34th Annual INCOSE international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024

Exploration of SysML V2 Capabilities to bridge the gap from System Modeling to Network Design

Tony Komar, Siemens

Objective

- Describe Operational Capabilities for Network based Product Feature for a System using SysML V2 (Use Cases)
  - System Modeling to Network Design
  - Network Design to System Model
- Highlight how SysML V2 could be used to improve capabilities highlighting
  - Domain Libraries
  - SysML V2 Server API
- Discuss elements of a high-level physical architecture to deliver these capabilities

Goal : Reduce the “Unhealthy Tension” between bus designer and system designer

Disclaimer: The content presented here is not commitment on future Siemens product.

Presentation Topics

- Understanding the Problem – Developing a new product feature
  - Operational Analysis Model of the Problem
    - Actors and Entities
    - Operational Capabilities
    - Actors and their Activities to Deliver Capabilities with SysML V2 related technology
      - Introduce an Example to Enhance understanding
- Describe an overview of Physical Architecture of solution
- Conclusions on impact of SysML V2 to System Engineering and Domain Engineering

Actors and Entities that we will be discussing in Operational Analysis

- Actors
  - Design Engineering Related Domain Roles
    - System Engineer – Oversee Functional Allocation to Hardware and Software
    - System Modeler – Develop a System Model reflecting the allocation of new feature to hardware and software components
    - Network Designer – Implements the network design for the system across various communication buses
  - Project Management
    - Product Owner - Collaborates with stakeholders and actors to implement the systems new product feature
  - Information Technology
    - Domain Application Architects
- Entities
  - Domain Tools (System Modeling and Network Archtiecture)
  - SysML V2 API Server
    - Delivers the V2 API's to be used by Actors
  - SysML V2 Domain Libraries

System Engineer/System Modeler Responsibilities

Evolve the System to Deliver the Feature

Understand Existing System and Desired Signals

Comprehend the Impact of new signals and changes

Understand New Functions and Signals

Perform Analysis on the New Functions and Signals

Define/Access the Allocation new Functions to Hardware vs Software

Access the impact of hardware and software components required to deliver the capability or feature.

Functional Analysis

Perform simulation and analysis of new functions.

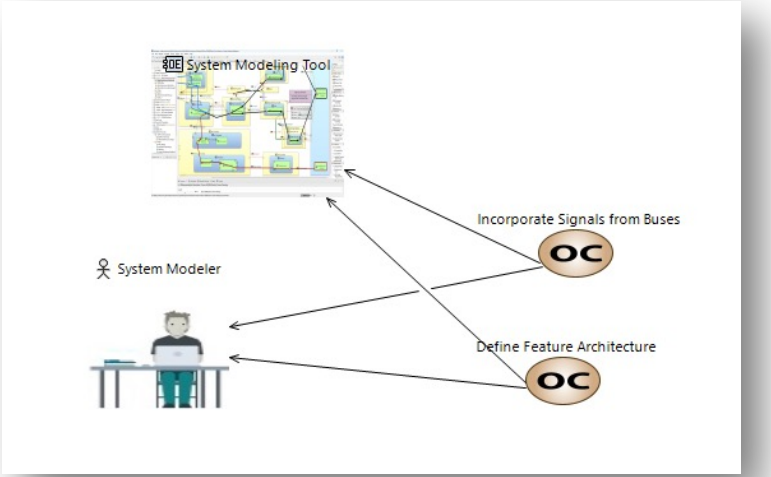
Trade Studies & Impact Analysis

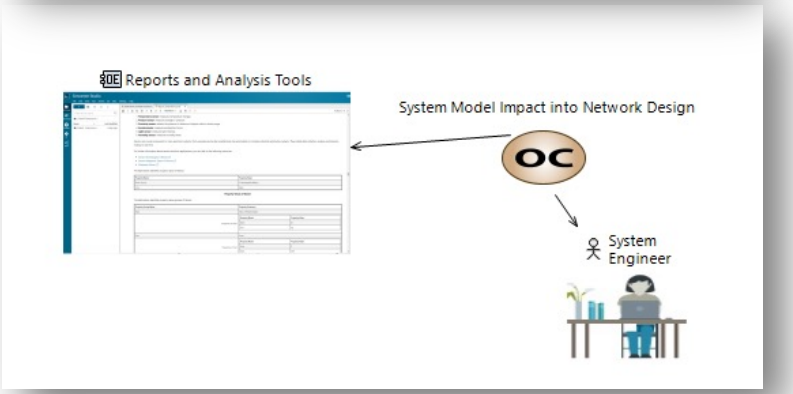
Deliver trade study based on the modeled content

Communicate Architecture to Downstream Implementation

Deliver the architectural decisions for design implementation

Focus for this analysis





Network Designer and Tool Responsibilities

Manage Architecture Platform

System Integration
Integrate multiple systems and components into a single platform to validate the connectivity

Signal to Carrier Assignment
Assignment of signals to network carriers and assisted interface and gateway management

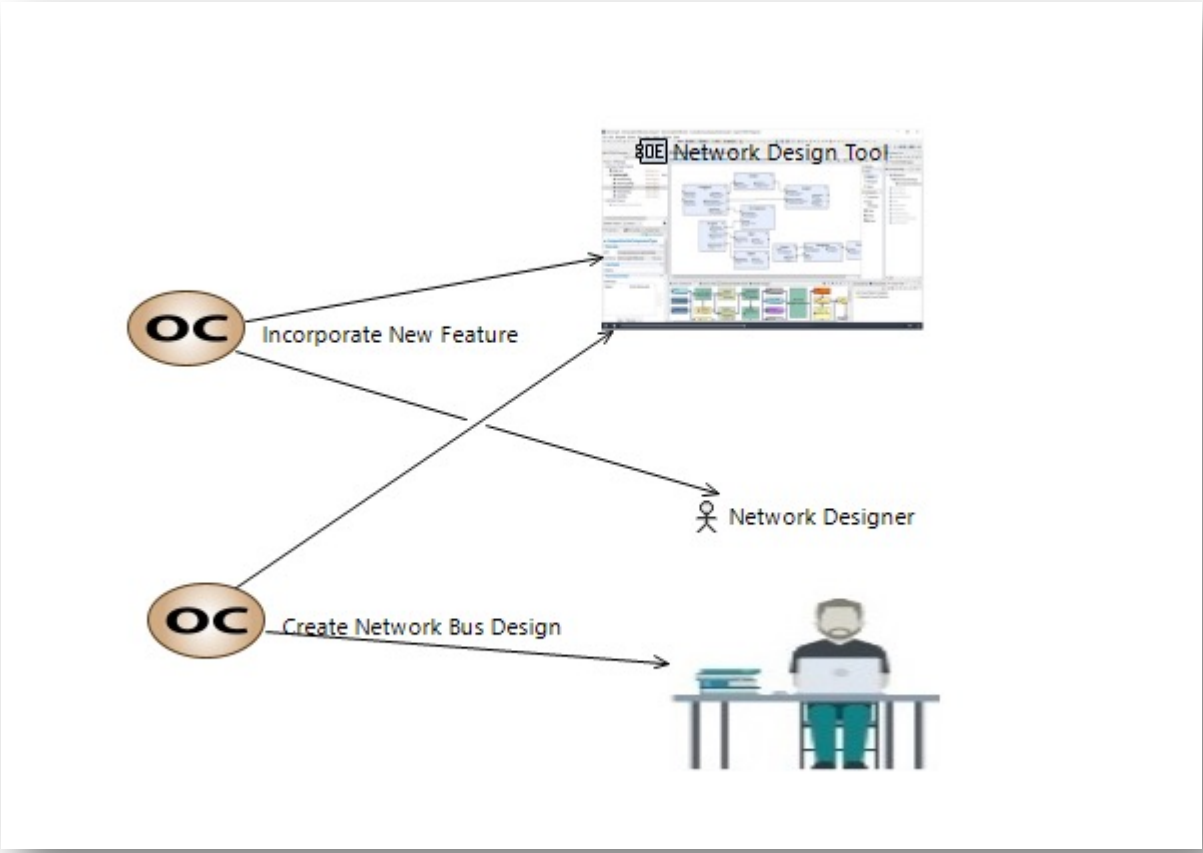
Function/Behavior Allocation
Allocation and re-deployment of behavior to physical components, including configuration and platform variants

Trade Studies & Impact Analysis
Real-time metrics reporting on efficacy of implementation proposals generated as functions and signals are allocated (e.g. data latency, bus utilization)

Device Management
Assign components to existing ICDs, create/update ICDs based on new/modified components internal/external connectivity

Generation of Downstream Implementation
Synthesis of models for NW/SW/EDS/HW design and implementation

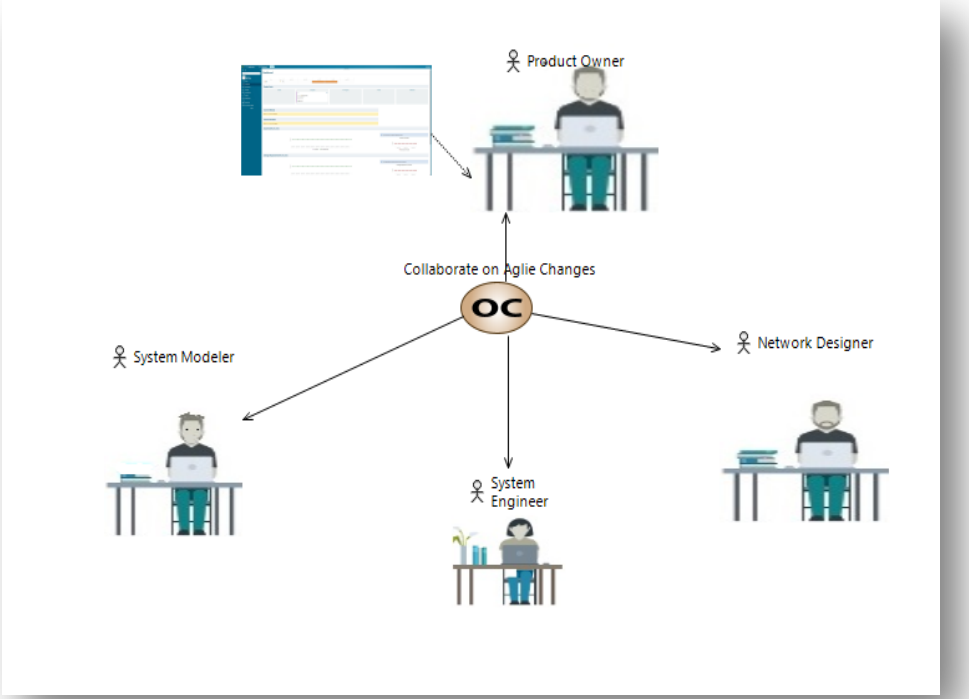
Focus for this analysis



Product Owner

Coordinate the development of features into the platform

- Developing and communicating the Product Goal
- Define the feature to be developed
- Developing and communicating product backlog items.
- Continuously refine and prioritizing product backlog items depending on any changes from ongoing feedback..
- Collaborate closely with team and stakeholders to ensure the Product Backlog is transparent, correctly prioritized and understood

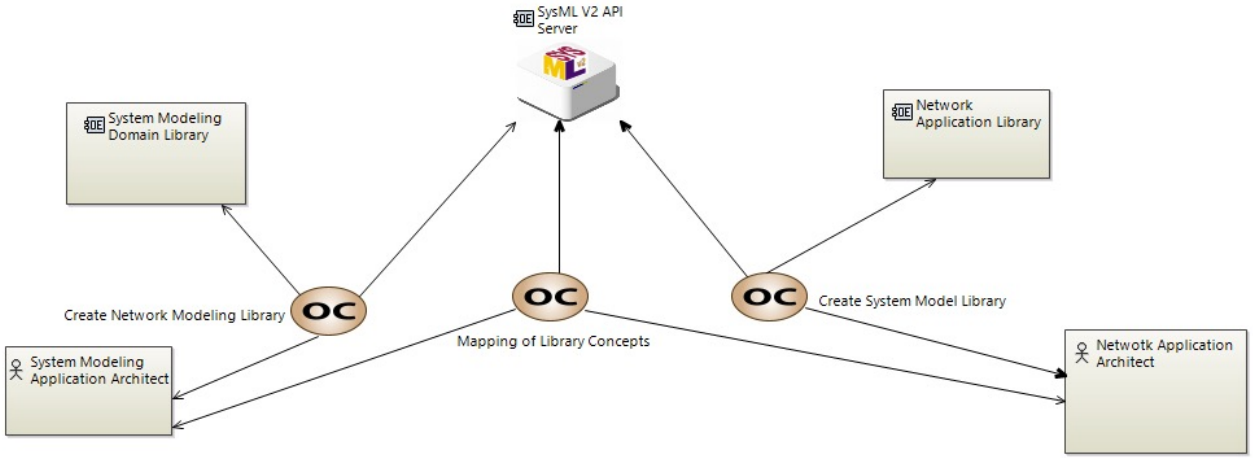


Application Architect Responsibilities

Support interoperability among application domains

- Create domain specific libraries
- Configure domain libraries for use in translation
- Support development of translator by creating a mapping on domain concepts

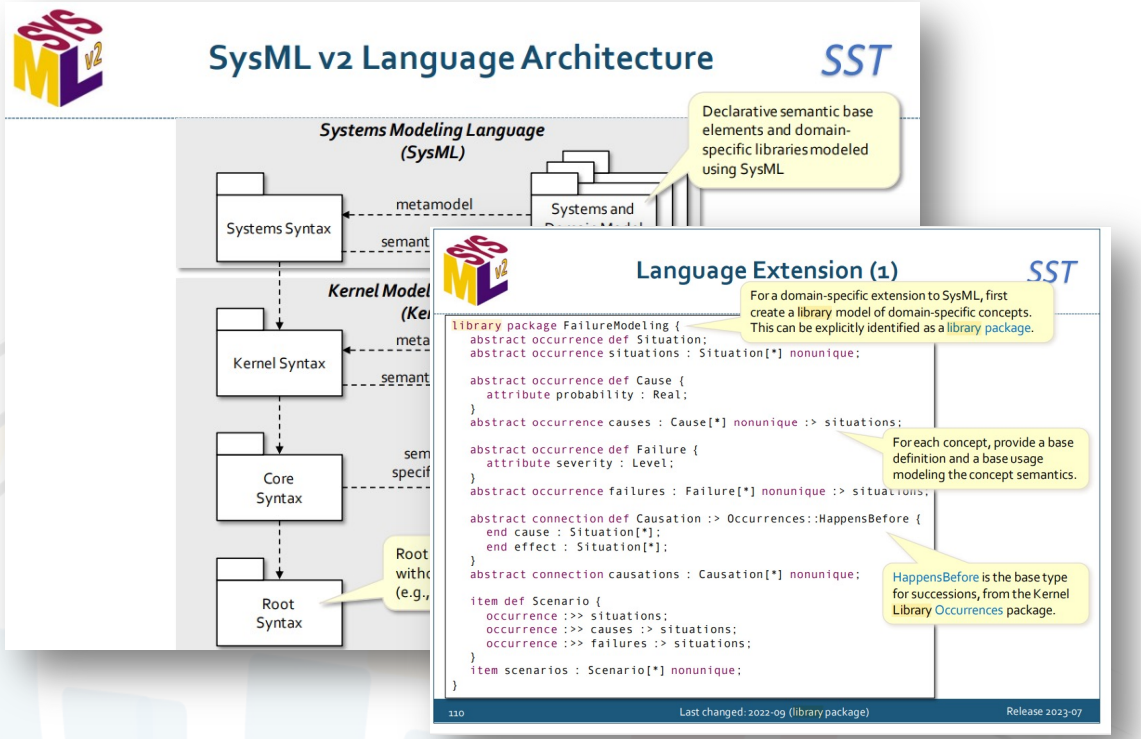
Subset for this Analyses



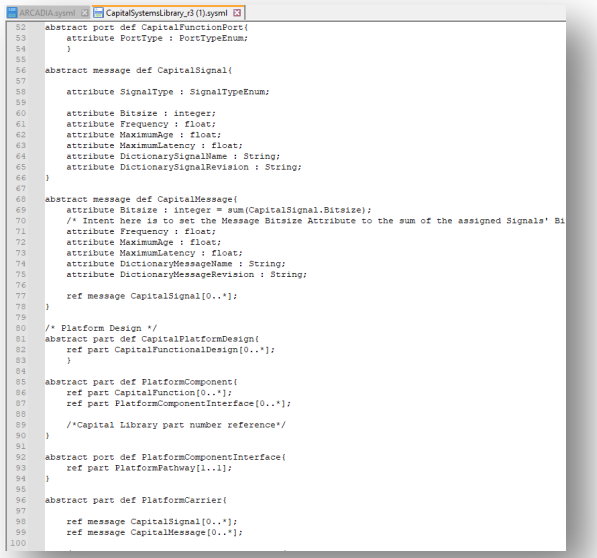
SysML V2 Libraries

Extends SysML V2 for each Domain

- Libraries provide a view of Domain Specific Topics (System Modeling, Network Design) presented in SysML V2 constructs
- They extend SysML V2 to the domain of the application

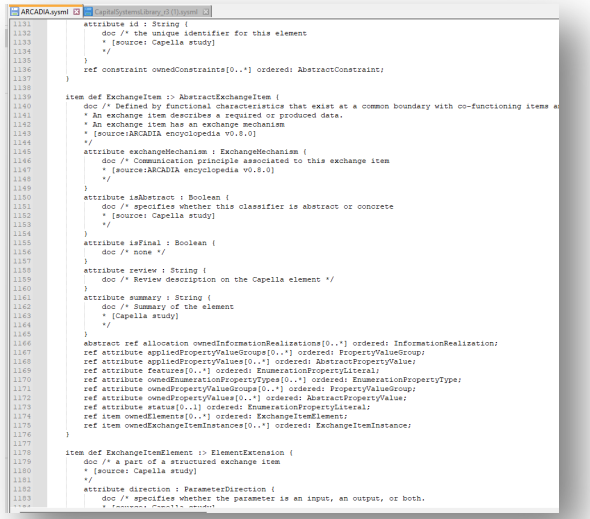


Network Application Domain Library



```
abstract port def CapitalFunctionPort{
    attribute PortType : PortTypeEnum;
    }

abstract message def CapitalSignal{

    attribute SignalType : SignalTypeEnum;

    attribute Bitsize : integer;
    attribute Frequency : float;
    attribute MaximumAge : float;
    attribute MaximumLatency : float;
    attribute DictionarySignalName : String;
    attribute DictionarySignalRevision : String;
}

abstract message def CapitalMessage{
    attribute Bitsize : integer = sum(CapitalSignal.Bitsize);
    /* Intent here is to set the Message Bitsize Attribute to the sum of the assigned Signals' Bi
    attribute Frequency : float;
    attribute MaximumAge : float;
    attribute MaximumLatency : float;
    attribute DictionaryMessageName : String;
    attribute DictionaryMessageRevision : String;

    ref message CapitalSignal[0..*];
}

/* Platform Design */
abstract part def CapitalPlatformDesign{
    ref part CapitalFunctionalDesign[0..*];
    }

abstract part def PlatformComponent{
    ref part CapitalFunction[0..*];
    ref part PlatformComponentInterface[0..*];

    /*Capital Library part number reference*/
}

abstract port def PlatformComponentInterface{
    ref part PlatformPathway[1..1];
}

abstract part def PlatformCarrier{

    ref message CapitalSignal[0..*];
    ref message CapitalMessage[0..*];
```

System Modeling Application Domain Library

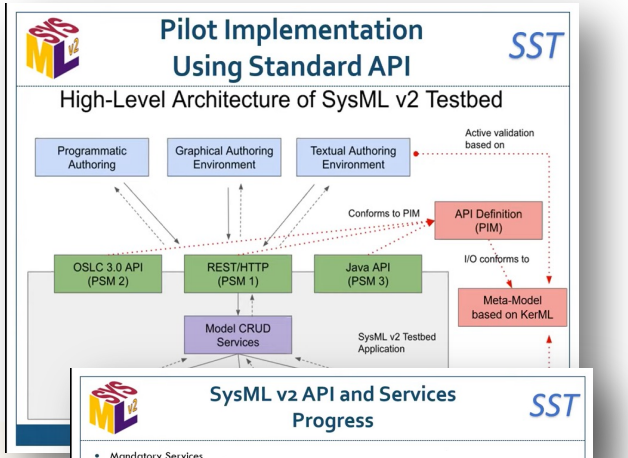


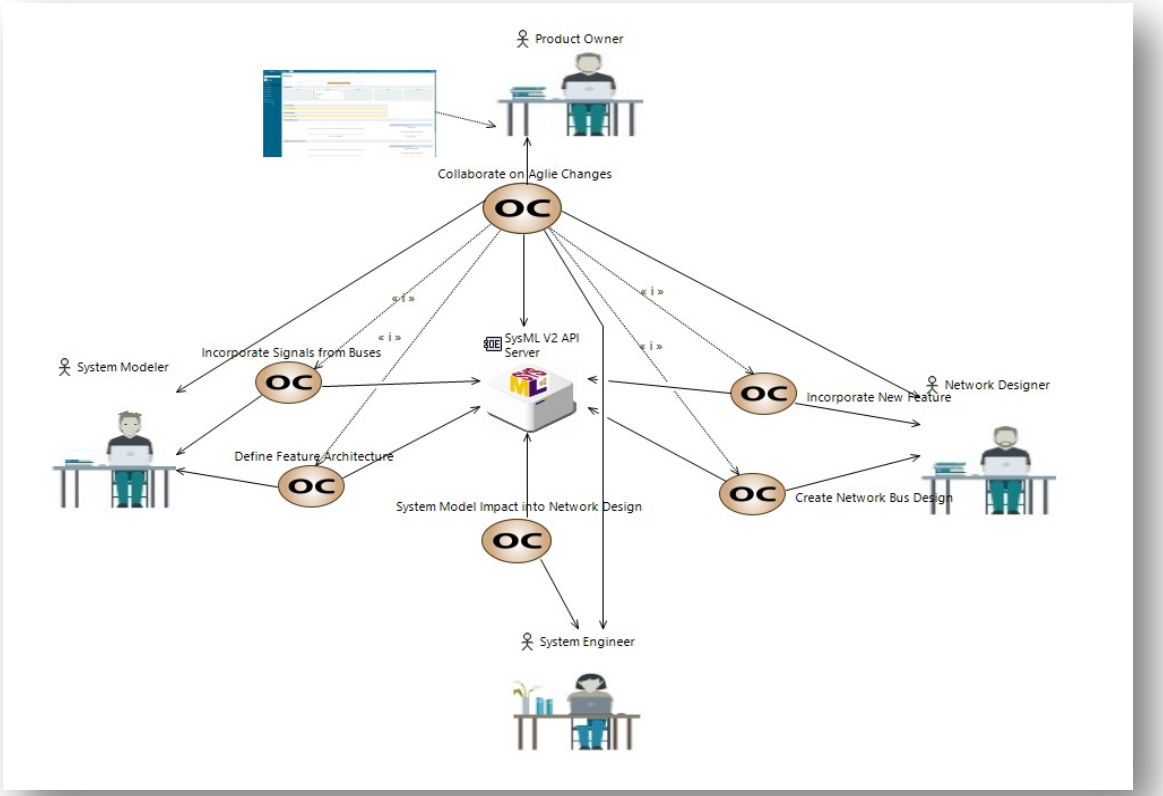
```
        attribute id : String {
            doc /* the unique identifier for this element
             * [source: Capella study]
             */
        }
        ref constraint ownedConstraints[0..*] ordered: AbstractConstraint;
    }

    item def ExchangeItem :> AbstractExchangeItem {
        doc /* Defined by functional characteristics that exist at a common boundary with co-functioning items a
         * An exchange item describes a required or produced data.
         * An exchange item has an exchange mechanism
         * [source:ARCADIA encyclopedia v0.8.0]
         */
        attribute exchangeMechanism : ExchangeMechanism {
            doc /* Communication principle associated to this exchange item
             * [source:ARCADIA encyclopedia v0.8.0]
             */
        }
        attribute isAbstract : Boolean {
            doc /* specifies whether this classifier is abstract or concrete
             * [source: Capella study]
             */
        }
        attribute isFinal : Boolean {
            doc /* none */
        }
        attribute review : String {
            doc /* Review description on the Capella element */
        }
        attribute summary : String {
            doc /* Summary of the element
             * [Capella study]
             */
        }
        abstract ref allocation ownedInformationRealizations[0..*] ordered: InformationRealization;
        ref attribute appliedPropertyValueGroups[0..*] ordered: PropertyValueGroup;
        ref attribute appliedPropertyValues[0..*] ordered: AbstractPropertyValue;
        ref attribute features[0..*] ordered: EnumerationPropertyLiteral;
        ref attribute ownedEnumerationPropertyTypes[0..*] ordered: EnumerationPropertyType;
        ref attribute ownedPropertyValueGroups[0..*] ordered: PropertyValueGroup;
        ref attribute ownedPropertyValues[0..*] ordered: AbstractPropertyValue;
        ref attribute status[0..1] ordered: EnumerationPropertyLiteral;
        ref item ownedElements[0..*] ordered: ExchangeItemElement;
        ref item ownedExchangeItemInstances[0..*] ordered: ExchangeItemInstance;
    }

    item def ExchangeItemElement :> ElementExtension {
        doc /* A part of a structured exchange item
         * [source: Capella study]
         */
        attribute direction : ParameterDirection {
            doc /* specifies whether the parameter is an input, an output, or both.
```

SysML V2 API Server

Supports Multiple Domain Collaboration

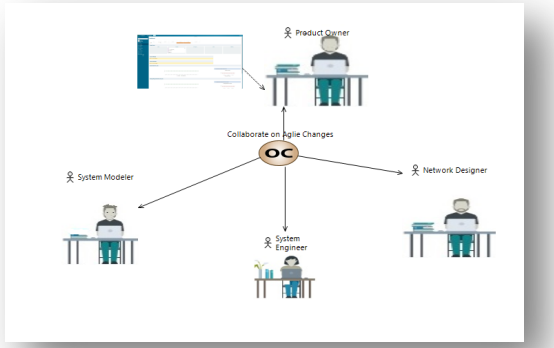
- Provide means to access the content of projects and models.
- Allows a means for Actors to save model in SysML V2
- Provides a means to query portions of a model based on SysML V2 constructs
- Saves libraries that are referenced by applications



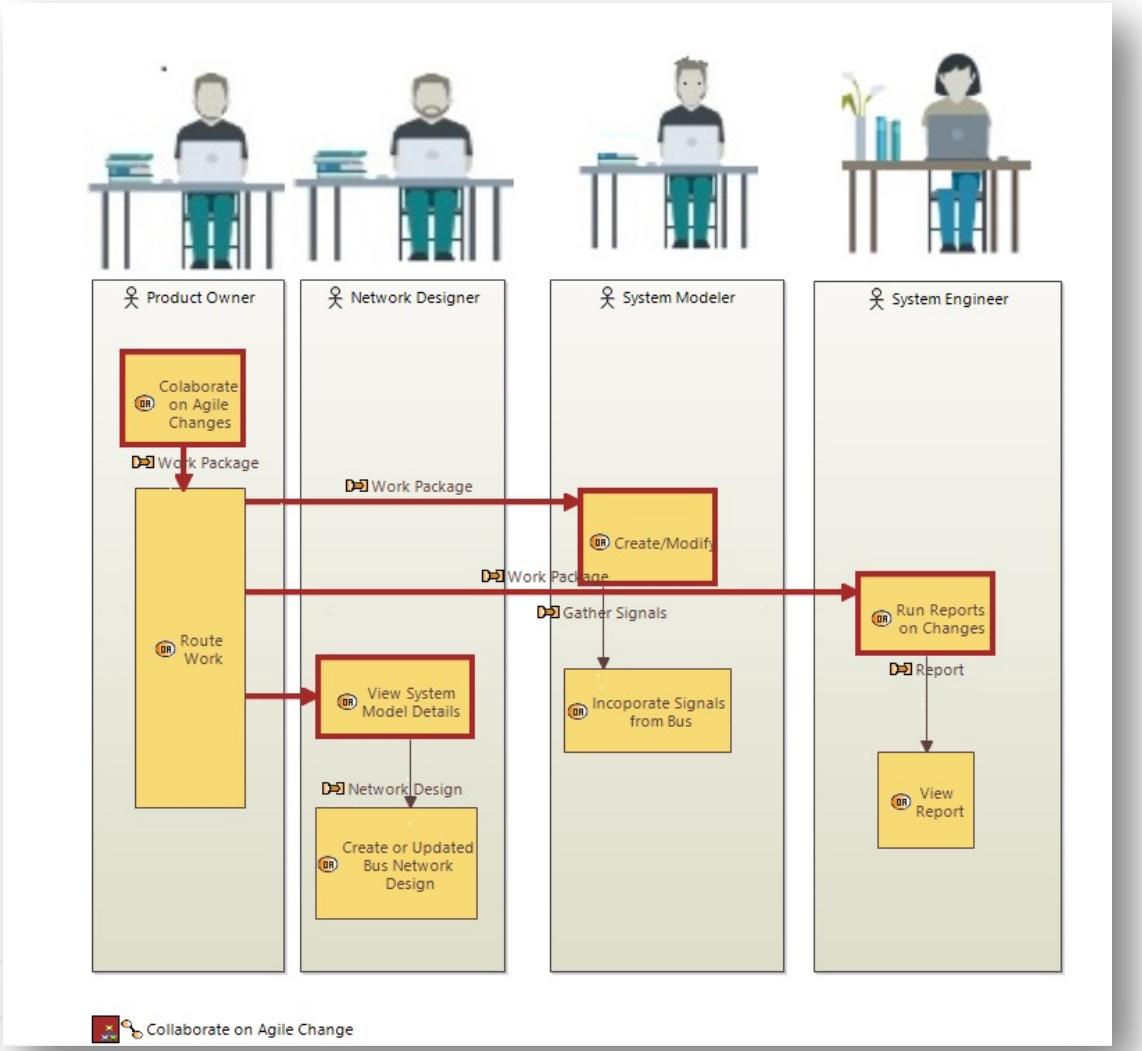


The V2 api services supported by a server/repository can provide means to retrieve commit related temporal model content

Product Owner Initiates Activity to Develop a new Feature for the System or Product



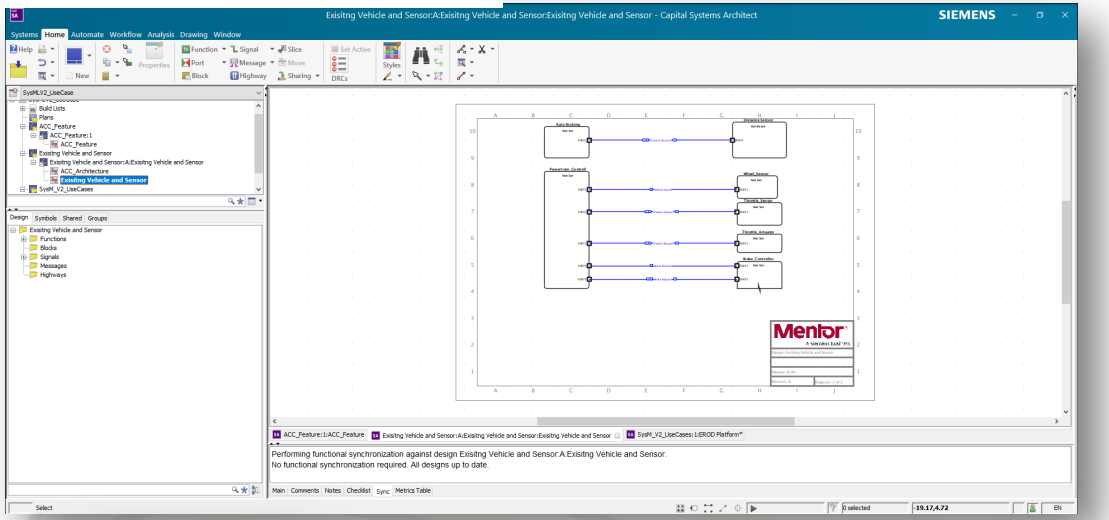
A simple automotive example will be used around an adaptative cruise control

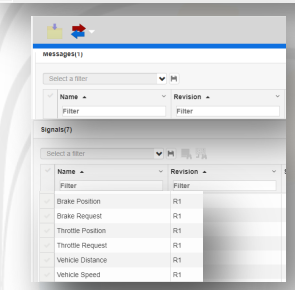


Network Architecture Platform Model Exists for the Vehicle

Capital Systems Modeler

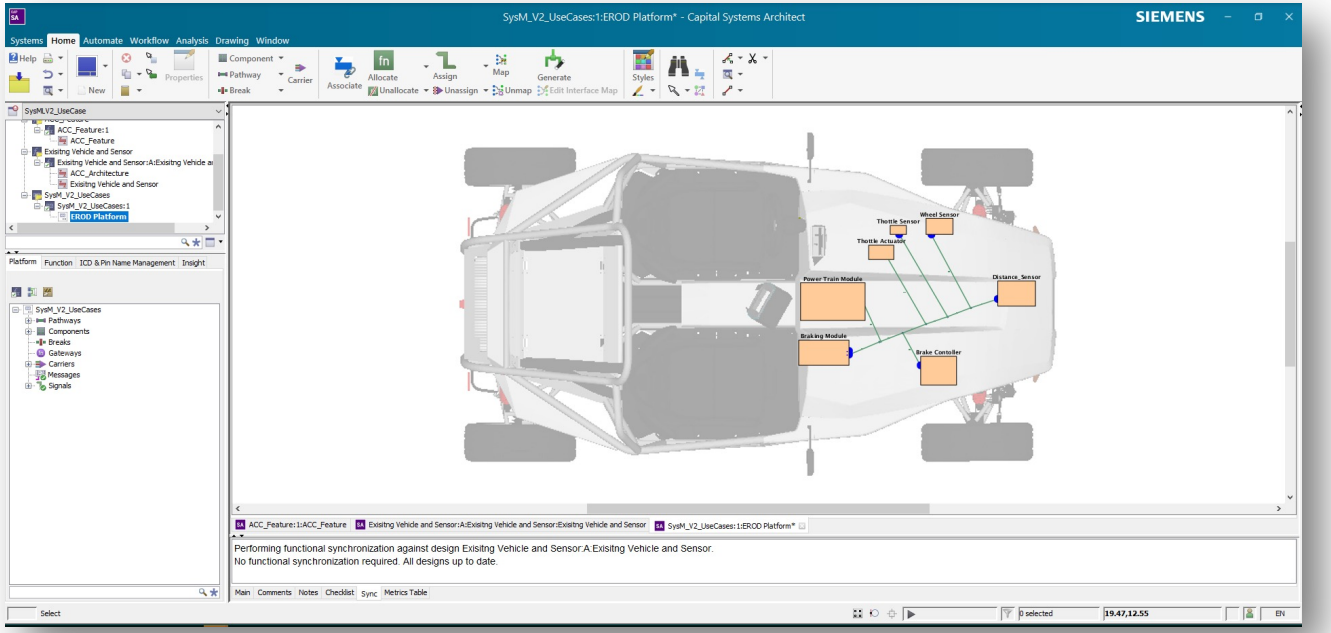
Functional design capture tool.



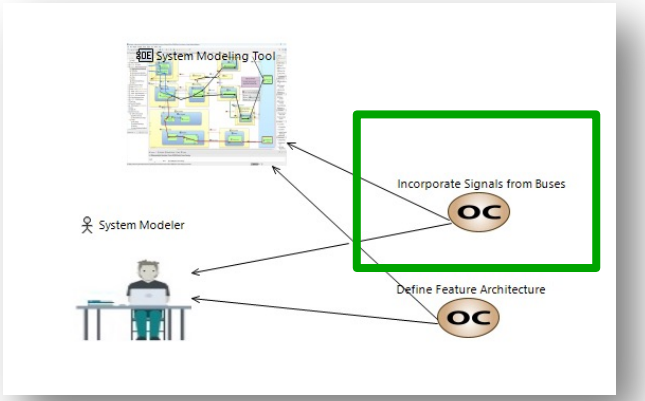


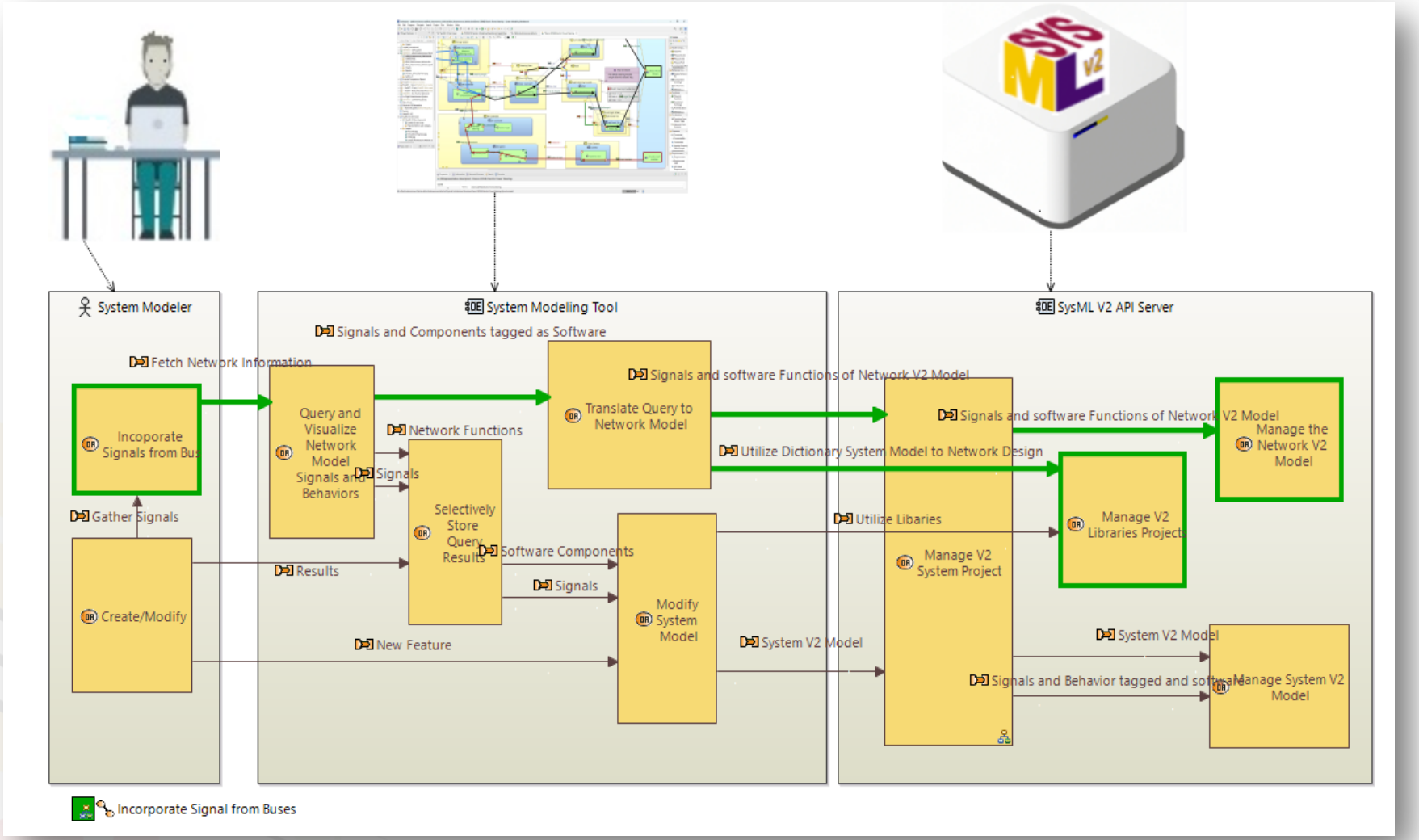
Capital Systems Architect

Functional implementation and planning.
Automated synthesis & metric feedback.



System Modeler – Incorporates Signals from Network Platform Architecture

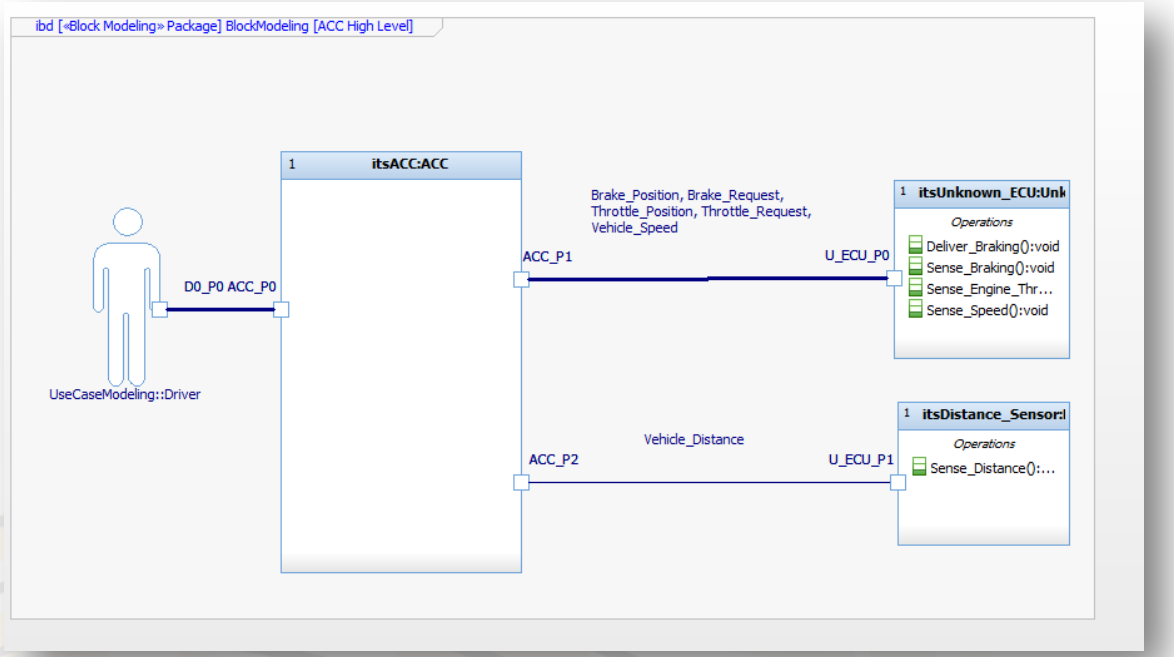


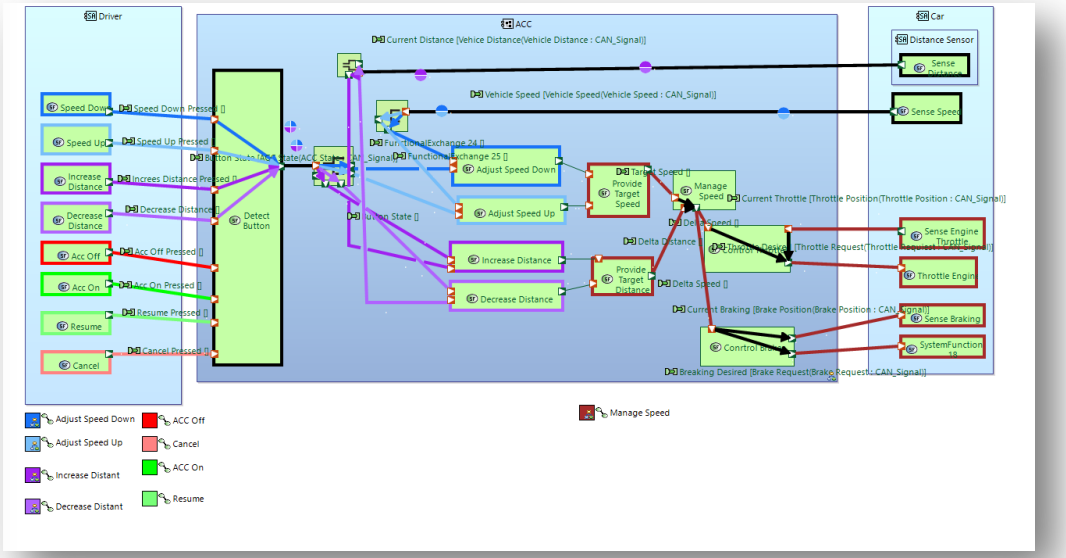


- Queries based on system modeling constructs provide the means to yield a targeted subset on a domain specific model
- Translations on specific model constructs defined in libraries yields interoperability between domains

System Analysis defines the context of the system feature to be developed

SysML v1

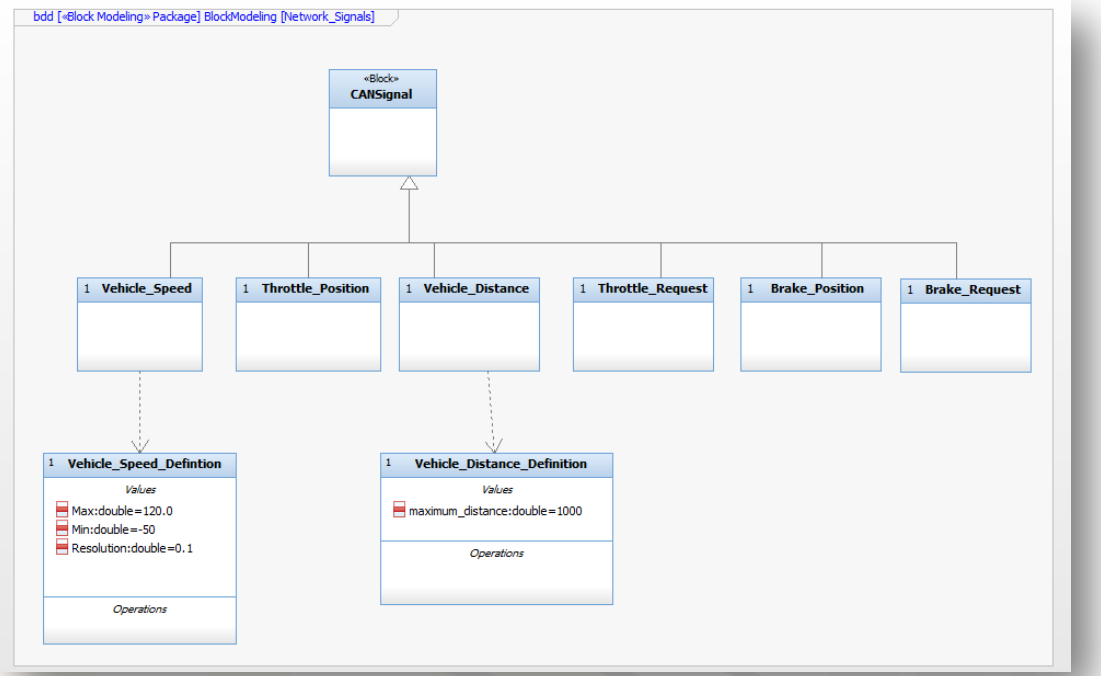




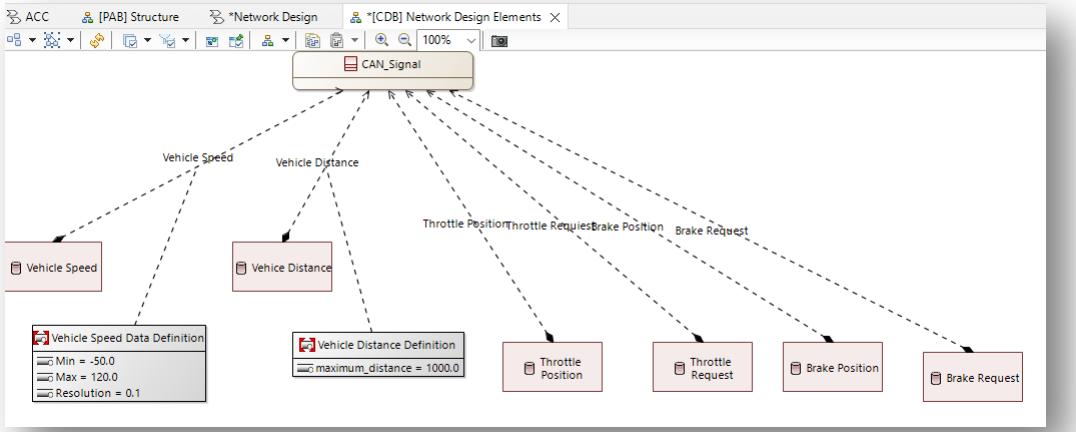
Capella

Results of a queries yield signals that can be selectively brought into the system modeling tool

SysML v1

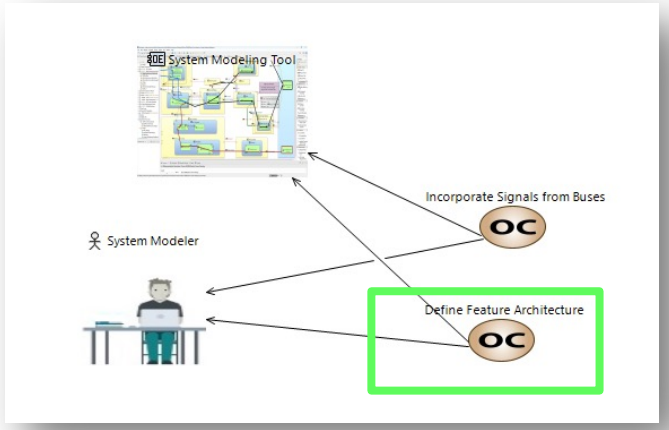


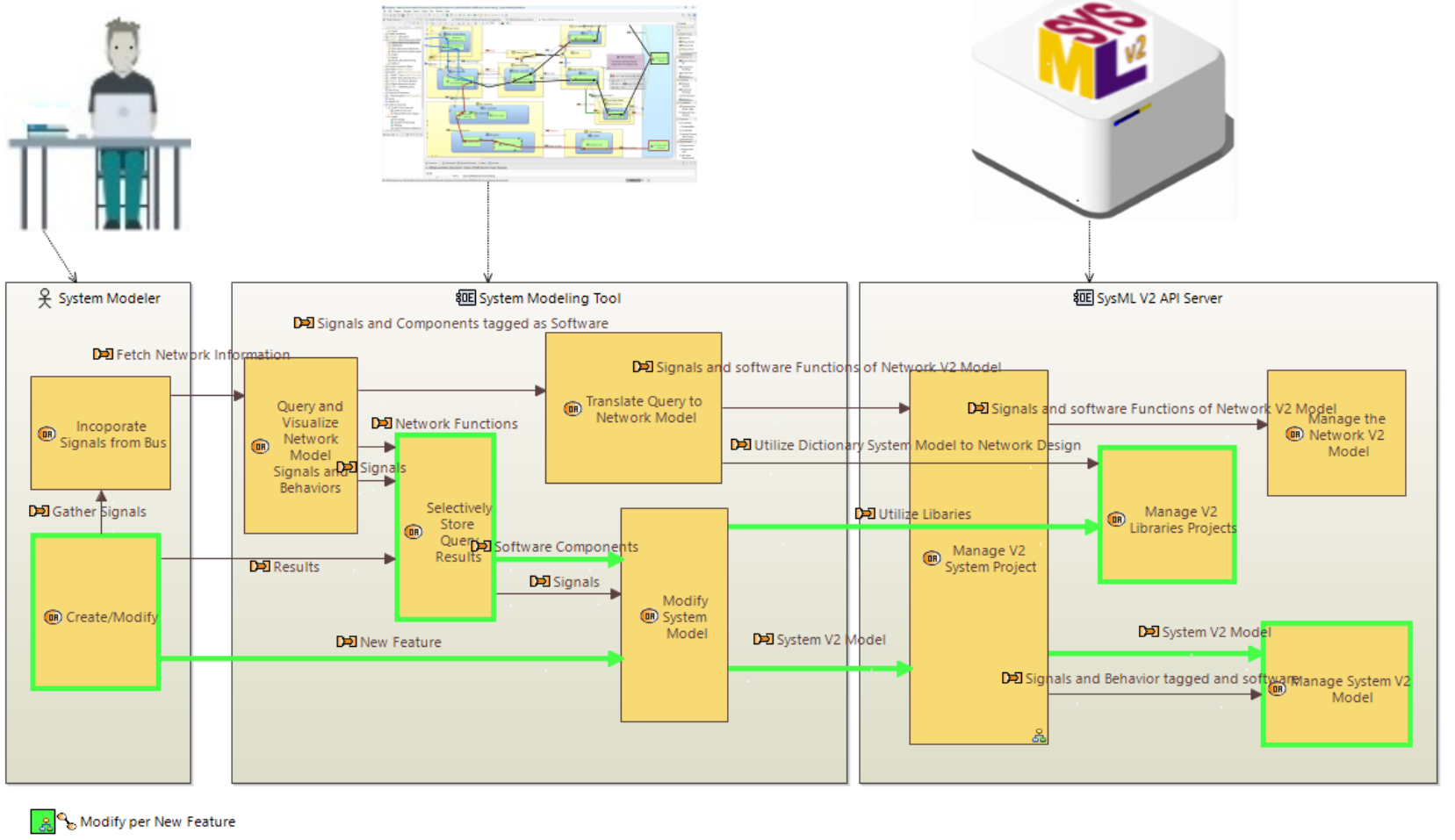
Result of Signal Query from Network Model



Capella

System Modeler Defines the Feature Architecture



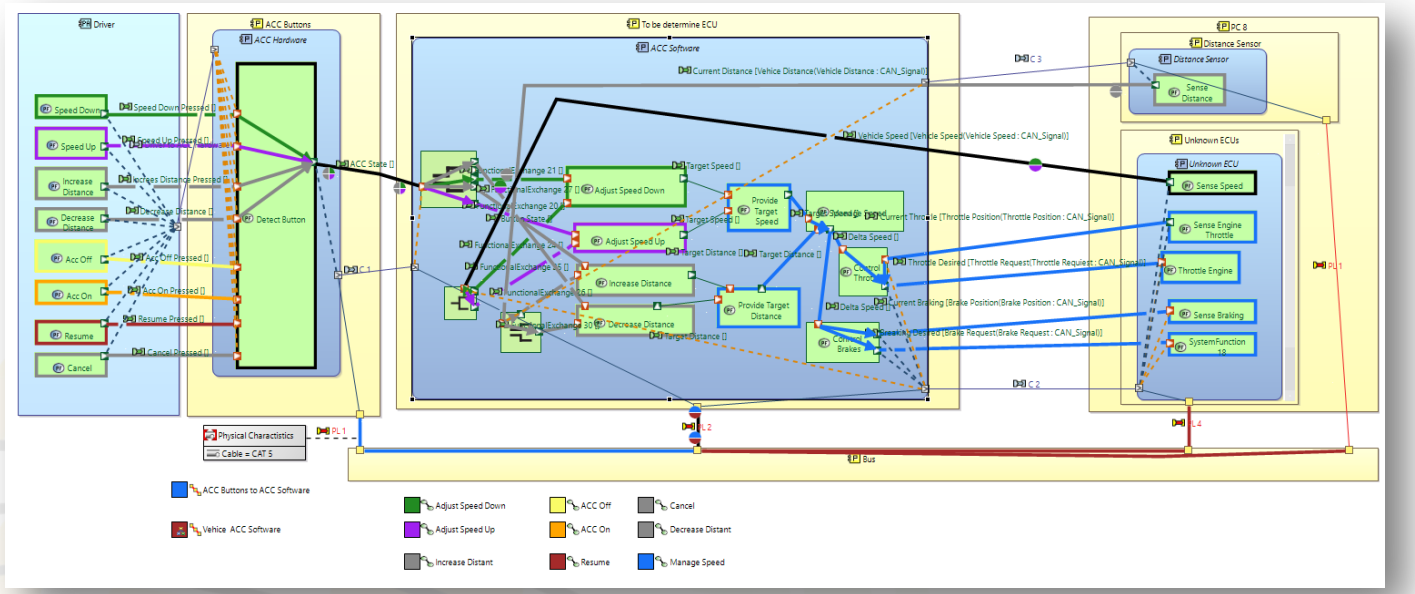


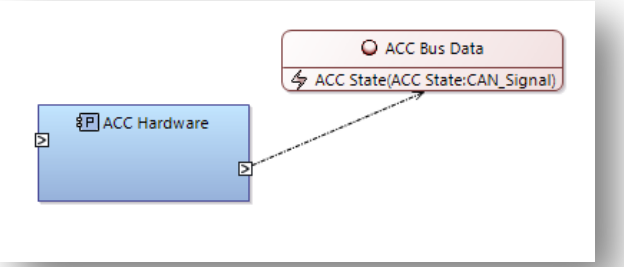
- Subsets of translated model constructs can be brought to Domain
- The model is then committed back to the repository

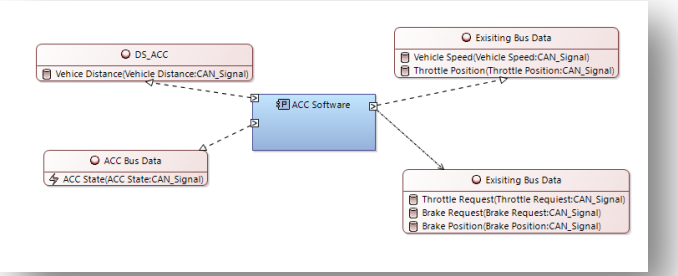
Architecture Development of system model results in functionality allocated to hardware and software components

Capella

New Components defined



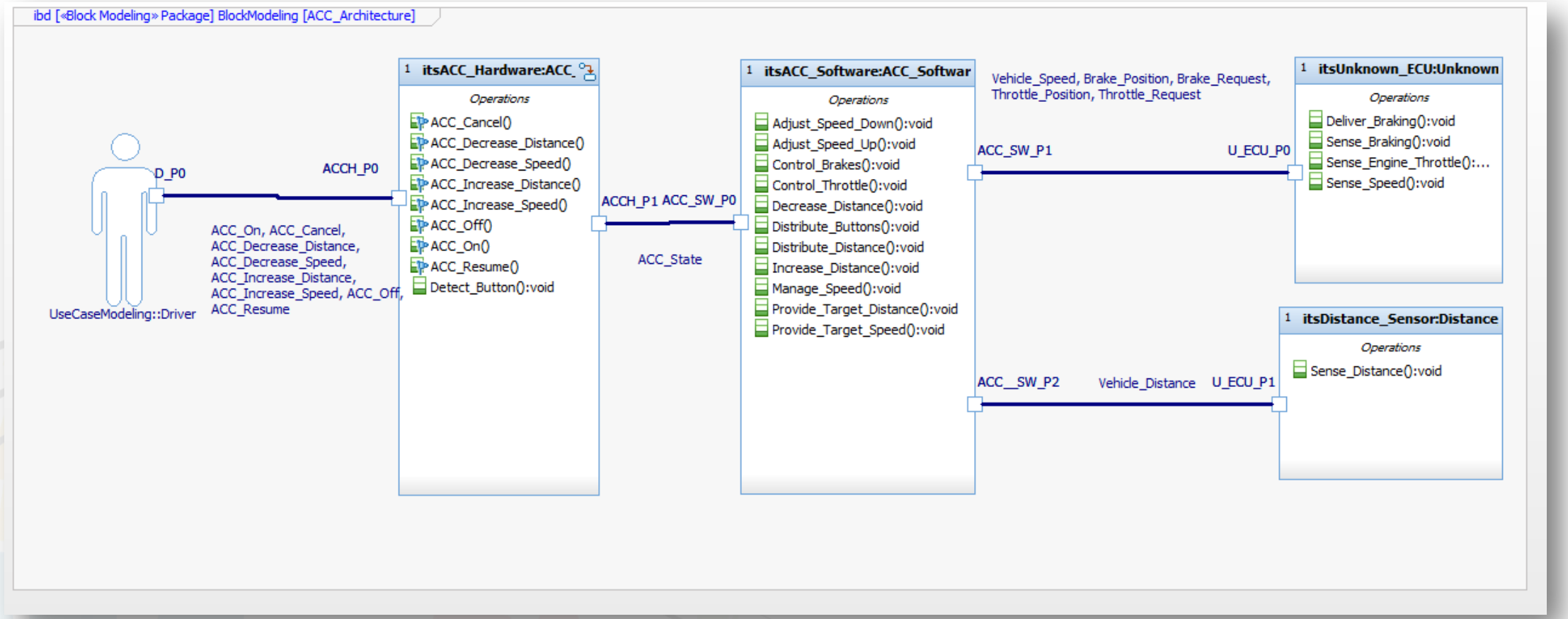




Architecture Development of system model results in functionality allocated to hardware and software components

SysML v1

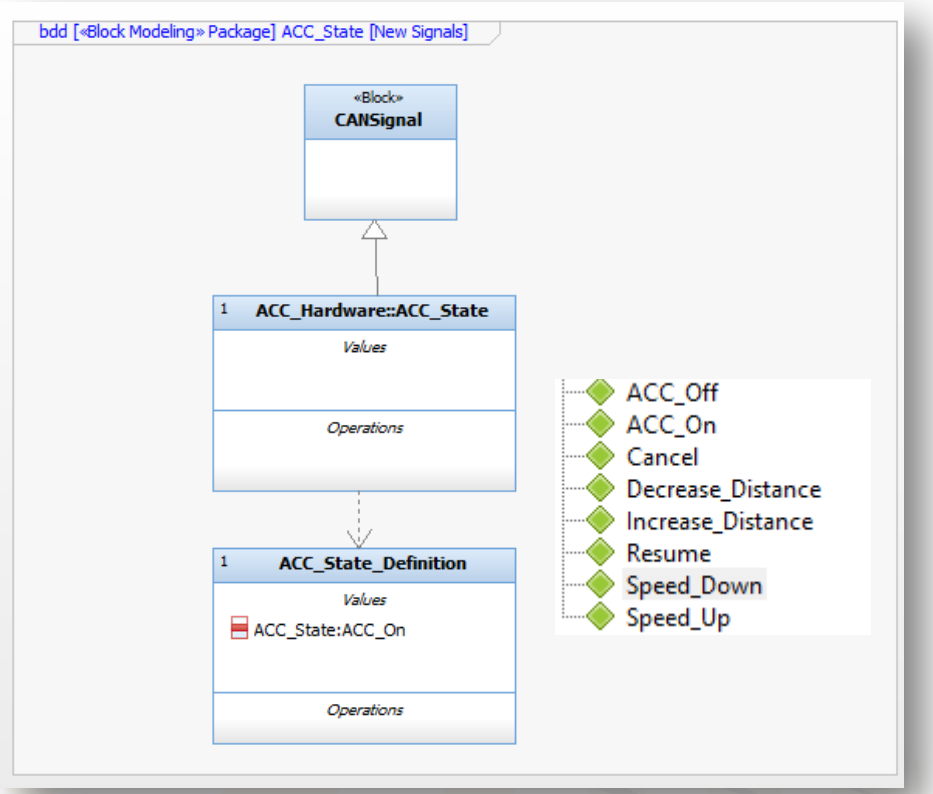
Components defined



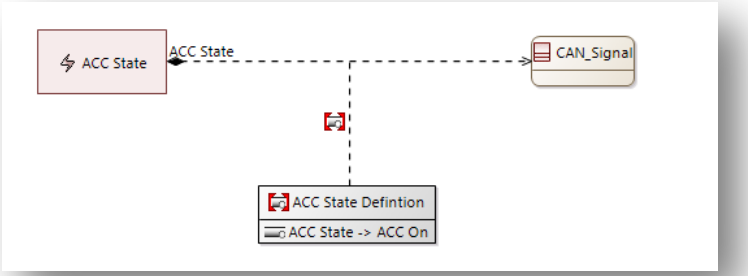
To Do : Need Updating

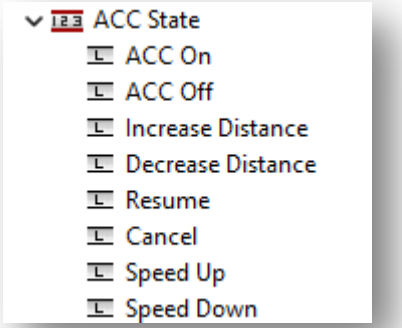
Architecture Development results in new network data required for the new feature.

SysML v1



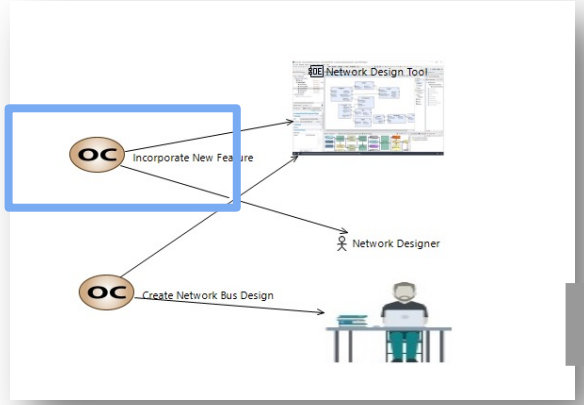
New Signal Identified





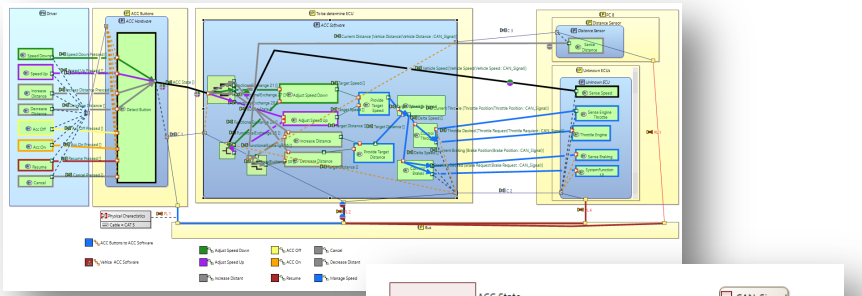
Capella

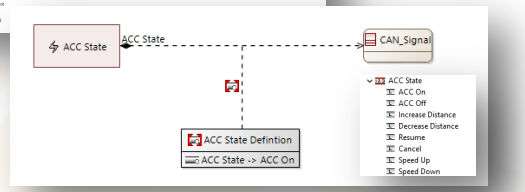
Network Designer – Incorporates the System Model Feature

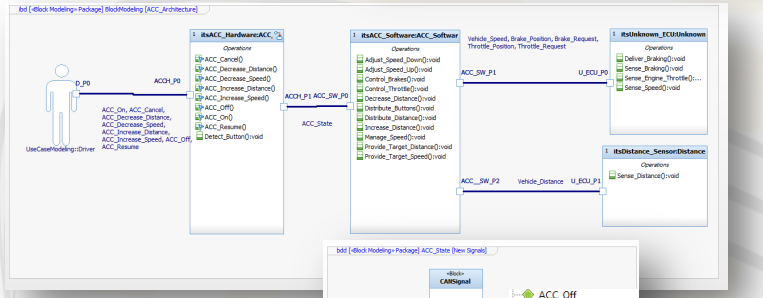


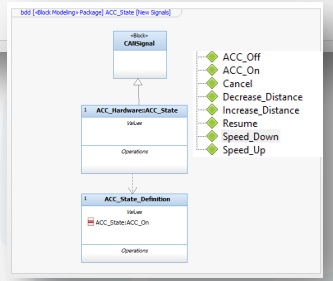
- Queries can provide the means get the system model components for the new feature
- Translations on specific model constructs yields interoperability between domains

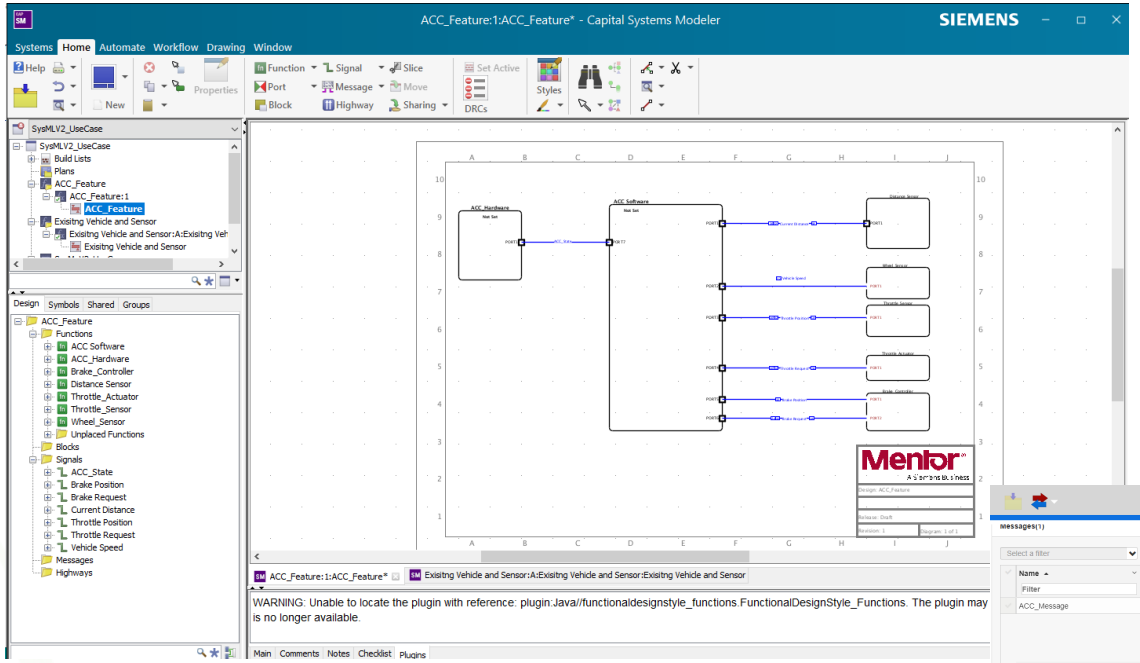
Network Design incorporate new ACC architecture in the Network Design System Model

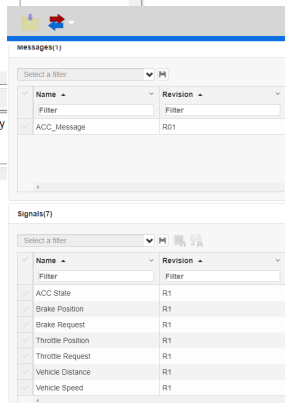




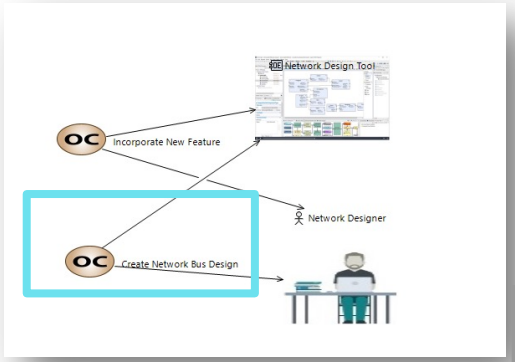




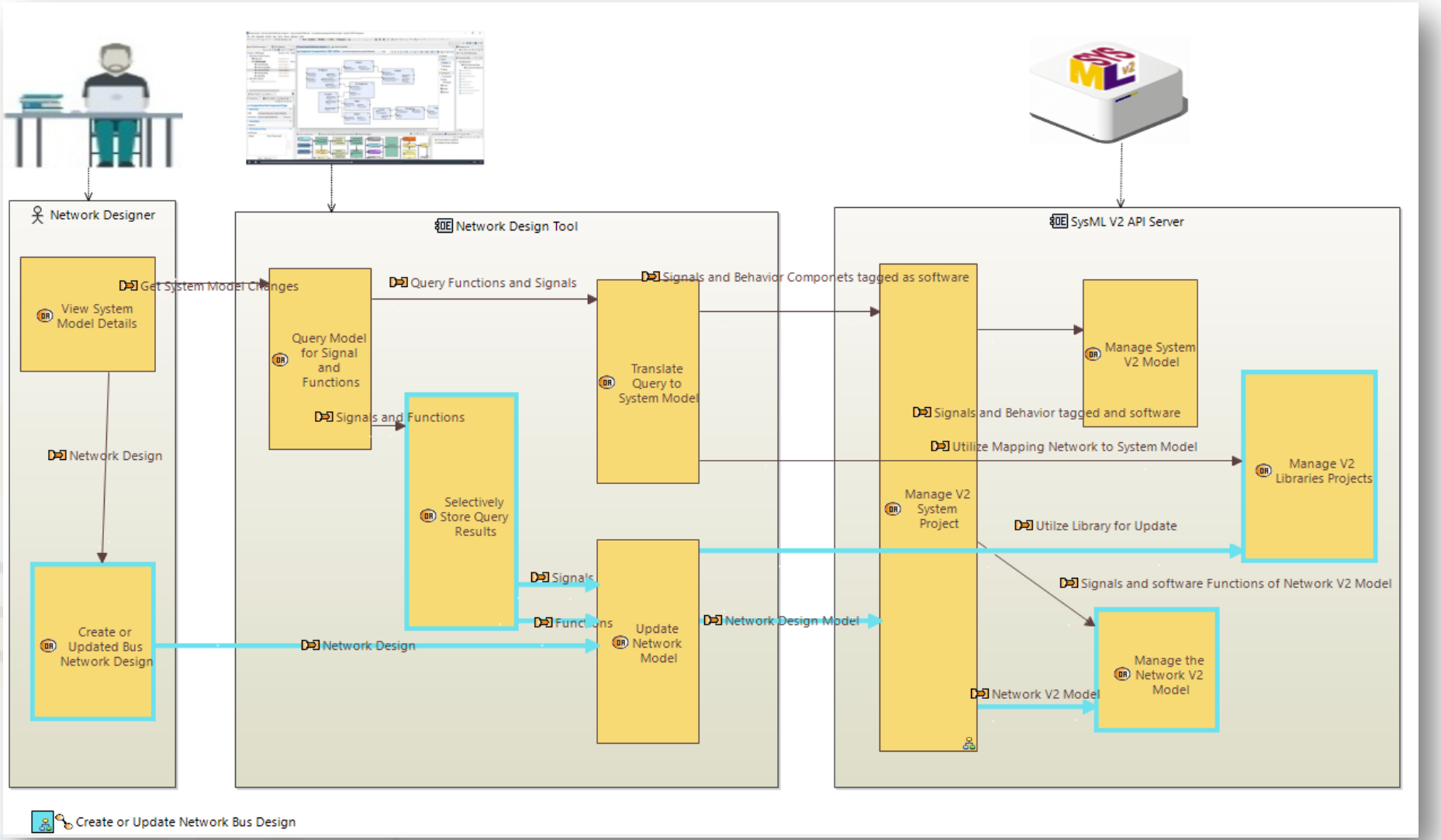




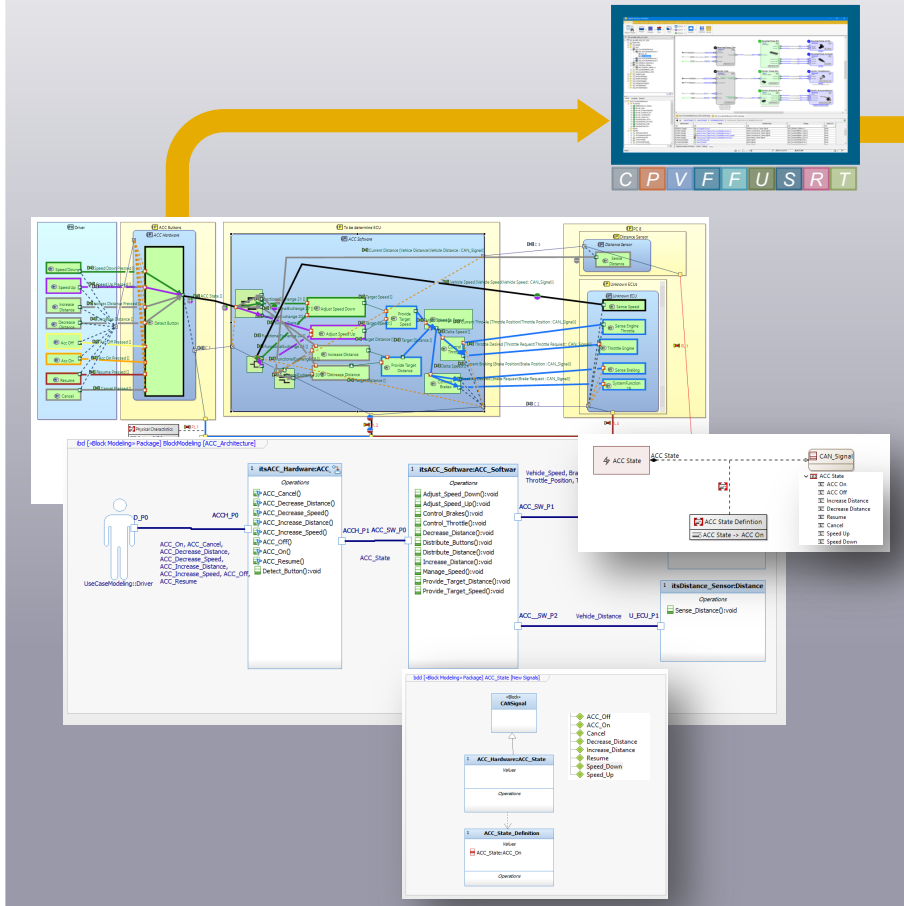
Network Designer – Creates the detailed network design content for the Platform

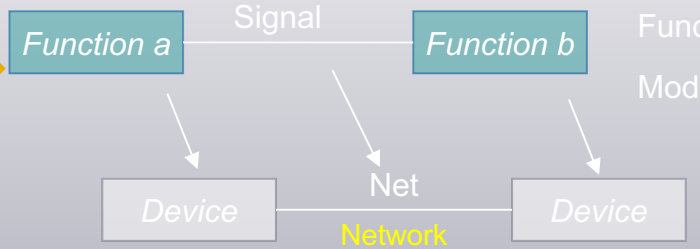


- Subsets of translated model constructs can be brought to Domain
- Network Model is saved to the V2 Repository



Network Designer brings the system Model Feature to the Platform





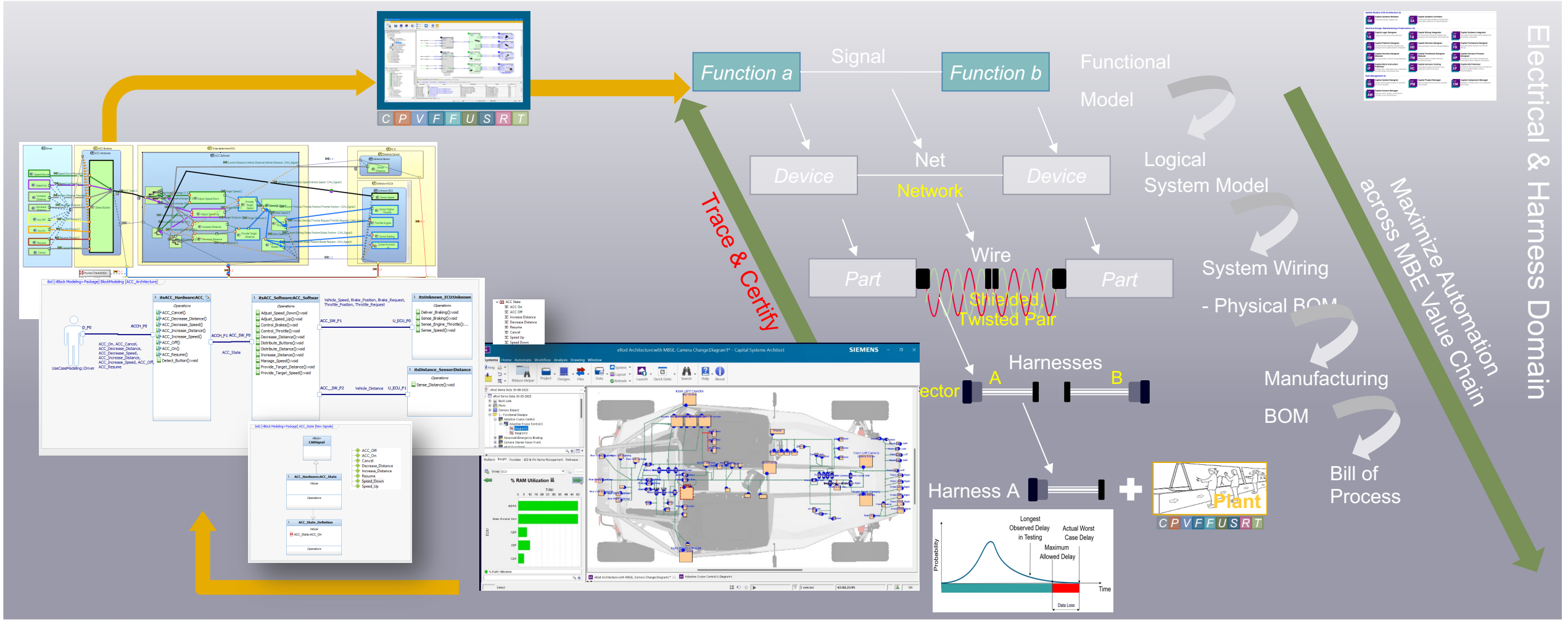
CAP SM **Capital Systems Modeler**
Functional design capture tool.

CAP SA **Capital Systems Architect**
Functional implementation and planning.
Automated synthesis & metric feedback.

Brings in only “new” signals and “new” functions

Electrical & Harness Domain

Network Designer expands the feature into the E/E content for the Platform

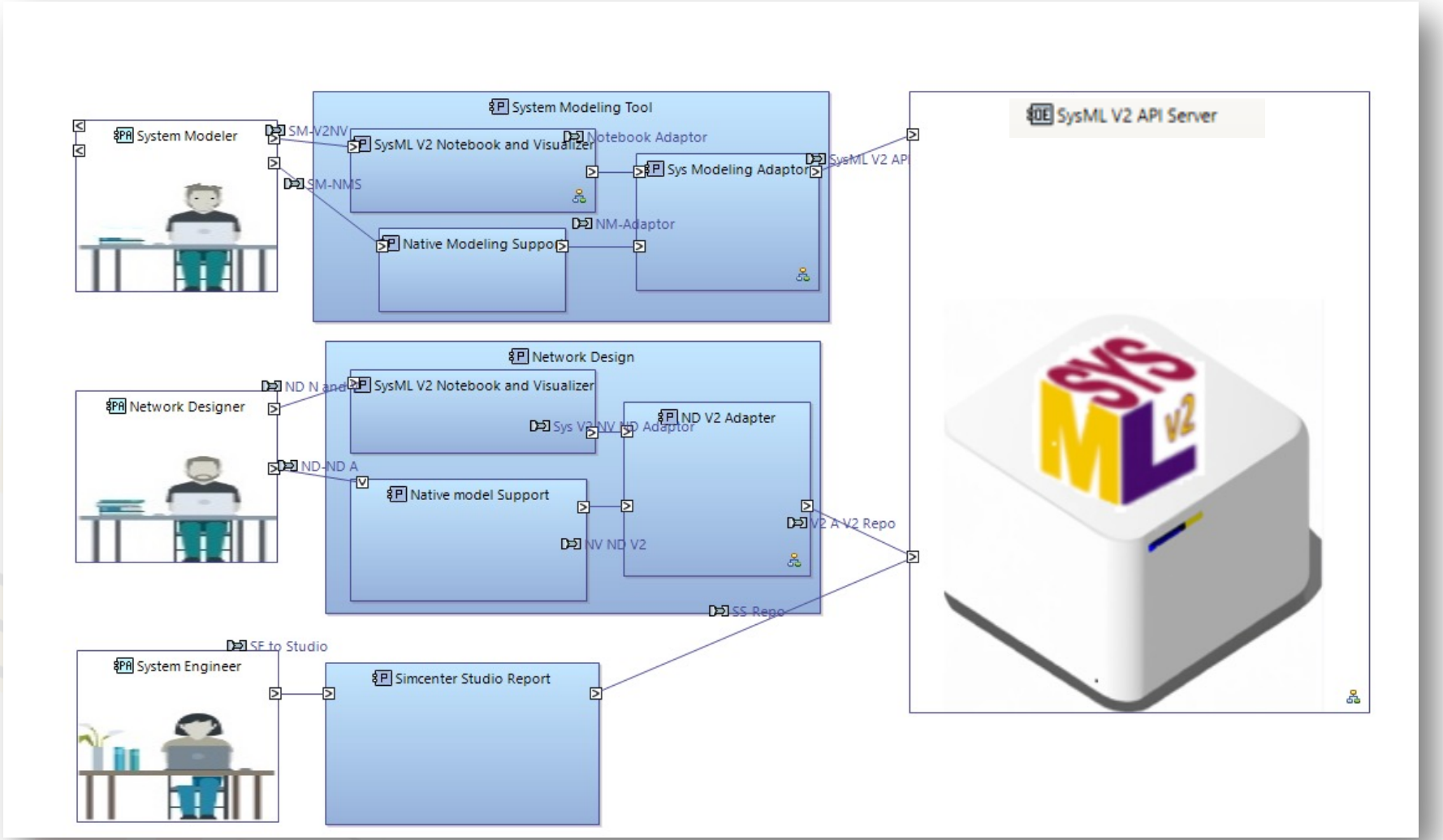


Presentation Topics

- Understanding the Problem
  - Operational Analysis Model of the Problem
    - Actors and Entities
    - Operational Capabilities
    - Actors and their Activities to Deliver Capability
      - Introduce an Example to Enhance understanding
- **Describe an overview of Physical Architecture of solution**
- Conclusions to System Engineering and Domain Engineering

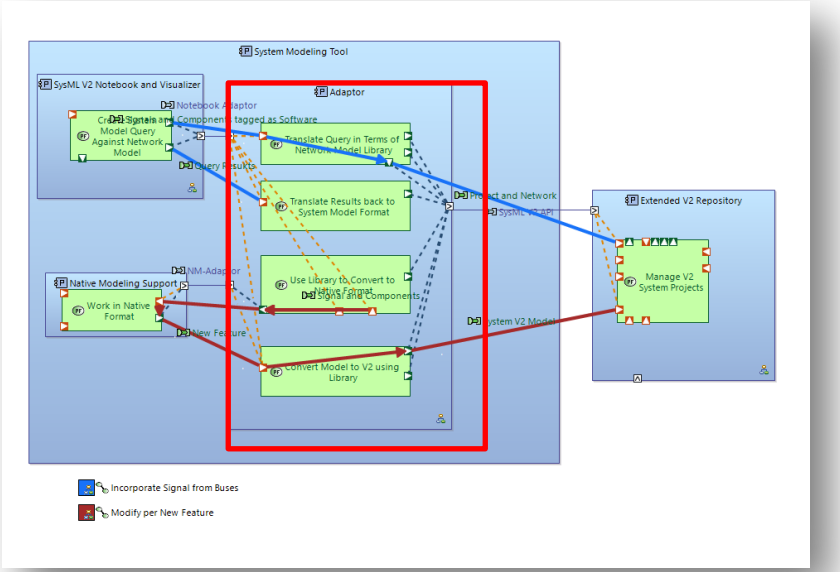
Physical Architecture Components

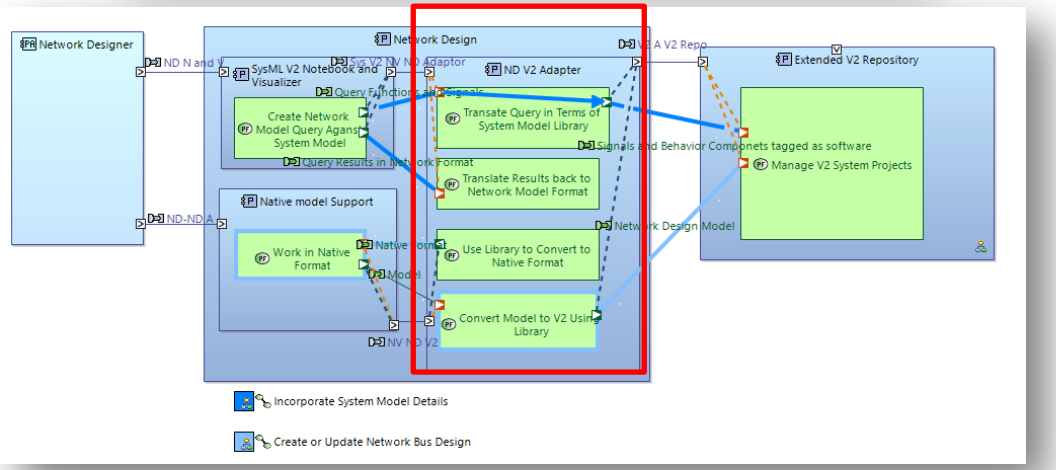
- Native Modeling Tool
- V2 Adapters
- SysML V2 Notebook and Visualization
- SysML V2 API Server



Domain Application Adapters

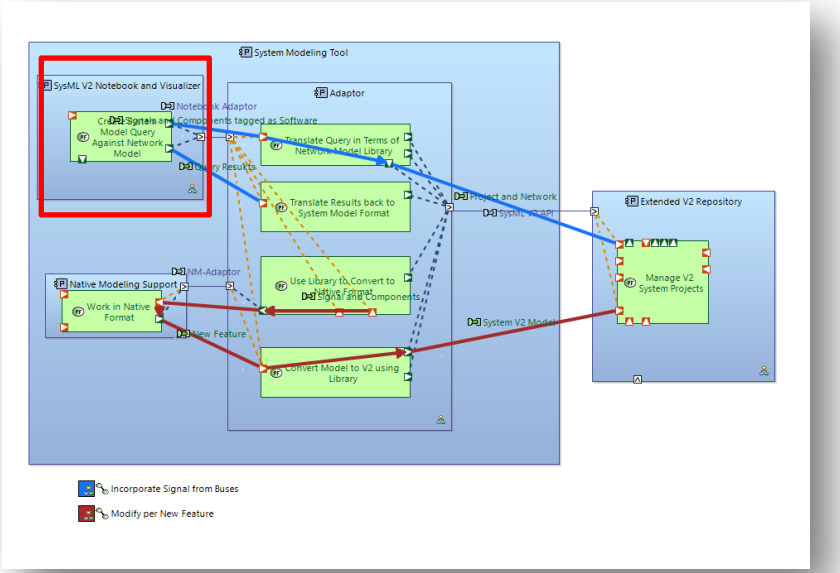
- Perform bi-directional translation from one V2 representation format to another
  - Utilize the V2 Libraries
- Connects to specific instance of V2 Compliant API Server

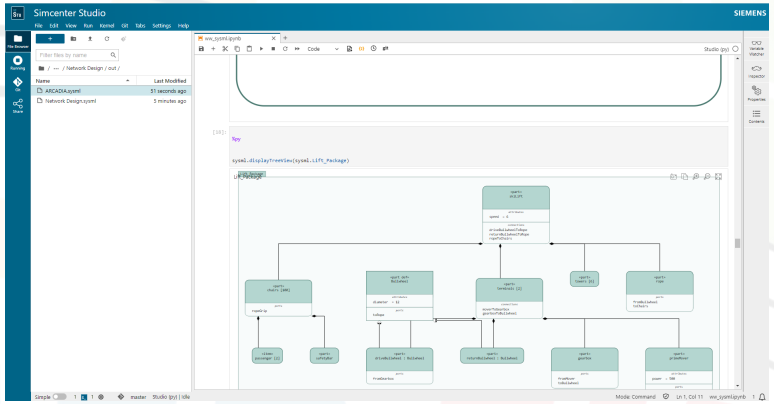


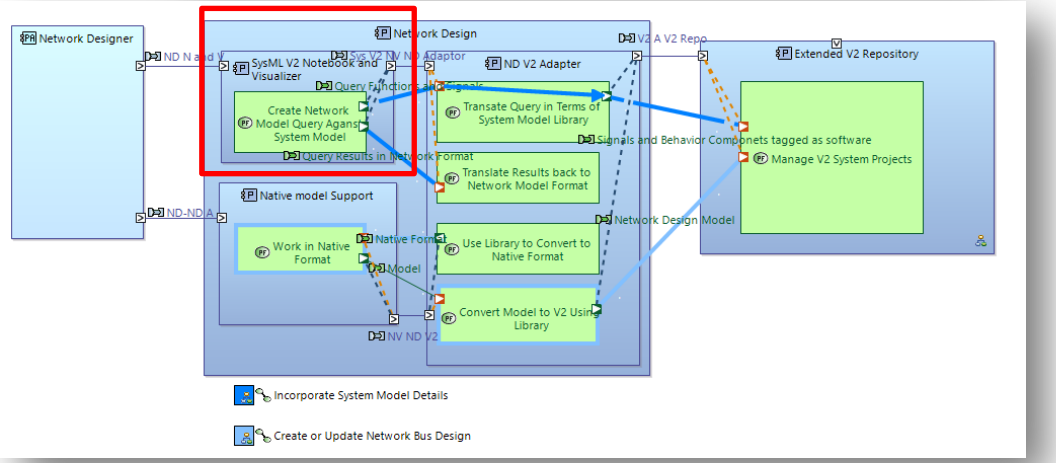


SysML V2 Notebook and Visualization

- Easily Adapted to Perform Repetitive Operations and to Explore retrieved model content
  - Used by System Engineer to generate reports and documents
  - Used to automate the translation and retrieval of model content
  - Used to pull subset information
  - Used to explore “new” concepts via studies
- Provide insight into model representation in a V2 neutral format







Conclusions

- SysML V2 holds representation holds “Authoritative Source of Truth” for each Domain
  - V2 compliant API server provides access to domain extended v2 format
- SysML V2 facilitates cross domain analysis and exchange
  - Domain application adaptors create and translate targeted subsets or information for integration with Native Authoring Tools
- SysML V2 has power to breakdown organizational barriers through leveraging and exchanging target information
- Future directions
  - Explore the potential of AI to analyze and translate domain specific representations

Acknowledgement

- Technical
  - Karen Ryan – Keeping me honest with V2 concepts
  - Ed Siedewitz – Bringing V2 to life
  - Jason Wickers – Bring Capella to V2 translation to life
  - Rich Morrell – Network Domain Knowledge
  - Shashank Alai – Encouragement to keep tackling hard problem

- Organizational Support
  - Kevin Drooger – Freedom to pursue the big problems for our customers

- Emotional Support
  - Heather Komar – Keeping the family running while I am at the keyboard