

COMPLETE VENDOR-NEUTRAL INSTRUMENTATION CONFIGURATION WITH IHAL AND TMATS XML

John Hamilton, Timothy Darr, Ronald Fernandes
Knowledge Based Systems, Inc.

Joe Sulewski
L-3 Telemetry East

Charles Jones
812TSS/ENTI
Edwards AFB, CA

ABSTRACT

Previously, we have presented an approach to achieving standards-based multi-vendor hardware configuration using the Instrumentation Hardware Abstraction Language (IHAL) and an associated Application Programming Interface (API) specification. In this paper, we extend this approach to include support for configuring PCM formats. This capability is an appropriate fit for IHAL since changes to hardware settings can affect the current telemetry format and vice versa. We describe extensions made to the IHAL API in order to support this capability. Additionally, we show how complete instrumentation configurations can be described using an integrated IHAL and TMATS XML. Finally, we describe a demonstration of this capability implemented for data acquisition hardware produced by L-3 Telemetry East.

KEYWORDS

IHAL, TMATS XML, PCM, Instrumentation configuration, metadata, web services, XML, API

INTRODUCTION

Current Instrumentation Support Systems (ISSs) are required to use vendor-specific languages in order to support configuring instrumentation systems prior to testing. Since there are a number of vendors selling various data acquisition, control, transmission and storage components, an ISS not only has to interface with these different systems, but also different vendor software that supports various instrumentation systems. Moreover, hardware vendors typically supply their own configuration software that works only with their own products. Thus, an instrumentation engineer who is trained on one vendor's software must learn a new software system in order to use a different vendor's hardware, raising the cost of switching vendors, reducing competition, and making interoperability almost impossible.

In 2006 [1] and 2008 [2], we introduced the Instrumentation Hardware Abstraction Language (IHAL) and subsequent extensions. IHAL is an XML-based language that provides a means to describe instrumentation hardware in a vendor neutral manner. In addition to basic physical descriptions, IHAL can be used to describe the functions each device performs, the settings available on each device, and the valid values for each setting. The focus of the language is on describing settings both at the capabilities level (i.e. how the device can be configured) and at the instance level (i.e. how is the device currently configured).

In 2010, we introduced an application programming interface (API) and methodology for interacting with vendor hardware configuration engines with IHAL. Additionally, we demonstrated the effectiveness of IHAL as a vendor-neutral hardware configuration language by implementing the API and methodology for data acquisition hardware produced by L-3 Telemetry East.

In this paper, we extend this approach to include support for interactively configuring PCM formats with immediate feedback from the vendor's configuration engine. This capability is an appropriate fit for IHAL since hardware capabilities can affect the rules by which telemetry formats may be configured. We describe extensions made to the IHAL API in order to support this capability. Additionally, we show how complete instrumentation configurations can be described using an integrated IHAL and TMATS XML. Finally, we describe a demonstration of this capability implemented for data acquisition hardware produced by L-3 Telemetry East.

INCORPORATING TMATS XML IN IHAL

IHAL is designed to describe hardware settings and their current values, while TMATS is best at describing PCM formats. In order to configure a data acquisition system to acquire measurements and transmit them to the ground station via a PCM stream, we must be capable of describing both. Towards this end, we determined that two key concepts from TMATS are needed in IHAL:

1. PCM Format Attributes: attributes that describe the PCM format, such as parity, sync patterns, word length, and other attributes often referred to as “frame metrics.”
2. Measurement placement: description of which measurements exist in a PCM format, and where in the frame they are placed

Due to the nature of IHAL and TMATS XML, we took two different approaches for incorporating these two concepts in IHAL.

PCM Format Attributes

As described in [1] and [2], IHAL descriptions are divided into two major types: “pool” descriptions and “use” descriptions. Pool descriptions specify the capabilities and configurability of a particular device. Use descriptions specify the specific configuration of a device at a given moment. For example, a pool description of a signal conditioner might specify that the gain can be set to values in the range 1 – 10, while a use description of the signal conditioner might specify that the gain is currently set to 6.

TMATS XML is designed to describe PCM formats at the “use” level. That is, a TMATS XML description can say that the PCM code of a particular format is currently set to NRZ-L, but it cannot say that the PCM encoder that generates the format also supports NRZ-S.

Since IHAL is optimized for these types of configurable attribute descriptions, we added new constructs to IHAL to describe both the configurability of PCM Format Attributes (“pool” level) as well as the current configuration of PCM Format attributes (“use” level).

Following the existing IHAL approach of a device being composed of channels, which are composed of functions, which are themselves composed of sub-functions and parameters, we created a "PCMEncoderFormat" element that is structured similar to IHAL’s existing “hardwareChannel” concept. This new element serves as the container for a new “PCMEncoderFunction” element, and is available as a child of both the IHAL DAU device and the IHAL pcmEncoderCard device. The PCMEncoderFormat element is shown as a child of the “dau” element in the schema diagram in Figure 1.

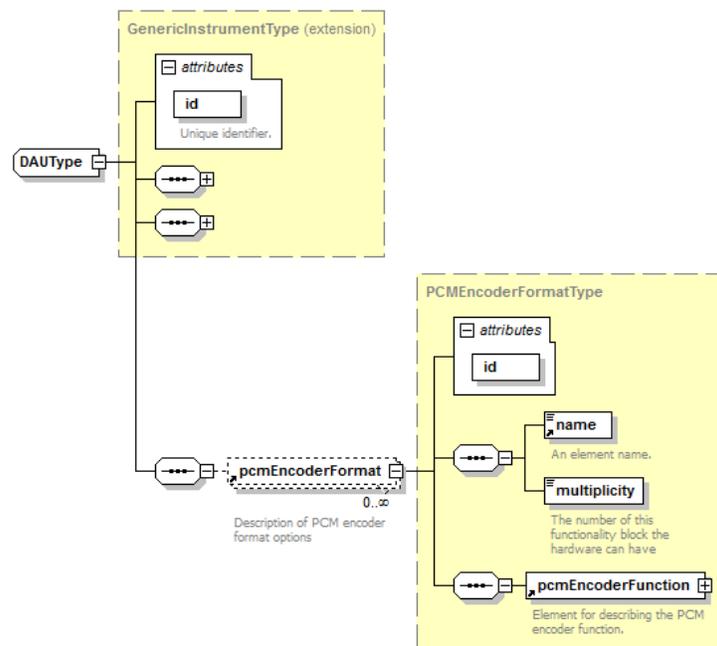


Figure 1: pcmEncoderFormat Element Added to the Pool-level DAU Description

The new PCMEncoderFunction in IHAL is composed of one sub-function (subFrameSynchronizationFunction), and the following attributes:

pcmCode	bitRate	polarity	commonWordLength
wordTransferOrder	parity	parityTransferOrder	numberOfMinorFrames
wordsPerMinorFrame	bitsPerMinorFrame	minorFrameSyncType	minorFrameSyncPattern

Each of these attributes is modeled as an IHAL “possiblyConfigurableParameter”, meaning that the descriptions of each attribute may contain a range or list of valid values allowed by the PCM encoder.

In order to support configuring the new PCM Encoder Function, we added a “formatParameterSettings” element to the instrument “use” structure in IHAL. This new element is composed of one or more “formatUse” elements that reference a particular format description in the pool and also specify a format number (for devices that support multiple formats). Each “formatUse” element is then composed of both “setParameter” and “restrictedParameter” elements which can be used to set each of the PCM format attributes to specific values, or to further restrict the range of valid values following the same approach described in [3]. A schema diagram of the new use-level structure is shown in Figure 2.

PCM Measurement Placement

IHAL already contained basic structures for describing hardware attributes and settings, so it was a fairly simple task to incorporate new attributes and settings associated with PCM Encoders. However, the same is not true for describing the contents of a PCM Format. Since the TMATS XML schema already includes structures for describing PCM Format contents, we decided to directly include a portion of the TMATS XML schema into the IHAL schema using an XML Schema “import” statement.

In order to import only the portion of the TMATS XML schema that we needed, it was necessary to make some structural changes to the TMATS XML schema. These changes included declaring “global” elements to make them accessible from outside the schema, defining namespaces to avoid collisions with similarly-named elements in IHAL, and splitting the schema into multiple sub-schemas to avoid importing more schema elements than we need. These changes follow the XML schema design considerations described in [4], and are currently being considered for inclusion in the next release of TMATS XML.

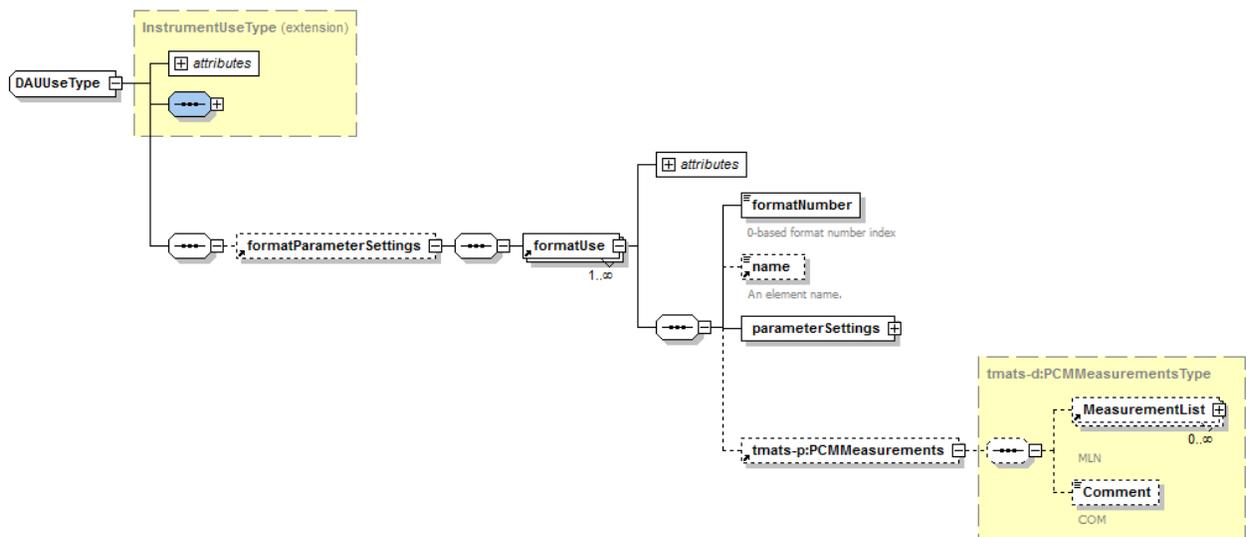


Figure 2: Inclusion of TMATS XML

In our IHAL “Instrument Use” schema, we import the TMATS XML “P” sub-schema and include the TMATS XML p:PCMMeasurements element as a child of the IHAL formatUse element described above. This enables a complete TMATS description of the contents of a PCM format to be included directly as a child of the IHAL description of the format. A schema diagram of this merging of the two schemas is shown in Figure 2.

Finally, we needed a simple structure in IHAL to describe the list of measurements available for encoding in a specific format. To this end, we added an “availableMeasurements” element at the top-level of the IHAL schema. This element, shown in Figure 3, is not included as part of an IHAL instrumentation description. Instead, it is used only in the return value of certain API calls, as described in the next section. The purpose of this element (and the corresponding API function) is to allow the vendor configuration engine to communicate to the Instrumentation Support System a list of measurement names that are available for placing into a PCM format.

The “availableMeasurements” element is composed of one or more “measurementRef” elements. The “measurementRef” element must contain a string that uniquely identifies a measurement in the vendor’s configuration engine. Currently, there are no restrictions in IHAL on the contents of the “measurementRef” element. However, in the future this element could be expanded by importing part of the measurement description in iNet’s Metadata Description Language (MDL) schema.

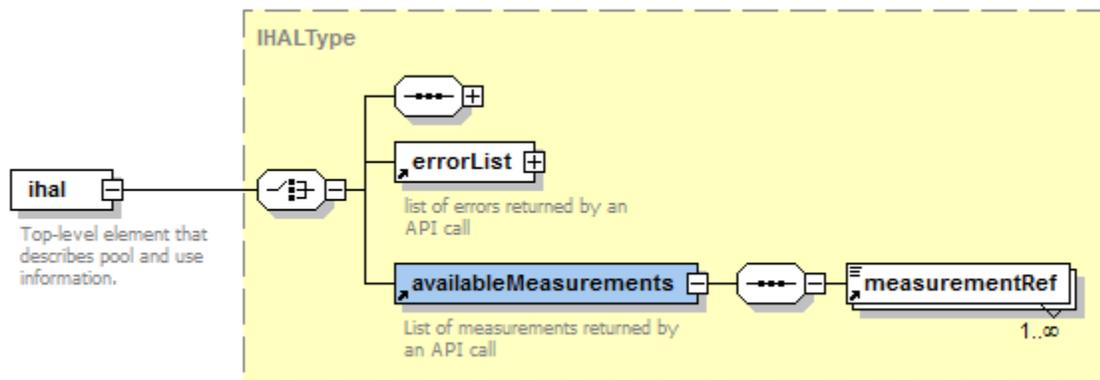


Figure 3: ‘AvailableMeasurements’ Element

ADDITIONS TO THE IHAL API

In [3], we describe five functions defined in the IHAL Application Programming Interface (API) Specification:

1. Retrieve vendor’s hardware “pool”
2. Create a new configuration in the vendor’s system
3. Retrieve the current configuration from the vendor’s system
4. Modify the current configuration
5. “Program” hardware

To support configuration of PCM encoders, we added four additional API calls for hardware configuration management. These API calls are described in detail below.

Retrieve list of measurements available to a PCM encoder (DAU, etc.)

Called before adding a measurement to a format, this API call returns a list of measurements that can be legally added to a DAU's PCM format. This function doesn't take any parameters as input, and outputs an IHAL "availableMeasurements" element. As mentioned in the previous section, IHAL places no restrictions on how these measurement names are represented. It is only required that the vendor configuration engine understand them, and that each measurement name be unique. Eventually, this part of IHAL could be modified to make use of existing measurement description standards.

The URL for accessing this function is:

http://<host>:<port>/formats/availableMeasurements/<deviceUse id>

where <deviceUse id> is the id of the dauUse or other instrumentUse element which is being queried.

Add a new PCM Format to a PCM encoder (DAU, etc)

This function can be used to add an empty format to a PCM Encoder, or to add a fully-specified format to a PCM Encoder. Either way, the resulting format will be added to the DAU as a new format. This function accepts a single parameter named "Ihal", which must contain a partial or complete IHAL description of the format, rooted at the <ihal:formatUse> element. The return value of this function is the "impact": An <ihal:dauUse> (or other instrument "use") element containing everything that has changed as a result of the call, including all format descriptions, all setParameters, and all restrictedParameters.

The URL for accessing this function is:

http://<host>:<port>/formats/<dUse_id>/new

where <deviceUse id> is the id of the dauUse or other instrumentUse element which is being queried.

Add measurements to a PCM Format

This function adds measurements to an existing PCM Format on a DAU. The placement of the measurements is specified using TMATS XML. The function accepts a single input parameter named "tmats", which must contain a complete TMATS XML <d:MeasurementList> element that describes the measurements and fragments to be added to the format. The output of this function is the "impact": An <ihal:formatParameterSettings> element containing everything that changed as a result of the call, including all format descriptions, all setParameters, all restrictedParameters and all measurements that have been added to the format. Due to the rules imposed by different versions of hardware, adding a measurement may result in many measurements being added depending on the common word size of the frame and the size of the measurement. In addition, hardware systems may impose limitations on the spacing of fragments.

The URL for accessing this function is:

```
http://<host>:<port>/formats/<deviceUse_id>/<format_id>/<format_#>  
  
/measurements/new
```

where <deviceUse id> is the id of the dauUse or other instrumentUse element which is being queried, <format id> is the pool-level id of the DAU's format, and <format #> is the number of the format to which measurements are being added.

Delete a measurement from a PCM format

This function removes a single measurement from all locations of an existing PCM format on a PCM encoder. This function accepts no input parameters, and returns the "impact": An <ihal:formatParameterSettings> element containing everything that changed as a result of the call, including all setParameters, and all restrictedParameters. Additionally, the <d:MeasurementList> portion will include only those measurements, fragments, and locations which have been removed.

The URL for accessing this function is:

```
http://<host>:<port>/formats/<deviceUse_id>/<format_id>/<format_#>  
  
/measurements/remove/<measurement_name>
```

where <deviceUse id> is the id of the dauUse or other instrumentUse element which is being queried, <format id> is the pool-level id of the DAU's format, <format #> is the number of the format to which measurements are being removed, and <measurement_name> is the unique name of the measurement to be removed.

DEMO WITH INSTRUMENTMAP™ AND VISTA TEC™

To validate the changes to the API and IHAL, we extended the proof-of-concept demonstration described in [3], which uses KBSI's InstrumentMap™ application as the Instrumentation Support System (ISS) and L-3 Communications' VistaTEC™ application as the vendor configuration system. For the demonstration, L-3 implemented the new API functions in the VistaTEC™ software, which is the program currently used to configure L-3 hardware. KBSI added support for calling the new API functions and for parsing the new language constructs to the InstrumentMap™ application. This enabled InstrumentMap to create a complete valid configuration of L-3 hardware, including setting up PCM formats, using only valid IHAL and TMATS XML as the representation language.

Using the same hardware and configuration as the previous demo ([3]), we demonstrated five additional scenarios:

1. Receiving the current PCM formats from the vendor in TMATS XML.
2. Configuring the PCM frame metrics in the vendor's system with IHAL.

3. Adding measurements to a PCM format using IHAL and TMATS XML.
4. Removing measurements from a PCM format using IHAL and TMATS XML.
5. Utilizing a lightweight client with the IHAL API

These five scenarios are described in detail in the following sections.

Receiving the Current PCM Formats from the Vendor in TMATS XML

In this scenario, the InstrumentMap™ application issues a request to the vendor’s API implementation for the current configuration. As in previous demonstrations, the vendor responds with a complete IHAL description of the current hardware configured in the vendor’s system. However, the use-level description returned by the vendor also includes a complete description of all of the PCM formats configured in the DAUs and PCM encoders in the system. These PCM format descriptions are included in the returned IHAL document and represented using the new merged IHAL and TMATS XML schemas.

Configuring the PCM Format Attributes in the Vendor’s System with IHAL

This scenario uses the same API functions defined previously for configuring hardware settings. However, we are now able to use the new IHAL constructs to configure PCM format attributes such as ‘number of minor frames,’ ‘words per minor frame,’ and ‘PCM code,’ among others. Just as in the previous demo, the response from the vendor includes the “impact” of making these changes on other hardware settings. Figure 4 shows the dialog in InstrumentMap™ for configuring format attributes. In this dialog, the user can select an attribute from the list, and then view or change the current value for that attribute.

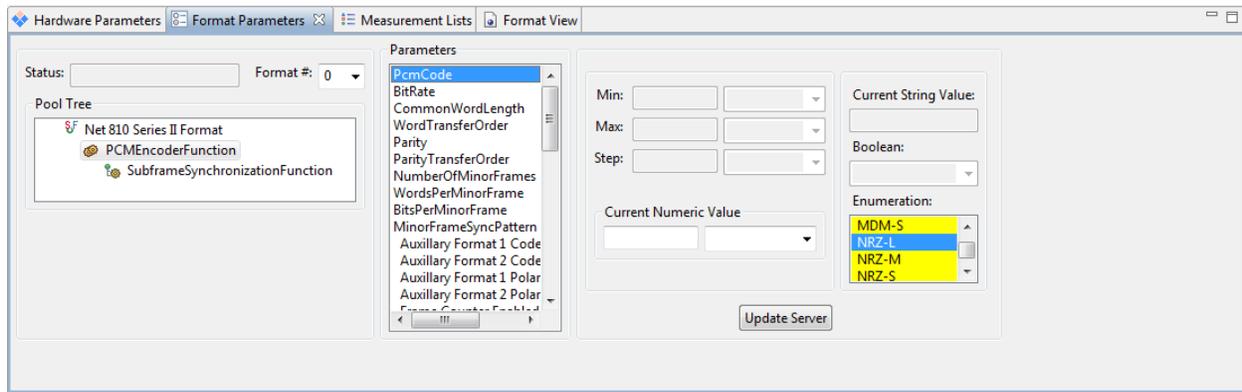


Figure 4: Configuration of New IHAL Format Attributes in InstrumentMap™

Adding Measurements to a PCM format using IHAL and TMATS XML

The third scenario utilizes the new API functions along with the merged IHAL and TMATS XML schemas to add measurements to a PCM Encoder’s (or DAU’s) PCM format. The measurement and its locations in the format are described using TMATS XML, and the vendor once again responds with the impact of adding this measurement, or an error message if adding the measurement is not possible. In order to provide the user with a list of measurements to add, the InstrumentMap™ application first makes a call to the API to retrieve a list of measurements that are available to the selected DAU or PCM Encoder. Figure 5 shows the InstrumentMap™

dialog for editing measurements in a PCM format. The measurements are displayed in a list, and when the user selects a measurement the current values of the TMATS attributes for measurement placement are displayed.

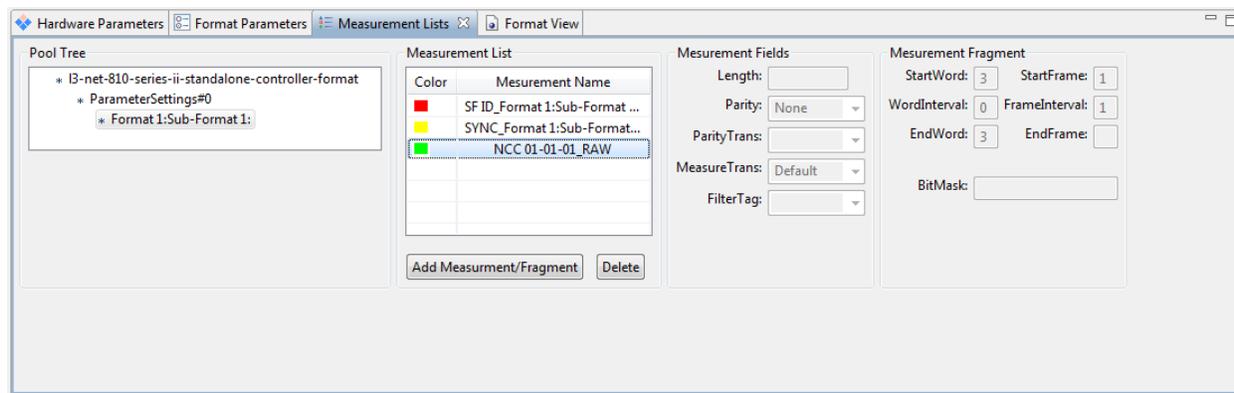


Figure 5: Editing Measurement Lists in InstrumentMap

Removing Measurements from a PCM Format using IHAL and TMATS XML

The fourth scenario for the demonstration involves removing existing measurements from a PCM format. In this scenario, the InstrumentMap™ application uses one of the new API calls to send the delete command, along with the measurement name and format identification, to the vendor’s API implementation. The vendor removes all instances of the given measurement from the specified format and responds with the impact, or an error message.

Utilizing a Lightweight Client with the IHAL API

The fifth and final scenario illustrates the effectiveness of a lightweight API designed to treat the vendor’s configuration rules as a “black box.” For this scenario, we developed a demonstration utilizing Microsoft® Excel® as the ISS with L3’s VistaTEC configuration engine. The purpose of the demonstration is to show how IHAL’s API and XML language can increase the utility of an application with which most users are familiar.

Many users use Excel to describe frame maps. However, with Excel alone users have no knowledge as to the validity of the frame map until they export the data and import it into the vendor’s tool. To make using Excel more effective, we developed Visual Basic macros to utilize the PCM modeling language constructs and API functions in IHAL to provide the users with immediate feedback about the validity of their frame maps. This demonstration allows the user to populate a frame map of any size by specifying values for TMATS frame location concepts such as “start,” “end” and “interval” for frame and word locations.

Once the information is entered in the spreadsheet, the user submits the data by clicking a button. At this time, the information is collected as TMATS XML data within the IHAL document, and the appropriate IHAL API function is called. The configuration engine parses the data and returns the impact and validation errors to the Excel macro. The Excel macro then parses the returned XML and displays either a message to indicate success or a list of errors to the user.

The key part of this demonstration is that the user can enter the information and get immediate validation feedback from the vendor's configuration engine. This immediate feedback provides a much faster turnaround time for the user by avoiding the traditional routine of exporting, importing, checking for errors, and trying again.

BENEFITS

Building a frame map is a well-understood exercise. However, building a frame map while respecting the limitations imposed by a vendor's hardware is a much more complicated task. These limitations vary not only between vendors but even between a single vendor's different product lines. The simple methods described in this paper coupled with standard IHAL XML descriptions enable third-party applications such as InstrumentMap™ to create a frame that respects the rules and restrictions of the underlying hardware. Examples of these restrictions include sampling rate restrictions, location restrictions of fragmented words, and collision detection (measurements trying to use the same location in the frame). More importantly, the API allows users to see the impacts and learn of the changes or violations immediately.

CONCLUSIONS

The existing IHAL language and API provide a vendor-neutral way to configure all hardware settings in an instrumentation system regardless of the vendor or the vendor's product line. However, no instrumentation configuration is complete without also setting up the output telemetry formats. We have modified the IHAL API and made use of the existing TMATS XML standard to enable configuration of both hardware settings and PCM telemetry formats in a vendor-neutral way from a single client application. We have validated our approach by developing a proof-of-concept demonstration using KBSI's InstrumentMap™ software and L-3 Communication's VistaTEC™ on a small hardware sample.

REFERENCES

- [1] Hamilton, Fernandes, Koola, and Jones, *An Instrumentation Hardware Abstraction Language*, Proc. International Telemetry Conf., Vol. XLII, (2006) Paper 06-10-02, San Diego, CA.
- [2] Hamilton, Fernandes, Graul, Darr, and Jones, *Extensions to the Instrumentation Hardware Abstraction Language (IHAL)*, Proc. International Telemetry Conf., Vol. XLIV, (2008) Paper 08-19-04, San Diego, CA.
- [3] Hamilton, Darr, Fernandes, Jones, and Sulewski, *IHAL and Web Service Interfaces to Vendor Configuration Engines*, Proc. International Telemetry Conf., Vol. XLVI (2010), Paper 10-08-01, San Diego, CA.
- [4] Darr, Hamilton, Fernandes, and Jones, *Design Considerations for XML-based T&E Standards*, Submitted to the International Telemetry Conf. (2011), Las Vegas, NV.