XML CONSTRUCTS FOR DEVELOPING DYNAMIC APPLICATIONS
OR TOWARDS A UNIVERSAL REPRESENTATION OF PARTICLE ACCELERATORS IN XML

J. Chrin, R. Krempaska, H. Lutz, G. Prekas, Paul Scherrer Institut, Villigen PSI, Switzerland
T. Pelaia, Oak Ridge National Laboratory, Tennessee, USA

ABSTRACT

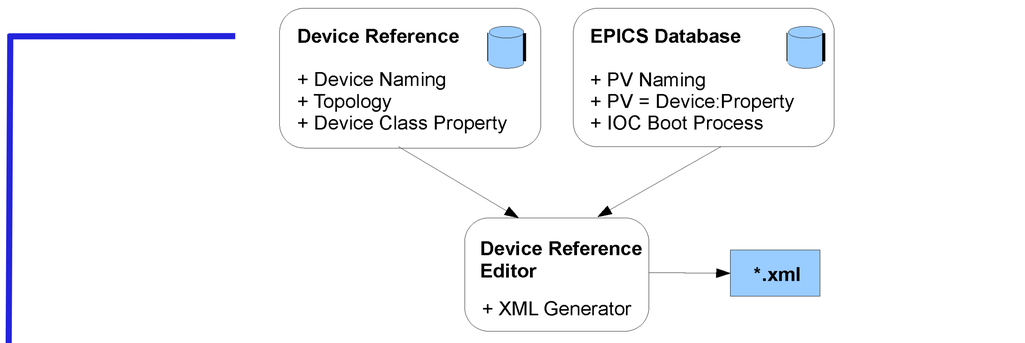
A recognized practice in the development of high-level beam dynamics applications is to separate data parameters destined for the configuration of the application from the programming language domain. The contemporary approach is to generate input files that provide the configuration parameters in a structured data format specified by the Extensible Markup Language (XML), enhancing flexibility, and simplifying code maintenance. Furthermore, a careful consideration to the form of XML syntactic constructs i.e. structured elements, attributes, etc., that map well to the various accelerator components, provides a basis for portability of configuration classes and high-level applications. This has been exemplified by the XAL application software package which initiated an XML description of the Standard Machine Format (SMF) accelerator object model. We have since adopted and optimized XML-SMF to provide an XML representation of both the Swiss Light Source (SLS) and the SwissFEL 250 MeV Injector Test Facility. We demonstrate how such a common set of XML constructs allows us to deploy the same, example orbit display application at both facilities. Our experience leads us to advocate a *Universal Machine Format (UMF)* that encompasses an all-inclusive XML vocabulary for the management of particle accelerator information pertaining to beam dynamics applications.

WHY XML?

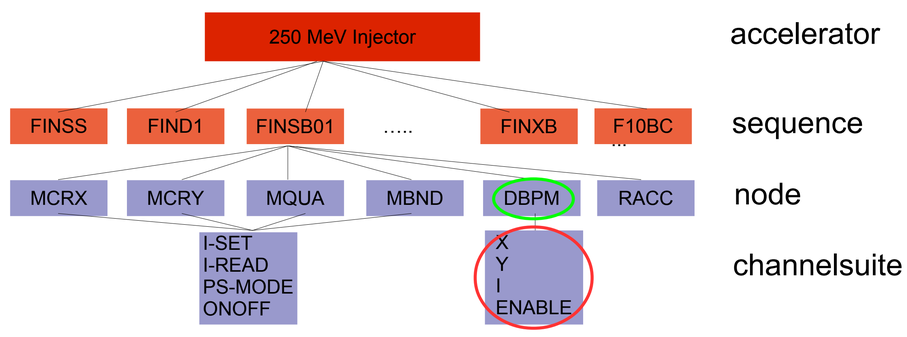
- Hierarchical data structures, *extensible:* handles new data in newer input format
- Supported by several categories of APIs: Stream oriented, e.g. SAX, Tree-traversal, e.g. DOM, XML data bindings, directly translates XML to language objects, XQuery/XPath for querying and traversing XML documents
- Portable across platforms and languages
- A de facto standard for data exchange and portability
- Widely accepted format in accelerator field:
  XAL application software package [1,2] provides an XML description of the Standard Machine Format (SMF) accelerator object model [3,4]
  Lattice files formatted in XML used in accelerator design and simulation [5]
- Applications easily adapted to changes in the accelerator system by simple modification to the configuration file; enhances portability of applications

XML CONFIGURATION

accelerator XML extracted from the Device Reference and Epics DBs



Hierarchical View of SwissFEL Test Injector XML Structure in SMF



XML-SMF (á la XAL) accelerator.xml

```
<sequence="FIND1">
<node type="DBPM" id="FIND1-DBPM10" pos="0.4370" len="0.0730">
   <properties>
     <info group="Diagnostic" description="DBPM" class="BPM"/>
   </properties>
   <channelsuite name="dbpmsuite">
        <channel handle="X"      signal="FIND1-DBPM10:X"      settable="false"/>
        <channel handle="Y"      signal="FIND1-DBPM10:Y"      settable="false"/>
        <channel handle="I"      signal="FIND1-DBPM10:I"      settable="false"/>
        <channel handle="ENABLE" signal="FIND1-DBPM10:ENABLE" settable="true"/>
   </channelsuite>
</node>
<node type="MCRX" ...> ... </node>
</sequence>
```

./xmlfilter *Node=DBPM* > CAFECollections.xml

Nodes of same type, e.g. DBPM are extracted from the accelerator XML to form collections that may be referenced in a control system interface as a single logical software entity.

```
<   <cafe:collection id="cDBPM">
      <cafe:description>Collection of Digital Beam Position Monitors for the 250 MeV SwissFel
                        Test Injector</cafe:description>
      <cafe:attributes>
          <cafe:attribute>X</cafe:attribute> <cafe:attribute>Y      </cafe:attribute>
          <cafe:attribute>I</cafe:attribute> <cafe:attribute>ENABLE</cafe:attribute>
       </cafe:attributes>
       <cafe:member pos="0.4370"> <cafe:device>FIND1-DBPM10  </cafe:device> </cafe:member>
       <cafe:member pos="7.7130"> <cafe:device>FIND100-DBPM10</cafe:device> </cafe:member>
       ......
       <cafe:member pos="64.1155"><cafe:device>F10D100-DBPM10</cafe:device> </cafe:member>
    </cafe:collection>
```

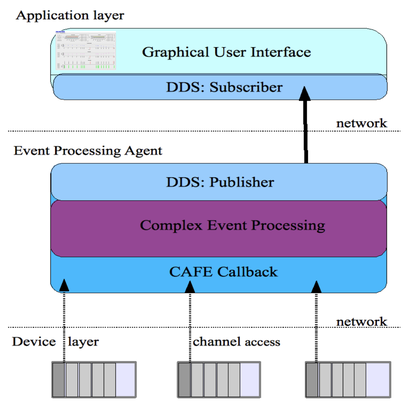
bpmDisplay.xml

CAFE configuration file defines a group of DBPM channels by reference to its collection identifier and attributes (X, Y, I, etc.).

```
<cafe:group id="gDBPM">
   <cafe:description>DBPM agent configuration</cafe:description>
      <cafe:collection> <cafe:id>cDBPM</cafe:id>
            <cafe:attribute>X</cafe:attribute><cafe:datatype>CAFE_DOUBLE</cafe:datatype>
         </cafe:collection>
         <cafe:collection> <cafe:id>cDBPM</cafe:id>
            <cafe:attribute>Y</cafe:attribute><cafe:datatype>CAFE_DOUBLE</cafe:datatype>
         </cafe:collection>
         <cafe:collection> <cafe:id>cDBPM</cafe:id>
            <cafe:attribute>I</cafe:attribute><cafe:datatype>CAFE_DOUBLE</cafe:datatype>
         </cafe:collection>
         …
</cafe:group>
```

A DYNAMIC BEAM DYNAMICS APPLICATION

An Event Driven Orbit Display Application Configured through XML

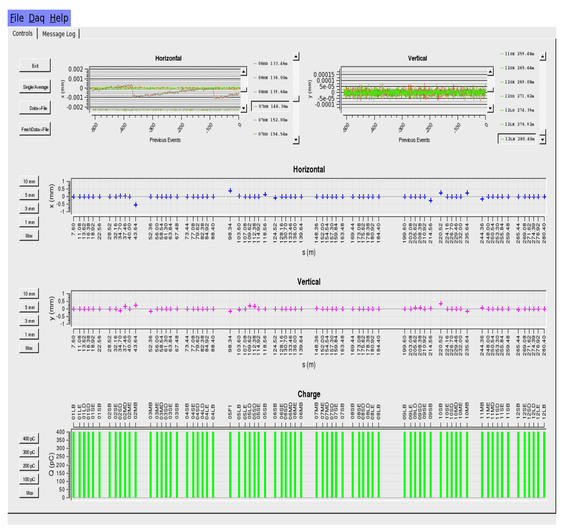


An Event Processing Agent (EPA) aggregates, verifies, analyzes low-level Digital Beam Position Monitor (DBPM) data and distributes summarized results through a Data Distribution Service (DDS) to the GUI. The EPA uses the CAFE API [6, 7], an in-house channel access library, to establish a callback mechanism to the EPICS control system.

EPAs are configured dynamically through XML:

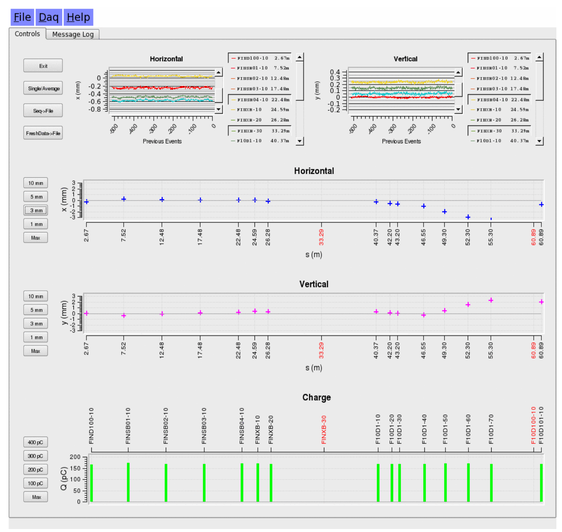
acccelerator XML provides information on the device type, i.e. number, names, positions of DBPMs.
CAFECollections XML groups related devices, e.g. DBPMs, into a single software entity. It is loaded by CAFE on initialization.
bpmDisplay XML is the CAFE configuration file for defining a group of DBPM channels, enabling synchronous/asynchronous interaction to EPICS through a single method invocation.

Example orbit displays from two implementations of identical code configured dynamically through their corresponding XML input files



Qt v. 4.6.3
Qwt v. 5.2.1

SLS Storage Ring



SwissFEL Test Injector (Phase 2)

A UNIVERSAL MACHINE FORMAT

The adoption of XML-SMF has allowed us to write applications that can easily adapt to changes in accelerator topology, an important consideration during the development phase of a facility such as the SwissFEL, and be equally applied to different accelerators at PSI. Our experience further leads us to recognize the benefits of an all-inclusive XML markup language to advance portability of certain standard beam dynamics applications. (For instance, additional `channelsuite` child elements could cater for local variations in definitions of a channel's state; node types may be mapped to customized classes.) This may be realized if communities of interest conscribe to using the same XML constructs. Such an XML vocabulary can be shared via a repository containing the appropriate elements of metadata, facilitating disclosure and usage. OpenXAL [8] which extends XML-SMF to other facilities, is a natural home base for discussion, and finalizing on a *Universal Machine Format (UMF).* Adapting standard machine applications to a particular accelerator may ultimately be reduced to selecting the required software packages and stacking configuration files with the adopted UMF constructs.

REFERENCES

[1] J. Galambos et al., “XAL Application Programming Framework”, ICALEPCS 2003, Gyeongju, Korea, 13-17 October 2003.

[2] A. Shishlo et al., “The XAL Infrastructure for High Level Control Room Applications”, ICAP 2009, San Francisco, California, USA, 31 Aug.-4 Sept. 2009.

[3] N. Malitsky et al., “A Proposed Flat Yet Hierarchical Accelerator Lattice Object Model”, Part. Accel. 55:313-327, 1996.

[4] N. Malitsky, T. Satogata, and R. Talman, “Configurable UAL-Based Modeling Engine for Online Accelerator Studes”, PAC 2003, Portland, Oregon, USA, 12-16 May 2003.

[5] D. Sagan et al., “The Accelerator Markup Language and the Universal Accelerator Parser”, EPAC 2006, Edinburgh, UK, 26-30 June 2006.

[6] J. Chrin and G. Prekas, “A Taste of CAFE”, ICALEPCS 2009, Kobe, Japan, 12-16 October 2009.

[7] J. Chrin and M.C. Sloan, “CAFE, A Modern C++ Interface to the EPICS Channel Access Client Library”, to be presented at ICALEPCS 2011, Grenoble, France, 10-14 October 2011

[8] OpenXAL, `http://xaldev.sourceforge.net.`

2nd International Particle Accelerator Conference, IPAC'11, 4-7 September, 2011, San Sebastián, Spain