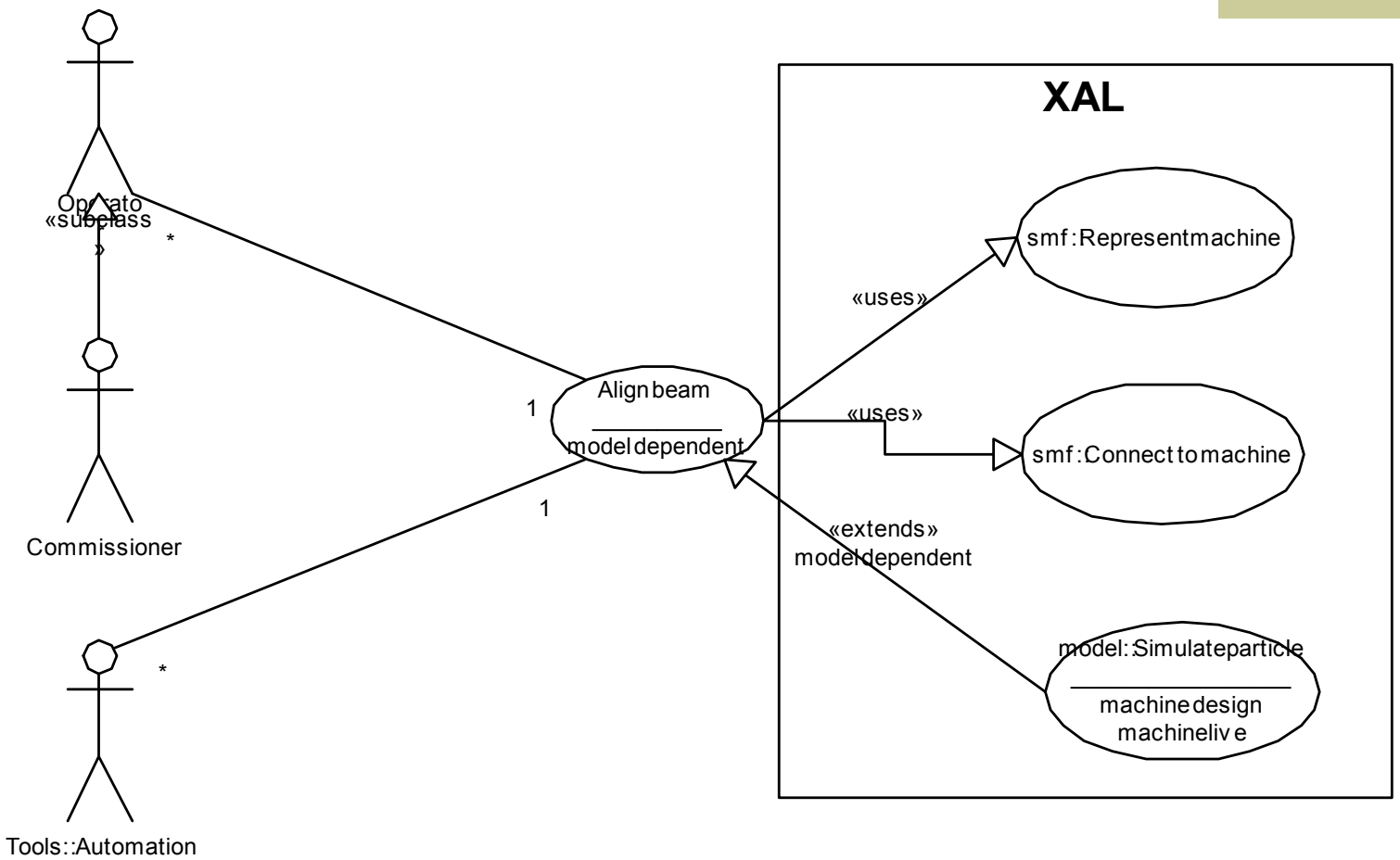
2. Applications using XAL

2.1 Beam alignment Use Case

2.2 Scripting applications

2.3 Remote commissioning of SNS

2.1 Use Case – Beam Alignment



2.2 Scripting and XAL

Jython

```
# read the accelerator
#acc_xml = "file:/home/jdg/xaldev/xal_xmls/sns.xml"
acc_xml = "file:./sns.xml"
acc = XmlDataSource.parseUrlAt(acc_xml, 0)

# get the some primary sequences from the accelerator

mebt = acc.getSequence("MEBT")
dtl1 = acc.getSequence("DTL1")
dtl2 = acc.getSequence("DTL2")

print " There are ", mebt.getAllNodes().size(), "nodes in the
sequence", mebt.getId()
```

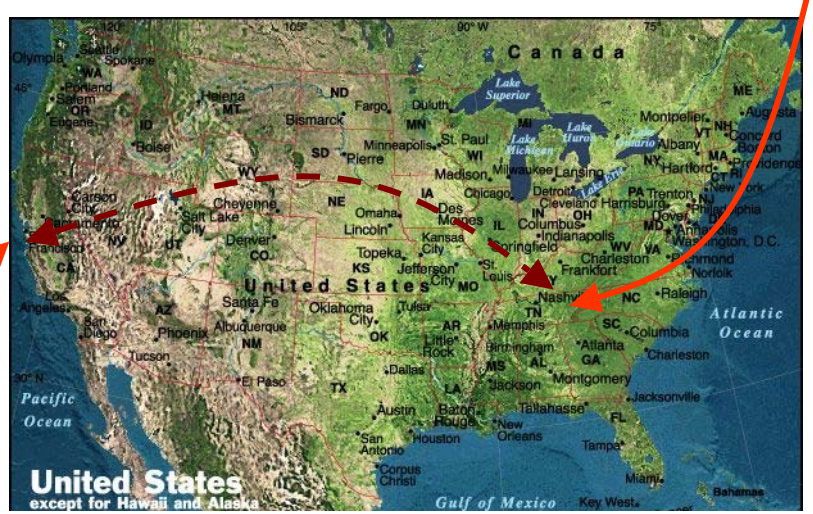
MatLab

```
%  scan the first quad
% monitor beam positions in the last MEBT BPM

for i =1:10
      fld(i) = field;
                  quad.setField(field);
                  va_chan.putVal(1); %  for virtual accelerator
                  pause(1); %  for virtual accelerator
                  xpos(i) = bpm.getXAvg;
                  ypos(i) = bpm.getYAvg;
                  field = field * 1.015; % increment field value
end

% Plot results
plot(fld, ypos)
```

2.3 Remote Commissioning of SNS Front End

Front End
at LBNL

Test control room
at ORNL



- Remote testing of high level applications from ORNL on the Front End System at LBNL

3. Specifications

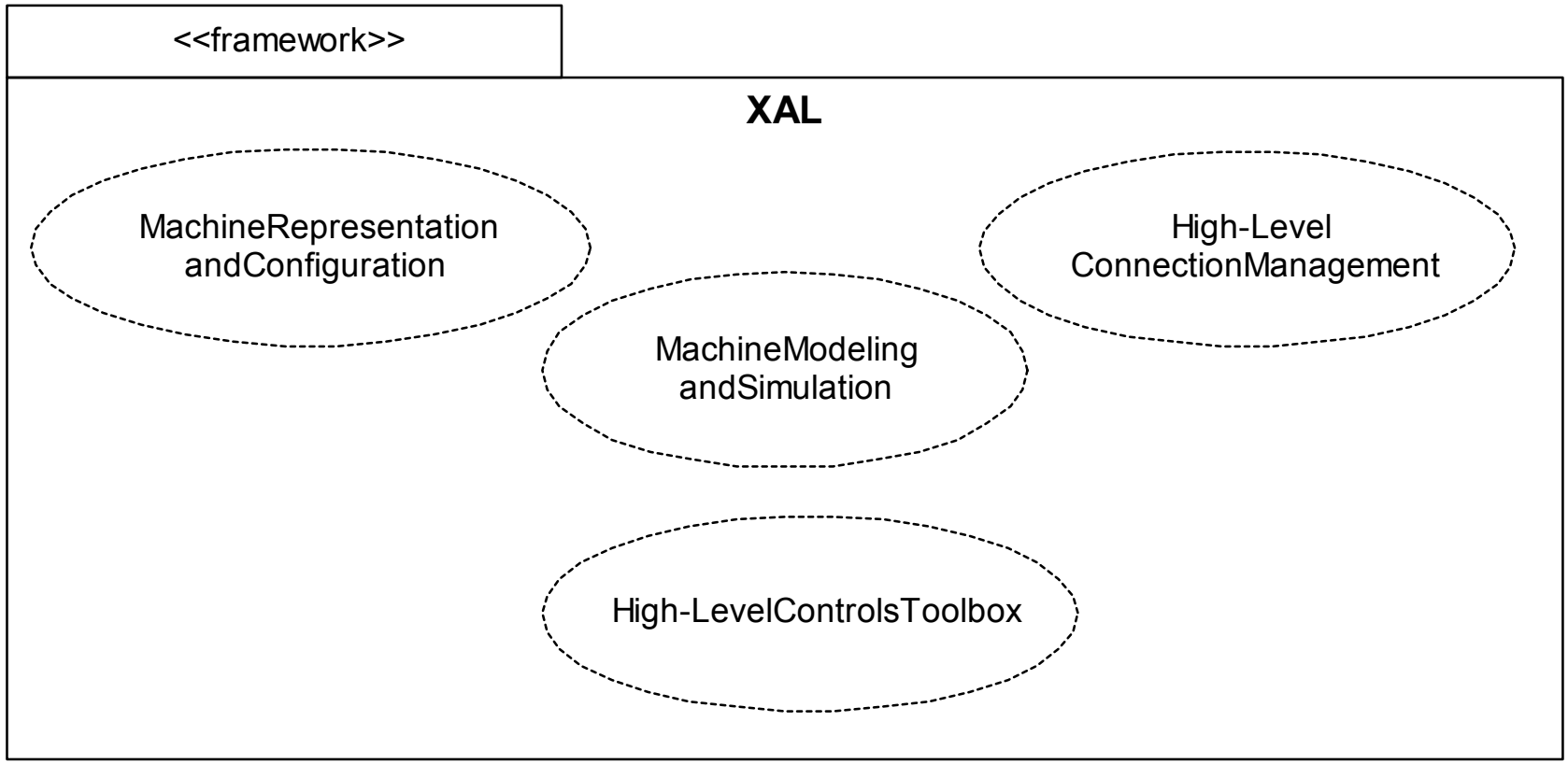
3.1 XAL Framework Diagram

3.2 Data Graph Example

3.3 XAL System Diagram

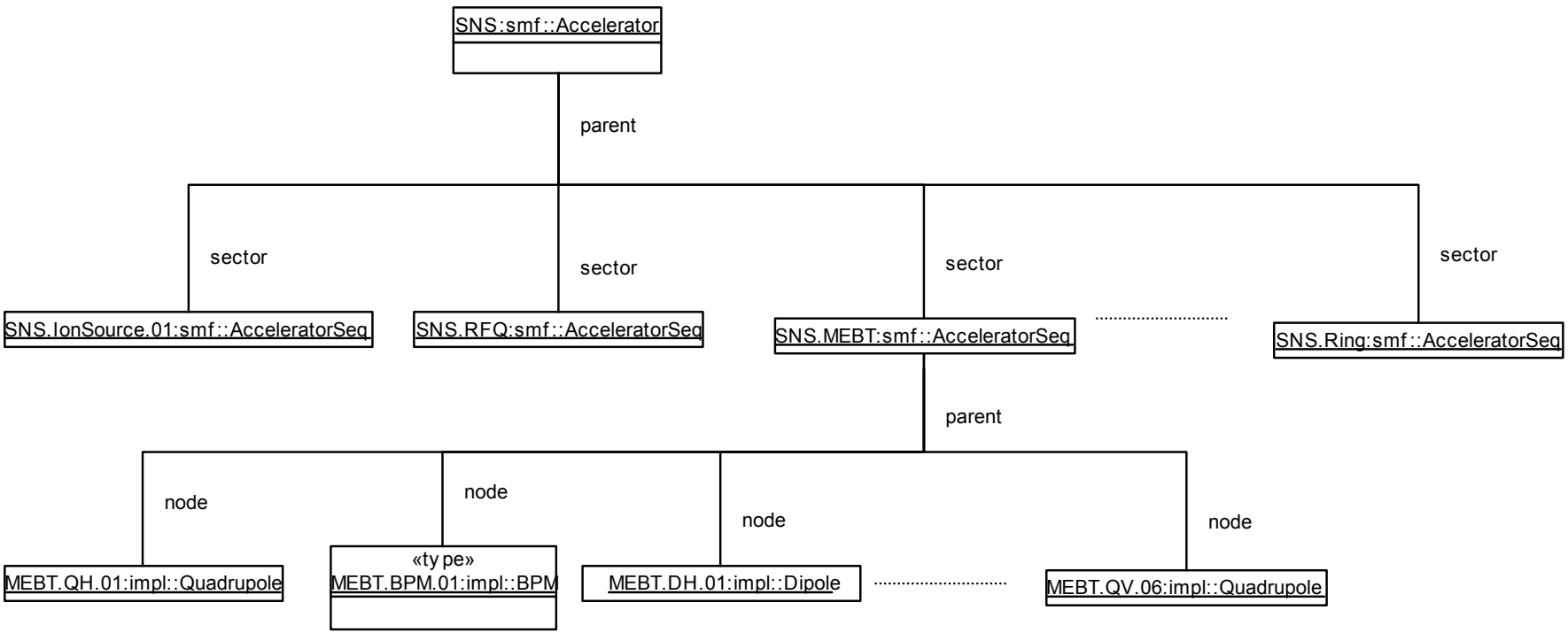
3.4 XAL Component Diagram

3.1 XAL Framework and Mechanisms

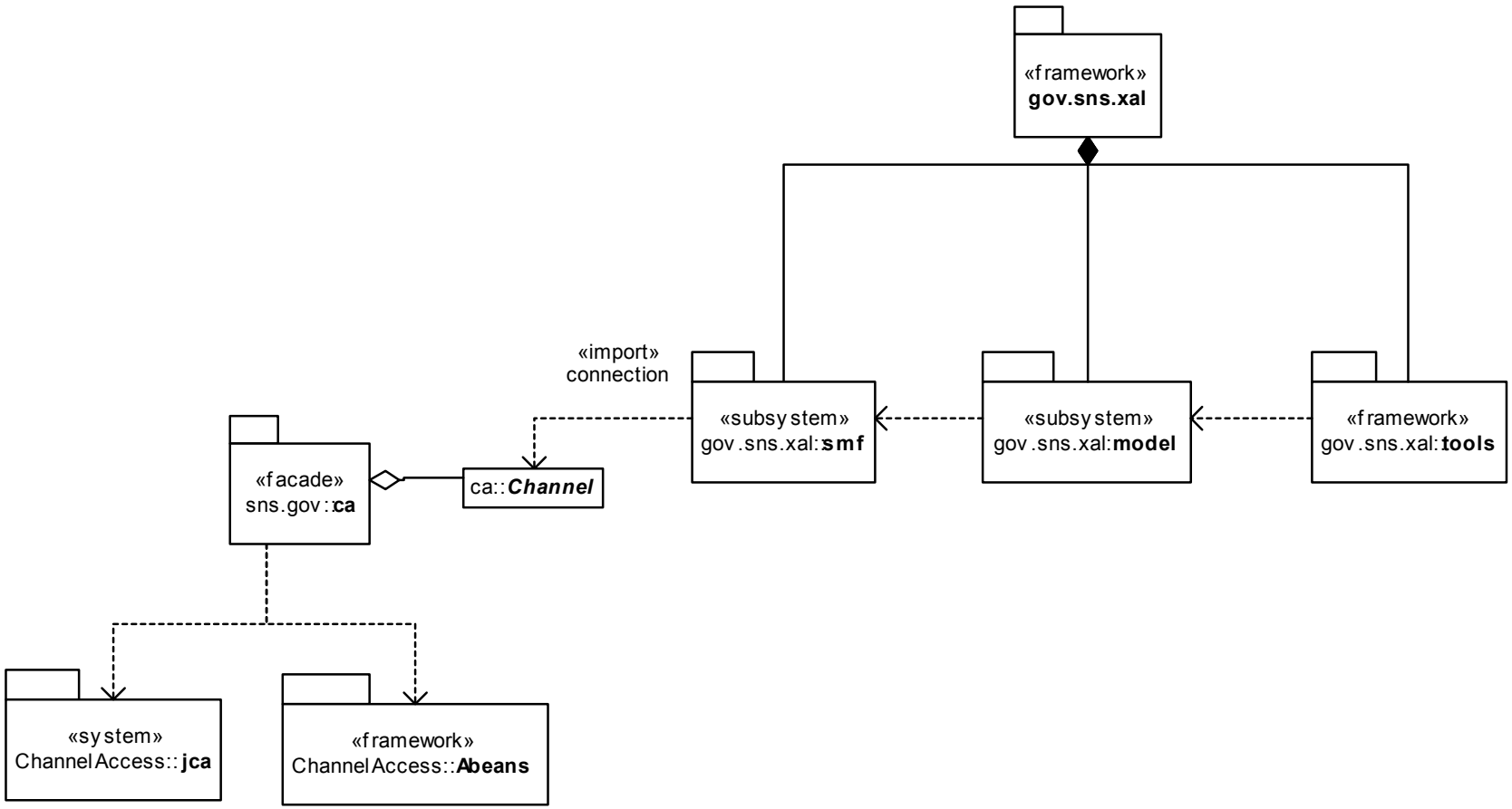


3.2 XAL Machine Representation

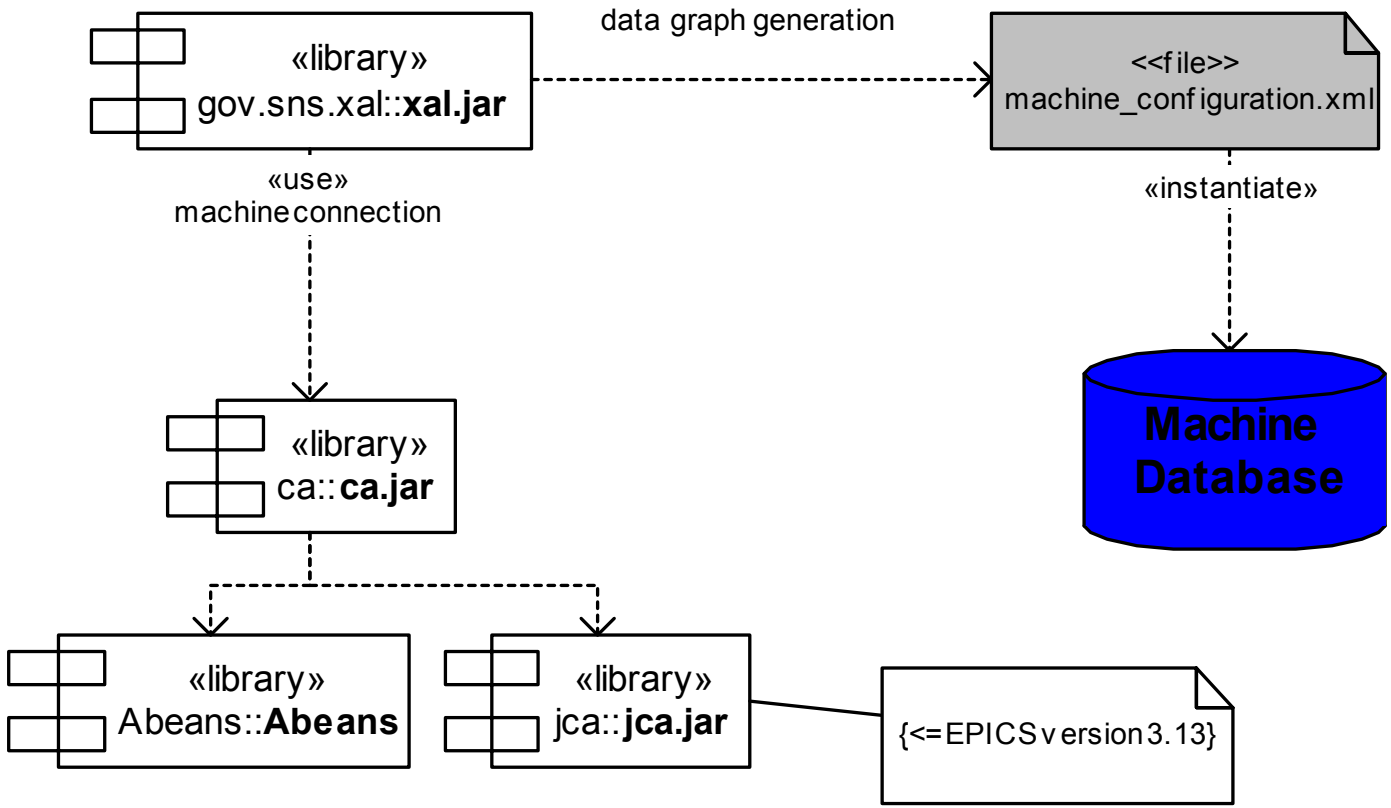
The XAL Data Graph



3.3 XAL System Diagram



3.4 XAL Component Diagram



4. XAL Status

- Machine representation completed
- Machined connection operational
- Modeling and simulation complete by 2003
- Control toolbox undeveloped

- Currently used at SNS for commissioning
  - XAL still a Version 1 product
  - Survived preliminary testing
  - Machine connection still changing

5. Future Directions

- Additions to the Toolbox

Accelerator high-level control is far behind the state of the art in control theory

- Auto pilots
- Guidance systems
- Spacecraft
- Cruise missile

Accelerator Automation

5. Challenge?

Modern Control Theory

- Dynamic Programming
- Optimal Control
- H^{∞} Control
- Robust Control
- Adaptive Control

Modern control techniques are over 50 years old yet rarely seen in accelerator applications

Let's put some in the toolbox!