intercax

System Lifecycle Handler for enabling a digital thread for smart manufacturing

NIST MBE Summit 2018

April 4, 2018

Manas Bajaj, PhD & **Dirk Zwemer**, PhD - Intercax

Tom Hedberg - NIST

Contents

- **System Lifecycle Handler - Foundations**
  - **Project goals**
  - **Use Cases**
  - **Graph foundations**
- System Lifecycle Handler – Proof-of-concept demonstrations
  - Querying the digital thread
  - Building the digital thread
  - Maintaining the digital thread
  - API for the digital thread
  - Tools, APIs, and open standards used in the PoC
- Next steps
- Questions and Comments

NIST Digital Thread Project

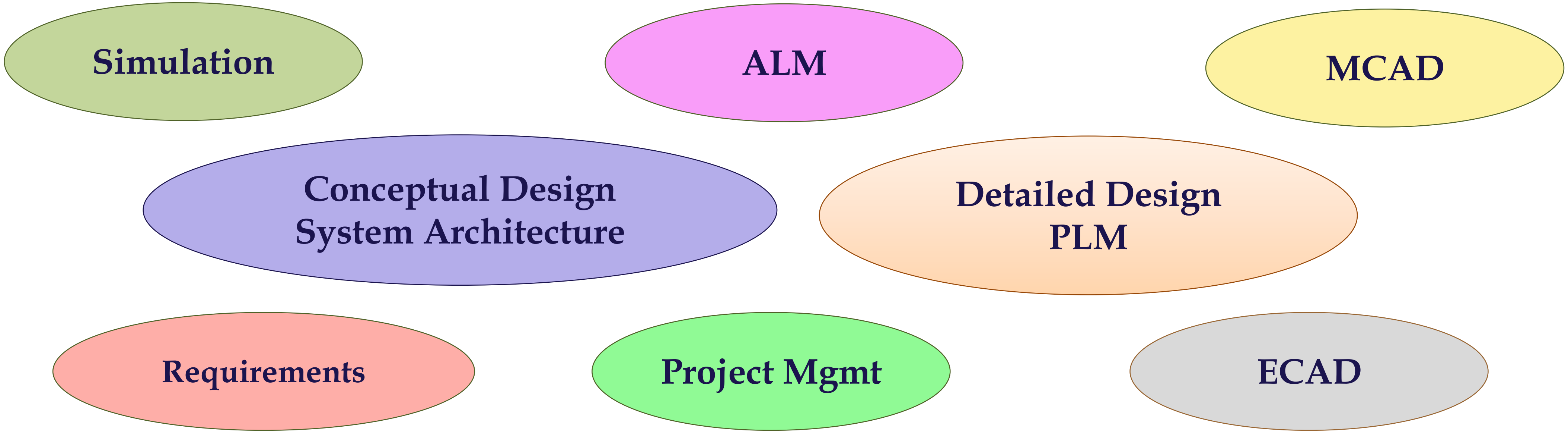
- Developing methods & open standards to support validating, certifying, and connecting engineering models across lifecycle
- Goals
  - Seamless traceability:
    System -> Design -> Manufacturing -> Operations -> Maintenance
  - High-quality manufacturing
  - Enterprise knowledge reuse
- Learn more at: http://www.nist.gov/el/msid/syseng/dtsm.cfm

System Lifecycle Handler (SLH) – Use Cases

- ***Build*** digital thread for systems
  - ***Connect*** to data across the enterprise (Systems, PLM, CAD, ALM, Project management, Manufacturing, Operations) and spin a digital thread
  - ***Generate*** models as information moves across disciplines
- ***Query*** *and* ***search*** the digital thread
- ***Manage*** the digital thread
  - ***Track changes*** in versioned models connected in the digital thread
  - ***Compare, synchronize, repair*** connections and models
- ***Visualize*** the digital thread

Silos of models and data

- Organizations deal with a diverse, multi-vendor engineering toolset.
- They create and store product/system data in a variety of tools, models and repositories: PLM, ALM, CAD, spreadsheets, databases, SysML models…



Simulation

ALM

MCAD

Conceptual Design
System Architecture

Detailed Design
PLM

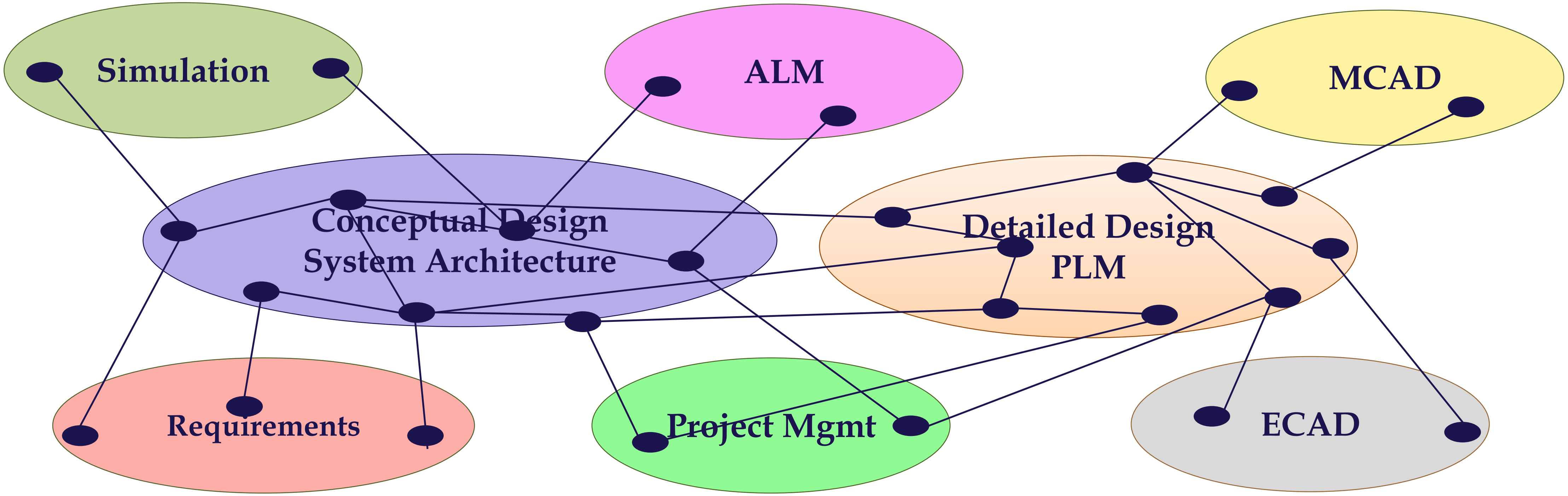
Requirements

Project Mgmt

ECAD

Digital Thread – Connecting the domains for true MBE

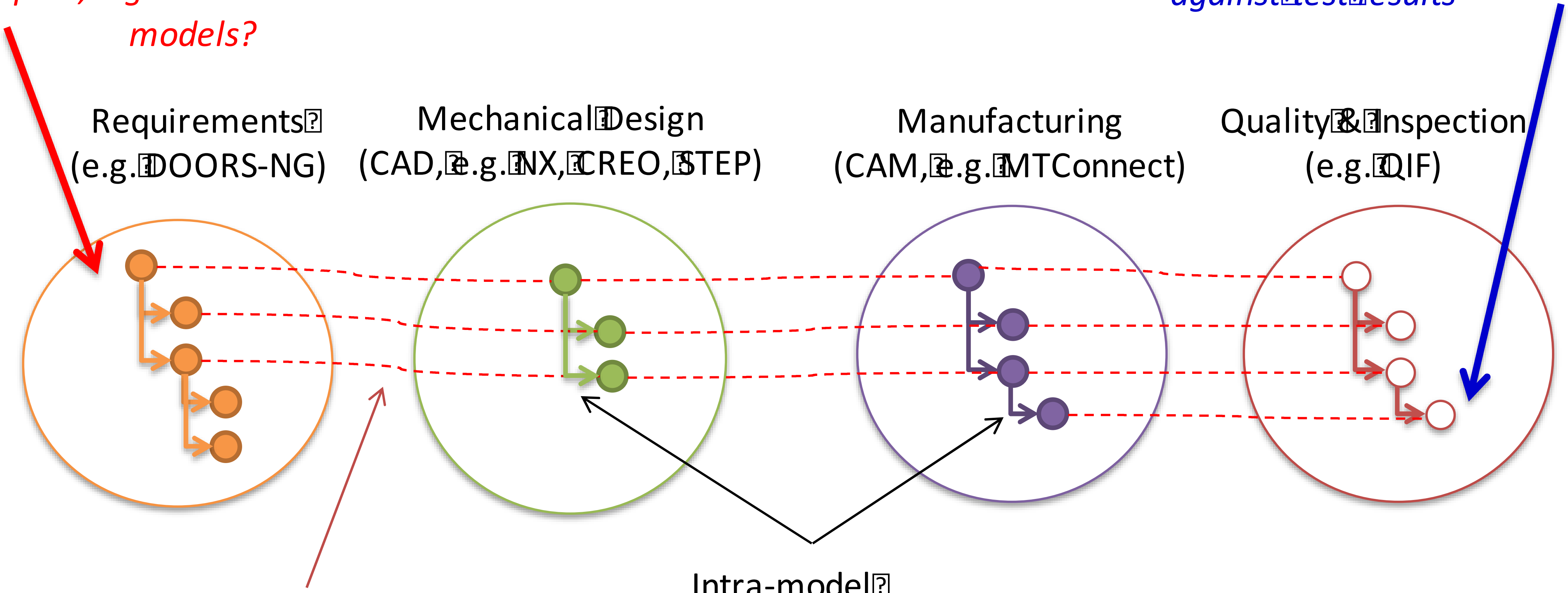
- Federating data and models across disciplines
- Connecting model elements and data at different levels of abstractions



Digital Thread – A simple example showing artifacts and connections

If I change this requirement, what is the downstream impact, e.g. to CAD and CAM models?

Trace the CAD and CAM models for this part and compare attributes against test results



Requirements (e.g. DOORS-NG)

Mechanical Design (CAD, e.g. NX, CREO, STEP)

Manufacturing (CAM, e.g. MTConnect)

Quality & Inspection (e.g. QIF)

Inter-model connections

Intra-model connections

Connections / Links in the Digital Thread

- What is the purpose of model-based connections?

↔ Reference Connections

Track/compare/sync versions of connected elements

↔ Data Map Connections

+ Track/compare/sync element attributes

↔ Function Wrap Connections

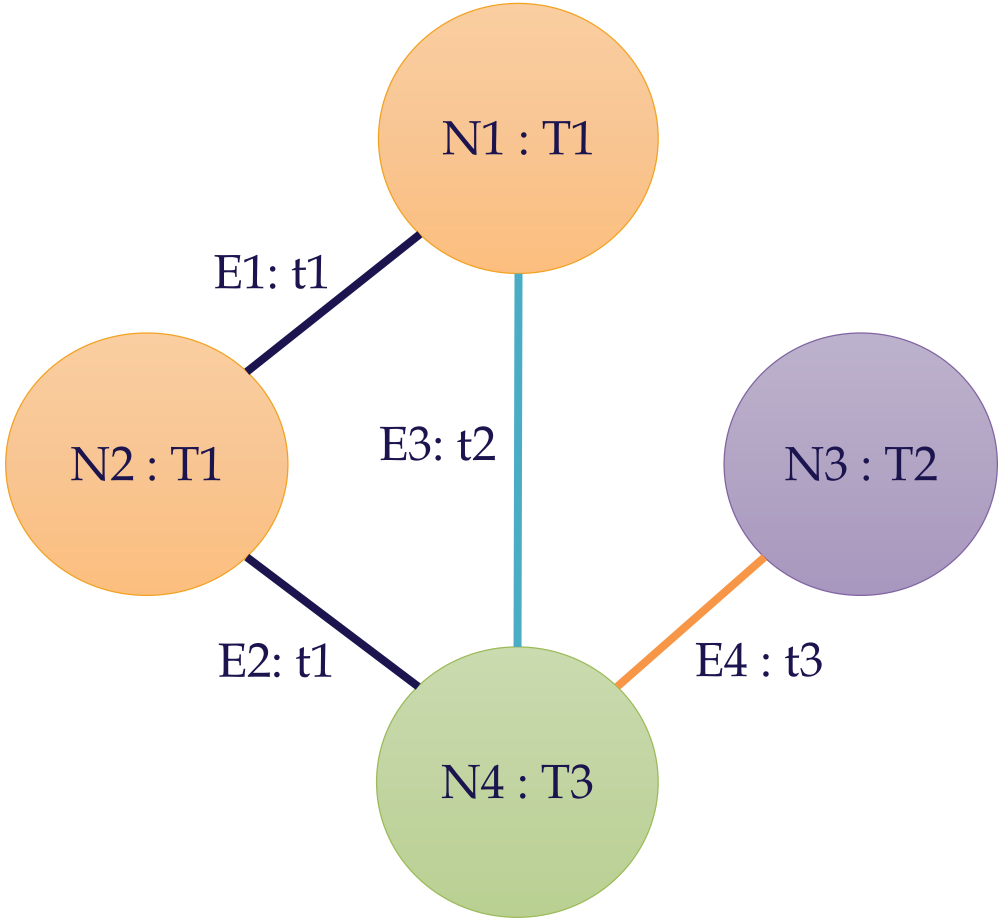
+ Track/execute connection elements

↔ Model Transform Connections

+ Track/compare/sync element structure (multi-level)

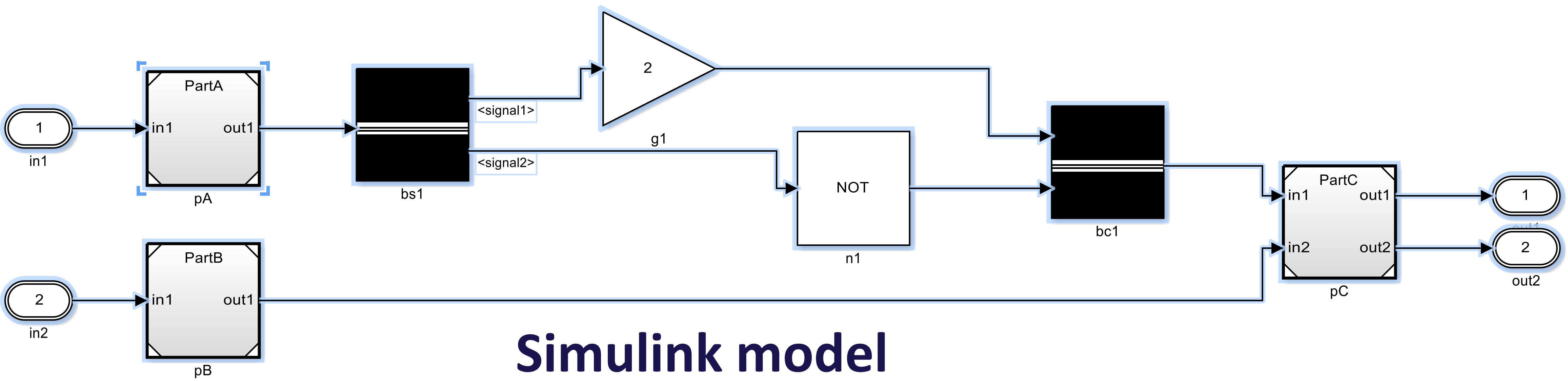
Digital Thread is a *Graph*

- Graph – Nodes and Edges
- Nodes and Edges may have
  - Name
  - Type (Typed Graph)
  - Properties (Property Graph)
- Edges may have
  - Direction (Directed vs. Undirected Graph)
- Graphs can be
  - Stored
  - Queried (Pattern matching)
  - Traversed (e.g. Breadth-first, Depth-first)
  - Generated and Transformed
  - Analyzed



Information models can be abstracted as graphs

- Projects
  - ⊞ Big Bang (BIG)



Simulink model

- ⊟ Watched by me
  - ⊟ ☑ SDB-655
    - Project: Syndeia Demo Box
    - Summary: Autopilot
    - Type: Task
    - Status: Open
    - Priority: Major
    - Updated: 2017-05-15T16:53:05
    - Assignee: dirkzwemer
    - Reporter: manasbajaj

odule

rtifacts

pecification Module

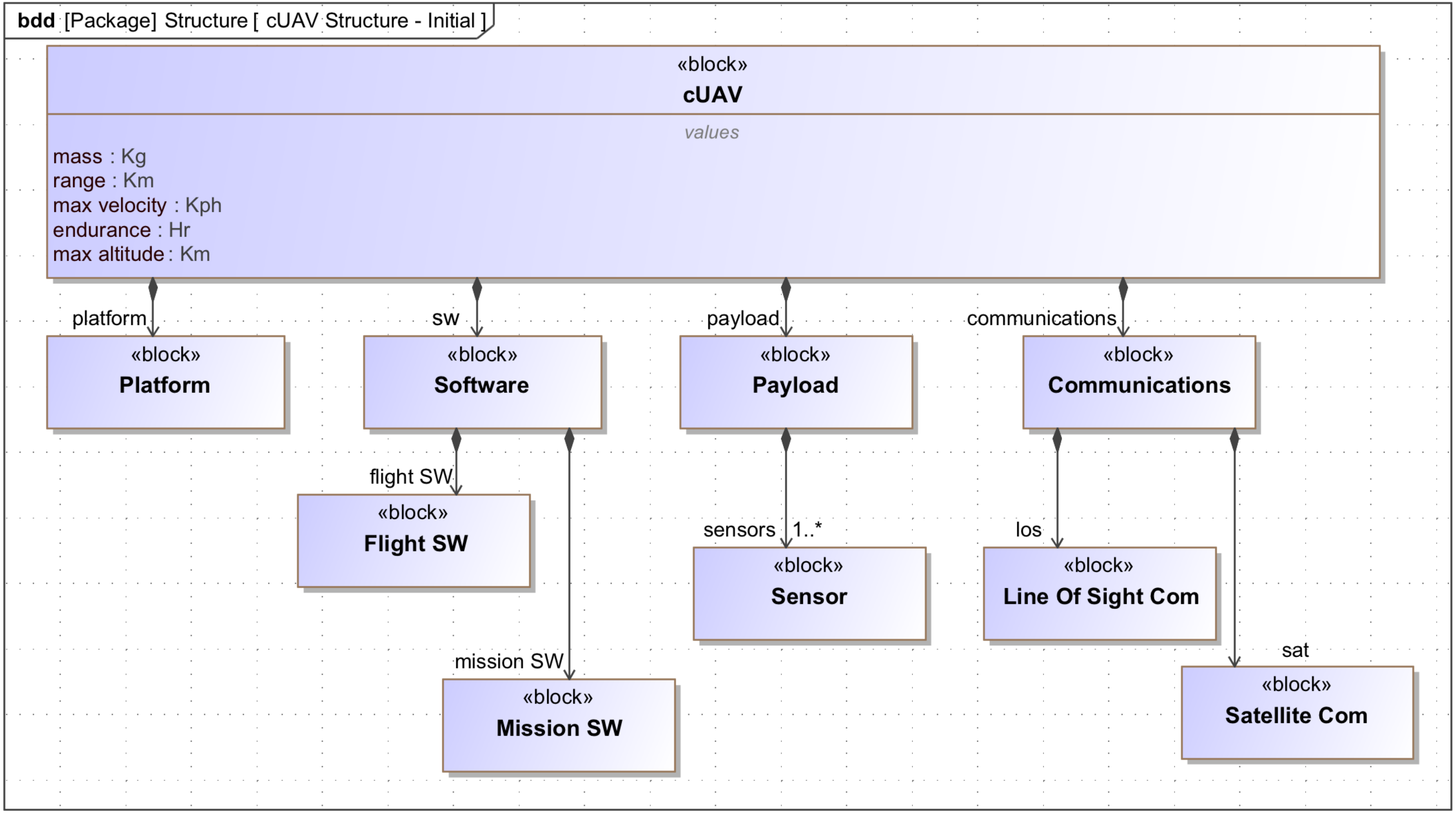
ge Load

System Model (SysML)

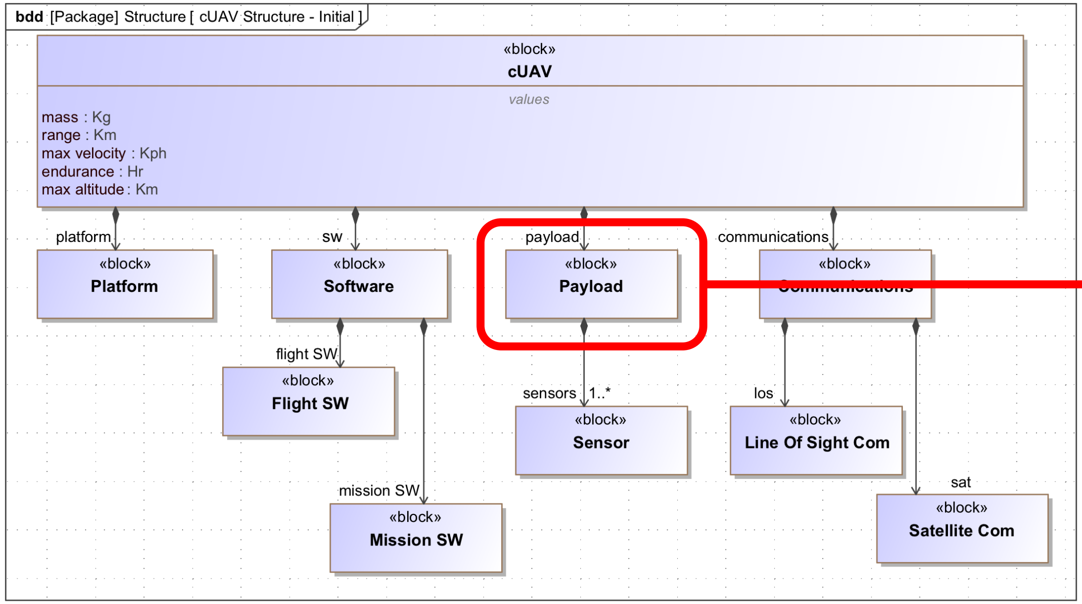
Contents

- System Lifecycle Handler - Foundations
  - Project goals
  - Use Cases
  - Graph foundations
- **System Lifecycle Handler – Proof-of-concept demonstration**
  - **Querying the digital thread**
  - **Building the digital thread**
  - **Maintaining the digital thread**
  - **API for the digital thread**
  - **Technology stack and open standards**
- Next steps
- Questions and Comments

Testbed model for proof-of-concept – Configurable UAV



Testbed model for proof-of-concept – Configurable UAV



bdd [Package] Structure [cUAV Structure - Initial]

«block»
cUAV

values
mass : Kg
range : Km
max velocity : Kph
endurance : Hr
max altitude : Km

platform
«block»
Platform

sw
«block»
Software

flight SW
«block»
Flight SW

mission SW
«block»
Mission SW

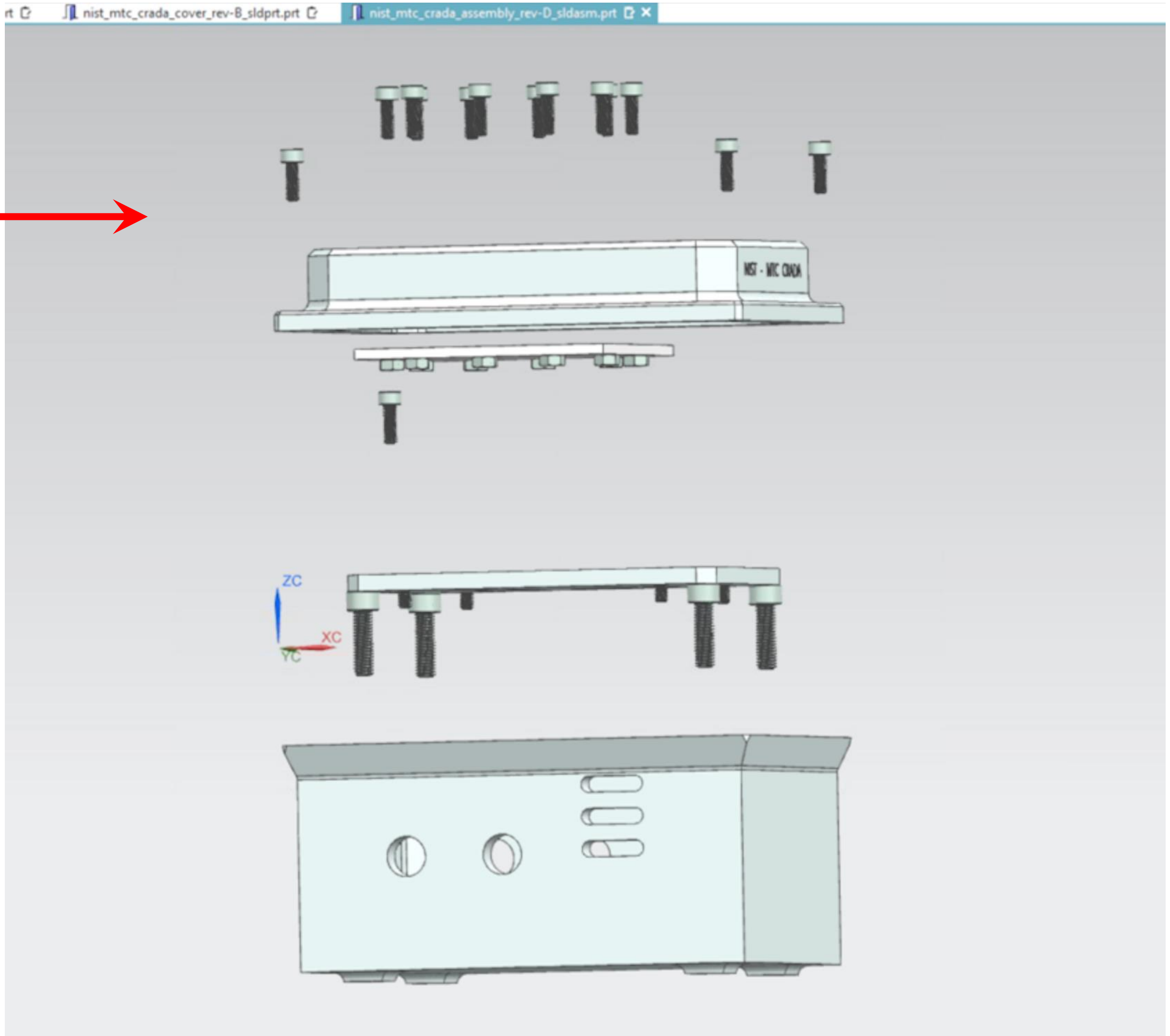
payload
«block»
Payload

sensors 1..*
«block»
Sensor

communications
«block»
Communications

los
«block»
Line Of Sight Com

sat
«block»
Satellite Com

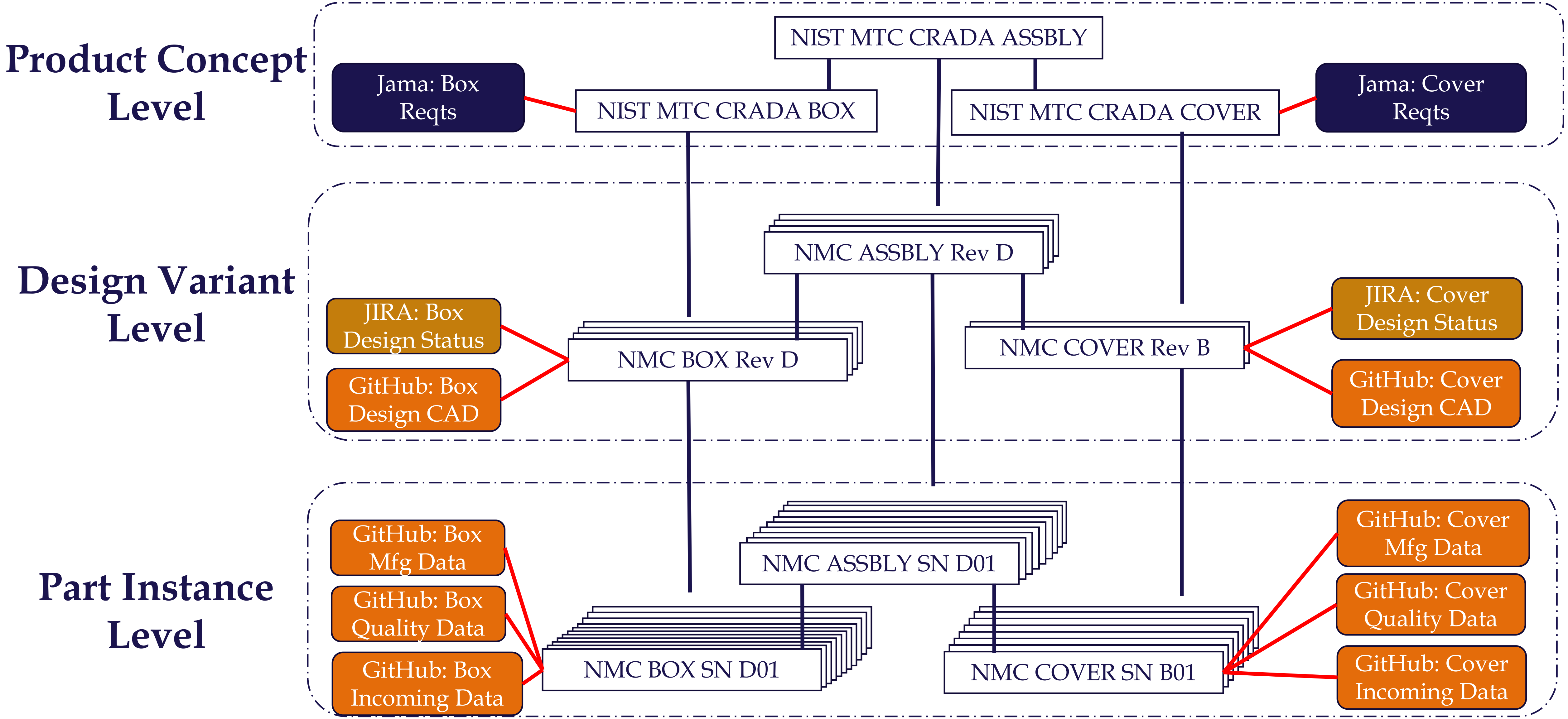

nist_mtc_crada_cover_rev-B_sldprt.prt
nist_mtc_crada_assembly_rev-D_sldasm.prt
NIST - MTC CRADA
ZC
XC
YC

Enclosure Box for an Avionics assembly used in the Configurable UAV

Available data and models for the Enclosure Box

- **SysML** model of the UAV and the payload
- CAD models for multiple variants and revisions of the Box (**SolidWorks files on GitHub**)
- Design flow management in **JIRA**
- 20 instances of each part are manufactured. For each instance:
  - Machine sensor data streams for each part instance (**MTConnect 1.3 XML on GitHub**)
  - NC code data - ISO 6983 **(G-code files on GitHub)**
  - First article inspection reports (**QIF 2.1 XML on GitHub**)
  - Receiving inspection reports (**QIF 2.1 XML on GitHub**)

Approach for organizing the digital thread – Schema and Linked Models



Product Concept Level

NIST MTC CRADA ASSBLY

Jama: Box Reqts

NIST MTC CRADA BOX

NIST MTC CRADA COVER

Jama: Cover Reqts

Design Variant Level

NMC ASSBLY Rev D

JIRA: Box Design Status

GitHub: Box Design CAD

NMC BOX Rev D

NMC COVER Rev B

JIRA: Cover Design Status

GitHub: Cover Design CAD

Part Instance Level

GitHub: Box Mfg Data

GitHub: Box Quality Data

GitHub: Box Incoming Data

NMC ASSBLY SN D01

NMC BOX SN D01

NMC COVER SN B01

GitHub: Cover Mfg Data

GitHub: Cover Quality Data

GitHub: Cover Incoming Data

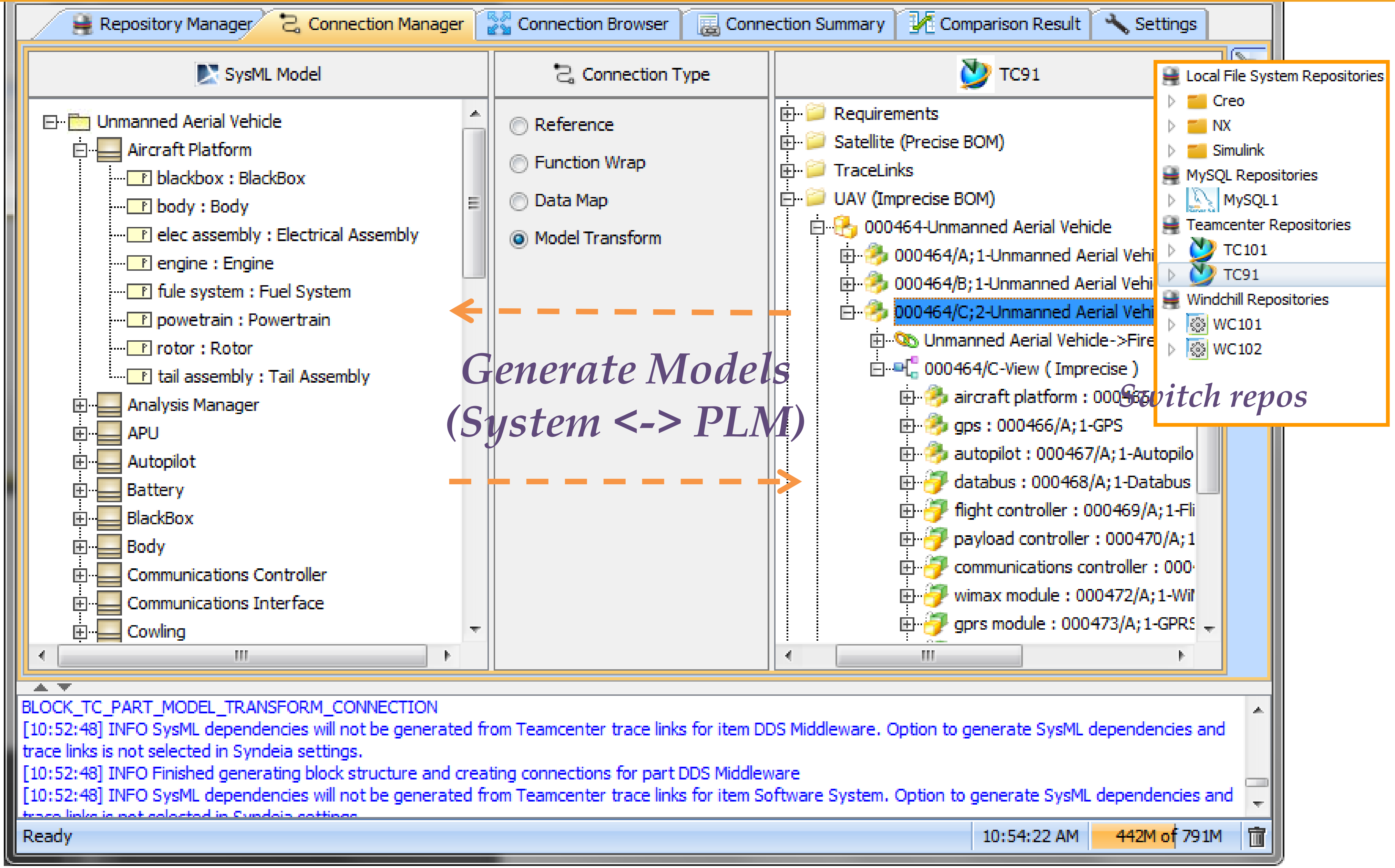
intercax

Demo video #1 – Querying the digital thread for the simple avionics box assembly

intercax

Demo video #2 - Building the digital thread for the simple avionics box assembly

Model Transformations – Automatically spinning the digital thread



Generate Models (System <-> PLM)

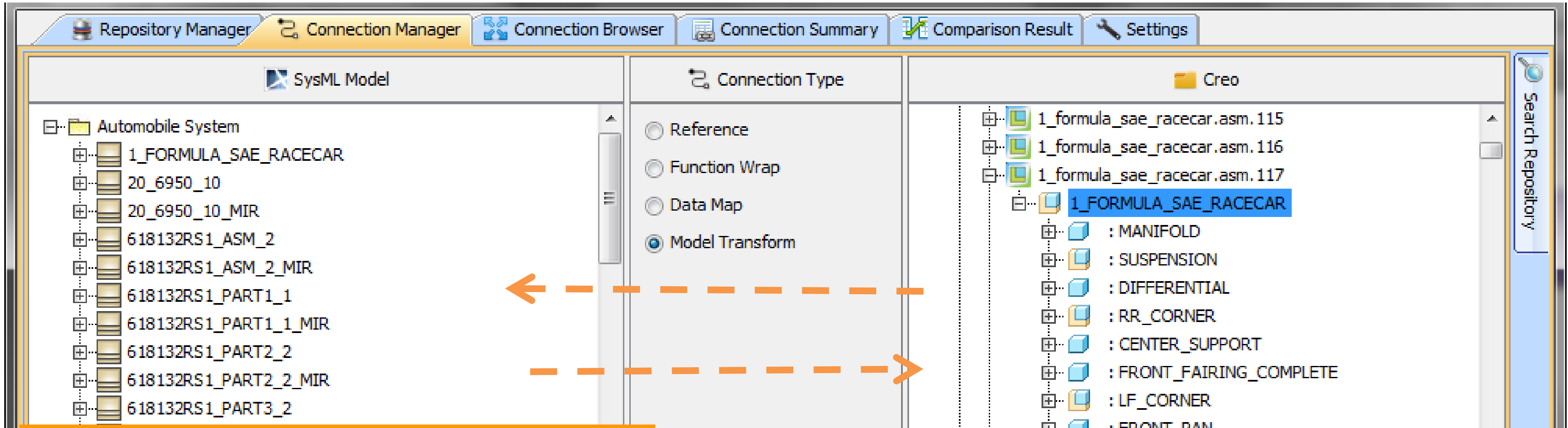
Switch repos

```
BLOCK_TC_PART_MODEL_TRANSFORM_CONNECTION
[10:52:48] INFO SysML dependencies will not be generated from Teamcenter trace links for item DDS Middleware. Option to generate SysML dependencies and
trace links is not selected in Syndeia settings.
[10:52:48] INFO Finished generating block structure and creating connections for part DDS Middleware
[10:52:48] INFO SysML dependencies will not be generated from Teamcenter trace links for item Software System. Option to generate SysML dependencies and
trace links is not selected in Syndeia settings.
```

Digital Thread as a conduit for information flow

Requirements
(SE -> ME)

Mass properties
(ME -> SE)



SysML Model

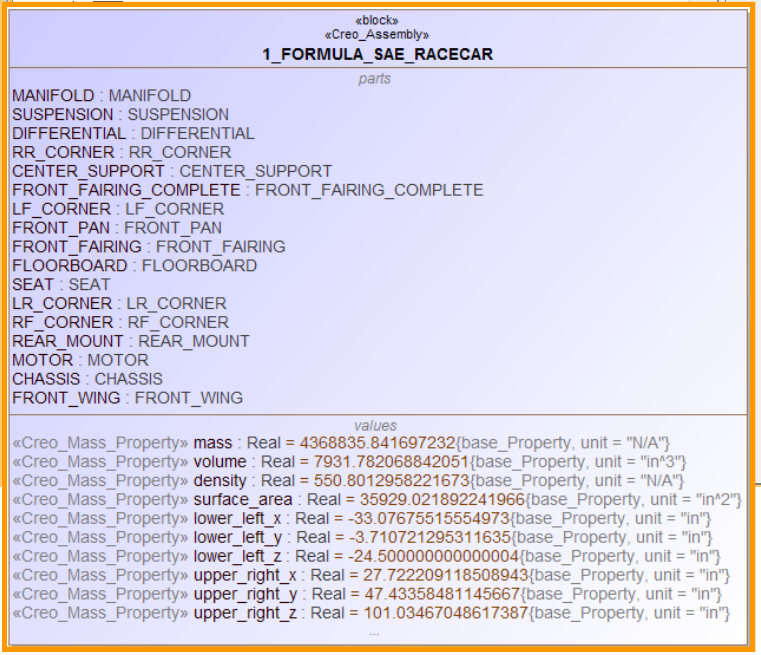
- Automobile System
  - 1_FORMULA_SAE_RACECAR
  - 20_6950_10
  - 20_6950_10_MIR
  - 618132RS1_ASM_2
  - 618132RS1_ASM_2_MIR
  - 618132RS1_PART1_1
  - 618132RS1_PART1_1_MIR
  - 618132RS1_PART2_2
  - 618132RS1_PART2_2_MIR
  - 618132RS1_PART3_2

Connection Type

○ Reference
○ Function Wrap
○ Data Map
● Model Transform

Creo

- 1_formula_sae_racecar.asm.115
- 1_formula_sae_racecar.asm.116
- 1_formula_sae_racecar.asm.117
  - 1_FORMULA_SAE_RACECAR
    - : MANIFOLD
    - : SUSPENSION
    - : DIFFERENTIAL
    - : RR_CORNER
    - : CENTER_SUPPORT
    - : FRONT_FAIRING_COMPLETE
    - : LF_CORNER
    - : FRONT_PAN



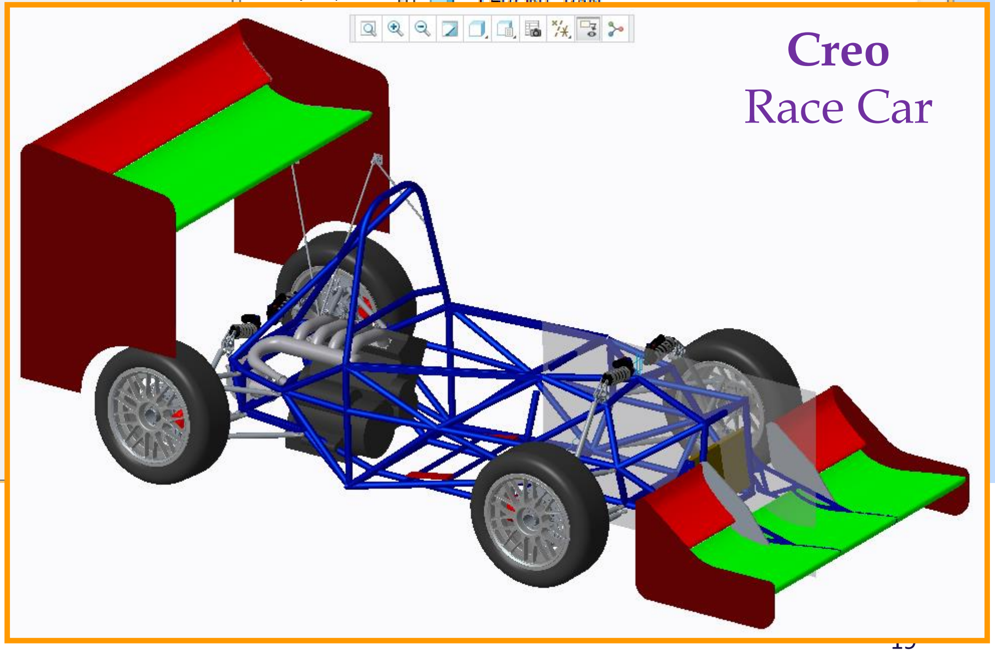
«block»
«Creo_Assembly»
1_FORMULA_SAE_RACECAR

parts

MANIFOLD : MANIFOLD
SUSPENSION : SUSPENSION
DIFFERENTIAL : DIFFERENTIAL
RR_CORNER : RR_CORNER
CENTER_SUPPORT : CENTER_SUPPORT
FRONT_FAIRING_COMPLETE : FRONT_FAIRING_COMPLETE
LF_CORNER : LF_CORNER
FRONT_PAN : FRONT_PAN
FRONT_FAIRING : FRONT_FAIRING
FLOORBOARD : FLOORBOARD
SEAT : SEAT
LR_CORNER : LR_CORNER
RF_CORNER : RF_CORNER
REAR_MOUNT : REAR_MOUNT
MOTOR : MOTOR
CHASSIS : CHASSIS
FRONT_WING : FRONT_WING

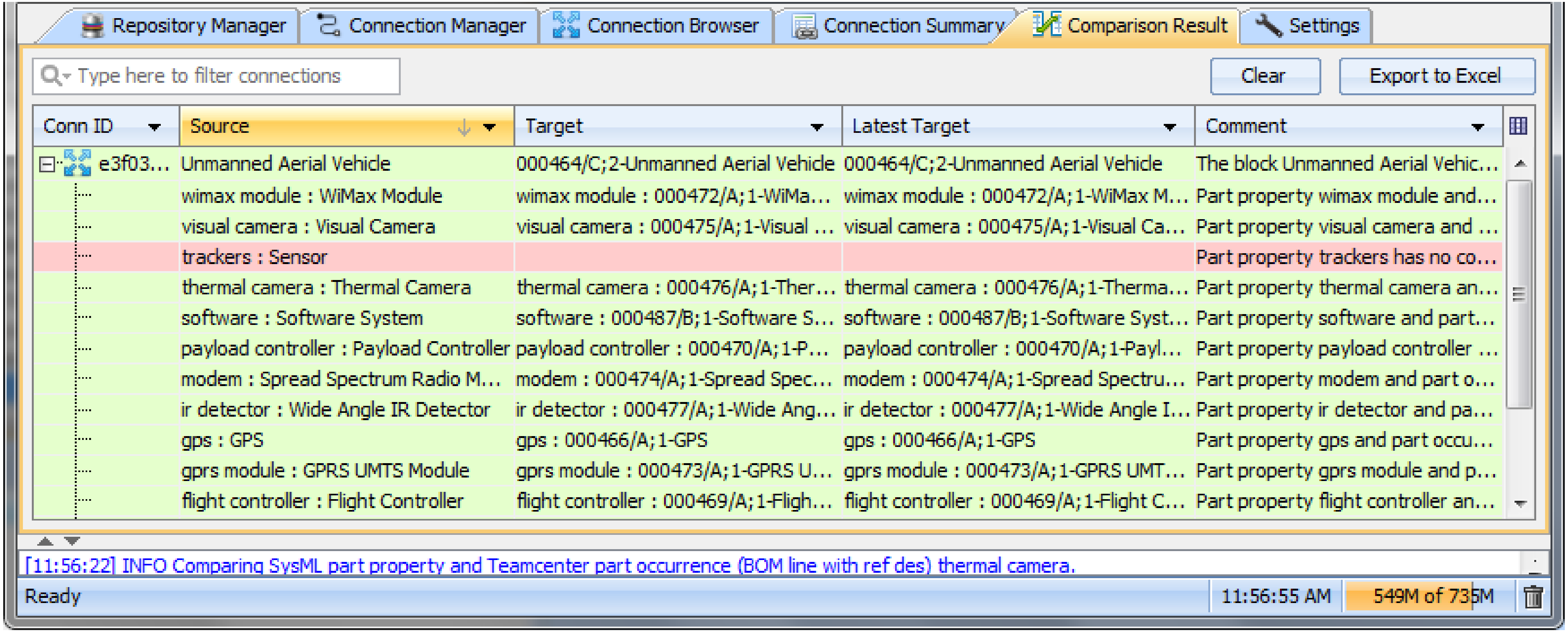
values

«Creo_Mass_Property» mass : Real = 4368835.841697232{base_Property, unit = "N/A"}
«Creo_Mass_Property» volume : Real = 7931.782068842051{base_Property, unit = "in^3"}
«Creo_Mass_Property» density : Real = 550.8012958221673{base_Property, unit = "N/A"}
«Creo_Mass_Property» surface_area : Real = 35929.021892241966{base_Property, unit = "in^2"}
«Creo_Mass_Property» lower_left_x : Real = -33.07675515554973{base_Property, unit = "in"}
«Creo_Mass_Property» lower_left_y : Real = -3.710721295311635{base_Property, unit = "in"}
«Creo_Mass_Property» lower_left_z : Real = -24.500000000000004{base_Property, unit = "in"}
«Creo_Mass_Property» upper_right_x : Real = 27.722209118508943{base_Property, unit = "in"}
«Creo_Mass_Property» upper_right_y : Real = 47.43358481145667{base_Property, unit = "in"}
«Creo_Mass_Property» upper_right_z : Real = 101.03467048617387{base_Property, unit = "in"}
...



Creo
Race Car

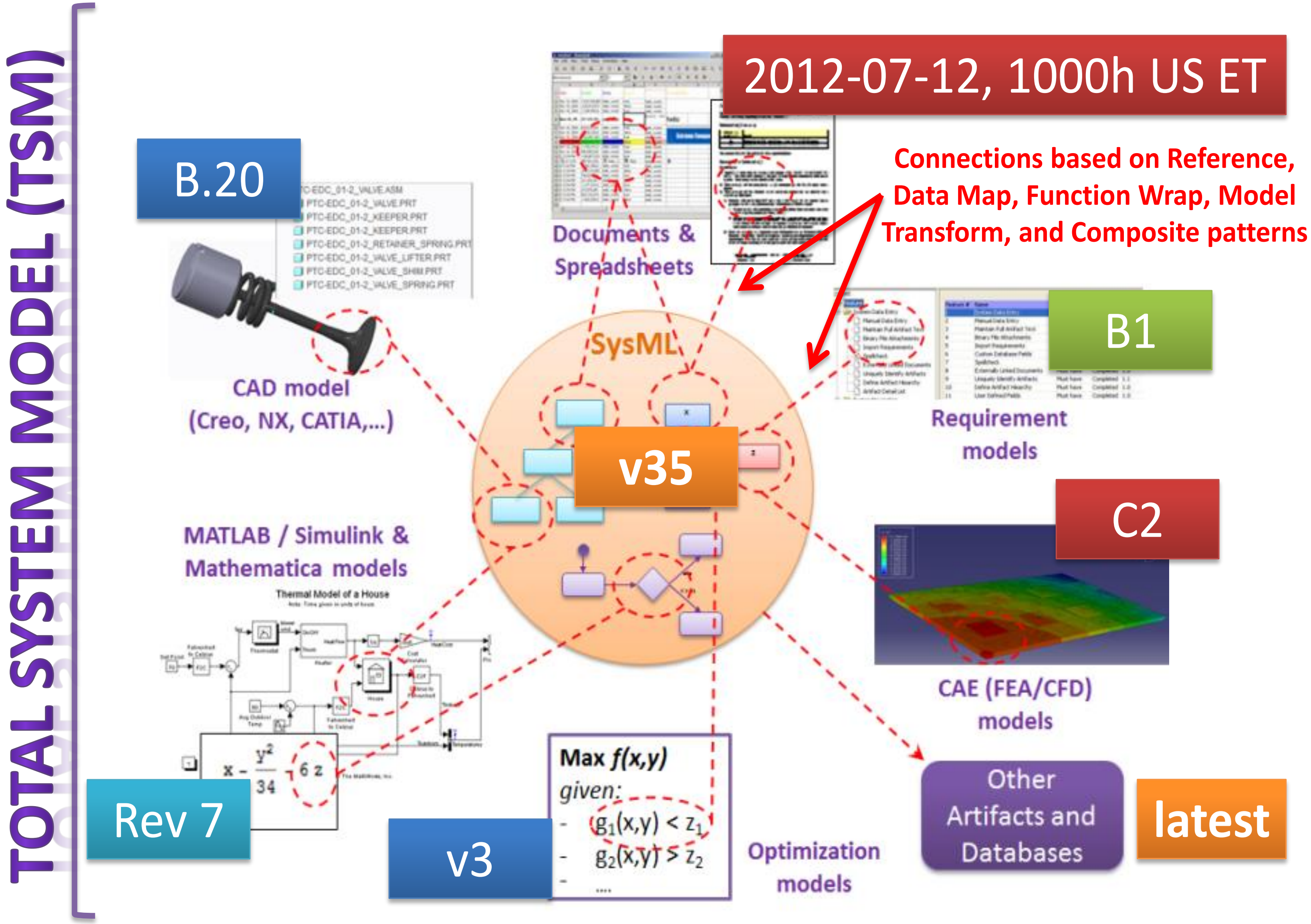
Tracking changes in connected models in the digital thread



Conn ID	Source	Target	Latest Target	Comment
e3f03…	Unmanned Aerial Vehicle	000464/C;2-Unmanned Aerial Vehicle	000464/C;2-Unmanned Aerial Vehicle	The block Unmanned Aerial Vehic…
	wimax module : WiMax Module	wimax module : 000472/A;1-WiMa…	wimax module : 000472/A;1-WiMax M…	Part property wimax module and…
	visual camera : Visual Camera	visual camera : 000475/A;1-Visual …	visual camera : 000475/A;1-Visual Ca…	Part property visual camera and …
	trackers : Sensor			Part property trackers has no co…
	thermal camera : Thermal Camera	thermal camera : 000476/A;1-Ther…	thermal camera : 000476/A;1-Therma…	Part property thermal camera an…
	software : Software System	software : 000487/B;1-Software S…	software : 000487/B;1-Software Syst…	Part property software and part…
	payload controller : Payload Controller	payload controller : 000470/A;1-P…	payload controller : 000470/A;1-Payl…	Part property payload controller …
	modem : Spread Spectrum Radio M…	modem : 000474/A;1-Spread Spec…	modem : 000474/A;1-Spread Spectru…	Part property modem and part o…
	ir detector : Wide Angle IR Detector	ir detector : 000477/A;1-Wide Ang…	ir detector : 000477/A;1-Wide Angle I…	Part property ir detector and pa…
	gps : GPS	gps : 000466/A;1-GPS	gps : 000466/A;1-GPS	Part property gps and part occu…
	gprs module : GPRS UMTS Module	gprs module : 000473/A;1-GPRS U…	gprs module : 000473/A;1-GPRS UMT…	Part property gprs module and p…
	flight controller : Flight Controller	flight controller : 000469/A;1-Fligh…	flight controller : 000469/A;1-Flight C…	Part property flight controller an…

```
[11:56:22] INFO Comparing SysML part property and Teamcenter part occurrence (BOM line with ref des) thermal camera.
```

Total System Model – A snapshot of the digital thread



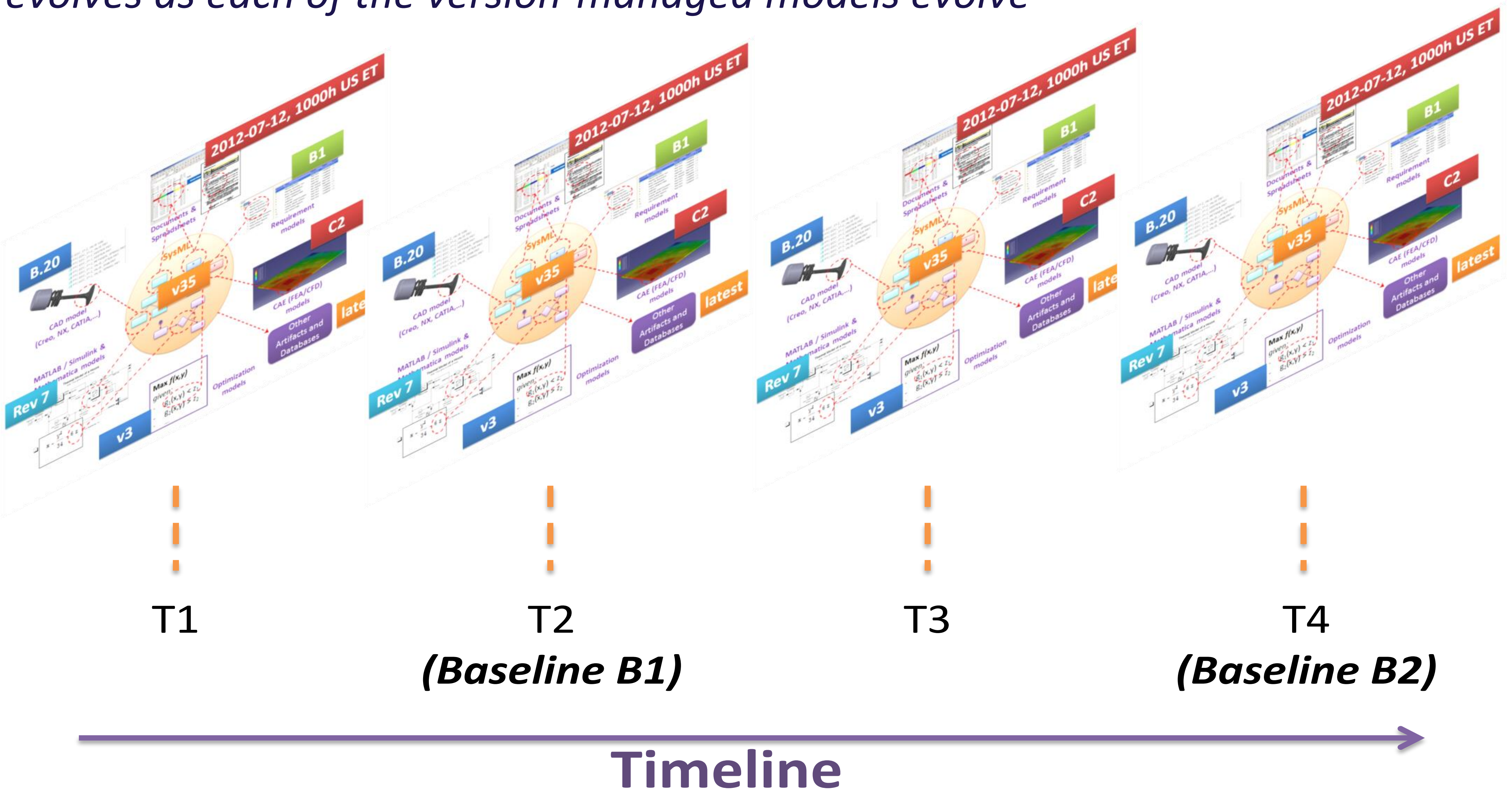
- Connect architecture model (SysML) with domain-specific models

- Total System Model (TSM) as a digital blueprint of the system connecting models across disciplines, tools, and version-management systems

- Goal: Seamless traceability between disciplines across the system lifecycle

Total System Model maturing though the lifecycle

TSM evolves as each of the version-managed models evolve



T1

T2
(Baseline B1)

T3

T4
(Baseline B2)

Timeline

Handler System – A registry for all participants in the digital thread

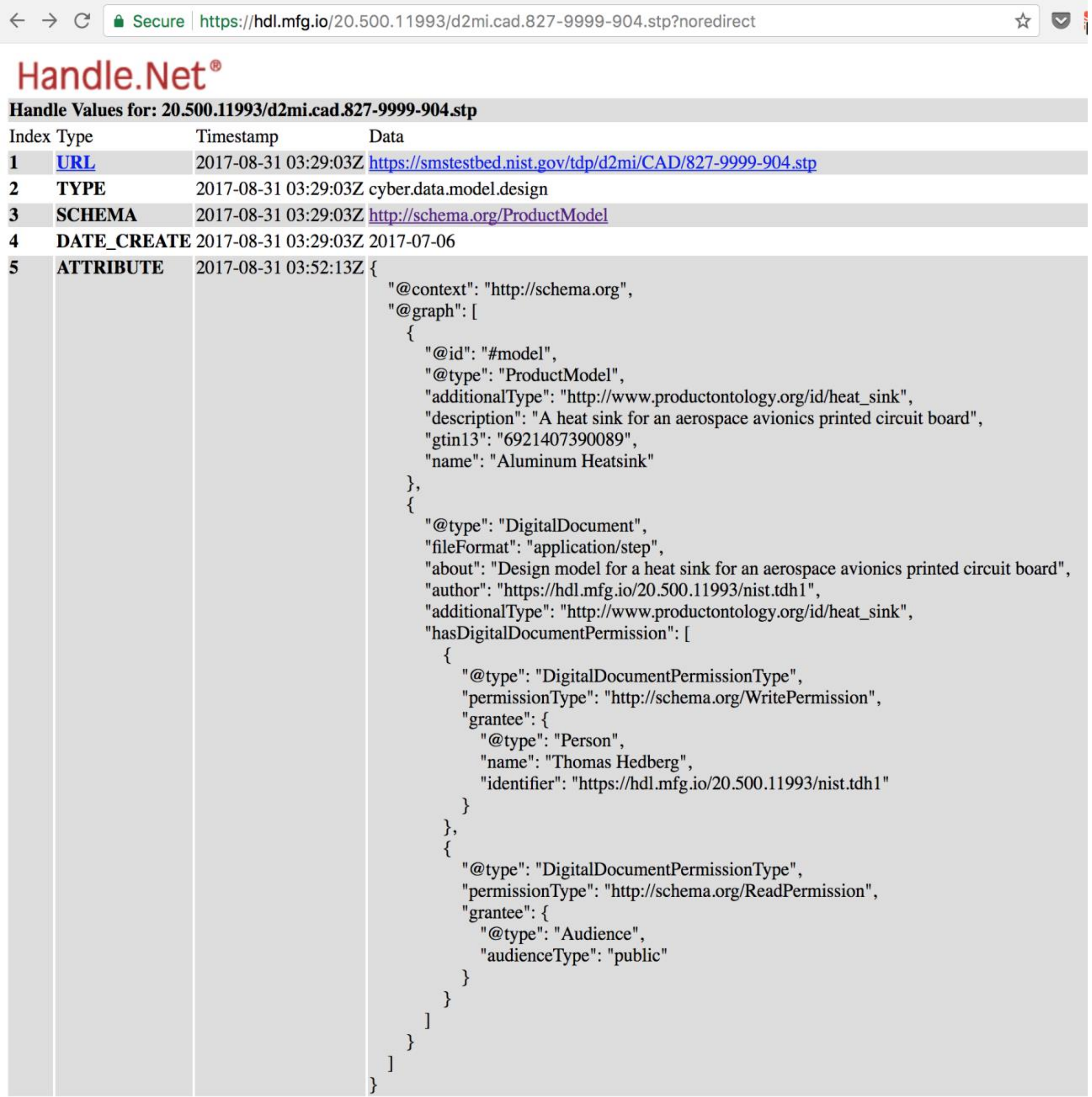
- Provides a unique registry for all artifacts in the digital thread (similar to DOI)
- Global ID system to identify artifacts (part, sensor, data, machines, …)
  - chain of addesses {A1, A2, A3, ...} similar to postal address. Addesses may be URIs.
- Basic meta-data for each artifact (full artifact data in native repository – PLM, ALM,...)
- Uses Handle.Net system

MFG.IO
PEOPLE
MACHINES
FEDERATED DATA
THINGS
SYSTEM INTEGRATION

Example entry in the Handle System for a Heat Sink part

Handler system
https://hdl.mfg.io

Entry for he Heat Sink part
https://hdl.mfg.io/20.500.11993/d2mi.cad.827-9999-904.stp?noredirect



Secure | https://hdl.mfg.io/20.500.11993/d2mi.cad.827-9999-904.stp?noredirect

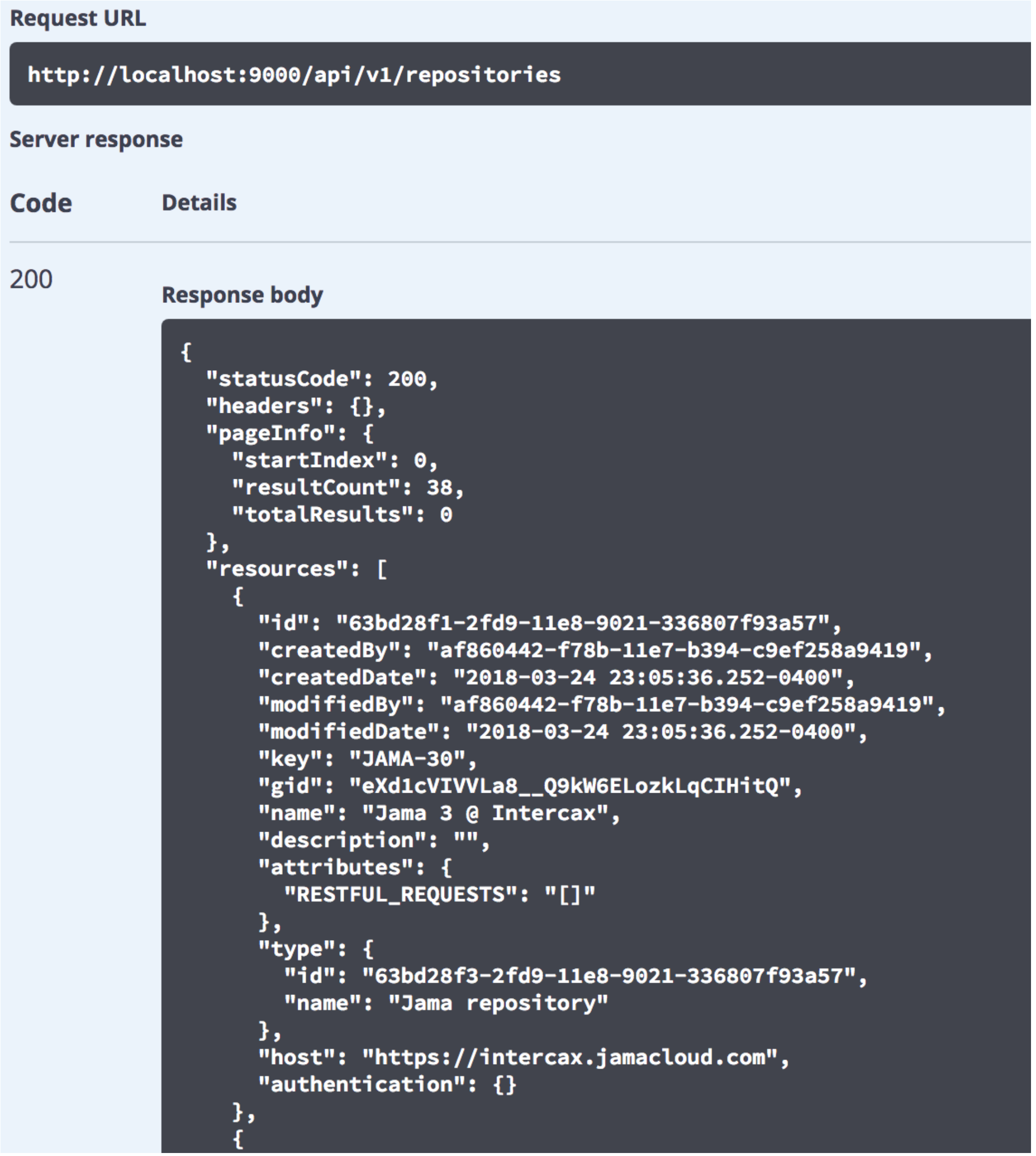
Handle.Net®

Handle Values for: 20.500.11993/d2mi.cad.827-9999-904.stp

Index	Type	Timestamp	Data
1	**URL**	2017-08-31 03:29:03Z	https://smstestbed.nist.gov/tdp/d2mi/CAD/827-9999-904.stp
2	**TYPE**	2017-08-31 03:29:03Z	cyber.data.model.design
3	**SCHEMA**	2017-08-31 03:29:03Z	http://schema.org/ProductModel
4	**DATE_CREATE**	2017-08-31 03:29:03Z	2017-07-06
5	**ATTRIBUTE**	2017-08-31 03:52:13Z	{

```
  "@context": "http://schema.org",
  "@graph": [
    {
      "@id": "#model",
      "@type": "ProductModel",
      "additionalType": "http://www.productontology.org/id/heat_sink",
      "description": "A heat sink for an aerospace avionics printed circuit board",
      "gtin13": "6921407390089",
      "name": "Aluminum Heatsink"
    },
    {
      "@type": "DigitalDocument",
      "fileFormat": "application/step",
      "about": "Design model for a heat sink for an aerospace avionics printed circuit board",
      "author": "https://hdl.mfg.io/20.500.11993/nist.tdh1",
      "additionalType": "http://www.productontology.org/id/heat_sink",
      "hasDigitalDocumentPermission": [
        {
          "@type": "DigitalDocumentPermissionType",
          "permissionType": "http://schema.org/WritePermission",
          "grantee": {
            "@type": "Person",
            "name": "Thomas Hedberg",
            "identifier": "https://hdl.mfg.io/20.500.11993/nist.tdh1"
          }
        },
        {
          "@type": "DigitalDocumentPermissionType",
          "permissionType": "http://schema.org/ReadPermission",
          "grantee": {
            "@type": "Audience",
            "audienceType": "public"
          }
        }
      ]
    }
  ]
}
```

API for the digital thread



Request URL

```
http://localhost:9000/api/v1/repositories
```

Server response

Code **Details**

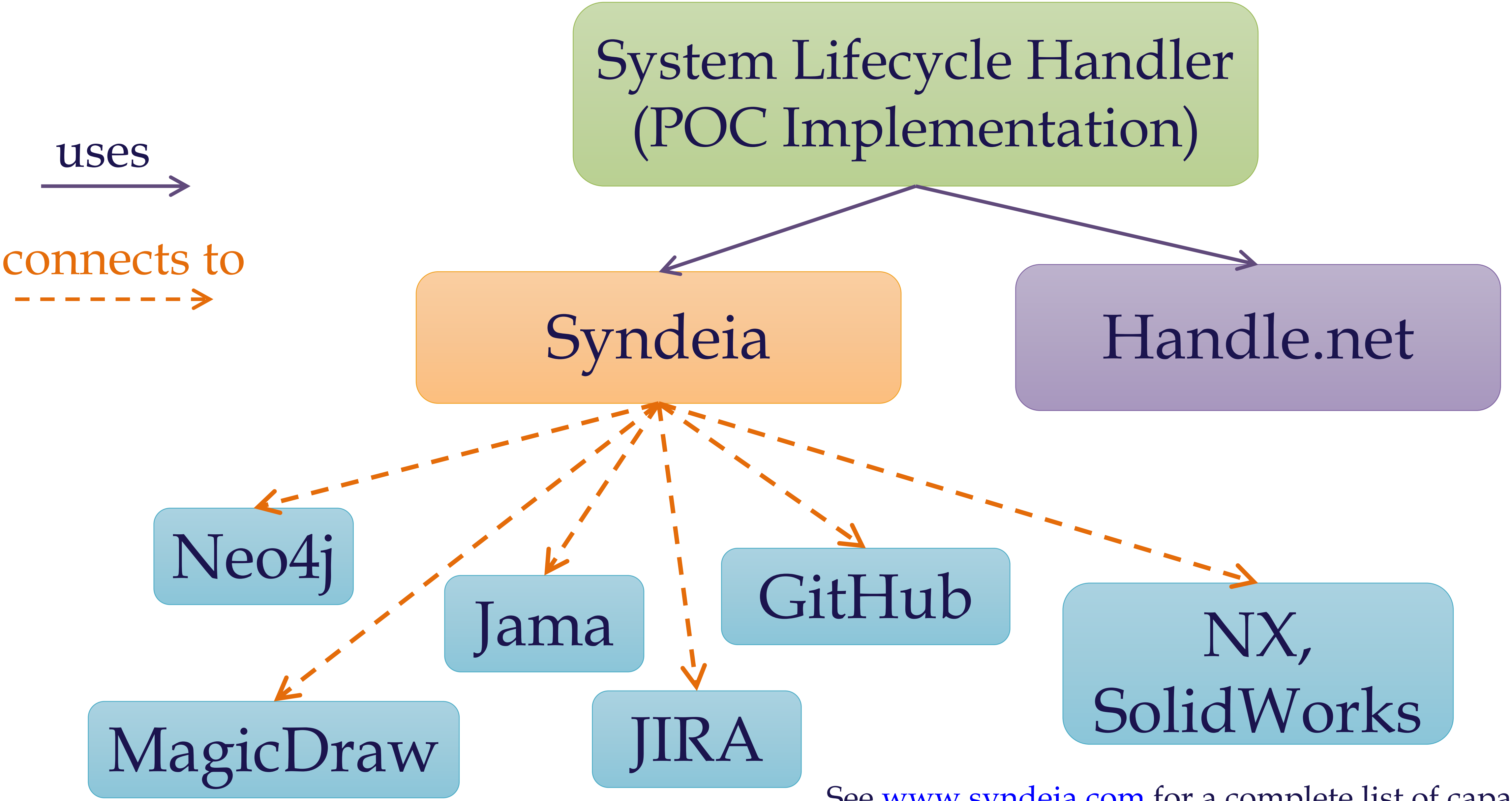
200

Response body

```
{
  "statusCode": 200,
  "headers": {},
  "pageInfo": {
    "startIndex": 0,
    "resultCount": 38,
    "totalResults": 0
  },
  "resources": [
    {
      "id": "63bd28f1-2fd9-11e8-9021-336807f93a57",
      "createdBy": "af860442-f78b-11e7-b394-c9ef258a9419",
      "createdDate": "2018-03-24 23:05:36.252-0400",
      "modifiedBy": "af860442-f78b-11e7-b394-c9ef258a9419",
      "modifiedDate": "2018-03-24 23:05:36.252-0400",
      "key": "JAMA-30",
      "gid": "eXd1cVIVVLa8__Q9kW6ELozkLqCIHitQ",
      "name": "Jama 3 @ Intercax",
      "description": "",
      "attributes": {
        "RESTFUL_REQUESTS": "[]"
      },
      "type": {
        "id": "63bd28f3-2fd9-11e8-9021-336807f93a57",
        "name": "Jama repository"
      },
      "host": "https://intercax.jamacloud.com",
      "authentication": {}
    },
    {
```

- REST/HTTP API to access data in the digital thread, such as:
  - Repositories, projects, model elements, and connections
  - Query capabilities to search for connections given type, source, target, etc. (basic graph navigation)
- Foundation for new apps that can be built to access, analyze, enhance the digital thread

Tools used in this proof-of-concept (POC)



See www.syndeia.com for a complete list of capabilities

Open Standards and APIs used in this POC

- SysML
- MTConnect
- QIF
- JSON
- REST/HTTP
- OSLC
- OAuth
- JDBC
- Other relevant open standards – STEP, FMI
- Native APIs, and Multiple open source Apache and Google libraries

Next Steps

- Common schema(s) for the artifacts and relationships in the digital thread
- Library of queries (FAQs) for the digital thread
- Tracking active lifecycle states – design, make, ops, service
- Explore multi-level change management scenarios, e.g. replay cascading changes that may happen if one artifact changes
- Test suites for V&V of the digital thread, automated testing and release builds (Technical Data Packages) of the digital thread

Questions and Comments

Manas Bajaj, PhD -- manas.bajaj@intercax.com

Dirk Zwemer, PhD -- dirk.zwemer@intercax.com

Intercax

Thomas Hedberg -- thomas.hedberg@nist.gov

NIST