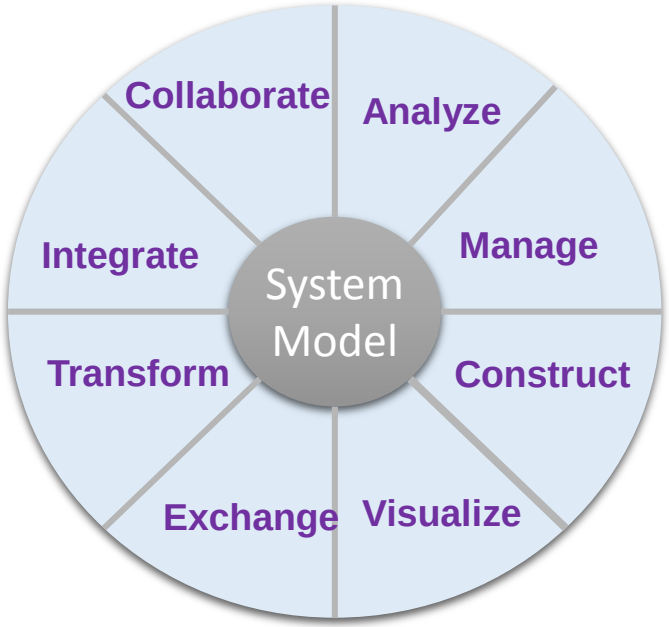
System Modeling Assessment & Roadmap
Joint OMG/INCOSE Working Group

Systems Engineering Model Construction Focus Area



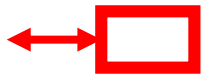
Snapshot
June, 2016

OMGSysML Model Construction Wiki

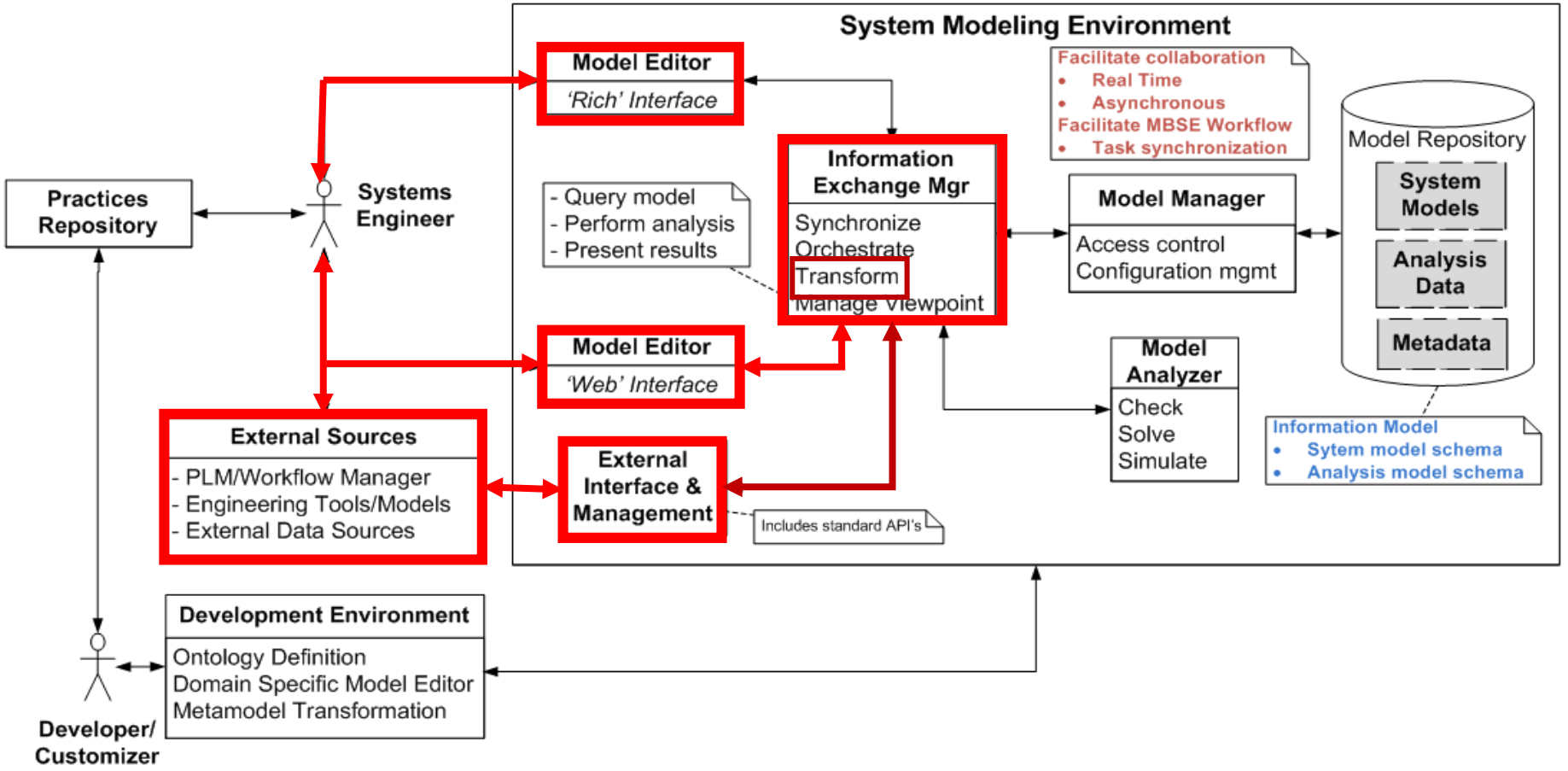
MBSE Capability: Construct Model

- **Task objective**
  - Elaborate concepts, requirements, and metrics for effective model construction that support the next generation system modeling language (SysML v2)
- **Use Cases**
  - Systems engineers and other discipline engineers contribute to the development and update of the system model throughout the lifecycle to support system specification, design, analysis, and verification activities
    - Hybrid SUV Change Scenario
- **MoE**
  - Ability to efficiently and intuitively construct models
- **High Level Intent/Driving Requirement:**
  - Intuitive and efficient model construction.
  - It often requires several clicks to capture a core concept in a model.
  - Reduce the time and effort to build and maintain a model.
  - Repeat common modeling patterns with reduced user input
- **Services**
  - Documented in wiki

Systems Modeling Environment
Conceptual Architecture - Construct Model



Initial Scope



Topics

- Follow-up on actions from the Reston WG meeting
  - Examples and linking to complete set of services
- Effectiveness measures and driving requirements
  - Related to model creation and update efficiency, intuitiveness, correctness, completeness
- Limitations of current SysML
  - Focused on the current hurdles related to creating models from scratch and/or leveraging existing models structures (libraries, patterns)
  - Creating and linking system test concepts (e.g. Use Case but no explicit, related Test Case)
- Key features and graphic of their concept
  - Included in this presentation for feedback
- Service requirements (e.g. functions)
  - Complete set for CRUDE (Create, Retrieve, Update, Delete, Execute) included in Standard API wiki page
- Illustration of how their concepts supports the Hybrid SUV scenario
  - Included in the Model Construction wiki page
- Plans for prototypes to demonstrate feasibility
  - No separate plans….related to Collaboration, Model Management prototypes
- Wiki status
  - Updated incrementally

Status

- **Status of the effort (including whether Wiki is current)**
  - Wiki is current
  - Updates made to Concept Model references SECM Concepts
    - Suggest a Use Case / Services driven walk through of the Concept Model by each lead to check for completeness
  - Link to common supporting services wiki page completed
  - Services and Use Cases consistent Model Construction Use Cases
- **Effectiveness measures and driving requirements**
  - Ease of use (model construction complexity measures)
  - Automation opportunities (import of external data)
  - Validation aspects (correctness, completeness of the model construction)
  - Model Quality Analytics (e.g. assessment of complexity, depends on method profile employed (SysMOD, Harmony, OOSE, etc.)
- **Limitations of current SysML**
  - Executable aspects of the system model (linkage to simulation, physics & dynamic models)
  - Constraints language for typical systems engineer (beyond OCL)
  - Parametrics equations (free text is inadequate....reference a standard and incorporate in tools)
  - Temporal modeling (some hooks, but should go much further to adequately model “time”
  - Spatial modeling (relative, logical and geospatial)
  - Viewpoint and View definitions and mechanisms
  - Accommodation of Patterns
  - Support for non-model data import (tables, text, etc.)
  - Ease of creating new “sub profiles” that are tailored to problem domains (vocabulary, constraints, patterns, etc.)
  - Availability of a human readable/understandable language to represent SysML (XMI is for machines not humans)

Status

- **Key features of the concept**
  - Automation mechanisms for structured & unstructured data import
  - Linkage with advanced visualization
  - Linkage with Model Management
  - Linkage with Collaborative Model Development
- **Service requirements (e.g. functions)**
  - see list at Interoperability Working group
- **Illustration of how the concepts supports the Hybrid SuV scenario**
  - see explanation below Hybrid SuV scenario
- Change Scenario Example - Model Construction Focus
- **Plans for prototypes to demonstrate feasibility**
  - No plans as yet, suggest combining with Model Lifecycle Management prototype plans

Follow up Actions

- Practical Examples for Model Construction
  - Batch Mode - Three Use Cases
  - Relating Methodology and Workflow - Coordinating with Workflow Group
  - Textual Input (Narrative View) -
    - Structured e.g. Tabular
    - Semi-Structured e.g Requirements specification
    - Unstructured e.g. Concept of Operations document

Change Scenario Example - Model Construction Focus

- The following Hybrid SUV Change Scenario will be used to illustrate the concept:
  - Vehicle design unable to meet a requirement (e.g., stopping distance, safety, stability)
  - Propose Requirement Change
  - Assess potential impact
  - Propose update to system design
  - Implement/update design
  - Verify system meets requirement

Next Generation Systems Modeling Environment – Model Construction

Highlights

From a Macro Model Sketch
to a High Fidelity
Micro Model

Physical – Technical & Non Technical

CRUDEA Services: Create, Retrieve, Update, Delete, Execute, Analyze

Context – Technical & Non Technical

CRUDEA Services: Create, Retrieve, Update, Delete, Execute, Analyze

Logical – Technical & Non Technical

CRUDEA Services: Create, Retrieve, Update, Delete, Execute, Analyze

Synchronized View Model Construction Unstructured Text

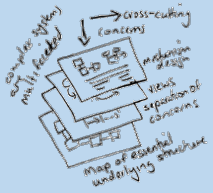
	A	B	D
1	id	name	text
2	REQ-0	HSUV Specification	
3	REQ-0.1	Performance	The HSUV shall have...
4	REQ-0.2	Capacity	The HSUV shall have the capacity...
5	REQ-0.1.1	Braking	The HSUV shall have the braking...
6	REQ-0.1.2	Fuel Economy	The HSUV shall have fuel economy...
7	REQ-0.1.3	Acceleration	The vehicle should have a 0-30 mph...
8	REQ-0.1.4	OffRoad Capability	The HSUV shall have the offroad capability...
9	REQ-0.3	Regenerative Breaking	Regenerative braking should not adversely impact...
9	REQ-0.4	Power	The power of the engine...
10			
Requirement		Problems | Rationale	TestCase

Size, Weight, Location, Interface

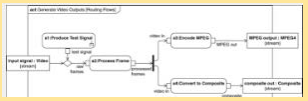
Macro Model Sketch

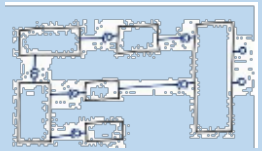
Environment, Location, Mission

Systems, Subsystems, Products, Components, Interfaces

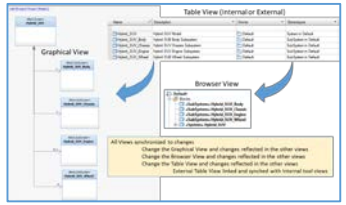


Micro Model High Fidelity





Synchronized View Model Construction with Tables



Time, Power, Temperature, Resource Consumption

Cost, Schedule, People, Political

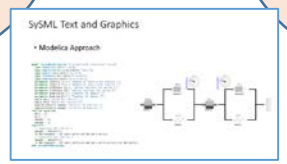
Sense, Control, Command, Effect, Communicate

CRUDEA Services: Create, Retrieve, Update, Delete, Execute, Analyze

Performance – Technical & Non Technical

CRUDEA Services: Create, Retrieve, Update, Delete, Execute, Analyze

Constraints – Technical & Non Technical



CRUDEA Services: Create, Retrieve, Update, Delete, Execute, Analyze

Functional – Technical & Non Technical

Model Construction with Tables

Highlights

Synchronized View Model Construction with Tables

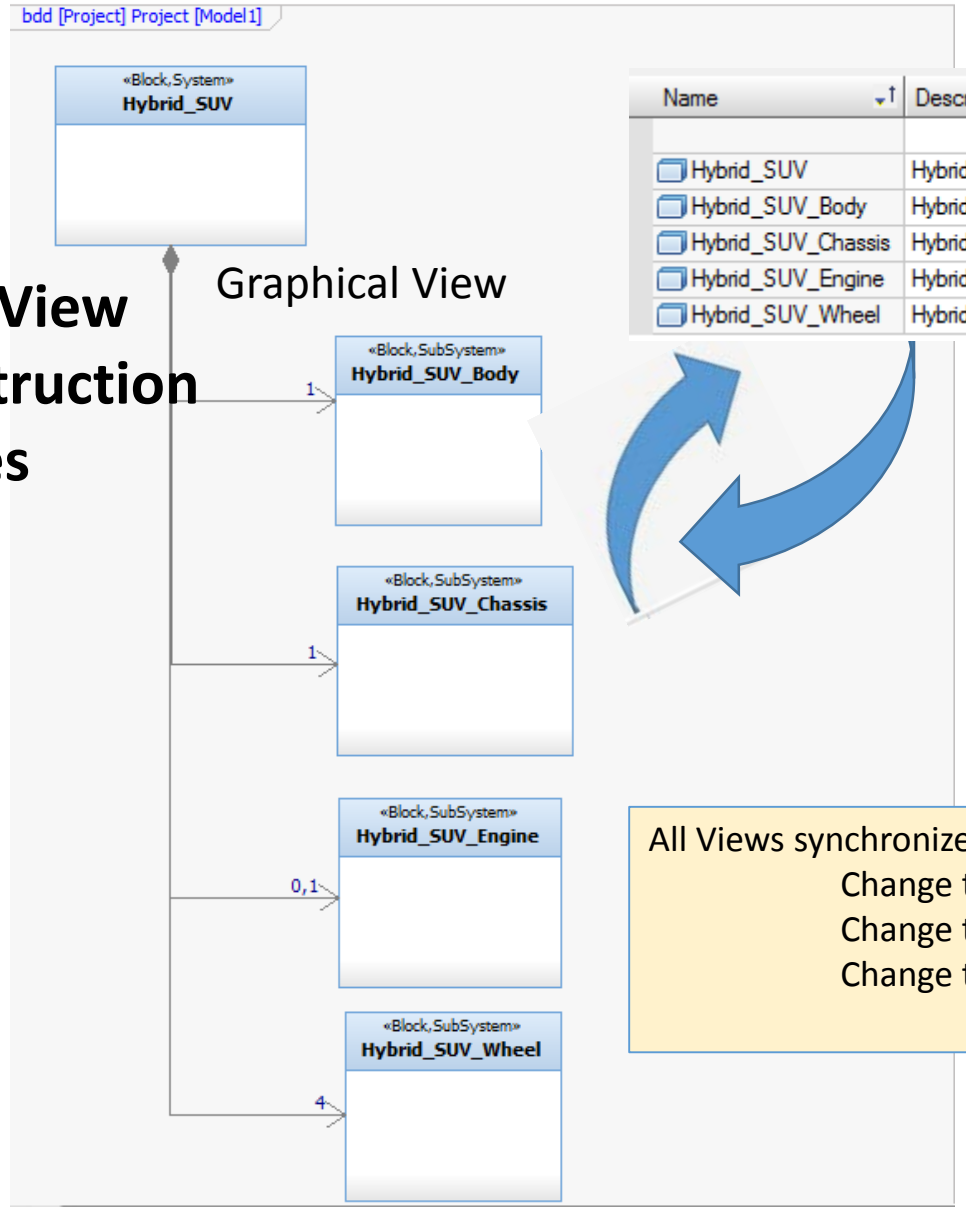
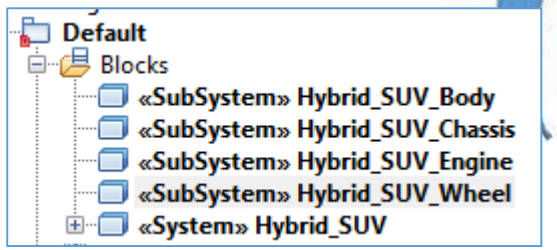


Table View (Internal or External)

Name	Description	Owner	Stereotypes
Hybrid_SUV	Hybrid SUV Model	Default	System in Default
Hybrid_SUV_Body	Hybrid SUB Body Subsystem	Default	SubSystem in Default
Hybrid_SUV_Chassis	Hybrid SUV Chassis Subsystem	Default	SubSystem in Default
Hybrid_SUV_Engine	Hybrid SUV Engine Subsystem	Default	SubSystem in Default
Hybrid_SUV_Wheel	Hybrid SUB Wheel Subsystem	Default	SubSystem in Default

Browser View



All Views synchronized to changes

Change the Graphical View and changes reflected in the other views

Change the Browser View and changes reflected in the other views

Change the Table View and changes reflected in the other views

External Table View linked and synched with internal tool views

Model Construction with Unstructured Text

Synchronized View
Model Construction
Unstructured Text

Unique Requirement Name Established

Unique Requirement ID Established

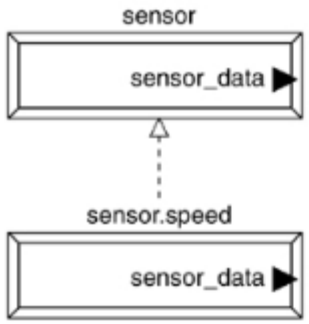
Requirement Text Semantically Parsed to capture system components, characteristics, functions, constraints, timing, value properties, etc

	A	B	D
1	id	name	text
2	REQ-0	HSUV Specification	
3	REQ-0.1	Performance	The HSUV shall have...
4	REQ-0.2	Capacity	The HSUV shall have the capacity...
5	REQ-0.1.1	Braking	The HSUV shall have the braking...
6	REQ-0.1.2	Fuel Economy	The HSUV shall have fuel economy...
7	REQ-0.1.3	Acceleration	The vehicle should have a 0-30 mph...
8	REQ-0.1.4	OffRoad Capability	The HSUV shall have the offroad capability...
9	REQ-0.3	Regenerative Breaking	Regenerative braking should not adversely impact...
9	REQ-0.4	Power	The power of the engine...
10	...	...	...

Requirement | Problems | Rationale | TestCase

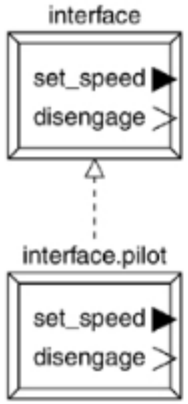
SysML Text (Narrative) and Graphics

- AADL Approach



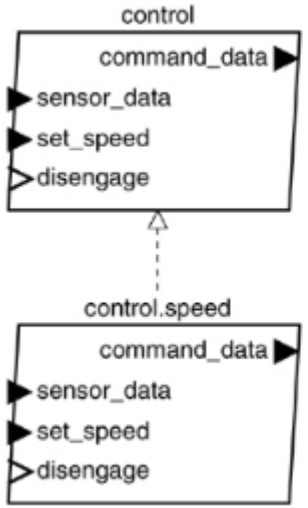

```
device sensor
  features
   sensor_data: out data port;
end sensor;


device implementation sensor.speed
end sensor.speed;
```



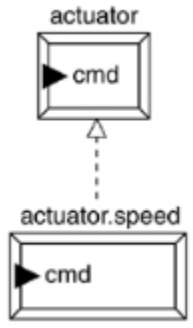

```
device interface
  features
   set_speed: out data port;
   disengage: out event port;
end interface;


device implementation interface.pilot
end interface.pilot;
```




```
process control
  features
   command_data: out data port;
   sensor_data: in data port;
   set_speed: in data port;
   disengage: in event port;
end control;


process implementation control.speed
end control.speed;
```




```
device actuator
  features
   cmd: in data port;
end actuator;


device implementation actuator.speed
end actuator.speed;
```

SySML Text (Narrative) and Graphics

- Modelica Approach

```
model SecondOrderSystem "A second order rotational system"
  type Angle=Real(unit="rad");
  type AngularVelocity=Real(unit="rad/s");
  type Inertia=Real(unit="kg.m2");
  type Stiffness=Real(unit="N.m/rad");
  type Damping=Real(unit="N.m.s/rad");
  parameter Inertia J1=0.4 "Moment of inertia for inertia 1";
  parameter Inertia J2=1.0 "Moment of inertia for inertia 2";
  parameter Stiffness k1=11 "Spring constant for spring 1";
  parameter Stiffness k2=5 "Spring constant for spring 2";
  parameter Damping d1=0.2 "Damping for damper 1";
  parameter Damping d2=1.0 "Damping for damper 2";
  Angle phi1 "Angle for inertia 1";
  Angle phi2 "Angle for inertia 2";
  AngularVelocity omega1 "Velocity of inertia 1";
  AngularVelocity omega2 "Velocity of inertia 2";
initial equation
  phi1 = 0;
  phi2 = 1;
  omega1 = 0;
  omega2 = 0;
equation
  // Equations for inertia 1
  omega1 = der(phi1);
  J1*der(omega1) = k1*(phi2-phi1)+d1*der(phi2-phi1);
  // Equations for inertia 2
  omega2 = der(phi2);
  J2*der(omega2) = k1*(phi1-phi2)+d1*der(phi1-phi2)-k2*phi2-d2*der(phi2);
end SecondOrderSystem;
```