A Few Words First

Audio Connection –Please mute phone (*6 toggle) –or your GM left-side name

Phone connections may be muted during presentation. Put questions in chat box.

Upcoming Meetings:

- July 10: Systems Engineering for Early Stage R&D Projects – Heidi Hahn, Los Alamos National Laboratory, and Ann Hodges, Sandia National Laboratories

- August 9: Summer Social, Bow & Arrow Brewing Co.

- September 11: Systems Engineering Challenge, Nexus Brewery

CSEP Courses by *Certification Training International:*
    Course details(with more locations and dates)
    Upcoming Course Schedule (somewhat nearby):
    2019 Aug 12-16| Austin, TX
    Chapter SEP mentors: Ann Hodges alhodge@sandia.gov, Heidi Hahn hahn@lanl.gov

First slide, not retained in recording but retained in pdf presentation.
And now - introductions

Enchantment Chapter Monthly Meeting

12 June 2019 – 16:45–18:00 MT

Enterprise Data Definition Management for a System of Systems

Barbara Mills, Sandia National Laboratories

Abstract: This presentation discusses a model-based approach to data design and management for a remote sensing ground station which is being developed as a system of systems. It is also discusses the evolution of an enterprise data management approach for that same "system".

Download slides today-only from INCOSE_GMSeven file library or any time from the Library at www.incose.org/enchantment

NOTE: This meeting will be recorded
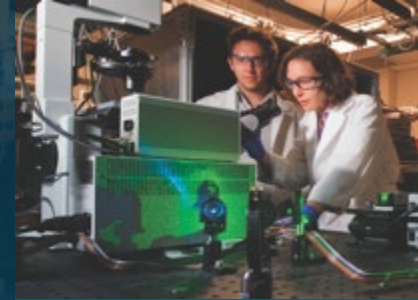
Today's Presentation

Things to think about

- How can this be applied in your work environment?

- What did you hear that will influence your thinking?

- What is your take away from this presentation?

Speaker Bio

_____

Barbara Mills is a Principal Staff Member at Sandia National Labs (SNL) in the remote sensing ground station area. Her focus in this group has been data design/management as well as system architecture. Her prior position, as technical lead of an OPIR Data Standards group, set the stage for the current data design work. Barbara received her masters of science in electrical engineering (MSEE) from the University of California, Berkeley and her BSEE from the Ohio State University. Her experience at SNL has been varied (hardware design, embedded software design, signal processing, data and system design) but all applied to various sensing systems. In every case some kind of model was created; it is puzzling that complex systems are still created without them.

Enterprise Data Definition Management for a System of Systems

PRESENTED BY

Barbara Mills, Sandia National Labs

June 12, 2019

Focus of this Talk

- The definition and management of interfaces in a system of systems implementation of a remote ground station.

- The interfaces are between:
  - Applications
  - Framework services and applications
  - Framework common data backbone and applications
  - Data output by the Framework with content provided by the applications

- The interfaces are implemented via:
  - Shared memory instantiation
  - Protocol buffers
  - JSON messages
  - Streaming binary data
  - Hierarchical Data Format V5 (HDF5) files

Context

- Realizing the vision of a flexible, extensible common sensor family processing capability requires
  - Establishment of a framework
  - Plans for maintenance/enhancement of that framework
  - Establishment and application of data standards
  - Plans for maintenance/enhancement of the data interfaces
- This talk puts forth a vision for that last bullet point in a system of systems environment where
  - Neither the hardware/IT provider, the framework provider, nor the application provider are the same
  - Multiple agencies or companies provide applications which depend on the framework

Recommendations: Bottom Line

- Recommendation 0:

  Retain control of ICDs but separate the Data Definition (referenced Data Spec) from the Physical Interface Definition (ICD body).

- Recommendation 1:

  Retain the benefits of model-based data definitions; govern the *UML data model* via comments to the generated ICDs/Data Specifications.

- Recommendation 2:

  Align data-model ownership along boundaries of domain, stakeholder groups, and mission

- Recommendation 3:

  Break up the UML model into multiple UML model projects, to align with multiple ownership and decrease coupling between programs.

> This requires a careful analysis of the data content & organization, mission-ownership, software development CONOPs, and the desired qualities of an enterprise solution.

Desired Qualities of An Enterprise Data Management Solution

- Support for
  - Common Data Definitions (facilitates fusion and common processing)
  - Interoperability
  - Developer CONOPS. Need to support
    - Both manual updates(*) and model-based
    - Development/testing as well as operational needs
  - System-of-Systems Enterprise
  - SME Control of Their Domain's Data Curation
  - Flexible to Changing Stakeholders
- Alignment of Data Ownership with Mission Ownership
- Improved Decoupling of Data Needs from Differing Program Boundaries and Delivery Cycles
- Adoption of Industry Best-Practices

_____
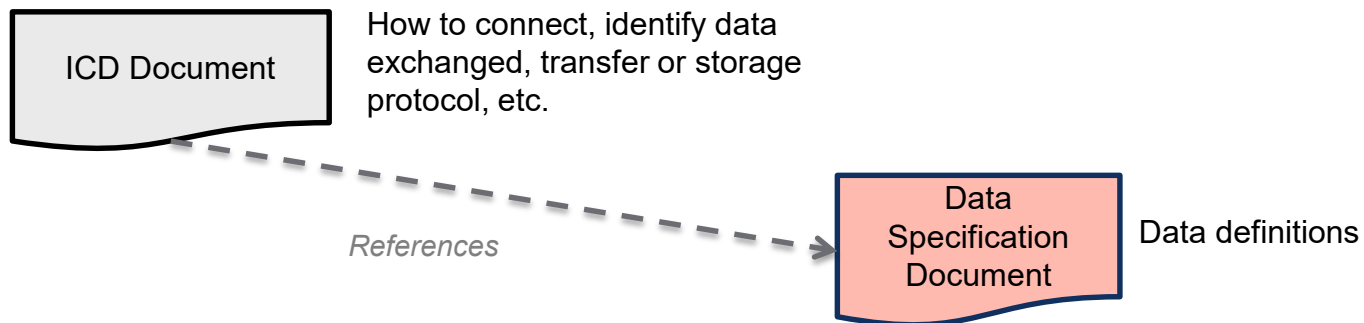
(*) Actually, all updates are model-based; these require a separate model updater.

End-to-End Data Management – Simple View
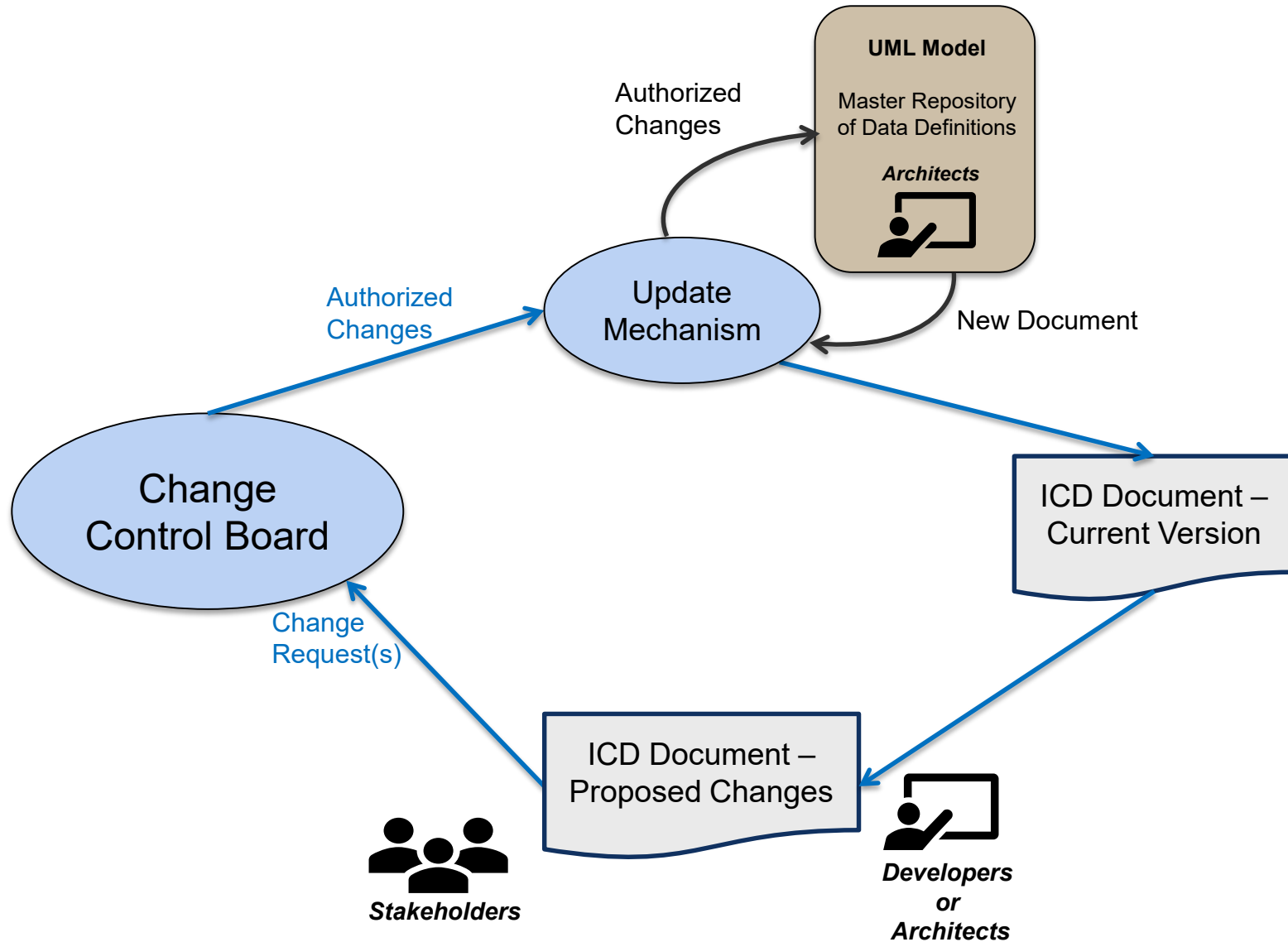
Breaking up the ICD to Support Data Re-Use

- Data tends to flow through systems
  - e.g., storage to streaming to database
- Separating the physical interface definition (ICD) from the data definition (Data Specs) is beneficial. It supports:
  - Re-use of the data definition in multiple interfaces (e.g., bent pipe with disparate transfer mechanisms on either end)
  - Instantiation of the data definition with multiple physical forms (e.g., protocol buffer streaming into a database for storage)

ICD Document

How to connect, identify data exchanged, transfer or storage protocol, etc.

References

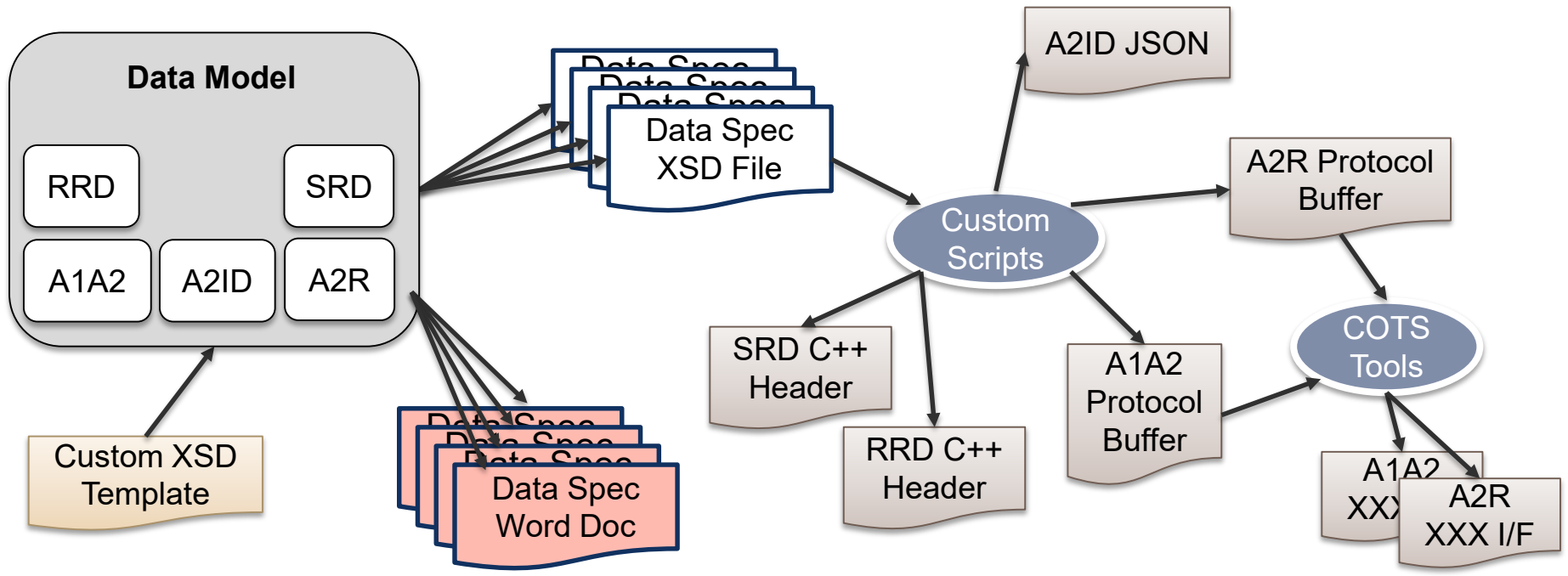Data Specification Document

Data definitions

Why use a Tool that Allows One to Model the Interfaces?

- Benefits of defining data with UML in a model (over documents)
  - Model management is scalable
  - Model provides means for complex analysis of relationships
  - Interface code can be generated from the model
    - The integration is less error-prone because both sides of the interface have the exact same data definition
  - Documentation can be generated from the model
    - This eases the burden of maintaining it as the interface code advances

- The documentation is updated once (in the model) and then reflected both in the Word documents produced as well as in the generated code

Data Specs Documents & Code are Generated from the Data Model

Master Repository → *Generated Outputs*

Data Model

- RRD
- SRD
- A1A2
- A2ID
- A2R

Custom XSD Template

Data Spec XSD File

Data Spec Word Doc

Custom Scripts

A2ID JSON

A2R Protocol Buffer

SRD C++ Header

RRD C++ Header

A1A2 Protocol Buffer

COTS Tools

A1A2 XXX

A2R XXX I/F

End to End Data Management – Support for Multi-Use Data & Developer/Integration Needs

(U) How to connect, transfer or storage protocol, etc.

ICD Document Proposed Changes

(U) Data definitions

Data Spec
Data Spec
Proposed Changes
Word Doc

(U) Data definitions

Data Spec
Data Spec
Proposed Changes
XSD File

Stakeholders

Change Request

Developers or Architects

Change Request

Change Control Board

Authorized Changes

Data Adjudication Staff

Data Model

Master Repository of Data Definitions

Architects

Data Spec Current Version Word Doc

Data Spec Current Version XSD File

Code generation

Collaborative Site

Stakeholders

Contractor Repo

Developers

14

Granular Data Models Require Granular Governance
Multi Sensor, Multi Mission, Multi System, Multi Stakeholder

Original Ground Station Model

SRD	RRD	A2R	A1A2
ODP	SDP	A2ID	COV
UI		EC	FW Services

Data Specification Models

Sensor-Specific Classes

- SF1-2 Specific
- SF3 Specific
- SF4 Specific
- SF11 Specific
- Classes unique to each Data Spec
- SF12 Specific
- Common RRD/SRD Classes
- Core Classes

Data Spec -Specific Classes

Classes Shared Across Data Specs & Logical Arch

Logical Architecture (Use Cases, UCRs, DFDs, etc.)

Logical Models

How to Manage All These Interfaces?

- First we thought we could assign a primary owner to each Data Specification that defines data which flows over an interface
  - Recognizing that there is always at least one other party who cares (a stakeholder)
  - So anyone who is a primary owner needs to involve other parties in authorizing changes
- We spent a few weeks negotiating that split; it is on the next slide.
- Then we realized there is the complication of classes which are part of multiple Data Specifications
  - Even some that underlie nearly all of them
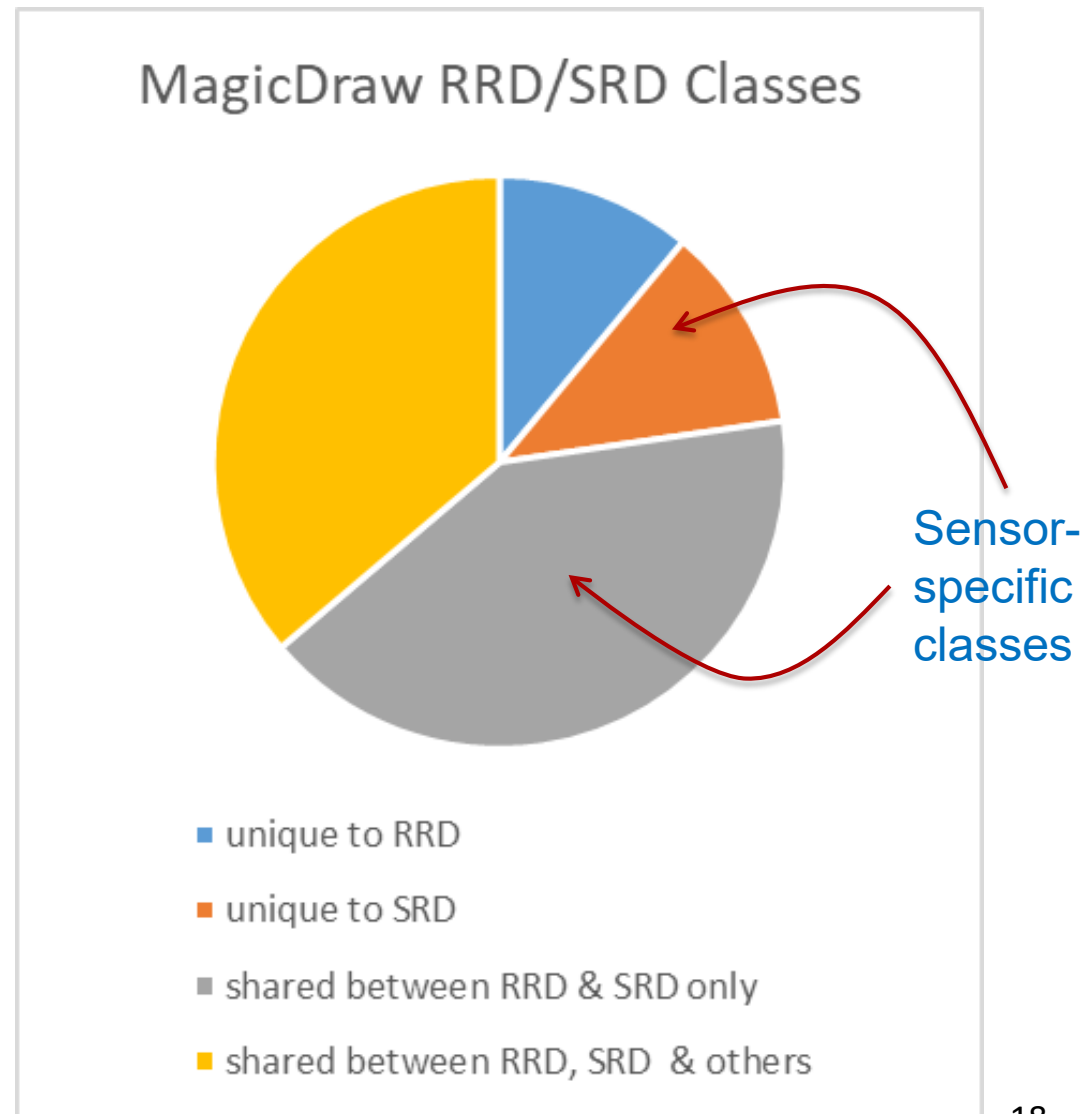 - That is expressed on the next++ slide

Recommended Governance of Data Specs – The Simplified View

Primary Governance Sphere	Data Spec	Data Spec Name
Enterprise (Multi-Sponsor)	RRD	Raw Raster Data
Enterprise (Multi-Sponsor)	SRD	Sensor Reference Data
App Set #1 Sponsor	ODP	Object Data Product
App Set #1 Sponsor	SDP	Scene Data Product
App Set #2 Sponsor	A1A2	App Set #1 to App Set #2 Data
App Set #2 Sponsor	A2ID	App Set #2 Internal Data
App Set #2 Sponsor	A2R	App Set #2 Results
Framework Sponsor	FW	Framework Services
Framework Sponsor	UI	User Interface Services
App Set #1 Sponsor	COV	Sensor Coverage
External Sponsor	EC	External Commanding

Sharing of data structures (classes)

- RRD is sharing roughly as many classes with downstream Data Types (as well as SRD) as with SRD alone

- This means it is not safe to assume one can modify any RRD/SRD class without affecting other Data Types

MagicDraw RRD/SRD Classes

Sensor-specific classes

- unique to RRD
- unique to SRD
- shared between RRD & SRD only
- shared between RRD, SRD & others

Core Classes Shared Across Multiple Data Specifications

- Generic science terms
  - Position, Velocity, Acceleration, GeodeticPosition, StateVector, SymmetricMatrix2d, SymmetricMatrix3d
  - Geolocation (Polynomials, LOS Uncertainties, Ephemeris)
  - UnitVectors (various types)

  Spatio-Temporal Support Classes

- Description of the Collection
- Description of the Detector (Band, Config, Id)
- Description of the ExposureUnit/Datastream
- Description of Rasters
- Time

UML Model Decomposition

- The unit of versioning in the tool we are using, MagicDraw, is the Project

- MagicDraw supports working with a collection of Projects, where there is a primary Project and other Projects which that Project depends upon.

    - Any party can generate code and documents using their own Project and Projects that others own.

- Provides clean delineation of authority and responsibility

- Provides a common mechanism for managing data across the enterprise

- Challenges will remain because conflicts may occur wherever there are multiple stakeholders with disparate concerns

Managing Shared & Unique Projects

- Owners manage independent development
  - Classes unique to a single Data Spec or the sensor-specific classes
- Co-owners will require governance procedures
  - Shared RRD/SRD Classes
  - Core Classes & selected Enumerations
  - Involves contractual issues
- Can support manual data-prototype development
  - Design & develop App stand-alone, using any relevant Data Specs
  - Modify XSDs by hand to prototype & test
  - Create Documentation manually
  - Rely on a Centralized MagicDraw Model Authority to update these Interfaces in the model

Breaking Up the MagicDraw Model
Plan In Progress: Also need separate Projects for Data Specs
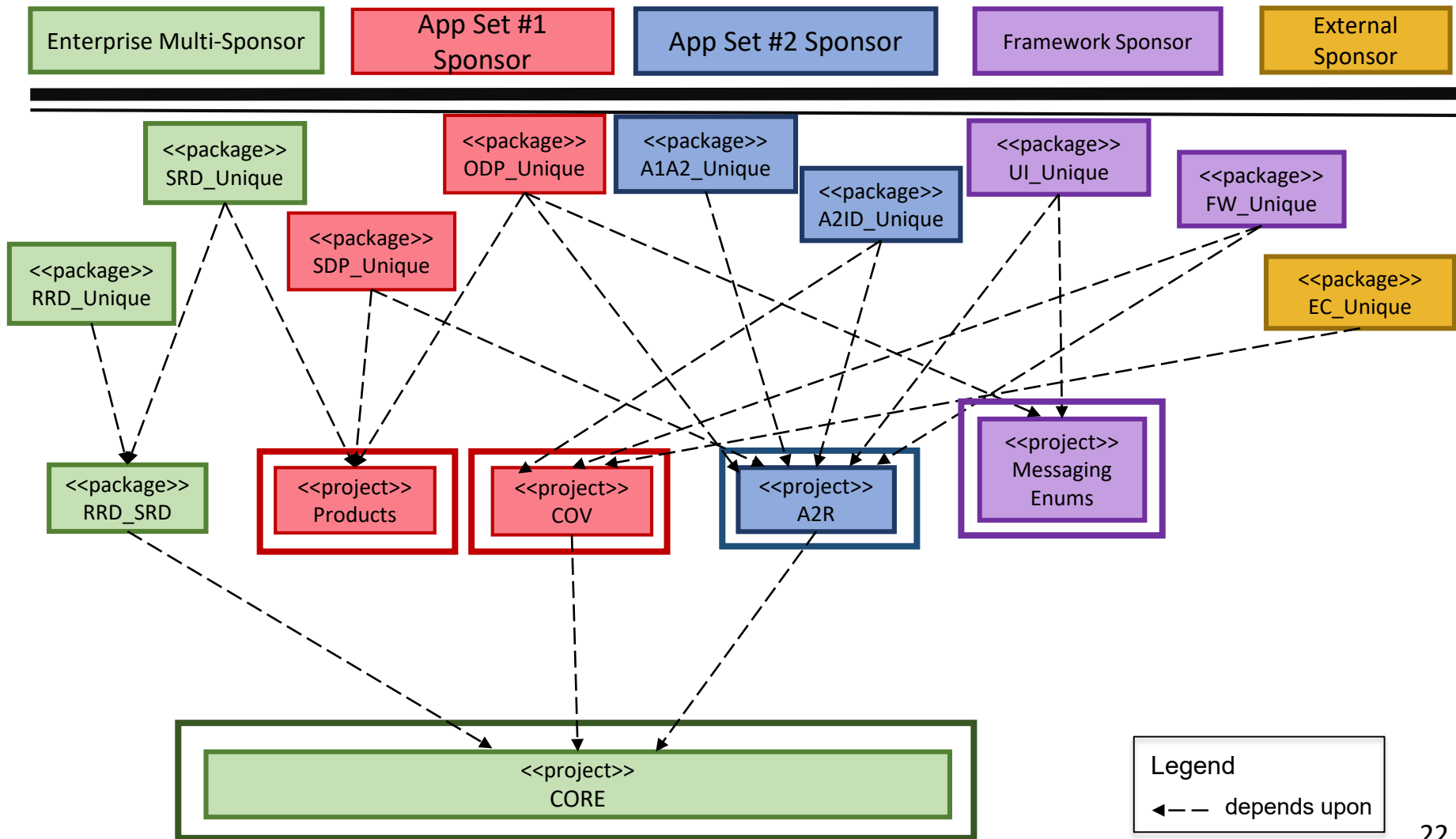
End to End Data Management – Support for Multi-Use Data, Developers, and Incremental Development
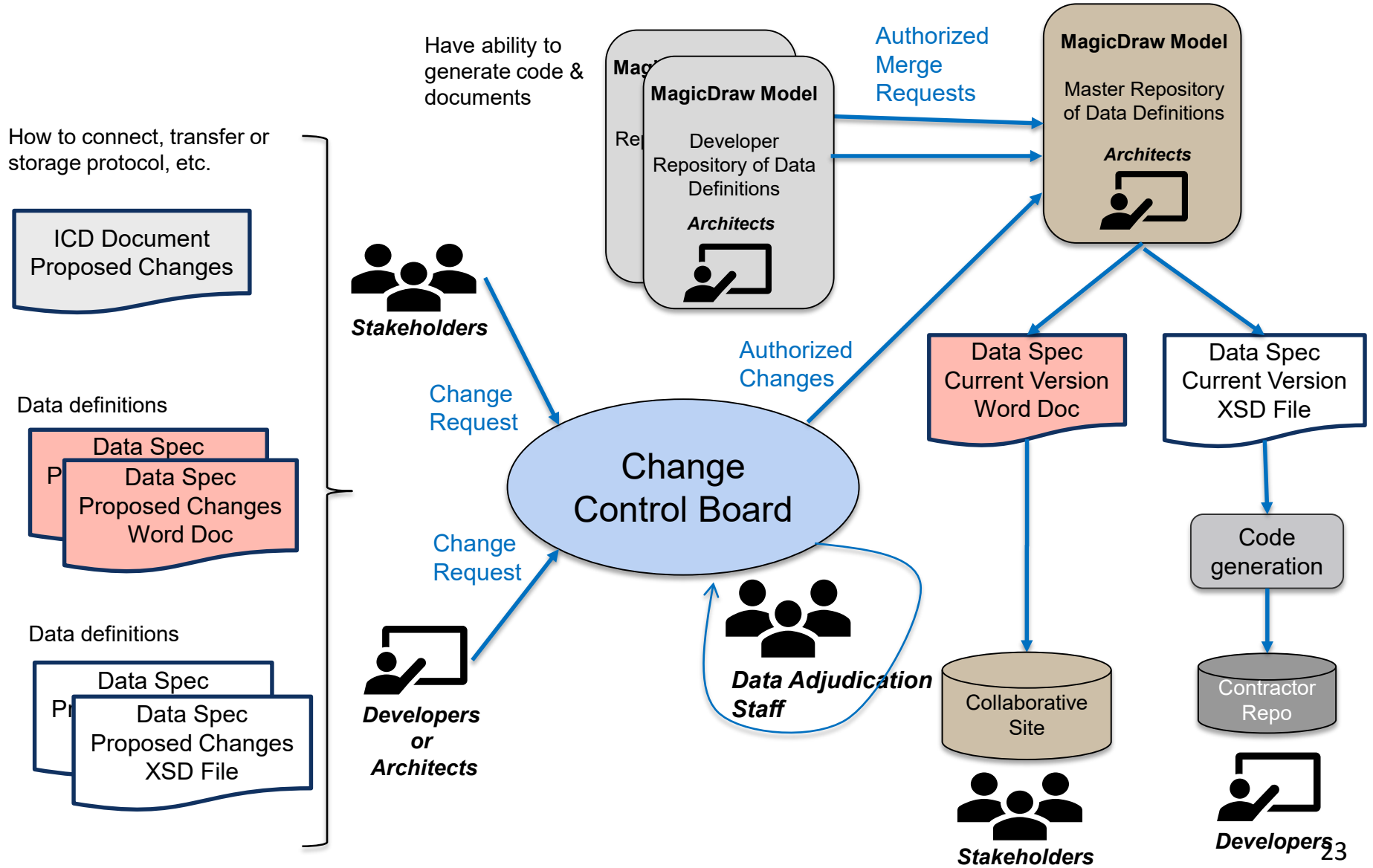
Current Status

- Developer Status
  - The definition of all the required Projects has been refined and circular or inappropriate class dependencies fixed
  - Currently defining the configuration management of the Projects
  - The master repository location/form and how it will be accessed needs to be determined

- Sponsor Status
  - Examining existing governance boards and mechanisms to determine their utility in this realm.
  - Determining which organization would support on-going data governance.

Summary

- Supporting a flexible, extensible, common sensor processing capability in a *Multi Sensor, Multi Mission, Multi System, Multi Stakeholder* environment is challenging. At a minimum, it requires common data definitions.

- This talk does not address all aspects of those challenges but focuses on the engineering aspects of supporting such data governance

- Key ideas to achieve this goal:
    - Separate physical interface definition from data definition
    - Use a UML modeling tool to define data and interfaces
    - Generate code and documentation for the interfaces from that modeling tool
    - Identify stakeholders and primary owners of interfaces
    - Separate the projects in the modeling tool along the lines of domain, interface, mission, and shared (common) data
    - Support Change Control so that stakeholders/owners retain control
    - Facilitate dev teams ability to prototype new data definitions with copies of the projects

Today's Presentation

Things to think about

- How can this be applied in your work environment?

- What did you hear that will influence your thinking?

- What is your take away from this presentation?

Please

The link for the online survey for this meeting is

- www.surveymonkey.com/r/2019_06_MeetingEval

- www.surveymonkey.com/r/2019_06_MeetingEval

Look in GlobalMeet chat box for cut & paste link

Slide presentation can be downloaded now/anytime from:

- The library page at: www.incose.org/enchantment

- Recording will be there in the library soon