DIM Tooling

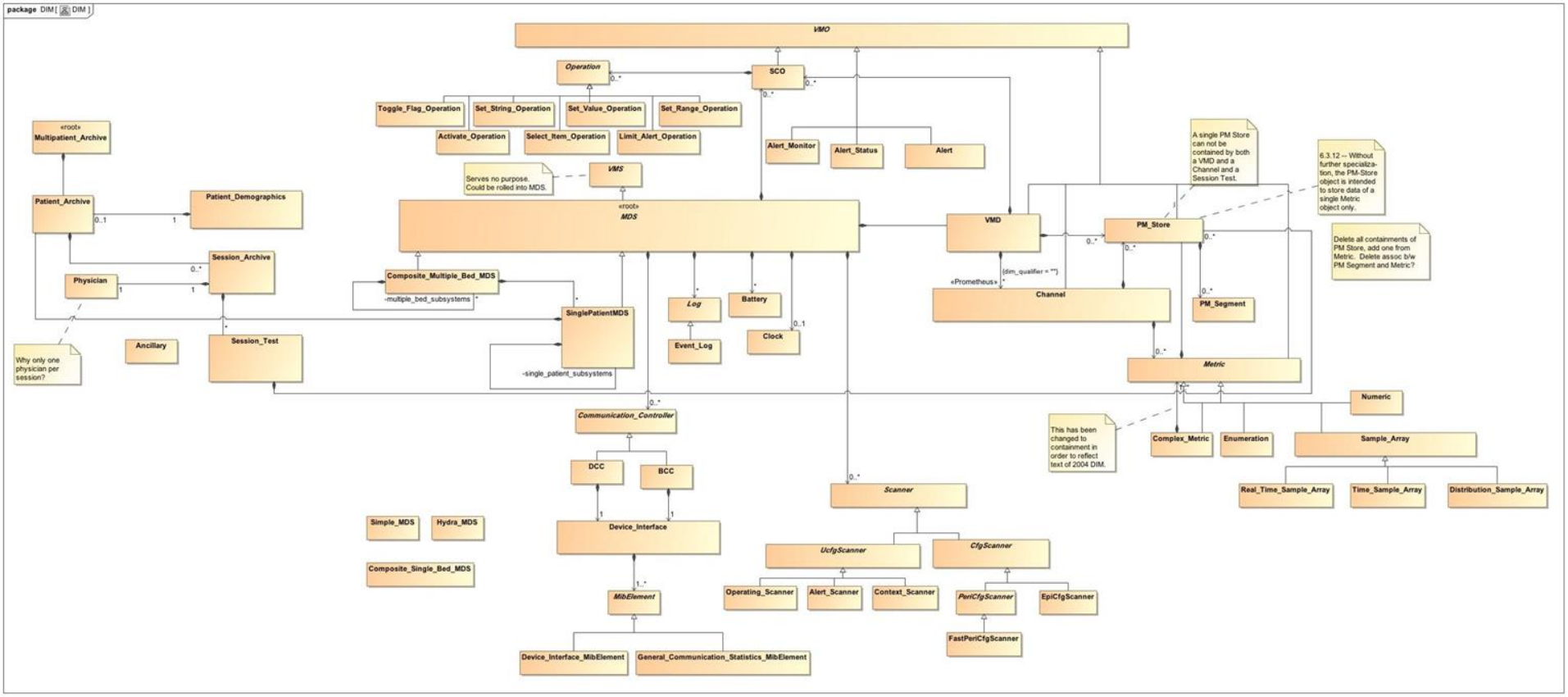
January 2015

Michael Faughn
Prometheus Computing LLC

DIM UML development

- Initial UML model programmatically derived from 11073-10201:2004 in early 2012
- Manual revisions performed
- Supporting models integrated
  - Nomenclature / RTMMS
  - 20101 (ASN.1)
  - Device Profiles
  - Metamodel
  - Conformance Statements (under construction)

DIM Classes



When ASN1 (not shown here) is included, there are 240* classes in the DIM model. The Device Profiling application relies on several other supporting models (MetaInformation, Nomenclature, DeviceProfile, etc.) that interact with the DIM for a total of over 300 classes.

* Will increase to 285 when enumerations are backed by classes.

The Model **is** the Standard

Why?

- Computable
- Artifacts derived from a common source
  - Printed Standard
  - Software tools
  - XML Schema
  - more

UML to Artifacts: Challenges

- UML has trouble expressing some constructs in a convenient way
  - BIT_STRING
  - Class instance variables
  - BNF (i.e. ASN.1)
- Each UML element type used has to be implemented for each builder plugin that produces an artifact.
- Keep the standard ‘pure’ vs. supporting the functionality that artifacts require

Programmatically Derived / Generated From UML

- XML Schema (soon to be updated)
- ASN.1
- Relational database schema
- Device Profile Editor web application
  - ~ 3,000 lines of code per classifier
- Rich Ruby API for interacting with DIM objects
- PDF of Standard (not bad, not perfect)
- JSON meta-information

Device Profile Editor Existing Capabilities 1/3

- Assemble DIM objects into containment trees.
  - Composition constrained by the standard
- Allow user to specify whether a given object attribute is used or not within the context of an object instance in a profile -- improved since last WG meeting
- Associate device profile elements with terms from RTMMS
- Fetch new and updated terms from RTMMS

API Example

```
p = MyDevice::PCDProfile.create(name: "Infusion", intended_use:
'Normative (11073)', purpose: 'Demonstration')

mds = DIM::System::SinglePatientMDS.create(name: "Infusion Pump MDS")

mds.set_ref_id "MDC_DEV_PUMP_INFUS_MDS"

p.profileroot = mds

p.save

v = DIM::Medical::VMD.create(name: "Infusion Pump VMD")

v.set_ref_id "MDC_DEV_PUMP_INFUS_VMD"

mds.vmds_add v

mds.save

delivery_ch   = DIM::Medical::Channel.create(:name => "Delivery
Channel")

v.channels_add delivery_ch

delivery_ch.set_ref_id "MDC_DEV_PUMP_INFUS_CHAN_DELIVERY”
```

API in REPL

main 003(0) > DIM::Medical::VMD.first

=> #<DIM::Medical::VMD @values={:id=>1, :class_reserved_id=>nil, :name_binding_id=>nil, :ref_id_id=>5818, :ext_obj_relations_id=>nil, :ext_obj_relations_class=>nil, :label_string_id=>nil, :label_string_class=>nil, :type_id=>nil, :locale_id=>nil, :locale_class=>nil, :alertoralertstatus_id=>nil, :alertoralertstatus_class=>nil, :position_id=>nil, :position_class=>nil, :compatibility_id_dim_id=>nil, :compatibility_id_dim_class=>nil, :operating_hours_id=>nil, :operating_hours_class=>nil, :operation_cycles_id=>nil, :operation_cycles_class=>nil, :instance_number_id=>nil, :vmd_model_id=>nil, :vmd_status=>nil, :measurement_principle=>nil}>

main 011(0) > DIM::Medical::VMD.attributes.keys #=> [:vmd_status, :measurement_principle]

main 012(0) > DIM::Medical::VMD.associations.keys

=> [:property_specializations, :property_uses, :valid_terms, :class_reserved, :name_binding, :ref_id, :ext_obj_relations, :label_string, :type, :handle, :locale, :alertoralertstatus, :position, :compatibility_id_dim, :operating_hours, :operation_cycles, :instance_number, :vmd_model, :parameter_group, :service_and_control, :channels, :pm_store, :production_specification, :mds, :deviceprofile]

https://rtmms.nist.gov/rtmms/getTermsJson.do?fromDate=01012014000000

Examples from RTMMS web service created by Nicolas Crouzier @ NIST:

[{"referenceId":"MDC_AREA_BODY_SURF_ACTUAL","termCode":188744,"systematicName":"Area | Actual | BodySurface | Body","commonTerm":"Patient body surface area","acronym":"","termDescription":"The actual body surface area of the patient, calculated from patient actual weight and patient actual length.","updateDate":"Oct 31, 2014 4:03:33 PM","status":"APPROVED","type":"METRIC","sources":**["HRTM","RTM"]**,"units":["MDC_DIM_SQ_X_M"]},
{"referenceId":"MDC_ATTR_AL_COND","termCode":68012,"systematicName":"","commonTerm":"","acronym":"","termDescription":"","updateDate":"Dec 17, 2014 5:49:55 PM","status":"APPROVED","type":"ENUMERATION","sources":["HRTM","RTM"],"enums":["MDC_EVT_SYRINGE_PATIENT_PRESSURE","MDC_EVT_SYRINGE_PLUNGER_POSITION","MDC_EVT_SYRINGE_FLANGE_POSITION","MDC_EVT_SYRINGE_BARREL_CAPTURE","MDC_EVT_SYRINGE_PRESSURE_DISC_POSITION","MDC_EVT_SYRINGE_END_OF_TRAVEL","MDC_EVT_SYRINGE_EMPTY","MDC_EVT_PCA_DOOR_UNLOCKED","MDC_EVT_PCA_HANDSET_DETACHED","MDC_EVT_PCA_MAX_LIMIT","MDC_EVT_PCA_PAUSED"]},
{"referenceId":"MDC_EVT_STANDBY_WARN","termCode":258048,"systematicName":"","commonTerm":"","acronym":"","termDescription":"","updateDate":"Nov 7, 2014 4:24:18 PM","status":"APPROVED","type":"LITERAL","sources":["RTM"]},
{"referenceId":"pump-stopped-transitioning","updateDate":"Nov 7, 2014 4:29:22 PM","status":"PROPOSED","type":"TOKEN","sources":["RTM"]},
{"referenceId":"_UOM_CONC_GAS","type":"**UNITGROUP**","sources":["RTM"],"units":["MDC_DIM_PERCENT","MDC_DIM_VOL_PERCENT","MDC_DIM_KILO_PASCAL","MDC_DIM_MMHG"]},
{"referenceId":"_MDC_ATTR_AL_COND_DELETED","type":"**ENUMGROUP**","sources":["RTM"],"enums":["MDC_EVT_SYRINGE_NUT_ENGAGED","MDC_EVT_SYRINGE_PATIENT_PRESSURE_ALARM","MDC_EVT_SYRINGE_PRESSURE_DISC","MDC_EVT_SYRINGE_PLUNGER","MDC_EVT_HANDSET_DETACHED","MDC_EVT_SYRINGE_FLANGE","MDC_EVT_SYRINGE_LEVER"]}]

* New features in bold.

3.2.14 Event_Log class

- Class: Event_Log
- Description: The Event Log o
  events in a free-text or in a bin
- Derived from: Log
- Name binding: Handle
- Registered as: MDC_MOC_LOG_EVENT

Programmatically Generated PDF of 11073:10201

Table 42: Attributes of Event_Log

Attribute name	Attribute ID	Attribute type	Remark	Qualifier
Event-Log-Entry-List	MDC_ATTR_-EVENTLOG_EN-TRY_LIST	EventLogEntry	Event entries; can be retrieved with GET service.	M
Event-Log-Info	MDC_ATTR_-EVENTLOG_-INFO	EventLogInfo	Static and dynamic specifications.	O
Type	MDC_ATTR_ID-TYPE	OCTET STRING	Further specification of log entry format.	O

```
Event_Log ::= SEQUENCE {
COMPONENTS OF Log,
Event-Log-Entry-List EventLogEntryList,
Event-Log-Info EventLogInfo,
Type OCTET STRING
}

EventLogEntryList ::= SEQUENCE OF EventLogEntry
```

Device Profile Editor Existing Capabilities 2/3

- Allow creation of Normative (11073-103xx) and User Defined device profiles
- Allow user to view metadata about DIM classes and attributes
- Use any device profile as a template for a new device profile via cloning*

Device Profile Editor Existing Capabilities 3/3

- Deliver Summary XML (Rosetta Containment Hierarchy) representation of a device profile
- Deliver detailed HTML report of containment and terminology
- Deliver comprehensive representation of Device Profile in JSON*

HTML Report

PCD Profile: Test Pulse Ox

Purpose: For testing
Intended Use: Normative (11073)
Owning Company: C4MI

Containment Tree

MDI Prototyping Project PulsOx MDS (SinglePatientMDS)	MDC_DEV_ANALY_SAT_O2_MDS
PulsOxim AlertMonitor (AlertMonitor)	MDC_MOC_VMO_AL_MON
Alert Scanner (AlertScanner)	Untyped
MDC_DEV_ANALY_SAT_O2_VMD (VMD)	MDC_DEV_ANALY_SAT_O2_VMD
MDC_DEV_ANALY_SAT_O2_CHAN (Channel)	MDC_DEV_ANALY_SAT_O2_CHAN
MDC_DEV_PULS_CHAN (Channel)	MDC_DEV_PULS_CHAN

Object Details

SinglePatientMDS: MDI Prototyping Project PulsOx MDS

MDC_DEV_ANALY_SAT_O2_MDS

AlertMonitor: PulsOxim AlertMonitor

MDC_MOC_VMO_AL_MON

AlertScanner: Alert Scanner

VMD: MDC_DEV_ANALY_SAT_O2_VMD

MDC_DEV_ANALY_SAT_O2_VMD

Channel: MDC_DEV_ANALY_SAT_O2_CHAN

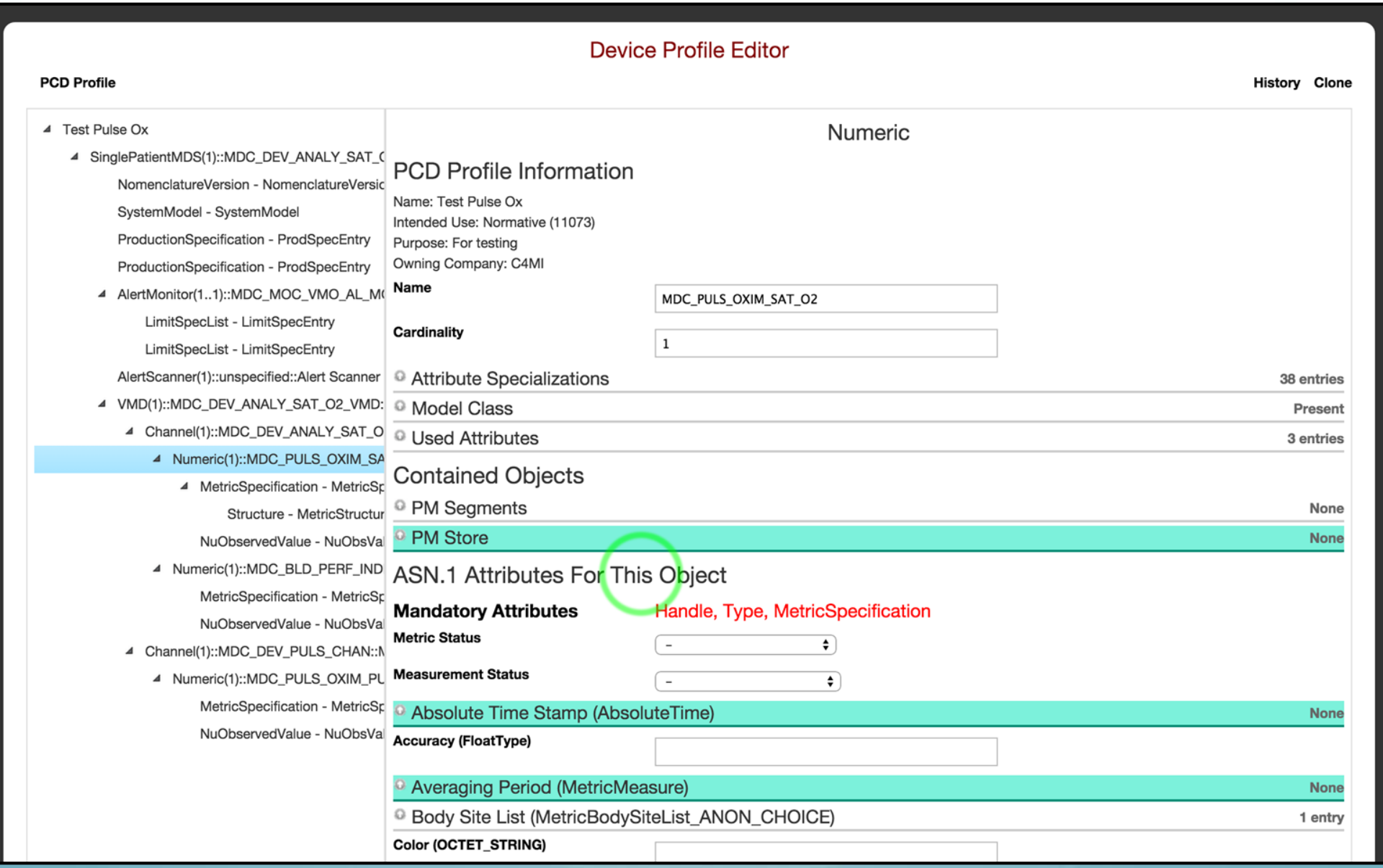
MDC_DEV_ANALY_SAT_O2_CHAN

Name	Reference ID	Units	Enumerations

Notable Improvements Since Chicago WG Meeting 1/2

- Unified code responsible for constructing the Ruby models and GUI specification
  - Increases assurance of fidelity to the model
- Improved containment tree visualization in UI
- Developed system that allows the user to use ref_ids in place of OIDType (i.e. integer) values…
  - In GUI
  - In XML
  - Does not change the underlying DIM model

New Features: Composition Tree Pane, Mandatory Attributes Listed, Attribute Types Shown (mostly)



Notable Improvements Since Chicago WG Meeting 2/2

- Increased speed of generation of XML and JSON
- XML being generated from complete traversal of device profile object graphs
- Improved facility for customization of XML & JSON output.
- Automatic management of ‘Used Attributes’
- Attribute types shown in GUI
- Mandatory attributes clearly shown in GUI

XML Output Example

```
<?xml version="1.0" encoding="UTF-8"?>
<PCDProfile company_owner="C4MI" intended_use="Normative (11073)" name="Test Pulse Ox"
   profile_mode="Baseline" profile_type="Agent" purpose="For testing" xml_date="2015-01-22
   01:06:15 +0000" description_content="" xmlns="http:prometheuscomputing.com/schemas/dim"
   xmlns:="http:prometheuscomputing.com/schemas/dim">
 <single_patient_mds cardinality="1" system_capability="sc-multiple-context"
   application_area="area-unspec" line_frequency="line-f-60hz" mds_status="disconnected"
   patient_type="adult" power_status="onMains" class="NO_REF_ID::Unknown"
   name_binding="MDC_ATTR_ID_HANDLE::2337" handle="">
   <system_type partition="nom-part-obj" code="MDC_DEV_ANALY_SAT_O2_MDS::4105"/>
   <nomenclature_version nom_major_version="majorVersion1" nom_minor_version="1"/>
   <system_model manufacturer="" model_number=""/>
   <production_specification spec_type="serial-number" prod_spec="" component_id_dim=""/>
   <production_specification spec_type="protocol-revision" prod_spec="" component_id_dim=""/>
   <locale charset="charset-unspec" country="" language="" str_spec=""/>
   <scanner cardinality="1" operational_state="" handle="55" reporting_interval="2"/>
   <vmds cardinality="1" vmd_status="vmd-standby" handle="">
     <type partition="nom-part-metric" code="MDC_DEV_ANALY_SAT_O2_VMD::4106"/>
     <channels cardinality="1" handle="">
       <type partition="nom-part-obj" code="MDC_DEV_ANALY_SAT_O2_CHAN::4107"/>
       <metric cardinality="1" handle="">
         <type partition="nom-part-metric" code="MDC_PULS_OXIM_SAT_O2::150456"/>
         <metric_specification access="sc-opt-normal" category="auto-measurement"
   relevance="rv-unspec" update_period="8192">
...
```

Imminent Improvements

- Better / Richer integration of RTMMS data
- More improvements / customizations to XML formats
- Resizable panes in GUI
- Easier access to metadata and information from the DIM standard

dim.prometheuscomputing.com

user: dim

password: 11073

Beta Testing

- Create base/normative profiles
- Create device profiles from cloned normative profiles
- Use output XML and provide feedback

FY2015 Goals 1/3

- **Users using Device Profile Editor (DPE)**
- Continue to make GUI friendlier
- Produce IEEE compliant PDF of Standard from web application
- Make snippets of DIM standard (PDF) available in DPE.
- Refine DPE output artifacts based on stakeholder needs.
  - Format(s) for round-trip profile data

FY2015 Goals 2/3

- Produce conformance statements in machine readable and human-readable formats (XML, JSON, PDFs)
- Alert user when and where a device profile is not in compliance with the DIM, 103xx, and/or Nomenclature
- Facilitate user and group level access to individual device profiles in DPE
- Cordon off application features specific to SDO
- Produce ‘Help’ documentation for DPE

FY2015 Goals 3/3

- Enumerations backed by model (they currently are not)
- Solve BIT_STRING issues
- Transition application code away from heavy reliance on launchtime code generation.
  - Will result in a codebase that is easier to understand and maintain for those unfamiliar with the Prometheus Computing toolchain.

Standard Document Manager

- Now
  - Can structure IEEE Standard
  - Can enter and edit text
  - Is integrated with Model Manager
- Future
  - Port LaTeX generators for PDF creation
  - Integrate with Profile Editor (read-only snippets)

Distant Goals 1/2

- Finish Model Manager application (supplanting MagicDraw)
  - Easier use
  - Output model
- Ability to add non-conformant attributes to objects within a device profile
- Validate profiles against their parent profiles (e.g. a device specialization or use-case based profile)*

Distant Goals 2/2

- Extension of MyDevice to PHD devices
- Tighter coupling between Profile Editor and RTMMS ?
  - One login, mutual hyperlinks
  - Verification of device profiles with respect to nomenclature hierarchy
- Integrate UML web application

Model Manager

- Now
  - The model is in the web application
  - Users can manipulate the model
  - Models are governing and informing Device Profile Editor
  - Is integrated with Standard Document Manager
- Future
  - Model will feed back into a UML editing tool
  - XMI and/or MagicDraw API

Questions? Comments?

m.faughn@prometheuscomputing.com