

11073

Device Profile Tooling

HL7 F2F September 2017

Michael Faughn
Prometheus Computing LLC

TL;DR

- The DIM is expressed as UML. The UML governs all.
- Device Profile Editor web app, printed standard, and XSD are programmatically generated.
- Web app is nominally usable now (prove me wrong).

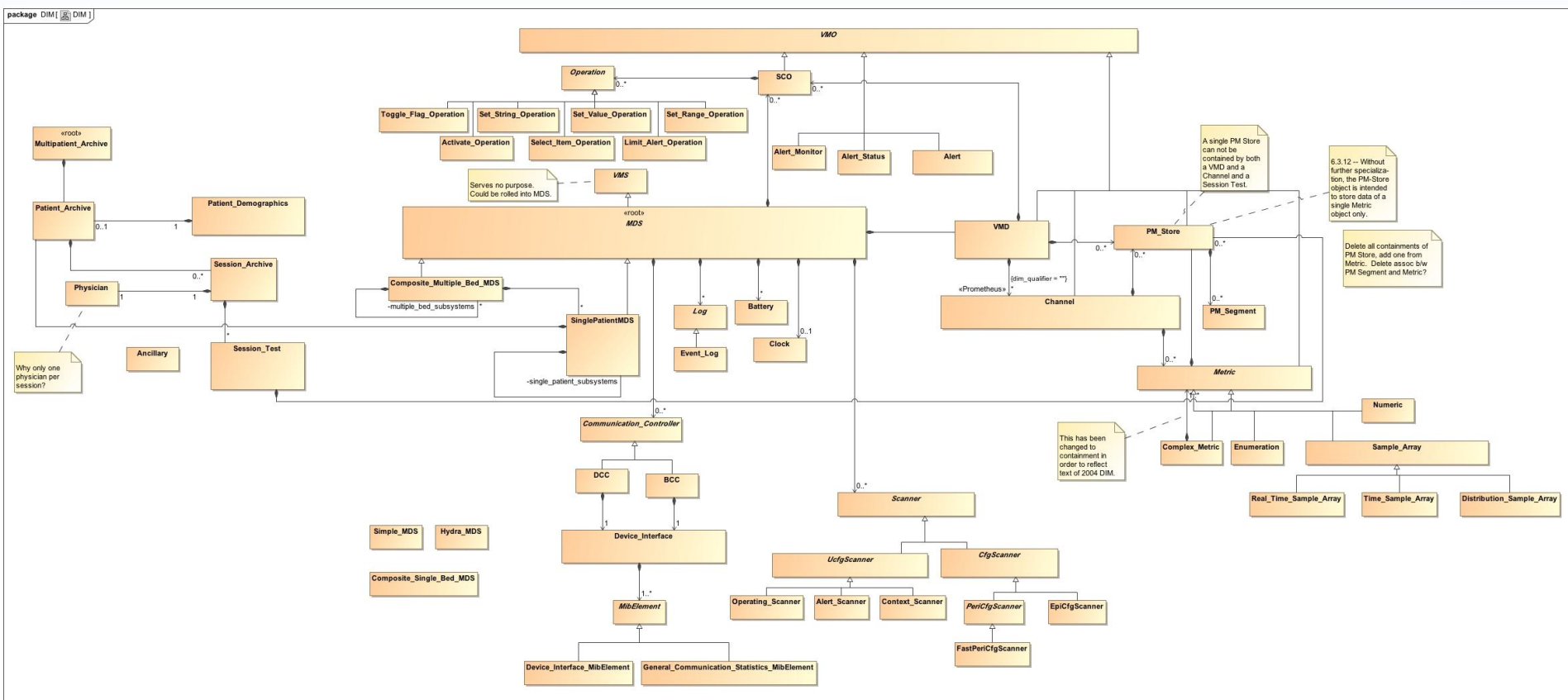
DIM UML development

- Initial UML model programmatically derived from 11073-10201:2004 in early 2012
- Work on applications begun FY 2013
- Manual revisions performed
- Ongoing development of additional UML

DIM UML development

- Supporting models integrated
 - IEEE11073:10101 Nomenclature (RTMMS)
 - IEEE11073:20101 (ASN.1 Simple Types)
 - Device Profiles
 - Metamodel (represents 10201 UML in web applications)
 - Printed Standard
 - IEEE11073:20601 Personal Health Devices
 - IEEE11073:10207 BICEPS

DIM Classes



This shows only the classes defined by the standard. There are ~350 classifiers total in the 10201 DIM model. There are 110 classifiers in the PHD 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 ~550 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

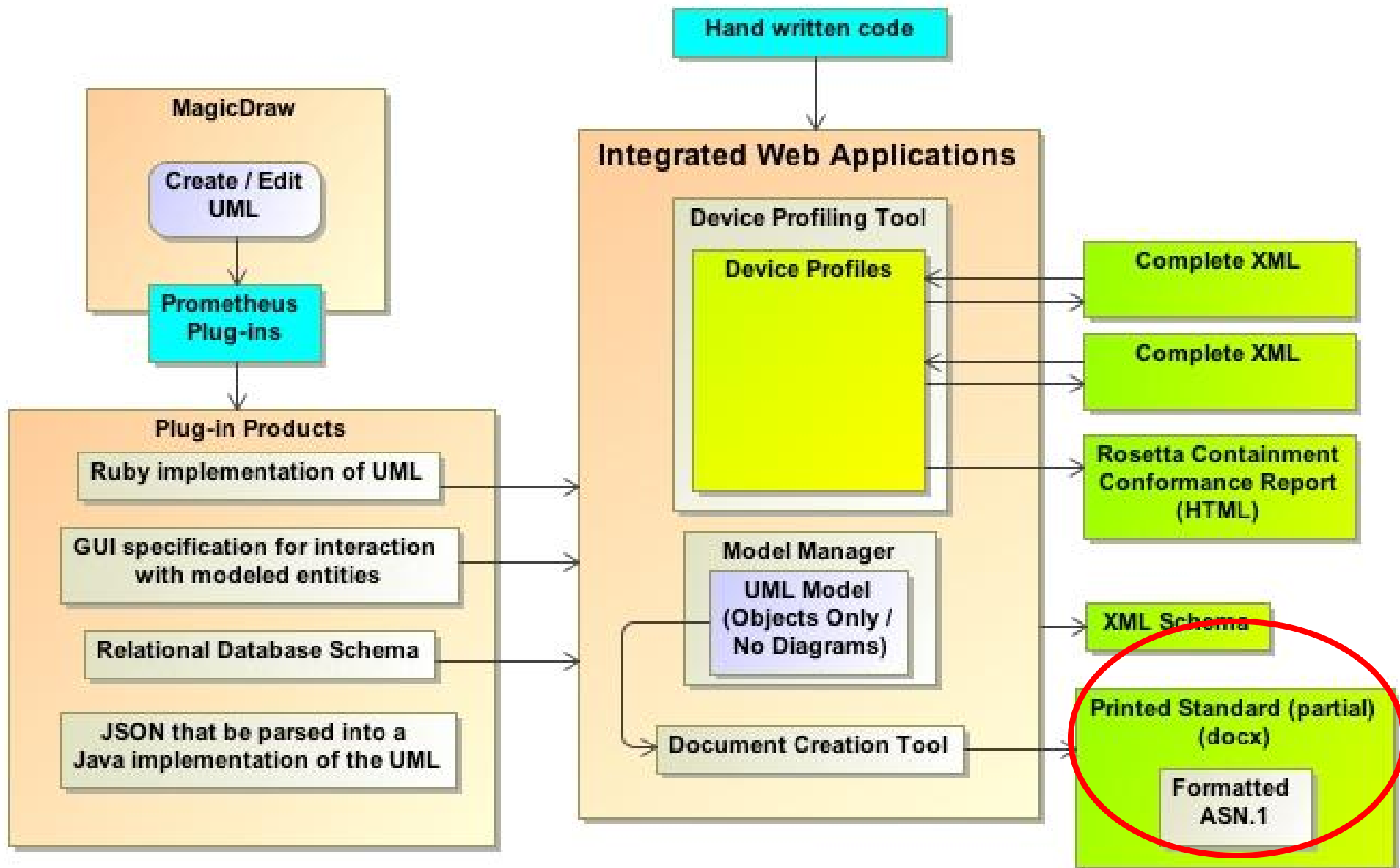
UML to Artifacts: Challenges

- 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 code module that produces an end product (web application, printed standard, etc.). Lots of work the first time you do it and every time you build a new module to produce a new end product.
- Keep the standard 'pure' vs. supporting the functionality that artifacts require.

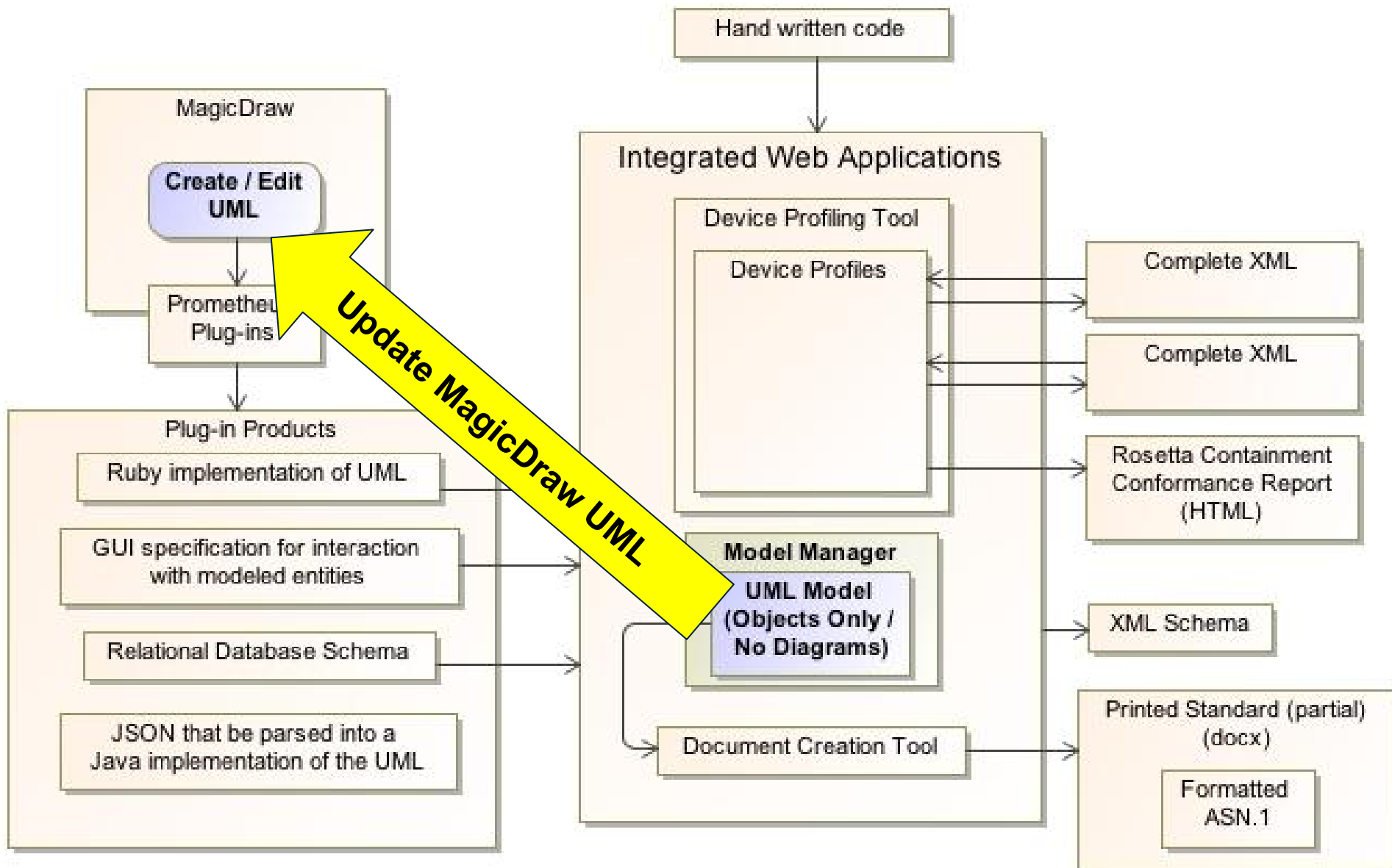
Programmatically Derived / Generated From UML

- 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
- .docx

R&D Workflow (Simplified)



Roundtrip UML



Existing Device Profiles

PCD

Pulse Oximeter

Infusion Pump

Vital Signs Monitor

Ventilator

Dialysis Machine

Microenvironment (Incubator)

PHD

Pulse Oximeter

Best Practices for Device Specialization Development

- Maintain terminology and containment structure independently.
- Every term definition must include a RefID.
- Every node in a containment tree must have a RefID.
- Content over format.
 - Massaging content into a format appropriate for upload into RTMMS and/or the Device Profile Editor is achievable. Application developers can assist you.
- For information about adding terms to RTMMS see the README files at: [IEEE 11073 Downloads](#)
 - Note: The Device Profile Editor will not upload terms into RTMMS for you.

(more) Best Practices for Device Specialization Development

Beta testers have found that it is easier to build device containment trees in XML rather than in the web application. They have leveraged a round-trip, iterative process of editing XML, uploading the XML to the web application to produce a new device profile, making a few edits and spot checks using the application, and downloading updated XML from the application.

XML containment tree from many sources can often be reformatted to be acceptable for upload into the Device Profile Editor.

Device Profile Editor

Existing Features 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
 - ⊖ ~~Entire containment tree must be cloned~~

Device Profile Editor

Existing Features 2/3

- Complete and Summary XML (Rosetta Containment Hierarchy) representation of a device profile
- 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.
- Support for the use of terms not found in RTMMS.

Device Profile Editor

Existing Features 3/3

- Create a Device Profile by uploading XML in Complete or Rosetta Containment Hierarchy (RCH) format
 - Original file is stored
 - (Non)Conformance messages provided after upload
- Round-trip capable XML - input and output - complete or summary (RCH)
- Visual cues provide conformance information
- HTML tabular report (dot level, units, enums, codes)

Recent Progress 1/2

- BICEPS work continues - Simon Baumhof
- **Major** refactoring of toolchain components
- Ready to test Profile Editor on JRuby
 - Eases deployment on NIST servers
- Better cloning (Channels, Metrics, etc.)
- Access Control implementation begun

Recent Progress 2/2

- Programmatically checked generated ASN.1 against published ASN.1
- Generated docx is ***pretty darned good***
- Updated UML figures for 10201
- ASN.1 in single document. DIM class pseudocode.

FY2017 Goals

- **Complete work to programmatically produce docx for 11073:10201**
- Revisit XML schema
 - Verify created XML is valid
- **Deploy application to NIST server**
- Produce user manuals and training material
- Author white paper(s) describing project methodologies and achievements.

More Goals

- **Finish implementation of r/w permissions in web application**
- Improve integration with RTMMS v2 (after it deploys)
- **Comprehensive audit of updated DIM model / application with respect to 11073:10201© 2004**
- **Improve access and presentation of metadata and information from the standard from device profile views**

User Oriented Goals

- Deploy on NIST server
- Implement Access Control
- Produce user manuals and training material
- Produce white paper(s) describing project methodologies and achievements

“Under the Hood” Tasks

- Update UML in UML tool (MagicDraw) with UML from web application
- Model and implement constraints (e.g. for Alerts)
- Implement more flexible device profiles
-

Future Device Profile Features

- Device components (e.g. VMDs, Channels) become reusable and shareable across multiple containing profiles
 - A change to one component will propagate to all occurrences
- Choices (any of, one of) within normative profiles
- Ad-hoc groupings of metrics

Draft Standard

- Review differences b/w classic, 2010 draft, & current draft
- Resolve critical issues on “Revisions & Comments”
- What else? What’s next?

Draft Standard: Material Changes

- Aggregation → Composition
- MDS types consolidated
- No more anonymous ASN.1 datatypes
- “Object” → “Class”

Draft Standard: Editorial Changes

- Attempting to make the text more uniform
- Identifiers CamelCased
-

Draft Standard: Laundry List

- Communication Diagram (Fig. 6.9)
- Should only immediate child classes be listed or should all child classes be listed?
- “The Top object is the common inheritance base for all objects in the DIM.” -- False.
 - DeviceInterface, MibElement, children
- ProdSpecType ::= INT-U16 { fda-udi(8)
 - Where did this come from?

Draft Standard: More Laundry

- SingleBedMDS & MultipleBedMDS need ref_ids.
- Handle is duplicate attribute for PatientDemographics
- Missing refid: MDC_ATTR_METRIC_ID_PART
- ref_id mismatches: assuming RTMMS is correct(?)

dim.prometheuscomputing.com

For login details contact:

dim@prometheuscomputing.com

End of Presentation

Questions? Comments?

`dim@prometheuscomputing.com`

XML Discussion

- Should term_code and cf_term_codes be used at all in the XML?
 - TYPE vs OID-Type
- Do we worry about ANY_DEFINED_BY?
-