VERIFICATION, VALIDATION AND COMPLETENESS SUPPORT FOR METADATA TRACEABILITY

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

Charles Jones
812TSS/ENTI
Edwards AFB, CA

ABSTRACT

The complexity of modern test and evaluation (T&E) processes has resulted in an explosion of the quantity and diversity of metadata used to describe end-to-end T&E processes. Ideally, it would be possible to integrate metadata in such a way that disparate systems can seamlessly access the metadata and easily interoperate with other systems. Unfortunately, there are several barriers to achieving this goal: metadata is often designed for use with specific tools or specific purposes; metadata exists in a variety of formats (legacy, non-legacy, structured and unstructured metadata); and the same information is represented in multiple ways across different metadata formats.

KEYWORDS

T&E metadata, validation, verification, inference, RDF/OWL, SPARQL, metadata traceability, IHAL, MDL, TMATS

INTRODUCTION

In its simplest definition, metadata is data about other data. For the purposes of this paper, we view the "T&E data" to be the measurements obtained during a test. Hence, "T&E metadata" is any information that provides additional description or context to the T&E data. This covers a broad spectrum of information, ranging from the initial requirements and motivation for the test, to the test article and instrumentation modifications required to perform the test, to the description of the packet format in which the data is transported.

Metadata traceability refers to the process of identifying the relationship between the initial requirements (i.e., measurements, test plans, test points, etc.) and the various sets of metadata that depend on these requirements. Metadata traceability can be used for forensics (identify missing or erroneous, pieces of metadata) and synthesis (constructing metadata). The requirements for a measurement defined by one metadata source can constrain information in other metadata sources. For example, the requirement for a temperature measurement in one metadata source constrains some temperature sensor description in a hardware description metadata source, which implies the existence of a temperature measurement in a PCM description metadata source. From a description of the requirements of a measurement in one piece of metadata, a placeholder for the sensor that collects that measurement and a placeholder for the PCM format definition for that measurement can be auto-generated or synthesized in separate pieces of metadata.

As another example, the "test points" defined in the initial test plan describe the maneuvers the pilot is supposed to perform. After the flight is over, there should be a record, such as a flight log, that indicates what maneuvers where actually flown by the pilot. Obviously, the test points and maneuvers that the pilot actually flew need to match, or at least be within a specified tolerance (it may be acceptable, for example, for the altitude to be off by a thousand feet).

This paper presents an approach to metadata traceability that includes the following: 1) a framework for transforming metadata (legacy, non-legacy, structured and unstructured) into a common representation; 2) example rules for Verification, Validation and Completeness (VVC) of metadata; and 3) the use of standards to promote interoperability. Our approach leverages existing technologies and standards to shift focus from developing enabling technologies to developing rules for metadata VVC traceability.

The approach in this paper has been used to implement a variety of intra- and inter-metadata rules. Specific examples include *inter-metadata element* verification and validation (V & V) of XML ID / IDREF pairs within an XML instance file, and *intra-metadata element* V & V among metadata elements such as iNET Metadata Description Language (MDL) [1], the Integrated Hardware Abstraction Language (IHAL) [2,3], and the TeleMetry Attributes standard (TMATS) [4].

METADATA TRACEABILITY

In this paper, the objective of *metadata traceability* is twofold: (i) to trace all the relevant metadata for a test from a source metadata element to a dependent metadata element, and (ii) to determine where each metadata element exists in a Word document, XML file, database or other source. For example, an instrumentation engineer might be interested in where the description of a signal conditioning card and the requirements of the measurement(s) conditioned by that card are found (answer: in the `signalConditionerUse` element of an IHAL file and the `Requirements` element of an iNET MDL file, respectively) and a test engineer might need to know where to find information about what objectives are met by a given test point (answer: in documents stored in a document management system).

Metadata Traceability Motivation

It is currently difficult to trace metadata elements because the information is distributed across so many different types of metadata (Word documents, spreadsheets, XML files, databases, etc.). Even when disparate metadata is stored in the same format, it is difficult to trace metadata elements through the various metadata formats. Figure 1 and Figure 2 illustrate this for traceability of measurement information.

Figure 1 illustrates the traceability of the requirements of a measurement from an MDL file to a TMATS file and an IHAL file. The MDL file specifies a range on the required resolution of a pressure measurement. This measurement is realized by some signal conditioning card whose resolution meets the requirements. Additionally, a TMATS file contains a definition of the same measurement for the purpose of defining the location in a PCM frame for that measurement.

Figure 2 illustrates the realization of a measurement requirement in each of the three metadata types. The MDL file defines the source node in an iNET network that conditions the measurement. The TMATS file defines the location of a telemetered measurement in a PCM frame. The IHAL file defines an instance (or use) element that defines the card that collects the measurement. The source node in the MDL file must match with an IHAL use element, which must also match with the measurement definition in the TMATS file, via an identifier (ID) / identifier reference (IDREF) pair.

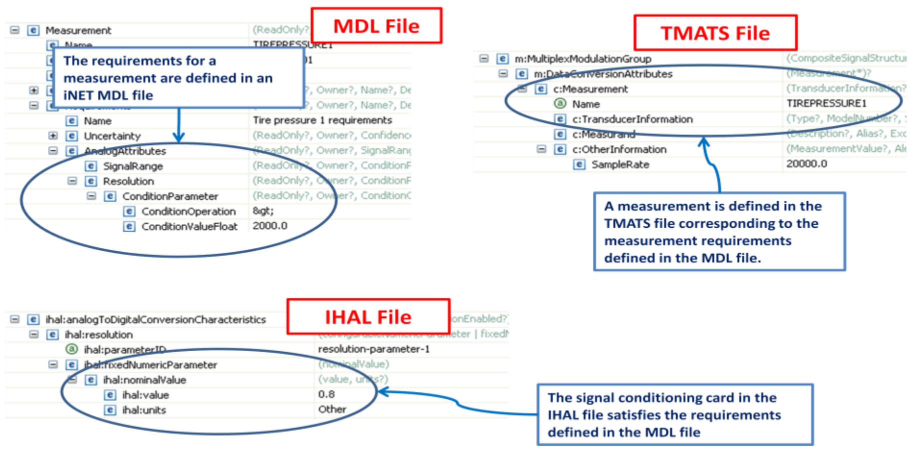

Figure 1 - Metadata Traceability Motivation

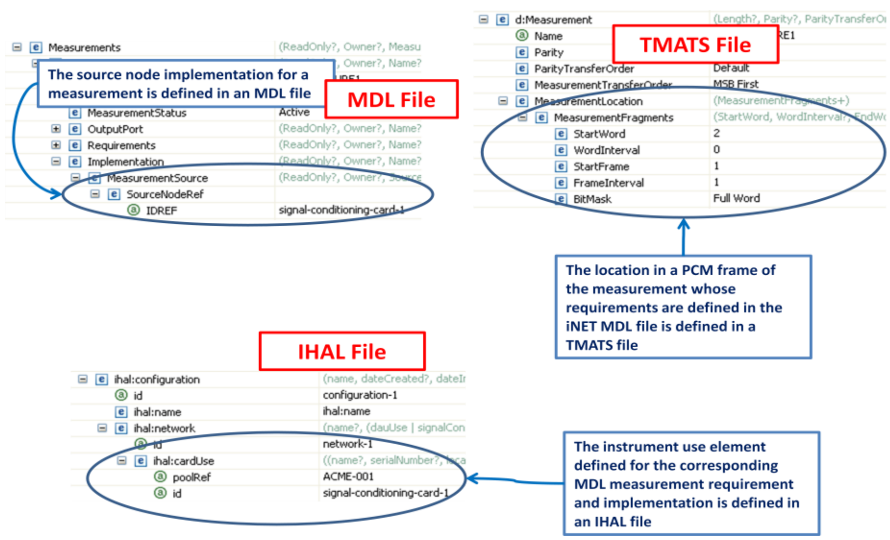

Figure 2 - TMATS Instance Data for Traceability Example

Metadata Traceability Solution

Figure 3 shows the general solution for metadata traceability. Given a collection of metadata that describes a test (MDL, TMATS, IHAL, Word document, etc), a set of rules are defined that trace the metadata element through each of the relevant metadata sources to collection information about the metadata element, or identify missing or incorrect information.

Figure 4 shows the system-level approach to metadata traceability. The source metadata elements (XML files, Word documents, Excel spreadsheets, etc) are converted to an *RDF/OWL* representation using a custom-developed or commercial off-the-shelf (COTS) translator. The Resource Description Framework (RDF) [5] and the web ontology language (OWL) [6] are standards, much like eXtensible

Markup Language (XML), that represent meaning in a programming language and database independent way. Using existing standards such as RDF / OWL does not require the introduction of new metadata formats. By the choice of RDF / OWL as the common representation format, we make available a wide variety of translators, inference engines, APIs and other tools that are supported by the RDF / OWL community. For example, KBSI has developed techniques for translating text documents into RDF / OWL format [7] and TopQuadrant's TopBraidComposer product line can translate XML files into an RDF / OWL format [8].

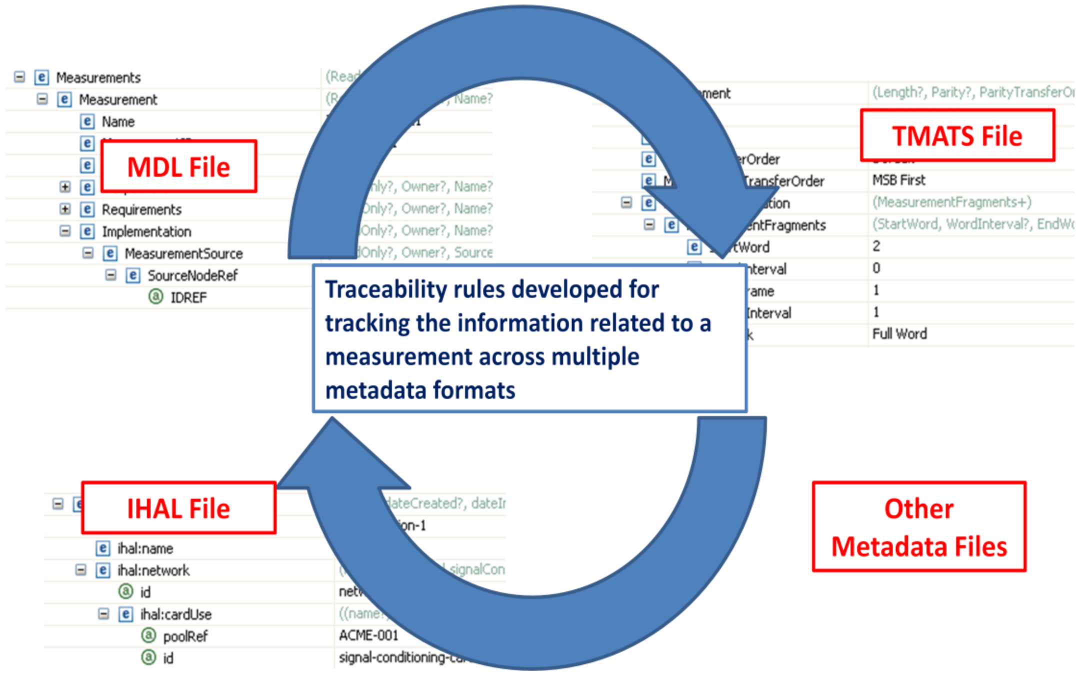

Figure 3 - Metadata Traceability Solution

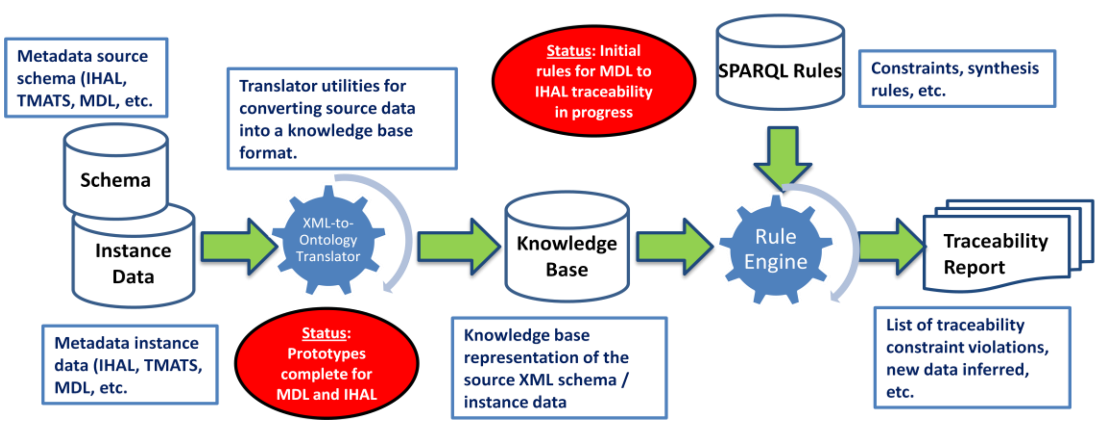

Figure 4 - Metadata Traceability Approach

RDF supports representation of graph data structures in the form of RDF triples; OWL, an extension of RDF, supports the definitions of concepts and relationships among concepts in such a way that applications can reason about meaning or semantics. *Rule-based inference* is a reasoning technique that allows for the creation of new relationships and links between concepts in a domain. Simple Protocol and RDF Query Language (SPARQL) [9] is a query language for the RDF triples. RDF / OWL representations and inference methods such as SPARQL provide a powerful way for accessing and manipulating rich, network-based data.

Once the metadata is converted to RDF/OWL format, SPARQL metadata VVC traceability rules are applied to the RDF/OWL metadata representation to generate a traceability report. This report includes some (or all) of the following:

- The result of a traceability query (e.g., "provide all information and its metadata source for the left front tire pressure"),
- A validation report (e.g., measurements whose requirements are not satisfied)
- A verification report (e.g., measurements that do not meet conditions specified by the test objectives)
- A completeness report (i.e., missing metadata elements)

Figure 5 shows a snippet of an RDF / OWL file translated from an MDL instance document. The topmost node[1] is a measurement node (type **`MDL:Measurement`**[2]) that represents a measurement collected by an instrumentation network. The measurement node has relationships to nodes that describe the implementation of the measurement (the property **`MDL:hasImplementation`** to nodes of type **`MDL:Implementation`**) and the requirements for collecting that measurement (the property **`MDL:hasRequirements`** to nodes of type **`MDL:Requirements`**).

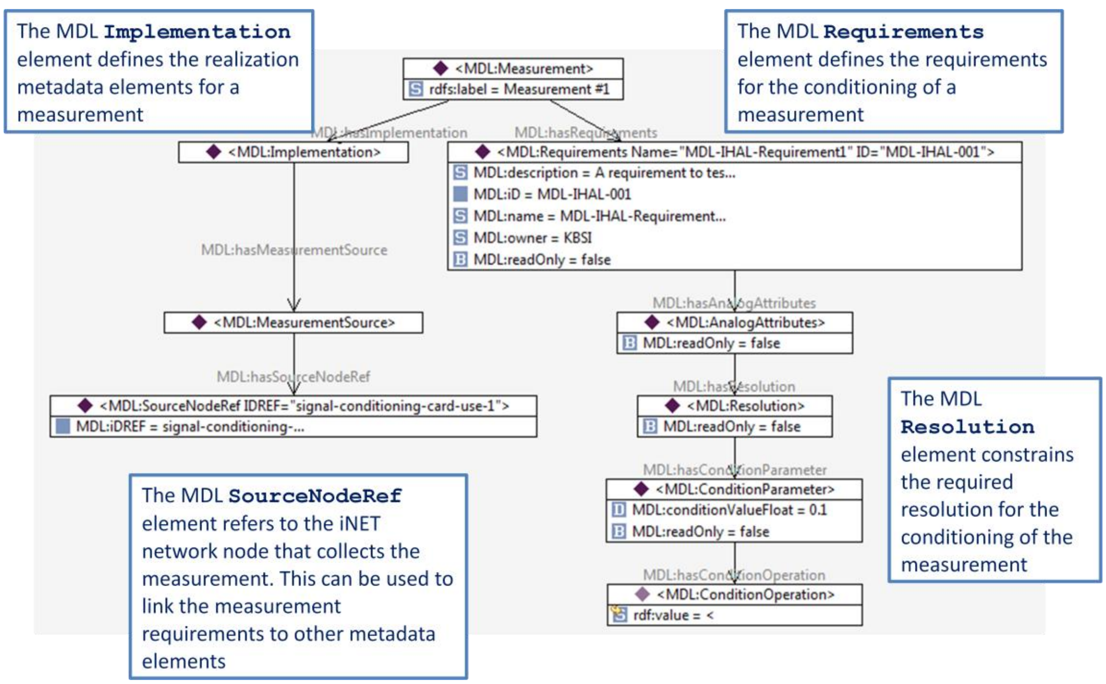

Figure 5 - RDF / OWL MDL Representation

---

1 In RDF / OWL terminology a node is used in the graph-theoretic sense in which nodes are connected to one another using properties. This is not to be confused with the MDL concept of a node, which is a piece of hardware on an IP-based network.

2 The syntax **`namespace:resource-name`** is used to describe RDF / OWL classes and properties. The **`namespace`** is used to segment metadata sources (MDL, IHAL, TMATS, etc.) and the **`resource-name`** is an RDF / OWL class or property for the given metadata elements.

The MDL implementation node has a relationship to a node that represents the measurement source (the property **`MDL:hasMeasurementSource`** to nodes of type **`MDL:MeasurementSource`**), which in turn has a relationship to a node that represents a reference to the measurement source (the property **`MDL:hasSourceNodeRef`** to nodes of type **`MDL:SourceNodeRef`**). The measurement source is fully defined somewhere else in the MDL XML instance document (typically, this is a PIN on a signal conditioning card). For the purpose of metadata traceability, the MDL source node reference can be used to link similar or identical concepts in other metadata sources, as will be illustrated in the following sections.

The MDL measurement node also has relationships to nodes that define the requirements for the measurement. This is shown on the right-hand side of Figure 5. The details of these relationships will not be described here, the key point being that the requirements for the resolution of the conditioned measurement are defined in this portion of the MDL RDF / OWL graph.

Figure 6 shows a snippet of an RDF / OWL file translated from an IHAL instance document. The left-hand side of the figure shows an IHAL card use node, which defines an instance of a signal conditioning card in an IHAL instrumentation network. The right-hand side of the figure shows an IHAL signal conditioning card node, which defines the characteristics of a signal conditioning card. These characteristics include the signal conditioning card analog-to-digital conversion resolution. The card use and pool elements in the XML are linked via an ID / IDREF pair: the use refers to the corresponding pool element by a reference to the pool element's unique ID.

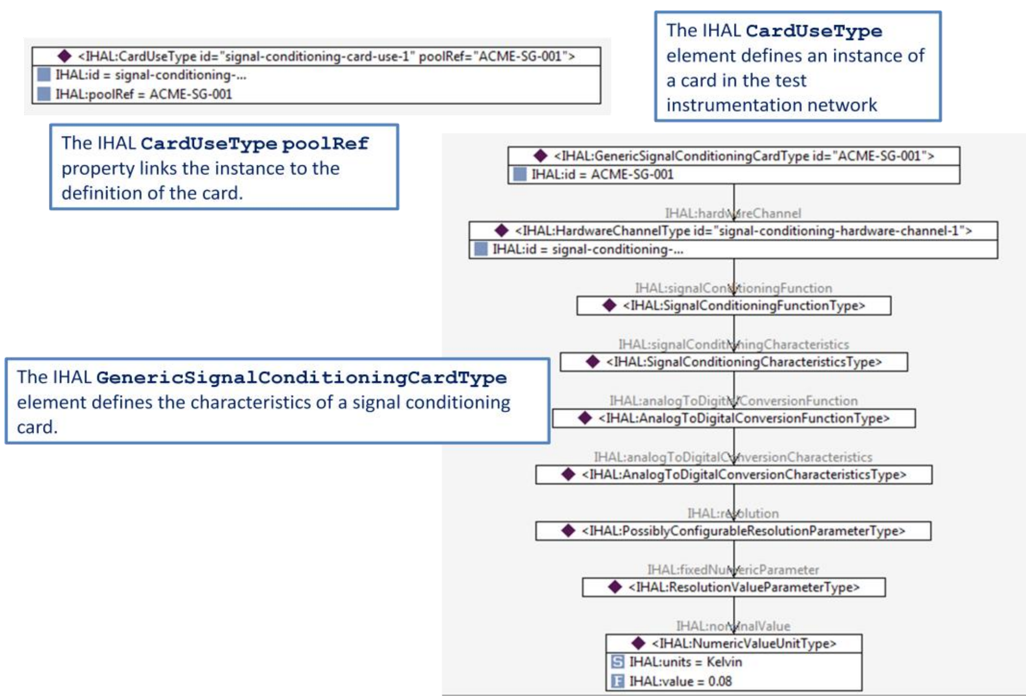

Figure 6 - RDF / OWL IHAL Representation

Verification and Validation

T & E metadata verification and validation are types of metadata traceability and will be used in the examples that follow. Verification and validation are defined as:

- **Verification**: a quality process that is used to evaluate whether or not a product, service, or system (in our case, the instances of the T&E metadata repository) complies with a regulation, specification, or conditions imposed at the start of a development phase or that exists in the organization.
- **Validation**: the process of establishing documented evidence of a high degree of assurance that a product, service, or system (in our cases the various metadata artifacts of a T&E program) satisfies its defining requirements. While this is often a "judgment call," involving acceptance of suitability by external customers, automation can provide significant assistance to the customers.

EXAMPLES

The examples provided in this section are taken from the domain that is most familiar to instrumentation engineers. Examples can easily be constructed for the test engineer or project engineer domains using the T & E reference model as a framework [10]. This section provides example V & V rules that show intra-metadata element V & V for MDL, IHAL and TMATS metadata elements.

Navigation Rules

This section illustrates the use of SPARQL / SPIN [11] rules for navigating across metadata elements. These rules are not rules in the sense of performing a verification or validation, but are used to identify nodes in the graph with specific relationships to other nodes in the graph. A SPARQL / SPIN rule consists of two parts: an ASK / SELECT / CONSTRUCT clause and a WHERE clause. The ASK / SELECT / CONSTRUCT clause is the action; the WHERE clause matches RDF / OWL sub-graphs. The semantics are such that the action is performed if the WHERE clause matches anything in the RDF / OWL sub-graph.

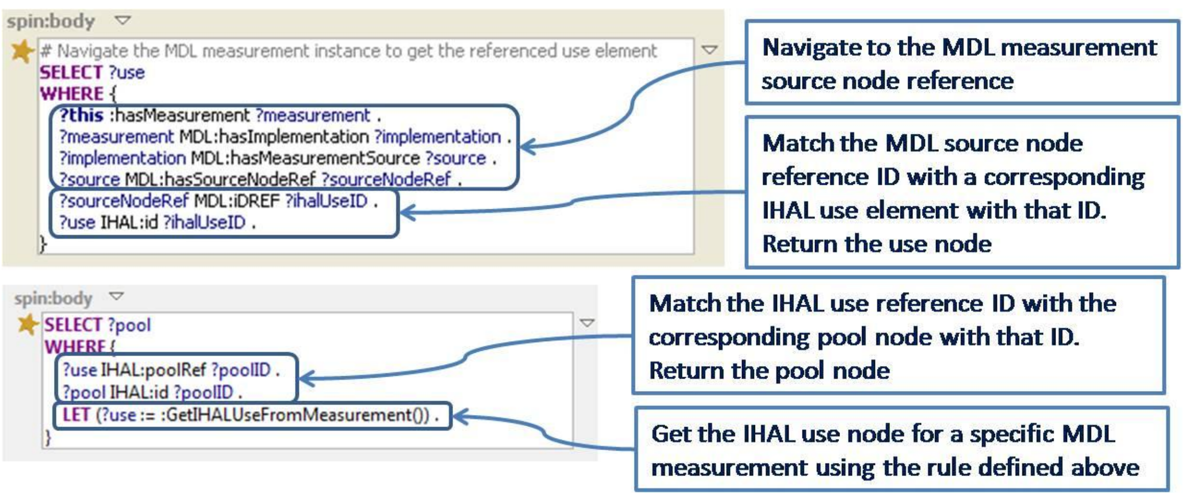

Figure 7 – Verification and Validation Navigation Rules

Figure 7 shows two navigation rules that are used to find an IHAL node that corresponds to the source node reference of an MDL measurement. This navigation rule is useful to support a validation rule that validates that the requirements specified in an MDL measurement description (Figure 5) are realized by the card that conditions that measurement (Figure 6). The navigation rule is broken into two parts: (i) select the IHAL use node that corresponds to an MDL measurement source node reference (topmost rule in Figure 7), and (ii) use the resulting "use" node to select the IHAL pool node that describes the characteristics of that "use" node (bottommost rule in Figure 7).

In the topmost rule in Figure 7, the first part of the WHERE clause matches the nodes and relationships from an MDL measurement node down to the MDL source node reference via the **`MDL:hasImplementation`** → **`MDL:hasMeasurementSource`** → **`MDL:hasSourceNodeRef`** path (Figure 5). The source node reference is bound to the variable **`?sourceNodeRef`**. The second part of the WHERE clause matches the MDL source node reference ID with a corresponding IHAL use node with that same ID. The IHAL use node reference is bound to the variable **`?use`** and is returned in the SELECT clause of the rule.

ID / IDREF Verification

One of the shortcomings of XML-based metadata is that the support for semantics is very limited. *XML schema validation* checks an XML instance document against a schema which defines the allowed instance types, the structure of the XML instance documents, and the existence of unique identifiers (ID) and identifier references (IDREF). The XML ID / IDREF modeling pattern used in IHAL illustrates the problem with XML metadata. The IHAL use element has an identifier reference to some element in an IHAL pool; the semantics are such that the use IDREF points to the IHAL pool element that defines its characteristics. For example, an IHAL card use element is supposed to point to some type of card (signal conditioning card) in the pool. Unfortunately, the ID / IDREF pattern only verifies that the IDREF exists, not that it refers to an element of the right type. It is perfectly valid for an IHAL card use IDREF to point to a thermocouple, GPS card, or even an instrumentation network.

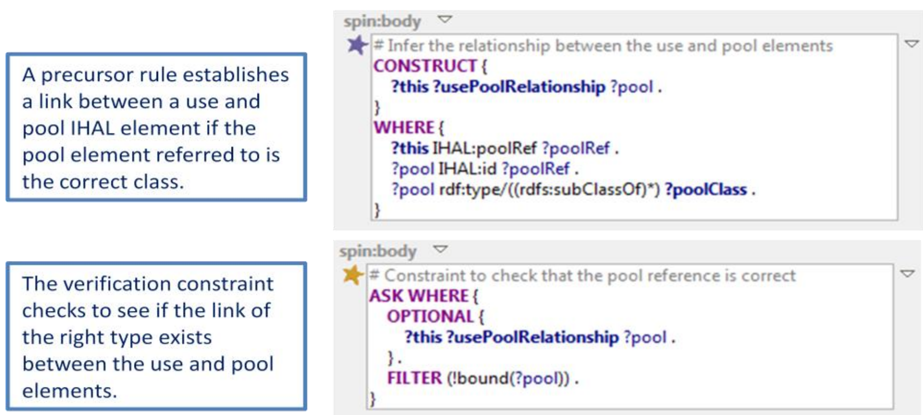

Figure 8 - IHAL ID / IDREF Verification Rule

Figure 8 shows rules for verifying that the IDREF of an IHAL use node refers to the right type of IHAL pool node (e.g., that an IHAL card use node references an IHAL card pool node). The topmost rule in Figure 8 establishes a relationship between the use and pool nodes and the bottommost rule checks for the existence of that relationship. The WHERE clause in the topmost rule matches the use node to the pool node if and only if the pool node is of the right type, where the expected pool node type is passed to the rule as the argument **`?poolClass`**. The CONSTRUCT clause establishes a link between the use and pool nodes via the argument **`?usePoolRelationship`**. Note that if the pool node is not of the expected type, then the CONSTRUCT action is not performed. The bottommost rule checks for the existence of the relationship that was created in the topmost rule; if this relationship does not exist, then the pool node is not the right type.

MDL / IHAL Validation

Figure 9 shows two rules that implement the validation of an IHAL use element against an MDL measurement requirement. Specifically, this rule checks that the maximum required conditioning resolution, as defined in MDL, is satisfied by the signal conditioning card that is selected to condition the measurement. This rule relies on several navigation rules (not shown) and is an ASK clause action rule that is satisfied if the selected card meets the requirements. The WHERE clause of this rule can be divided into three parts, as shown in the figure:

- The maximum required resolution as defined by the MDL is bound to the variable **`?maxRequiredResolution`**; this is achieved via the **`:mdlMaxResolutionRequirement`** navigation property (a special type of navigation function)
- The maximum resolution provided by the signal conditioning card that corresponds to the MDL measurement description is bound to the variable **`?maxResolution`**; this is achieved via the navigation function **`GetIHALOutputPortFromMDLMeasurement`**, which returns the signal conditioning card port or channel, defined in IHAL, and the navigation function **`GetIHALMaxResolution`**, which returns the maximum resolution provided by the signal conditioning card
- The maximum required resolution is compared against the maximum provided resolution as the key condition imposed by the constraint

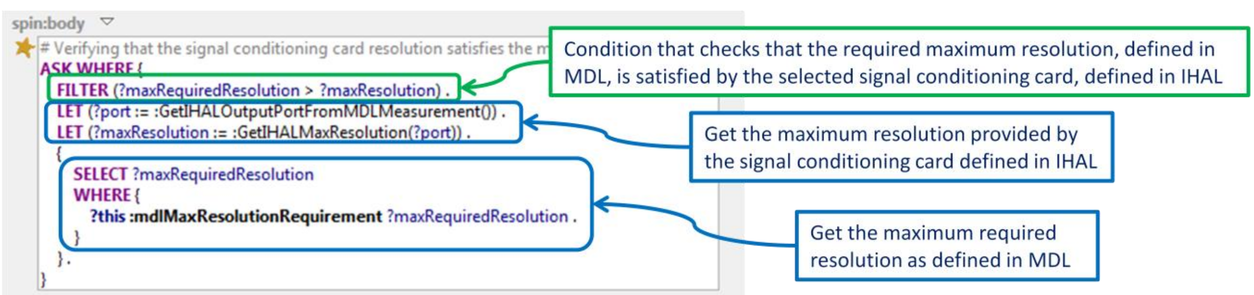

Figure 9 - MDL / IHAL Validation Rule

SUMMARY

The work presented in this paper is an extension of the work presented at a previous ITC conference [12]. We have motivated the reason that verification, validation and completeness is necessary for metadata traceability and the need to employ standards-based representations and technologies to achieve the goal of intra- and inter-metadata traceability. We have proposed RDF / OWL as the representation format and SPARQL / SPIN a the inference rule representation.

Future work includes the identification or development of tools for translating source metadata into RDF format, the development of rule libraries for metadata traceability (navigation functions, constraints, etc.) and the identification and documentation of design patterns, best practices and lessons learned in order to make these techniques accessible and usable within the T & E community.

ACKNOWLEDGEMENTS

The ideas and results described in this paper have been developed as part of a Phase I and Phase II Small Business Innovative Research (SBIR) project funded by Edwards Air Force Base AFFTC.

REFERENCES

[1] https://www.inetprogram.org/Documents/Standards/Metadata%20Standard/MD_Standard_Proposed_v0.5.pdf, accessed April 29, 2010.

[2] Hamilton, Fernandes, Koola, and Jones, *An Instrumentation Hardware Abstraction Language*, Proc. International Telemetering Conf., Vol. XXXXII, (2006) Paper 06-10-02, San Diego, CA.

[3] Hamilton, Fernandes, Graul, Darr, and Jones, *Extensions to the Instrumentation Hardware Abstraction Language (IHAL)*, Proc. International Telemetering Conf., Vol. XXXXII, (2008) Paper 08-19-04, San Diego, CA.

[4] https://wsmrc2vger.wsmr.army.mil/rcc/manuals/106-07/chapter9.pdf, accessed April 29, 2010.

[5] http://www.w3.org/RDF/, accessed April 29, 2010.

[6] http://www.w3.org/TR/owl-features/, accessed April 29, 2010.

[7] Benjamin, Mayer, et al., *Methods for Ontology-Driven Integration*, Proc. 2006 International Conference on Artificial Intelligence, Las Vegas, NV, June, 2006.

[8] http://www.topquadrant.com, accessed April 29, 2010.

[9] http://www.w3.org/TR/rdf-sparql-query/, accessed April 29, 2010.

[10] Hamilton, Fernandes, et. al. *A Model-Based Methodology for Managing T&E Metadata.* Proc. International Telemetering Conf. 2009. Las Vegas, NV.

[11] http://spinrdf.org, accessed April 30, 2010.

[12] Darr, Fernandes, et. al., *Semantic Web Technologies for T&E Metadata Verification and Validation*, Proc. International Telemetering Conf. 2009, Paper 09-08-03, Las Vegas, NV.