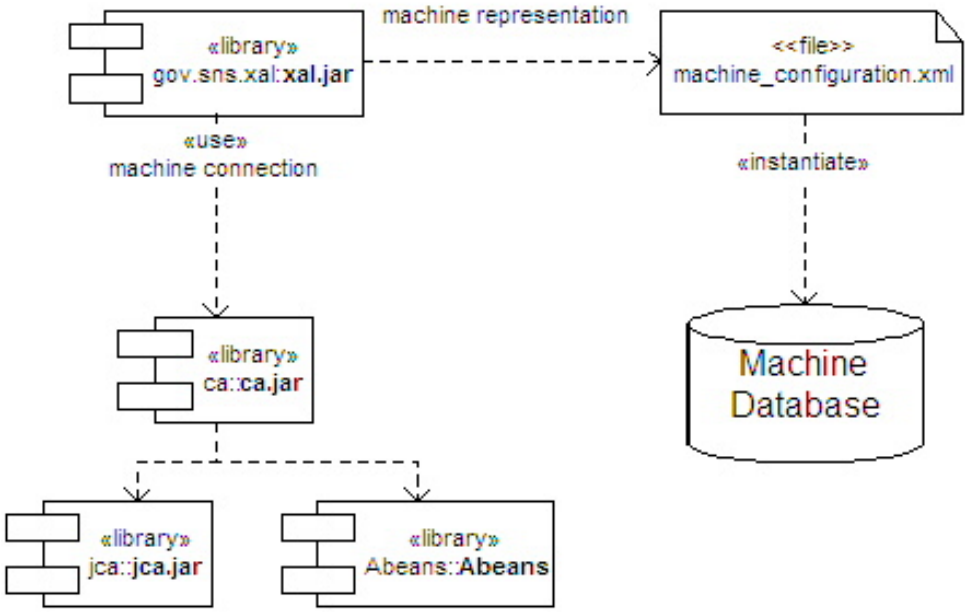
Control Room Accelerator Physics

Day 1

XAL Configuration: XML Configuration Files

Outline

1. XAL Configuration Overview
2. Master File
   1. Optics File
   2. Timing File
   3. Model Parameters File
3. Probe Files
4. Review



XAL Configuration

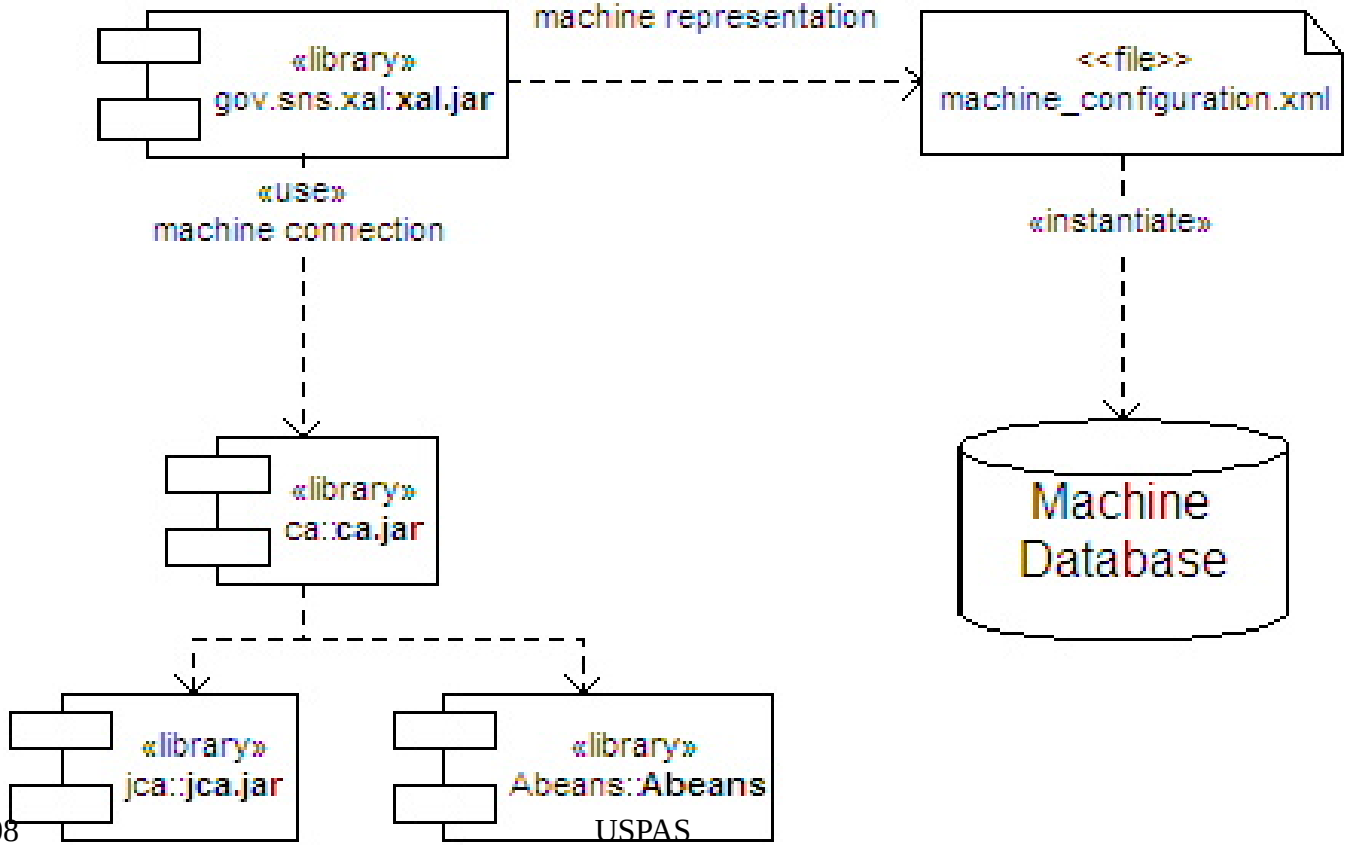
Overview

- XAL applications configure themselves to the accelerator hardware while launching
  - Allows all XAL applications to respond to changes in machine hardware
- Configuration mechanism is a common XML file
  - Typically generated by a database
  - Requires a specific format
- Here we present an overview of this XML file

XAL Configuration

Overview

The `xal.jar` file looks for the XML configuration file in order to produce the accelerator representation (and connections) used by XAL-based applications.



XAL Configuration

The XML File Intermediary

We recommend that some type of intermediary mechanism exist between the database and (all) the applications

(That is, like the XAL XML configuration file)

- If the database goes down, it brings down all the applications with it.
- Databases are very robust in general, however, during schema development refactoring is common and the query mechanisms can fail
- You can run XAL **without any** databases (i.e., for prototyping, etc.)
- For quick changes in the “middle of the night”
- Convenient way to preserve the hardware state

XAL Configuration File

Format

There are five basic XML files (formats) needed for XAL configuration.

- We highlight each in succession.
- For a more complete description of these file details see the following technical report:

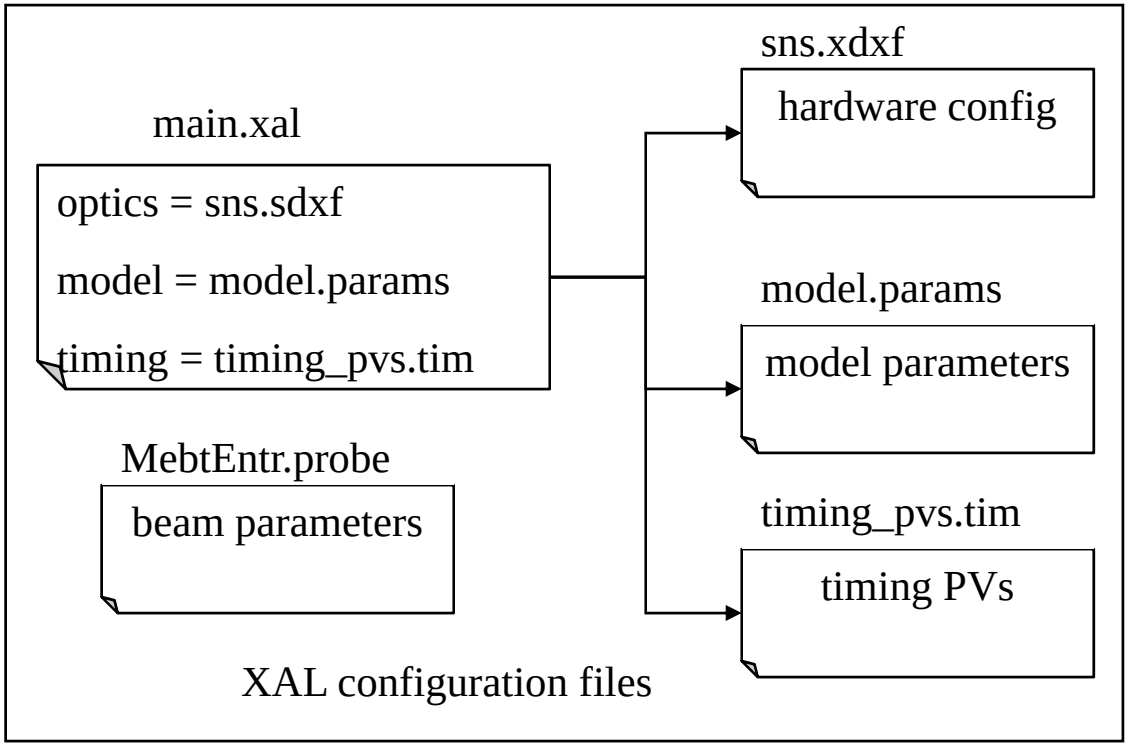
C.K. Allen, C.M. Chu, J.D. Galambos, T.A. Pelaia, and A. Shishlo, "Configuration and Initialization of the XAL High-Level Application Development Environment", LANL Tech. Report LA-UR-06-5303.

XAL Configuration File

Format

Five basic XML files

- main.xal – points to…
  - sns.xdxf
  - model.params
  - timing_pvs.tim
- BeamParam.probe
  - Optional
  - Beam parameters for online model



XAL Main Configuration File

main.xal

The `main.xal` file is the beginning of the XAL configuration mechanism. It is small – it simply points to the other configuration files

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE sources SYSTEM "xdxf.dtd">
<sources>
        <optics_source name="optics" url="sns.xdxf"/>
        <timing_source name="timing" url="timing_pvs.tim"/>
        <tablegroup_source name="modelparams" url="model.params"/>
</sources>
```

Excerpt : contents of main.xal

The XAL Optics File

sns.xdxf

File `sns.xdxf` is typically the largest file – it contains a complete description of all accelerator hardware and connection process variables.

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE xdxf SYSTEM "xdxf.dtd">
<xdxf system="sns" ver="2.0.0" date="Fri Jan 27 14:08:32 EST 2006">
  <comboseq id="MEBT-DTL">
          <sequence id="MEBT"/>
          <sequence id="DTL1"/>
          <sequence id="DTL2"/>
          <sequence id="DTL3"/>
          <sequence id="DTL4"/>
          <sequence id="DTL5"/>
          <sequence id="DTL6"/>
  </comboseq>
```

Excerpt : example XAL optics file

The XAL Optics File

sns.xdxf

Within the `sns.xdxf` the basic hardware object is the `<node>` element. It describes basic hardware components.

```
<node type="QH" id="MEBT_Mag:QH01" pos=".128" len=".061" status="true">
  <attributes>
    <magnet len=".061" polarity="-1" dfltMagFld="-34.636"/>
      <align x="0.0" y="0.0" z="0.0" pitch="0" yaw="0" roll="0"/>
      <aperture shape="0" x=".016"/>
   </attributes>
  <ps main="MEBT_Mag:PS_QH01"/>
  <channelsuite name="magnetsuite">
    <channel handle="fieldRB" signal="MEBT_Mag:QH01:B” settable="false"/>
  </channelsuite>
</node>
```

The XAL Optics File

sns.xdxf

Composite hardware objects are described by the `<sequence>` element. It usually represents the sectors of an accelerator system.

```
<sequence type="Bnch" id="MEBT_RF:Bnch01" pos=".528" len=".13" status="true">
   <attributes>
      <align x="0.0" y="0.0" z="0.0" pitch="0" yaw="0" roll="0"/>
      <sequence predecessor=”null”/>
      <rfcavity                 amp="1.3"
             TTFCoefs=".445, 0, 0"
             TTFPrimeCoefs="0, 0, 0"
             STFCoefs="0, 0, 0"
             STFPrimeCoefs="0, 0, 0"
             TTF_endCoefs=".445, 0, 0"
             TTFPrime_EndCoefs="0, 0, 0"
             STF_endCoefs="0, 0, 0"
             STFPrime_endCoefs="0, 0, 0"
             structureMode="0"
             phase="-90" freq="402.5"/>
   </attributes>
   <channelsuite name="rfsuite“>
….
```

The <timing_source> File

timing_pvs.tim

The `<timing_source>` file is very simple, it lists the channels carrying timing information. This information is contain in a single `<channelsuite>` element.

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<!DOCTYPE timing SYSTEM "xdxf.dtd">
<timing>
    <channelsuite name="timingsuite">
        <!-- beam trigger PV: 0=Trigger, 1=Counting -->
        <channel handle="trigger" signal="ICS_Tim:Gate_BeamOn:SSTrigger"/>
        <!-- beam trigger mode PV: 0=Continuous, 1=Single-shot -->
        <channel handle="mode" signal="ICS_Tim:Gate_BeamOn:SSMode"/>
        <!-- specify how many beam pulse(s) -->
        <!-- readback while it's counting down -->
        <channel handle="count" signal="ICS_Tim:Gate_BeamOn:SSCount"/>
        <!-- readback of overall rep rate -->
        <channel handle="repRate" signal="ICS_Tim:Gate_BeamOn:RR"/>
    </channelsuite>
</timing>
```

Excerpt : XAL Timing File

Specifying Model Parameters

A Specific Use of Table Groups

- Parameters for the online model (in particular, parameters describing the beam) are stored as a specific `<tablegroup>` record in the file pointed to by the entry

```
<tablegroup_source name="modelparams" url="model.params"/>
```

  in the main.xal file.

- *Table groups* are essentially miniature databases unique to XAL. They are used for passing arbitrary data to XAL applications, as well as persistence.
- Although table groups may be in any format, the one holding the model data is specific.

Specifying Model Parameters

A Specific Use of Table Groups

- Describing table groups, or the table group schema for specifying the model parameters, is beyond our scope.
  - For details on these topics refer to the document indicated previously.
- Use of the model parameters file provides a particular convenience
  - The online model automatically uses the appropriate parameters for the accelerator sequence that is being simulated.
- An alternative method is to specify beam parameters in a *probe file*.

Specifying Model Parameters

Probe Files

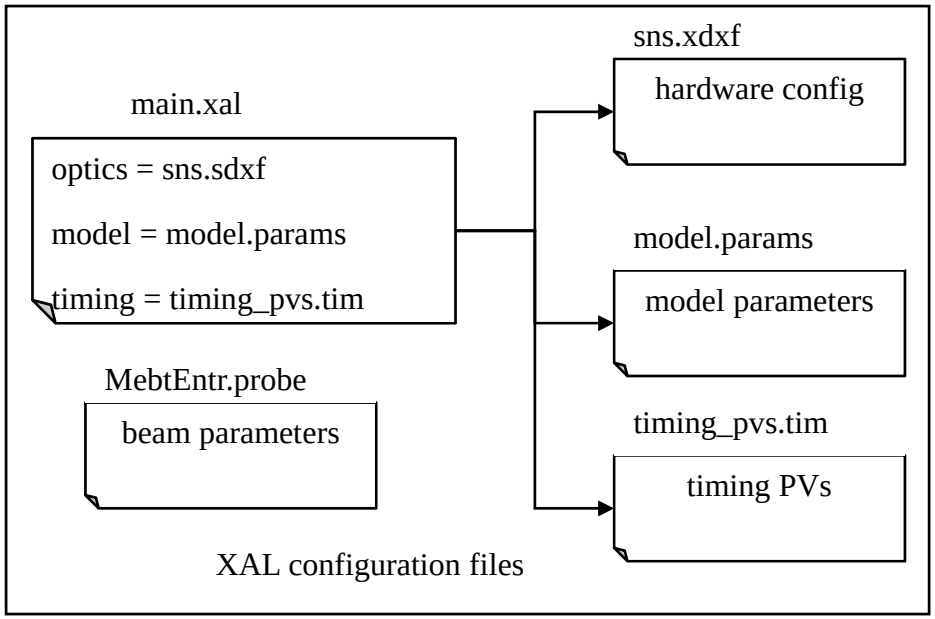
Probe files describe the beam parameters at a particular beamline location

- Online model beam representations are called *probes*
- Practical use of probe files is lessened by the addition of the model parameters table group

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<probe time="Feb 9, 2004 2:42:00 PM" type="gov.sns.xal.model.probe.EnvelopeProbe">
    <comment text="XAL model probe representing rms envelope state at HEBT entrance"/>
    <algorithm type="gov.sns.xal.model.alg.EnvTrackerAdapt" ver="2">
        <tracker debug="false" update="1"/>
        <adapt errortol="1.0E-5" initstep="0.031278684521382855"
          maxstep="0.0" norm="0" order="2" slack="0.05" maxiter="50"/>
    </algorithm>
    <state id="" type="gov.sns.xal.model.probe.traj.EnvelopeProbeState">
        <location W="2500000.0" elem="" s="0.0"/>
        <species Er="9.393014E8" q="-1."/>
        <beam I="0.020" Q="4.96894E-11"/>
        <envelope
            alphaX="-1.962" betaX="0.183" emitX="2.73e-6"
            alphaY="1.768" betaY="0.161" emitY="2.73e-6"
            alphaZ="0.0196" betaZ="0.5844" emitZ="3.8638e-6"/>
    </state>
</probe>
```

Configuring XAL

Review

- XAL applications configure themselves to the hardware when launched.
- The configuration information is stored in XML files, of which, there are five different formats
- There is a “master file” which points to other data files
- There is an optional file type that describes beam information

- main.xal
  - sns.xdxf
  - model.params
  - timing_pvs.tim
- BeamParam.probe