DIM UML development

- Initial UML model programmatically derived from 11073-10201:2004 in early 2012
- Work on applications begun FY 2013
- Manual revisions performed
- Supporting models integrated
  - IEEE11073:10101 Nomenclature (RTMMS)
  - IEEE11073:20101 (ASN.1 Simple Types)
  - Device Profiles
  - Metamodel (represents 10201 UML in web applications)
  - Printed Standard

Conformance Statements (work in progress)
This shows only the classes defined by the standard. There are ~350 classifiers total in the DIM model. The Device Profiling application relies on several other supporting models (MetaInformation, Nomenclature, DeviceProfile, etc.) that interact with the DIM model. The Device Profiling application implements ~440 classes specific to the application.
The Model is the Standard

Why?

● Computable

● Artifacts programmatically derived from a common source help to ensure harmonization.
  ○ Printed Standard
  ○ Software tools (Device Profiling, Validation, ...)
  ○ XML Schema
  ○ Conformance Statements
UML to Artifacts: Challenges

● UML (or UML tools) has trouble expressing some constructs in a convenient way
  ○ Class instance variables
  ○ BNF (i.e. ASN.1)
  ○ BIT STRING

● Each UML element type used has to be implemented for each builder plugin that produces an artifact. Lots of work the first time you do it and every time you build a new plugin.

● Keep the standard ‘pure’ vs. supporting the functionality that artifacts require.
Programmatically Derived / Generated From UML

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

MagicDraw
Create / Edit UML

Prometheus Plug-ins

Plug-in Products
- Ruby implementation of UML
- GUI specification for interaction with modeled entities
- Relational Database Schema
- JSON that can be parsed into a Java implementation of the UML

XML Schema

Handwritten code

Integrated Web Applications
- Device Profiling Tool
- Model Manager
- Document Creation Tool

PDF of Standard
- Formatted ASN.1
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 010(0) > DIM::Medical::VMD.first

=> #<DIM::Medical::VMD @values={:id=>1, :class_reserved_id=>nil, :name_binding_id=>nil, :ref_id_id=>5818, ...}>

Device Profile Editor
Existing Capabilities 1/3

● Assemble DIM objects into device profile containment trees.
  ○ Composition constrained by the standard
  ○ View the containment tree

● Allow creation of Normative (11073-103xx) and User Defined device profiles.

● Use any device profile as a template for a new device profile via cloning
Device Profile Editor
Existing Capabilities 2/3

- Allow user to view metadata about DIM classes and attributes (i.e. what is found in the paper standard)
- Associate device profile elements with terms from RTMMS.
- Fetch new and updated terms from RTMMS.
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_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.
Device Profile Editor
Existing Capabilities 3/3

- Summary XML (Rosetta Containment Hierarchy) representation of a device profile
- Detailed HTML report of containment and terminology
- Comprehensive representation of Device Profile in XML
- Comprehensive representation of Device Profile in JSON (same info as XML, different format).
PCD Profile: Test Pulse Ox

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

Containment Tree

<table>
<thead>
<tr>
<th>MDI Prototyping Project PulsOx MDS (SinglePatientMDS)</th>
<th>MDC_DEV_ANALY_SAT_O2_MDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PulsOxim AlertMonitor (AlertMonitor)</td>
<td>MDC_MOC_VMO_AL_MON</td>
</tr>
<tr>
<td>Alert Scanner (AlertScanner)</td>
<td>Untyped</td>
</tr>
<tr>
<td>MDC_DEV_ANALY_SAT_O2_VMD (VMD)</td>
<td>MDC_DEV_ANALY_SAT_O2_VMD</td>
</tr>
<tr>
<td>MDC_DEV_ANALY_SAT_O2_CHAN (Channel)</td>
<td>MDC_DEV_ANALY_SAT_O2_CHAN</td>
</tr>
<tr>
<td>MDC_DEV_PULS_CHAN (Channel)</td>
<td>MDC_DEV_PULS_CHAN</td>
</tr>
</tbody>
</table>

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
Recent Efforts

- NIST Policy Machine integration for managing multi-tenancy / object permissions (Sam Dana)
  - rolling out soon
- Improvements to XML / XSD generation
- Cataloged differences between the current model with 2004 standard
- Tree-view and breadcrumbs improved
- RTMMS model updated to be more comprehensive / useful
FY2016 Goals 1/2

- XML / XSD generation for DIM still needs work - Q1
- Map Classic to revised DIM in order to support backwards compatibility....?
- Enable PHD profile creation
  - Analyze DIM / PHD congruence
- Schema generation for device specializations
- Generate camera ready IEEE standard(?)
  - Generate modular artifacts (tables, ASN.1, ...)
- Complete integration of NIST Policy Machine
FY2016 Goals 2/2

- Move from Alpha to Beta testing to release
- User documentation / help for device profiling application
- Easier access to metadata and information from the DIM standard for device profile users
- Release for general use
- Update Nomenclature model as necessary
- Conformance statements from the profile editor
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
  ○ Generate .docx
  ○ Integrate with Profile Editor (read-only snippets)
dim2.prometheuscomputing.com

user: dim
password: 11073
XML Challenges

- Namespacing by package?
- Attributes typed as interfaces
Questions? Comments?

m.faughn@prometheuscomputing.com
Alpha / Beta Testing

- Create base/normative profiles
- Create device profiles from cloned normative profiles
- Use output XML and provide feedback
- Use XML schema as input to C4MI tools
3.2.14 Event_Log class

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

Table 42: Attributes of Event_Log

<table>
<thead>
<tr>
<th>Attribute name</th>
<th>Attribute ID</th>
<th>Attribute type</th>
<th>Remark</th>
<th>Qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event-Log-Entry-List</td>
<td>MDC_ATTR_.EVENTLOG_ENTRY_LIST</td>
<td>EventLogEntry</td>
<td>Event entries; can be retrieved with GET service.</td>
<td>M</td>
</tr>
<tr>
<td>Event-Log-Info</td>
<td>MDC_ATTR_.EVENTLOG_INFO</td>
<td>EventLogInfo</td>
<td>Static and dynamic specifications.</td>
<td>O</td>
</tr>
<tr>
<td>Type</td>
<td>MDC_ATTR_ID-TYPE</td>
<td>OCTET STRING</td>
<td>Further specification of log entry format.</td>
<td>O</td>
</tr>
</tbody>
</table>

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

EventLogEntryList ::= SEQUENCE OF EventLogEntry
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
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