Programming Interfaces for Reconfigurable Instruments
Matej Kenda, Hinko Kočevar, Tomaz Beltram, Aleš Bardorfer, Instrumentation Technologies d.d., Solkan, Slovenia

Abstract
Application Programming Interfaces (APIs) provide the manufacturer of the instrument for the accelerators as a very important part of the functionality. There are many interface standards (EPICS, TINE, TANGO) and even same standardized can be used in certain ways.

Important features of modern instruments are reconfigurability and embedded computing. The developers of instruments need to be connected to a control system, thus facing different requirements: adherence to standard protocol and support of reconfigurable instruments with diverse capabilities within a common interface.

Instrumentation Technologies has implemented a self-accepting solution with its proprietary Control System Programming Interface (CSP) type and adapter for each standard protocol. There are use challenges like reconfigurability, quality of service, discoverability and maintainability that are being addressed with improved Measurement and Control Interface (MCI).

Control System and Software Interfaces
The service that a Control System provides is defined on certain Interface requirements towards instruments that must cover following areas:

- Device discovery, identification and capabilities
- Operation mode control and configuration
- Events, alarms and health state monitoring
- Data acquisition and attributes (data type, size, offset, time-stamp)
- Error handling

From the reverse point of view, instrument can be used in different environments, e.g. for data coming from different sources for different purposes.

- Control System: Different types of control system protocols
- Other Instruments: Instrument interoperability, multiple instruments working together, clustering, shared processing
- Central System: Integration, testing of new, updated instruments
- Maintenance: Diagnostics, repair

Instrument Manufacturer’s View
Using embedded computers in the instruments enables instruments to behave as network attached devices with built-in control system interfaces.

- Embedded computer can be used to:
  - control the instrument’s operation
  - perform a part of digital signal processing
  - provide remote access to the instrument

The embedded computer is one of the important components of an instrument, because it provides convenient way to bring all of the parts (hardware modules, FPGA, software) of an instrument together into a working application.

- Increasing computing power (multi core, SIMD, GPU) in certain cases will be necessary to perform certain data processing which is usually done by specialized DSP processors and FGPA.

Reconfigurable Instruments
Physical setup and behavior of the instrument is not completely defined during manufacturing.

Properties of reconfigurable instruments:
- Reuse of modules: hardware module MOD_A can be used in instrument INS_A, INS_B, ...
- Behavior of the hardware module MOD_A can be altered by loading different FPGA design
- MODIFY_A can comprise variable number of modules MOD_B, MOD_C, thus defining different variations of the instrument

In general, the responsibilities of the instrument software can be split in several semi-independent layers:
- managing hardware platform
- instrument application logic
- external interfaces

Hardware flexibility influences all of the software layers, including external interfaces.

Control System Programming Interface

<table>
<thead>
<tr>
<th>Signals</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data access</td>
<td>Published, view (on demand)</td>
<td>Published by instrument (on request)</td>
</tr>
<tr>
<td>Size</td>
<td>Large</td>
<td>Small</td>
</tr>
<tr>
<td>Example</td>
<td>Times by Sum, ADC</td>
<td>Time acquisition, Fast acquisition events</td>
</tr>
</tbody>
</table>

Instrumentation Technologies develops families of specialized instruments for use in the accelerators. They are all equipped with embedded computers and have network connectivity.

Instruments can be divided into two classes: Platform A, Platform B. Main difference in hardware is the level of modularity, reconfigurability and computing power. Platform A instruments contain energy efficient ARM based embedded computer with limited computing power and provide control and signal acquisition through the API called Control System Programming Interface (CSP).

Platform B instruments are modular and reconfigurable (OTCA, OPH and other standards) and comprise a self-accepted computer. Access is provided through the Measurement and Control Interface (MCI).

The goal of both programming interfaces is similar: implementation of as much functionality as possible in a common fashion and connecting that information to a specific control system protocol as site as possible.

Both types of interfaces provide access to the semantic types of information described above.

Service Interfaces

<table>
<thead>
<tr>
<th>CSP</th>
<th>MCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network API</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of signals</td>
<td>8000</td>
</tr>
<tr>
<td>Processing</td>
<td>Control loops, on-access, on-update, on-access, on-update</td>
</tr>
<tr>
<td>Configuration</td>
<td>Registry, parameters, nodes, tree, group, signals, attributes</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Complete, passive, active</td>
</tr>
<tr>
<td>ELI</td>
<td>Yes</td>
</tr>
<tr>
<td>Control system</td>
<td>EPICS</td>
</tr>
<tr>
<td>REPS</td>
<td>TJ</td>
</tr>
<tr>
<td>XUI</td>
<td>EPICS, TUNE, Matlab, custom as</td>
</tr>
</tbody>
</table>

CSP

- Hardware configuration of Platform A instruments is defined at manufacturing. Available data and the API are coupled together.

MCI

- Provides interfaces for:
  - Monitoring, controlling the instrument through a number of parameters. They are all integer numbers and identified by number, like. The set of parameters is fixed for a certain instrument.
  - Acquisitions of the signals. Functions to easily access pre-defined number of signals are available.
  - Setting of values. Some of the function can be registered, which is called with the ID of the parameter that was modified.

Remote access

- Provided by:
  - Generic server, transparent CSP API access over the TCP/IP.
  - Embedded EPICS driver (Instrumentation Technologies and Diamond Light Source)

Remote access (External) Server (Orx)

MO:

- Dynamic nature of Platform B instrument required different design approach of the software and its API. MO has separated classes and functions of the API from the information that they are used to access. MO is networked by design.

The following concepts have been introduced in the API:

- Registry
  - Information is presented in a tree structure, individual nodes are identified by names (similar to directories and files) and have pre-defined data type
  - The tree is populated by the instrument software dynamically, depending on the hardware setup and the configuration of the instrument
  - Nodes values can be stored persistently in XML file, can have different flags (readable, writable, hidden, constants) that define access mode to data
  - Nodes can emit notifications (for example value change). Callback functions can be registered to nodes to receive notifications
  - Registry can be used by the instrument application software (local access) or remotely (network access)

- Data streams
  - Available data streams (signals) are enumerated in the registry and defined by instrument application software
  - Data stream classes provide access to different types of signals
  - The classes simplify processing of the signals (either in the embedded computer or by the client)

Remote access

- Provided by:
  - Directly by: EPICS, RFE
  - EPICS adapter lightweight server without a database maps MO registry and signals to EPICS Pvs

Examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPICS API supported for low level and instrument parameter communication. Use of low level EPICS API is necessary when instrument has no database.</td>
<td></td>
</tr>
<tr>
<td>RFE</td>
<td>Remote frame extender (for the Control System)</td>
</tr>
<tr>
<td>DSP</td>
<td>DSP interface between the controller and the instrument application software.</td>
</tr>
<tr>
<td>TANGO</td>
<td>TANGO database interface (for the Control System)</td>
</tr>
<tr>
<td>EMA</td>
<td>EMA database interface (for the Control System)</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>INSTRUMENT database interface (for the Control System)</td>
</tr>
<tr>
<td>LIBERA</td>
<td>LIBERA database interface (for the Control System)</td>
</tr>
</tbody>
</table>

See also:

- Control System Programming Interface (CSP)
- Measurement and Control Interface (MCI)
- EPICS, EPICS EDM, Matlab, custom as

Instrumentation Technologies, d.d., Veleja pot 42, 52050 Solkan, Slovenia. P: +386 5 335 26 01, F: +386 5 335 26 01, E: info@itech.si, sales@itech.si, support@itech.si, W: http://www.itech.si