Model-Based Systems Engineering $\rightarrow$ Semantics

Mark A. Austin

University of Maryland

austin@umd.edu

ENCE 688R, Spring Semester 2023

February 6, 2023

Overview

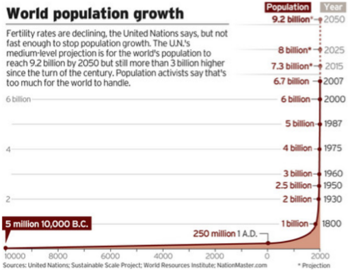
Systems Engineering Drivers

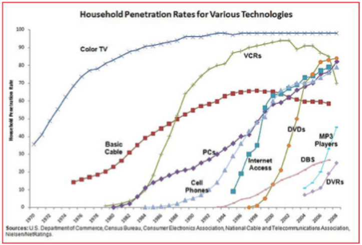
Need for Model-Based Systems Engineering (MBSE) and Software Development

Systems Engineering Drivers

Systems Engineering Drivers

- Increasing demand for limited resources;
- Rapid changes in technology;
- Fast time-to-market most critical;
- Increasing higher performance requirements;
- Increasing complexity of systems/ products;
- Increasing pressure to lower costs;
- Increased presence of embedded information and automation systems that must work correctly;
- Failures due to lack of systems engineering.

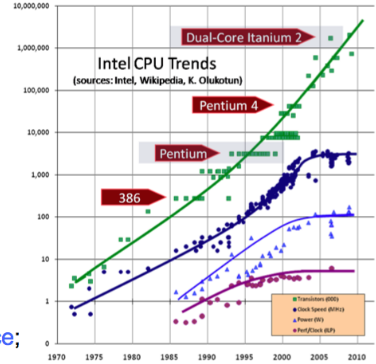




Systems Engineering Drivers

Features of a good design:

- Works correctly;
- Has a wide range of functionality;
- Has great performance;
- Is economical;
- Is resilient to attack;
- Easily adaptable to new functionality.



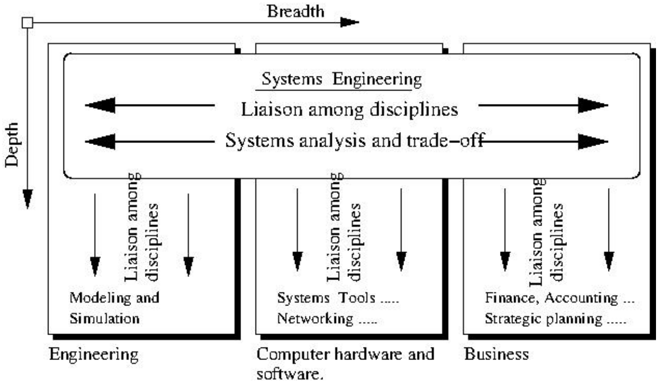
Opportunities for Systems Engineering

- Enhanced levels of attainable performance;
- Create new forms of functionality;
- Improved economics and operational efficiency (zero-energy)
- Improved resiliency and agility ...
- New processes and supply chains for creating systems.

Model-based Systems Engineering

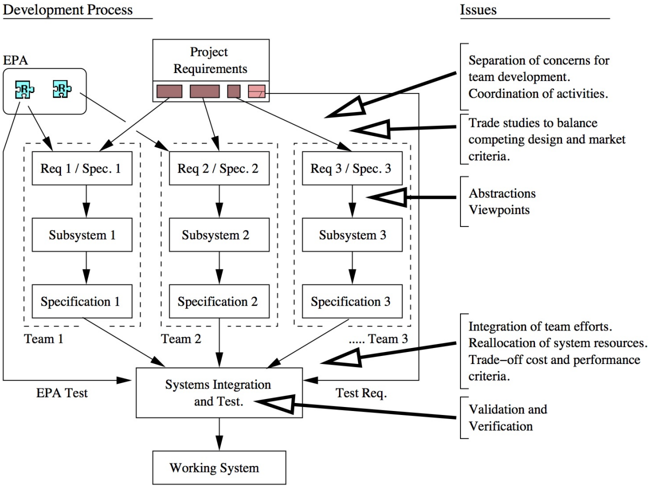
MBSE Concerns

Focus on liaison among disciplines supported by formal methods for systems analysis and design.



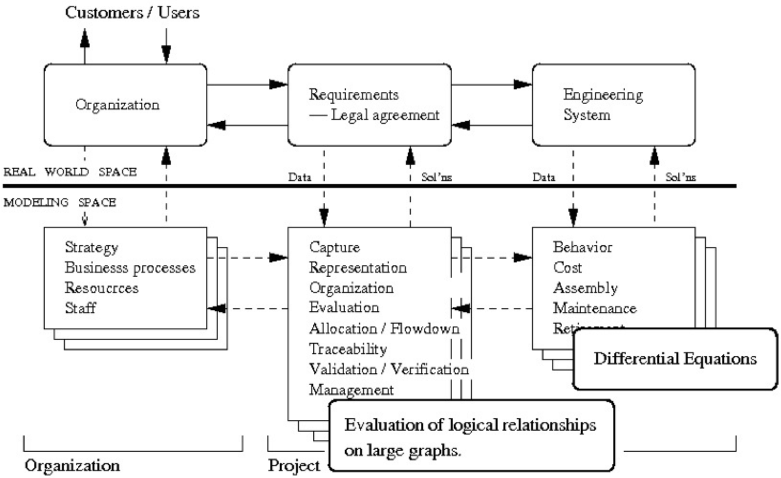
MBSE Concerns

Systems are developed by teams of engineers who must be able to understand one-another's work.



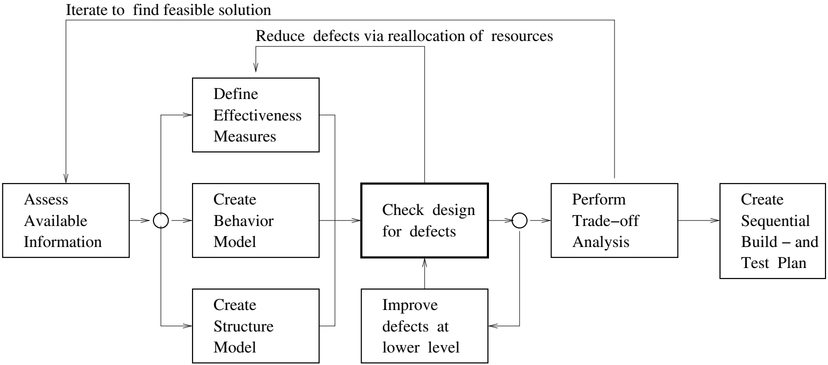
System Modeling Techniques

Organization-Requirements-Engineering Pipeline:



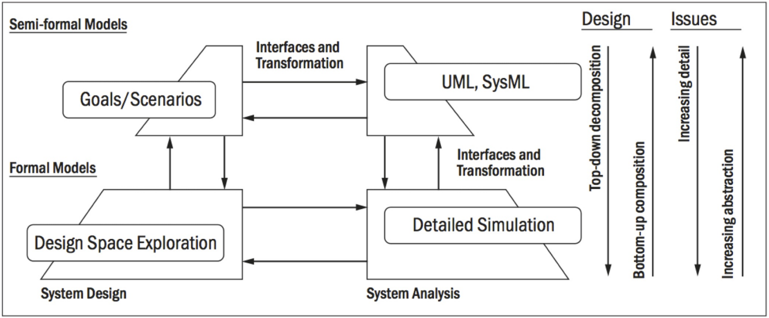
System Modeling Techniques

Core Technical Processes at General Electric:



System Modeling Techniques

Use multi-scale approaches to system modeling:



- Semi-Formal Models: View the complete system (efficiency).
- Formal Models: Detailed view of the actual system (accuracy).

System Modeling Techniques

Semi-Formal Models:

- Provide efficient representation of ideas (e.g., goals and scenarios) and preliminary/tentative design.

Formal Models:

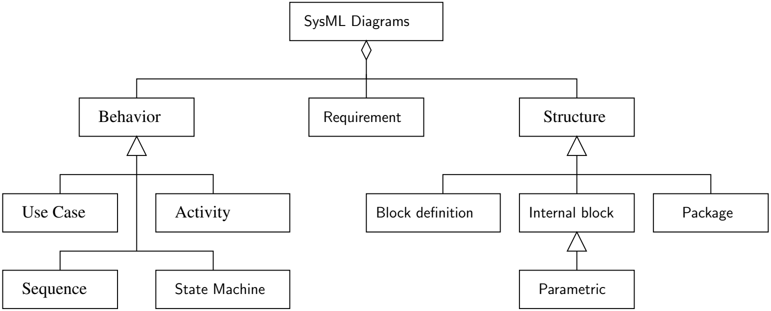
- Formal Models: To help prevent serious flaws in detailed design and operation, design representations and validation/verification procedures need to be based on formal languages having precise semantics.

Abstraction:

- Eliminate details that are of no importance when evaluating system functionality, system performance, and/or checking that a design satisfies a particular property.

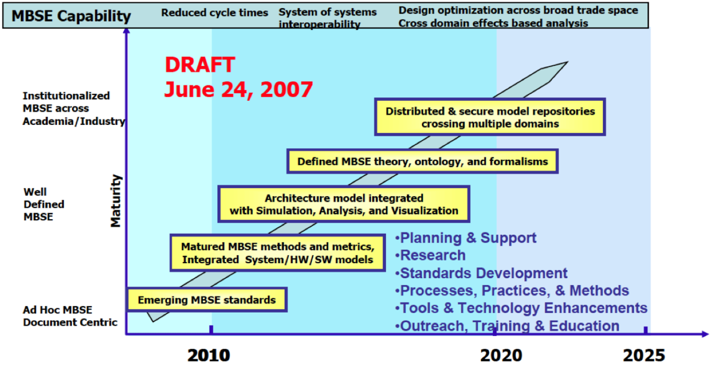
System Modeling Techniques

Taxonomy of diagrams in SysML:



Pillars of SysML: Structure, Behavior, Requirements, and Parametric Diagrams.

INCOSE: MBSE Capability 2020-2025



Notice: Use of AI is implied, but not explicitly stated. No mention of data mining. No mention of machine learning.

Ontologies and Ontology-Enabled Computing

Definition of an Ontology

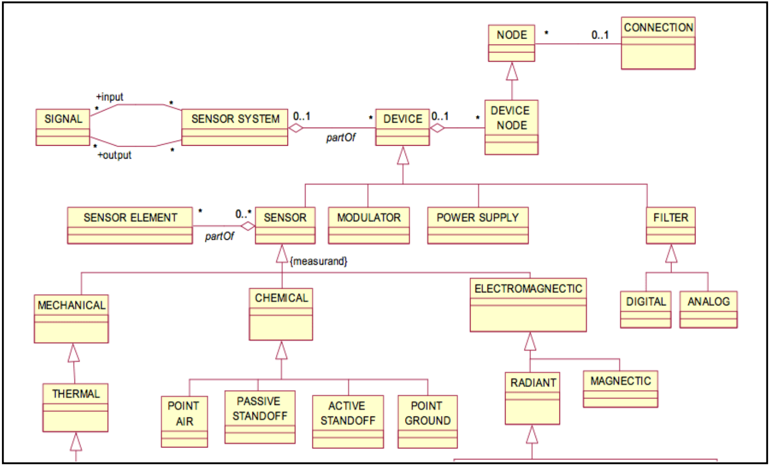
Definition (Ontology)

An ontology is a set of knowledge terms, including the vocabulary, the semantic interconnections, and some simple rules of inference and logic for some particular topic or domain.

Three Goals:

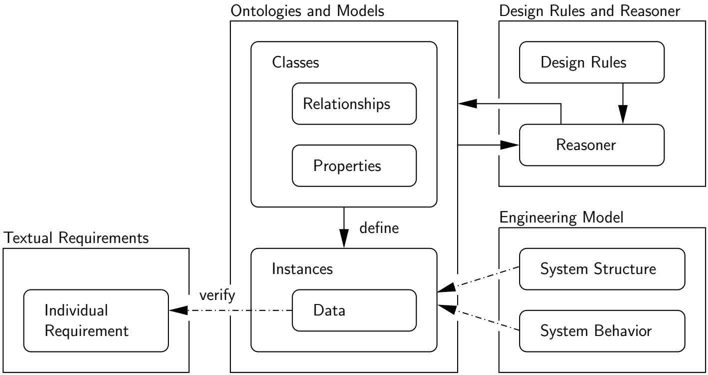
- Provide a semantic representation of each entity and its relationships to other entities;
- Provide constraints and rules that permit reasoning within the ontology;
- Describe behavior associated with stated or inferred facts.

High-Level Sensor Ontology



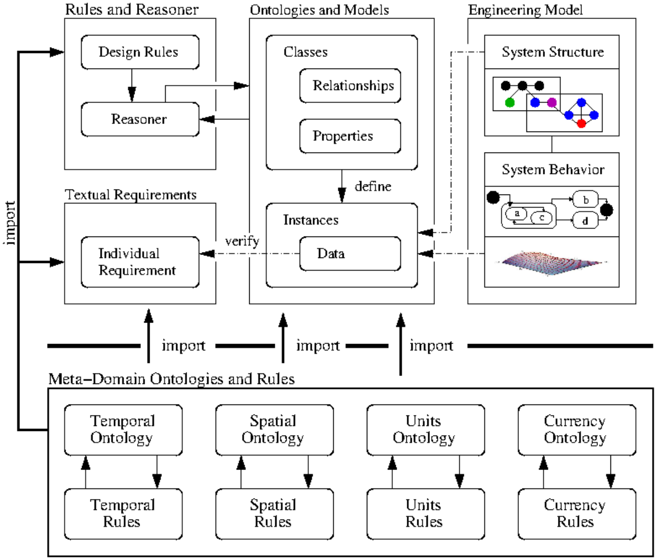
Ontologies and Rule Sets

Framework for Ontology-Enabled Design Assessment (Version 1):



Source: Parastoo Delgoshaei, MSSE Student, 2010-2012. Ph.D. Student in Civil Systems, 2013-2017.

Framework for Model-Based Design



Ontologies and Rule Sets

Benefits of Rule-Based Approaches to Problem Solving:

- Rules that represent policies are easily communicated and understood,
- Rules retain a higher level of independence than logic embedded in systems,
- Rules separate knowledge from its implementation logic, and
- Rules can be changed without changing source code or underlying model.

Benefits of Rules

A rule-based approach to problem solving is particularly beneficial when the application logic is dynamic.

Semantic Web Support for Ontologies

Goals of the WWW

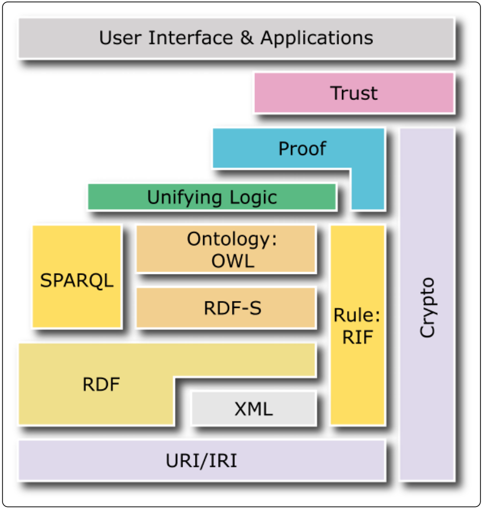
In his original vision for the World Wide Web, Tim Berners-Lee described two key objectives:

- To make the Web a collaborative medium, and
- To make the Web understandable and, thus, processable by machines.

Goals of the Semantic Web

Give information a well-defined meaning, thereby creating a pathway for machine-to-machine communication and automated services based on descriptions of semantics.

Semantic Web Support for Ontologies



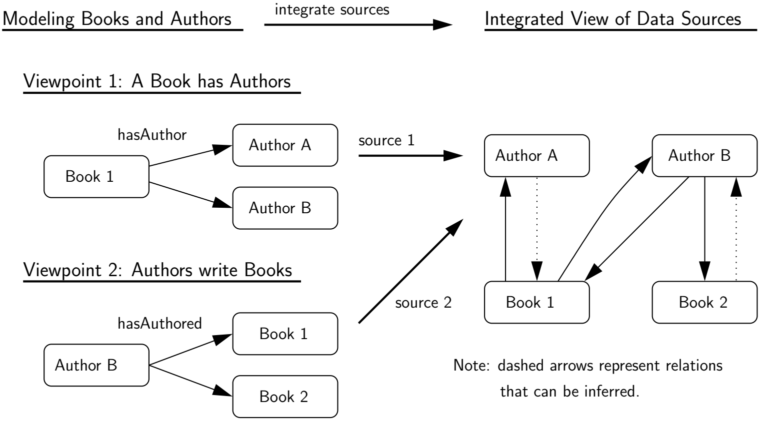
Semantic Web Support for Ontologies

Key Technologies:

- URI – Addresses on the Web.
- XML – Hierarchical storage (tree structures) of data with eXtended Markup Language.
- RDF – Model graphs of resources on the web with resource description framework.
- Crypto – Security and encryption.
- SPARQL – Rdf query language.
- OWL – Web ontology language.
- Logic – Reasoning with rules.
- Proof – Formal verification of goals.
- Trust – How can you believe what you read on the Web?

Semantic Web Support for Ontologies

Process for merging trees of data into graphs:



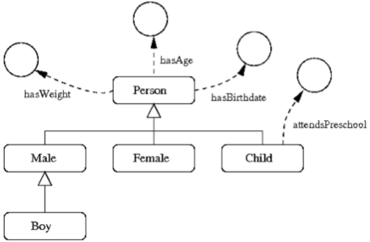
Example 1. A Simple Family Model

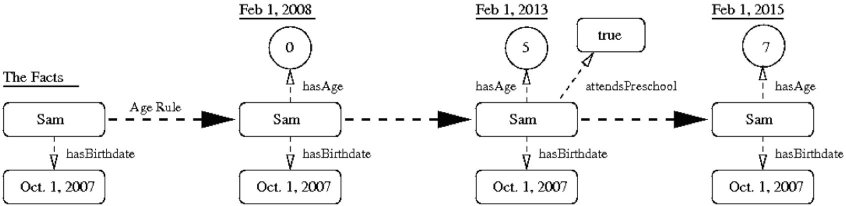
Fact. Sam is a boy. He was born October 1, 2007.

Rule 1: For a given date of birth, a built-in function getAge() computes a person's age.

Rule 2: A child is a person with age < 18.

Rule 3: Children who are age 5 attend preschool.





Example 1. Family Semantic Model

Create Family Individuals:

```
mark =   male.createIndividual(ns + "Mark");
sam  =    boy.createIndividual(ns + "Sam");
nina = female.createIndividual(ns + "Nina");

// Statements "Sam has birthdate 2007-10-01" and "Sam has weight 35"

Literal    dob01 = model.createTypedLiteral("2007-10-01", ...XSDdate );
Statement samdob = model.createStatement( sam, hasDOB, dob01 );
model.add ( samdob );

Literal weight35 = model.createTypedLiteral("35.0", ...XSDdouble );
Statement samw35 = model.createStatement( sam, hasWeight, weight35 );
model.add ( samw35 );
```

Facts in the Simple Family Model:

```
<rdf:Description rdf:about="http://austin.org/family#Sam">
   <j:hasWeight    rdf:datatype="http://www.w3.org/2001/XMLSchema#double"> 35.0 </j:hasWeight>
   <j:hasBirthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date"> 2007-10-01 </j:hasBirthDate>
   <rdf:type rdf:resource="http://austin.org/family#Boy"/>
</rdf:Description>
.........
```

Example 1. Family Rules (Apache Jena Rules)

Apache Jena Rules:

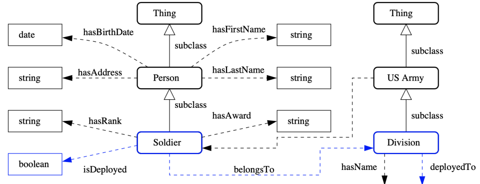
```
@prefix af:  <http://austin.org/family#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

// Rule 01: Propogate class hierarchy relationships ....

[ rdfs01: (?x rdfs:subClassOf ?y), notEqual(?x,?y) ->
        [ (?a rdf:type ?y) <- (?a rdf:type ?x)] ]

// Rule 02: Identify a person who is also a child ...

[ Child: (?x rdf:type af:Person) (?x af:hasAge ?y) lessThan(?y, 18) ->
         (?x rdf:type af:Child) ]

// Rule 03: See if a child attends preschool ...

[ Preschool: (?x rdf:type af:Child) (?x af:hasAge ?y)
             equal(?y, 5) -> (?x af:attendsPreSchool af:True) ]

// Rule 04: Compute and store the age of a person ....

[ GetAge: (?x rdf:type af:Person) (?x af:hasBirthDate ?y)
          getAge(?y,?z) -> (?x af:hasAge ?z) ]
```

Example 1. Query Transformed Semantic Model

```
Statements: Sam ...
======================================================================

Statement[1] Subject   : http://austin.org/family#Sam
             Predicate: http://austin.org/family#hasAge
             Object    : "5.0^^http://www.w3.org/2001/... #double"

Statement[2] Subject   : http://austin.org/family#Sam
             Predicate: http://www.w3.org/1999/02/... s#type
             Object    : http://austin.org/family#Child

Statement[3] Subject   : http://austin.org/family#Sam
             Predicate: http://austin.org/family#attendsPreSchool
             Object    : http://austin.org/family#True

Statement[4] Subject   : http://austin.org/family#Sam
             Predicate: http://austin.org/family#hasWeight
             Object    : "35.0^^http://www.w3.org/2001/... #double"

Statement[5] Subject   : http://austin.org/family#Sam
             Predicate: http://austin.org/family#hasBirthDate
             Object    : "2007-10-01^^http://www.w3.org/2001/... #date"

Statement[6] Subject   : http://austin.org/family#Sam
             Predicate: http://www.w3.org/1999/02/... #type
             Object    : http://austin.org/family#Boy

======================================================================
```

Example 2. Modeling Forrest Gump

Step 1: Design Ontologies and Rules



Rules

[Rule 1: (?x rdf:type Soldier) (?y rdf:type Division) (?x af:belongsTo ?y) —–>
(?y army:deployedTo ?z) —–> (?x af:isDeployed "True")

[Rule 2: (...) (...) (...) —–> (...)(...)]

Step 2: Add Data (1944)

First Name: Forrest
Last Name: Gump
DOB: June 6, 1944
Address: Greenbow, Alabama

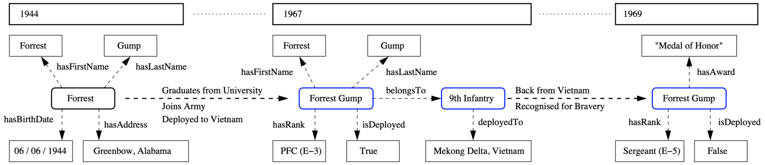
Military Deployment Data (1967)

Rank: PFC (E-3)
Division: 9th Infantry
Deployed: Mekong Delta

Post Deployment Data (1969-)

Rank: Sergeant (E-5)
Awards: Medal of Honor

Step 3: Event-Driven Execution of Semantic Graphs



Example 2. Modeling Forrest Gump

Key Concepts:

- Ontology classes can be organized into hierarchies, e.g., Soldier is a subclass of Person, Person is a subclass of Thing,
- Data properties (e.g., boolean, double, String, date).
- Object properties express association relationships between classes, e.g., Soldier belongsTo Division (a subclass of US Army).
- Ontology classes can inherit properties via the class hierarchy with which they belong, e.g., Soldier inherits the data property hasLastName from Person.
- Jena rules can reason with data and classes belonging to multiple hierarchies.
- Event-driven execution of semantic graphs.

Distributed System Behavior Modeling

Small Networks of Semantic Graphs
Employ Software Design Patterns

MSSE/Ph.D. (Civil Systems) Students

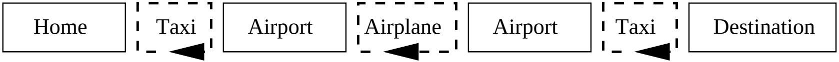
1. Parastoo Delgoshaei (2013-2017);
2. Maria Coelho (2015-present).

Motivation

ENCE 688P: Behaviors in the built environment are distributed and concurrent:

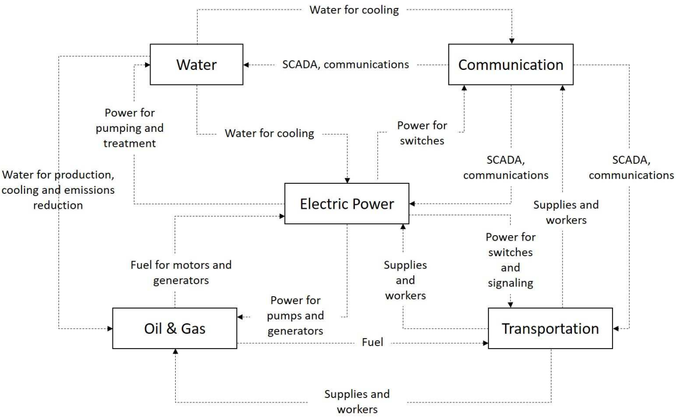
- Cities are system of systems.
- City subsystems may have a preference to operating as independently as possible from the other subsystems.
- Strategic collaboration among subsystems is often needed to either avoid cascading failures across systems and/or recover from a loss of functionality.

Systems-of-systems need not be complicated:



Motivation

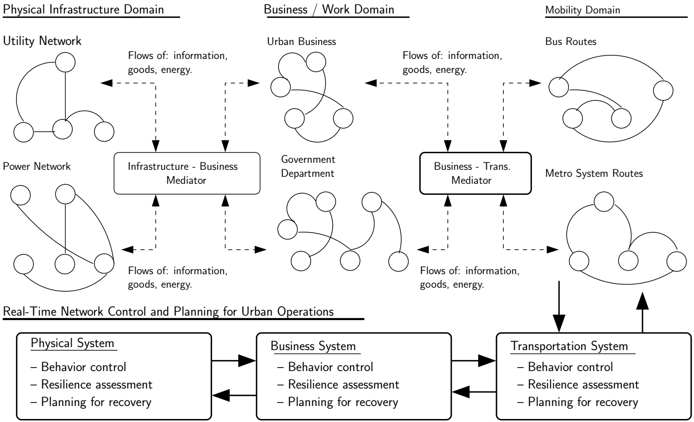
Dependency Relationships Among Different Infrastructures



Source: Gao et al., 2015.

Motivation

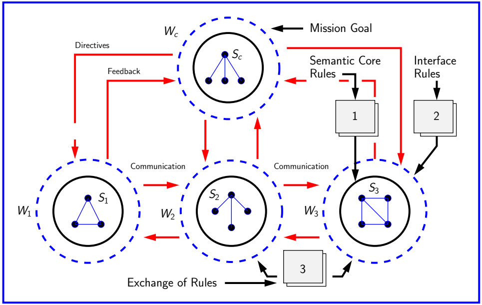
Architecture for Multi-domain Behavior Modeling



Source: Coelho, Austin, and Blackburn, 2017.

Distributed System Behavior Modeling

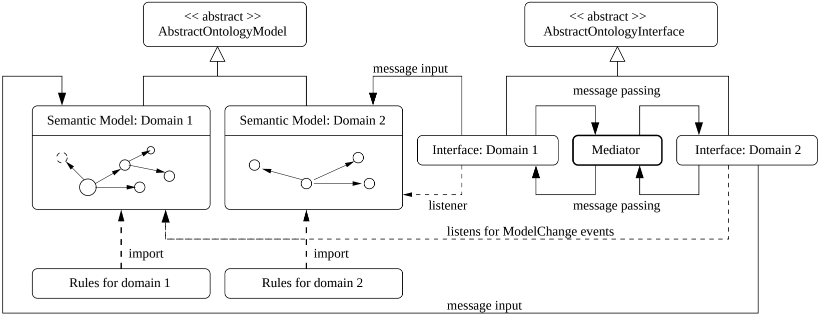
Basic Idea: Model distributed system behavior as a network of communicating semantic graphs.



Wrap entities with interfaces that can respond to events and rule-based reasoning. Enable communication among entities with message passing.

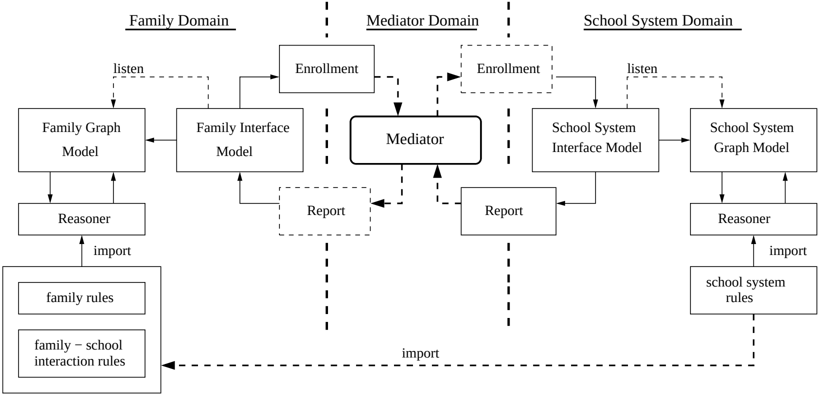
Distributed System Behavior Modeling

Prototype Architecture (2014): Use semantic graphs to model behavior of individual entities (e.g., an organization).



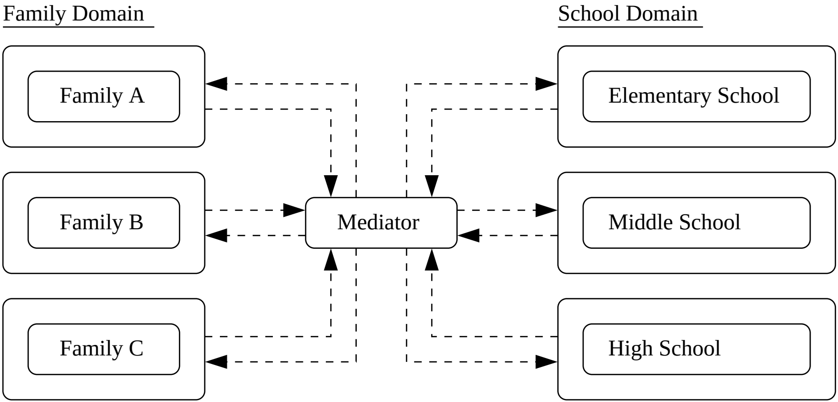
Individual semantic graphs are wrapped with interfaces, and respond to events and rule-based reasoning.

Example 3. Family-School System Dynamics



Note: Exchange of rules to cover admission, day-to-day operations.

Example 3. Framework for Communication



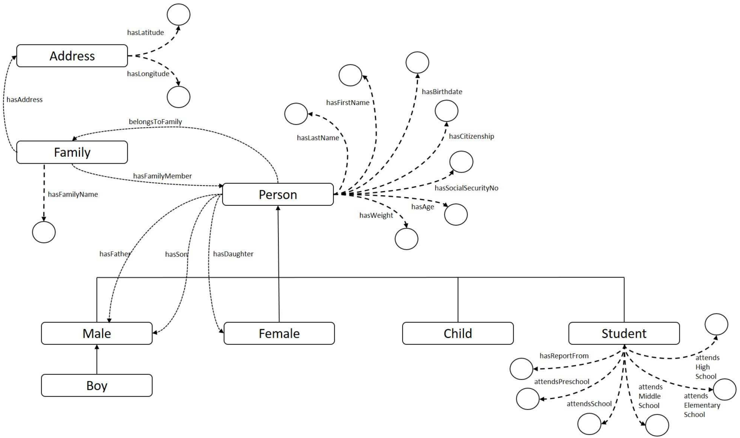
Example 3. Family Datafile (XML)


```xml
<?xml version="1.0" encoding="UTF-8"?>
<FamilyModel author="Maria Coelho" date="2017" source="UMD">
<Family>
    <attribute text="FamilyName" value="Austin"/>
    <attribute text="Address" value="6242 Heather Glen Way, Clarksville, MD 21029"/>
    <Person>
        <attribute text="Type" value="Male"/>
        <attribute text="FirstName" value="Mark"/>
        <attribute text="MiddleName" value="William"/>
        <attribute text="LastName" value="Austin"/>
        <attribute text="BirthDate" value="1704-06-10"/>
        <attribute text="Weight" value="170.0"/>
        <attribute text="Citizenship" value="New Zealand"/>
        <attribute text="SocialSecurity" value="111"/>
    </Person>
    <Person>
        ... description of other Austin family members ....
    </Person>
</Family>
<Family>
    <attribute text="FamilyName" value="Jones"/>
    <attribute text="Address" value="5807 Laurel Leaves Ln, Clarksville, MD 21029"/>
    <Person>
        ... description of Jones family members....
    </Person>
</Family>
</FamilyModel>
```

Example 3. School Datafile (XML)

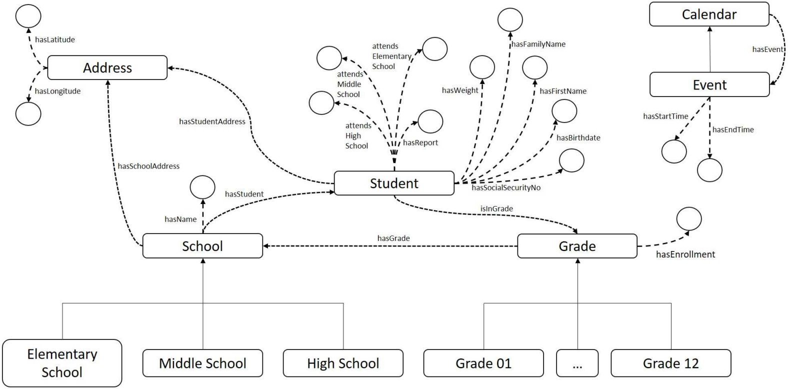

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SchoolSystemModel author="Maria Coelho" date="2017" source="UMD">
   <School>
       <attribute text="Type" value="High School"/>
       <attribute text="Name" value="River Hill High School"/>
       <attribute text="Grade" value="Grade09"/>
       <attribute text="Grade" value="Grade10"/>
       <attribute text="Grade" value="Grade11"/>
       <attribute text="Grade" value="Grade12"/>
       <attribute text="Report Period Start Time" value="2016-09-01T00:00:00"/>
       <attribute text="Report Period End Time" value="2020-10-20T00:00:00"/>
   </School>
   <School>
       ... description of Clarksville Middle School ...
   </School>
   <School>
       ... description of Pointers Run Elementary School ...
   </School>
</SchoolSystemModel>
```

Example 3. Family and School Ontologies



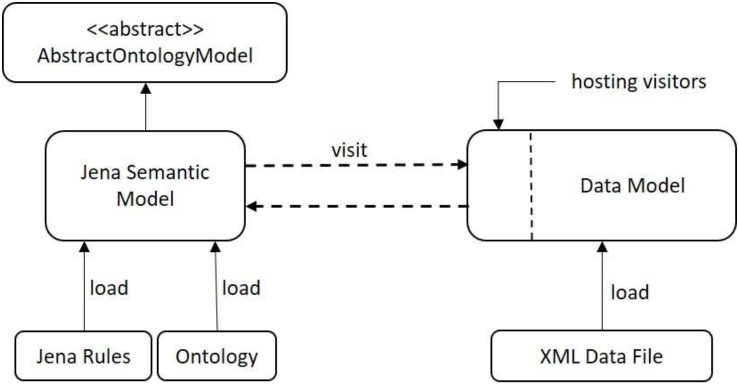
Implementation: Apache Jena, Jena Rules, OWL, RDF and XML.

Example 3. Family and School Ontologies



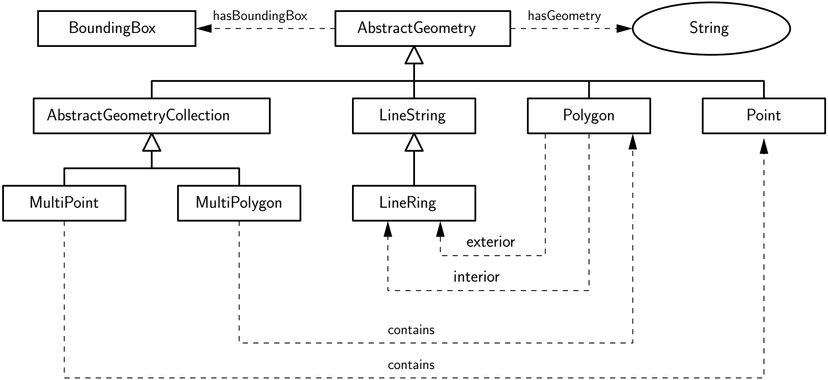
Source: Maria Coelho, MS Thesis, 2017.

Example 3. Populating Models with Data

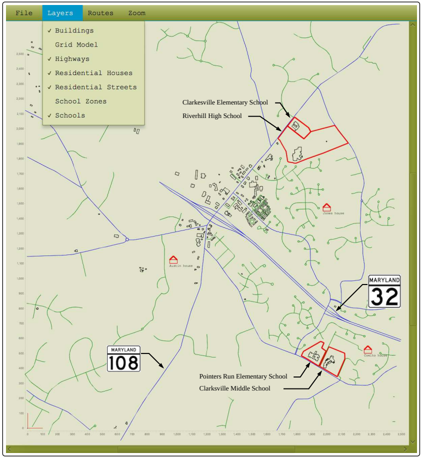


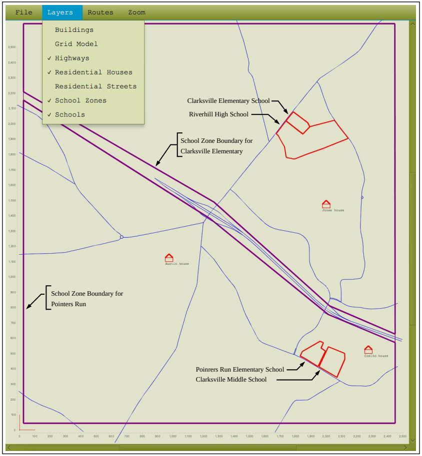
Example 4. Spatial Ontology

Abbreviated Spatial Ontology:



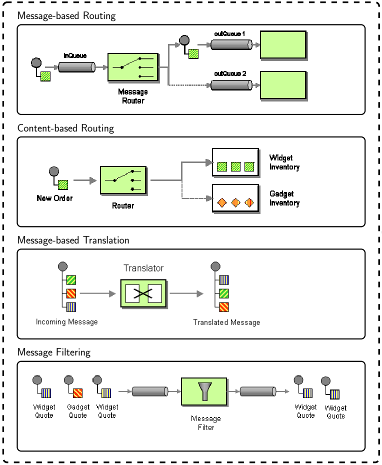
Example 4. Family-School-Urban-Geography Dynamics



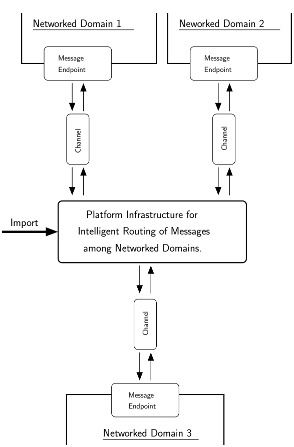


Future Work. Smart Messages with Apache Camel

Mechanisms for Message Transmisson and Processing in Apache Camel.



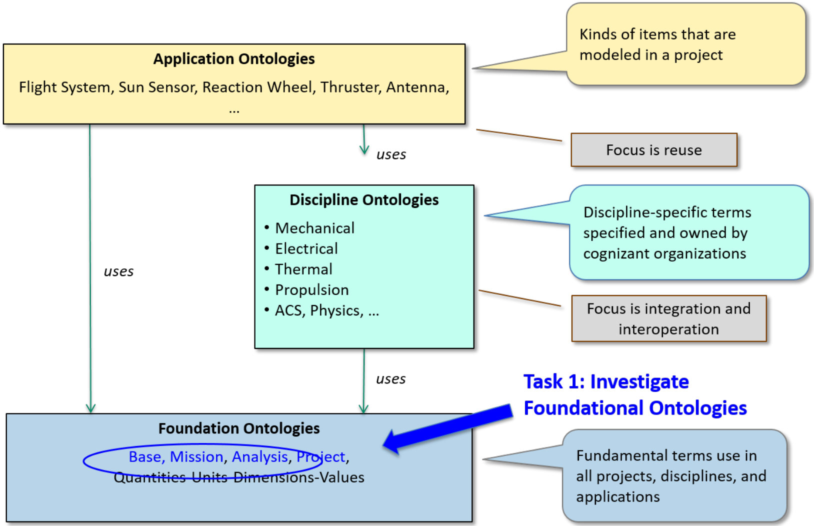
Distributed System Behavior Modeling



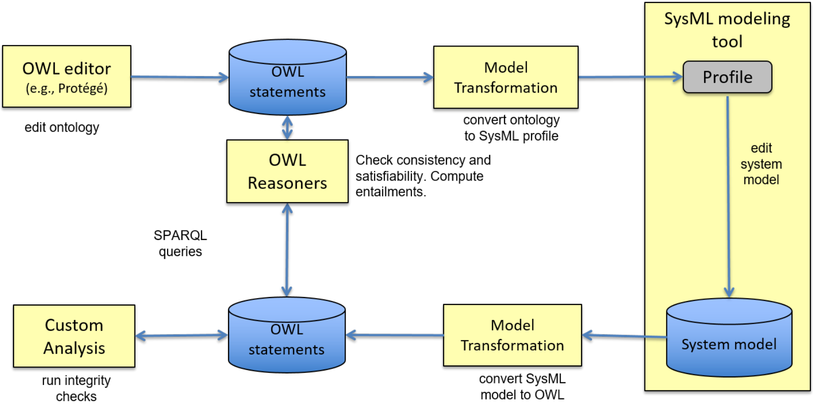
Ontology-Enabled Computing at JPL

Time frame: 2000-2006

Side-by-Side: Semantic/SysML Modeling at JPL



Side-by-Side: Semantic/SysML Modeling at JPL



Task 2: Investigate opportunities adding value to the MBSE process through integration of OWL ontologies and reasoning mechanisms with state-of-the-art SysML tools such as MagicDraw. How well does the proposed interaction of OWL and SysML actually work? What is actually be transformed in the model transformations? Is the model transformation process robust?

Analysis Procedure at UMD

1. Load the individual ontologies in Jena (e.g., base.owl, analysis.owl, mechanical.owl, etc, etc).
2. Systematically traverse the semantic graph.
3. For each class, print:
    1. Name of the class.
    2. The list of super classes.
    3. The list of subclasses classes.
    4. The list of data properties and object properties.
4. Record the number of classes and model size (i.e., number of statements in semantic graph).
5. Identify SWRL rules (if they exist).
6. Use VOWL to visualize the ontology (classes, data properties, object properties).

Note: At this point there are no individuals.

Analysis Procedure at UMD

Here's what a typical class looks like:

```
--- Full Name: http://imce.jpl.nasa.gov/foundation/analysis/analysis#Analysis

--- Superclass: http://imce.jpl.nasa.gov/foundation/analysis/analysis#Explanation ...

--- Subclass: http://imce.jpl.nasa.gov/foundation/analysis/analysis#TradeStudy ...
--- Subclass: http://imce.jpl.nasa.gov/foundation/analysis/analysis#KeyRequirementsExplanation ...
--- Subclass: http://imce.jpl.nasa.gov/foundation/analysis/analysis#DrivingRequirementsExplanation ...
--- Subclass: http://imce.jpl.nasa.gov/foundation/analysis/analysis#CostEstimate ...

--- Data Property Name: http://imce.jpl.nasa.gov/foundation/base/base#hasShortName ...
---             Domain: http://imce.jpl.nasa.gov/foundation/base/base#IdentifiedElement ...

... six data properties removed ...

--- Data Property Name: http://imce.jpl.nasa.gov/foundation/base/base#hasIndexEntry ...
---             Domain: http://imce.jpl.nasa.gov/foundation/base/base#IdentifiedElement ...

--- Object Property: http://imce.jpl.nasa.gov/foundation/analysis/analysis#isCharacterizedBy ...
---           Range: http://imce.jpl.nasa.gov/foundation/analysis/analysis#Characterization ...

... nine object properties removed ...

--- Object Property: http://imce.jpl.nasa.gov/foundation/analysis/analysis#isExplainedBy ...
---           Range: http://imce.jpl.nasa.gov/foundation/analysis/analysis#Explanation ...
```

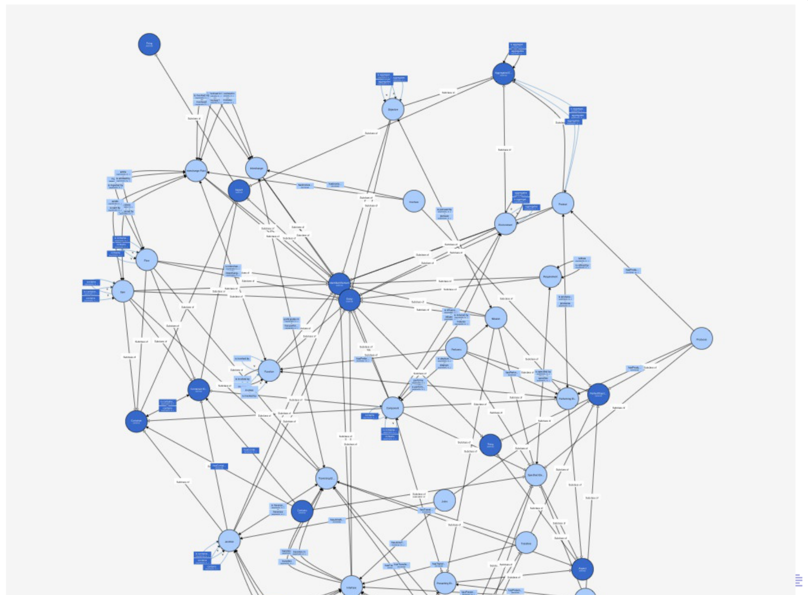
IMCE Ontologies (Number of Classes/Model Size)

Foundation Ontologies	Number of Classes	Model Size
Analysis.owl	101	2,769
Base.owl	13	–
Mission.owl	64	1,991
Project.owl	227	4,920
Time.owl	48	1,000

Discipline Ontologies	Number of Classes	Model Size
Mechanical.owl	105	–
Electrical.owl	243	5,074

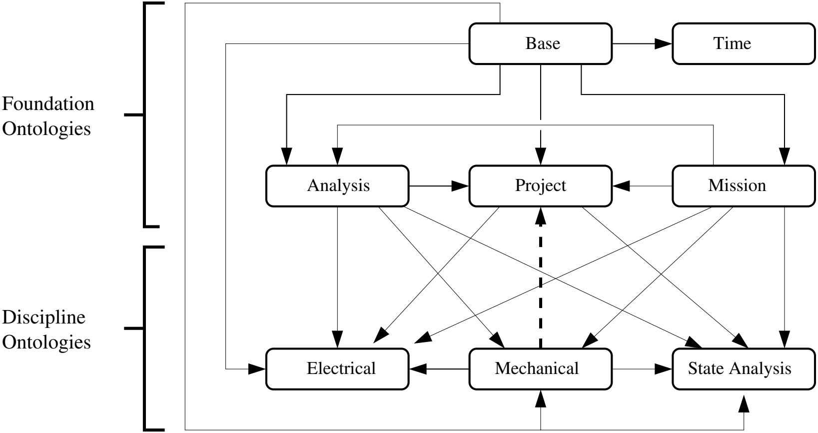
Miscellaneous Ontologies	Number of Classes	Model Size
SysML.owl	877	21,079

Panoramic View of Mission Ontology



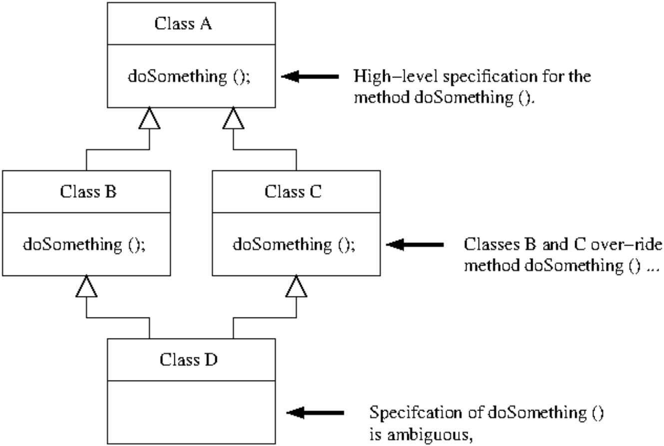
Concern 1: Dependencies Among Ontologies

What happened to notions of modularity?



Concern 2: Multiple Inheritance Relationships

Use of sultiple inheritance relationships in software:



Concern 2: Multiple Inheritance Relationships

Excessive use of multiple inheritance:

```
Named Class(79): Item

--- Full Name: http://imce.jpl.nasa.gov/foundation/mission/mission#Item

--- Superclass: http://imce.jpl.nasa.gov/backbone/imce.jpl.nasa.gov/foundation/mission/mission#Entity ...
--- Superclass: http://imce.jpl.nasa.gov/foundation/base/base#ContainedElement ...
--- Superclass: http://imce.jpl.nasa.gov/foundation/base/base#Container ...
--- Superclass: http://imce.jpl.nasa.gov/foundation/base/base#IdentifiedElement ...
--- Superclass: http://imce.jpl.nasa.gov/foundation/mission/mission#TraversingElement ...

--- Subclass: http://imce.jpl.nasa.gov/foundation/mission/mission#MaterialItem ...
--- Subclass: http://imce.jpl.nasa.gov/foundation/mission/mission#Message ...

--- Data Property Name: http://imce.jpl.nasa.gov/foundation/base/base#hasShortName ...
---             Domain: http://imce.jpl.nasa.gov/foundation/base/base#IdentifiedElement ...
--- Data Property Name: http://imce.jpl.nasa.gov/foundation/base/base#hasDescription ...

... etc ...
```

The Data-Ontology-Rule Footing

Building Block for Semantic Modeling and Event-driven Execution of Multi-Domain Systems

MSSE/Ph.D. (Civil Systems) Students

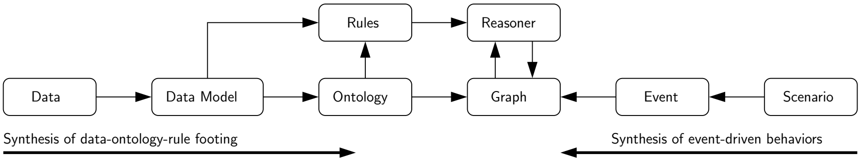
1. Parastoo Delgoshaei (2013-2017);
2. Maria Coelho (2015-present).

Data-Driven Approach

Guiding Principles:

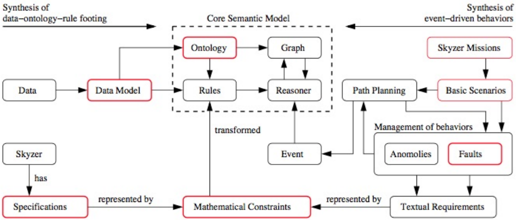
1. One footing for ontologies, rules and data ...
2. Use (but do not extend) foundational level ontologies ...
3. Ontologies visit data models to get individuals ...
4. Semantic graph dynamically responds to incoming events ...
5. Enhance power of rules with backend functions ...

Preliminary Schematic:

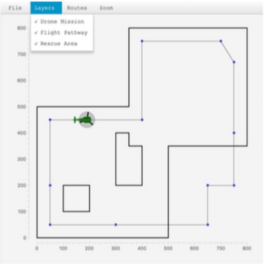


Data-Driven Approach (Synthesis of UAV Operations)

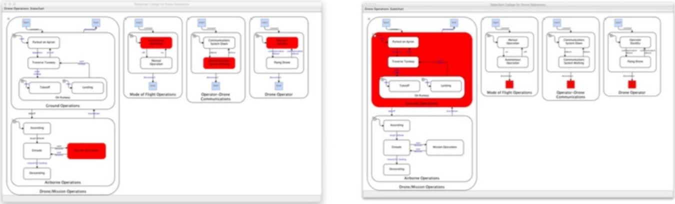
Synthesis of data-ontology-rule footing + event-driven behaviors.



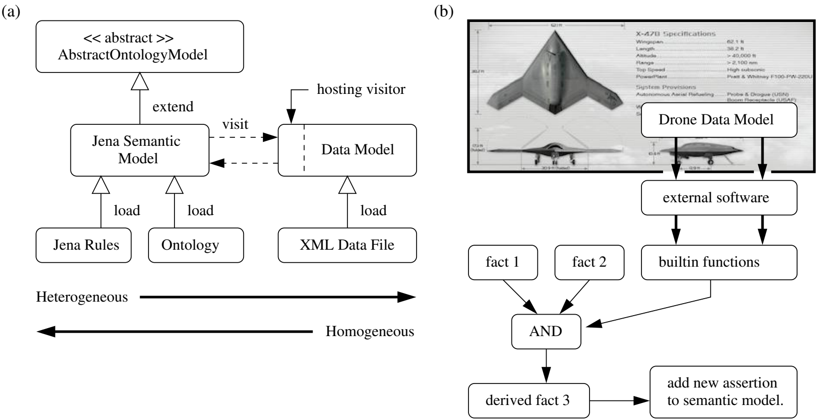
Simulation in Whistle ...



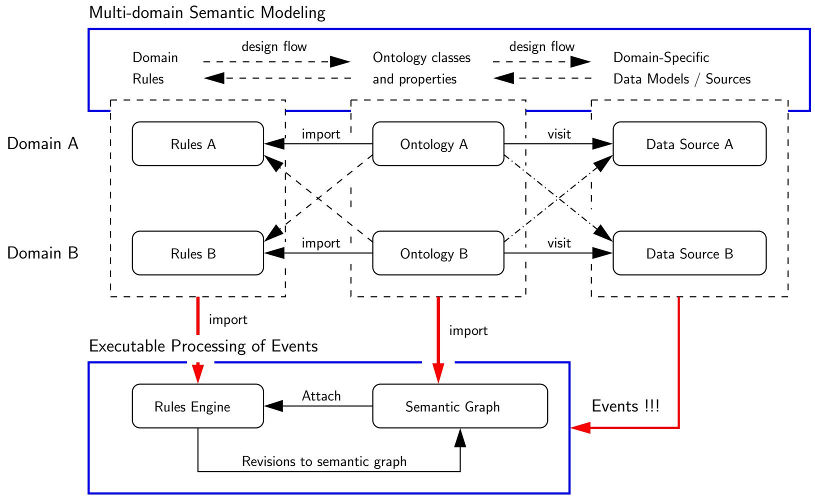
Visualization of subsystem behaviors ....



Data-Driven Approach (Populating Models with Data)



Template for Semantic Modeling + Processing of Events



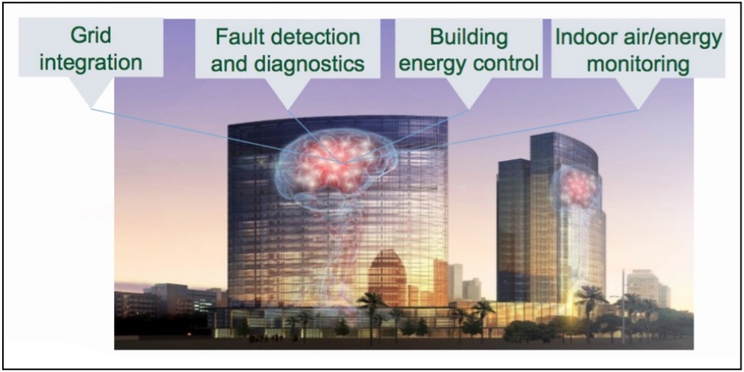
Case Study

Detection and Diagnostic Analysis of Faults in HVAC Equipment

Source: Delgoshaei and Austin, 2017.

Fault Detection in Buildings

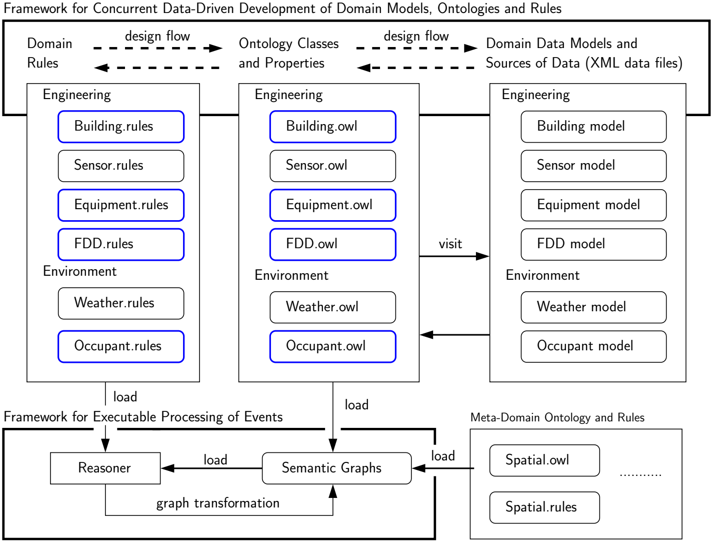
Example 1: Buildings that Think! (Work at NIST / UMD, 2017)



Research Question: How to use AI / Semantics to bring data, context and algorithms together for decision making?

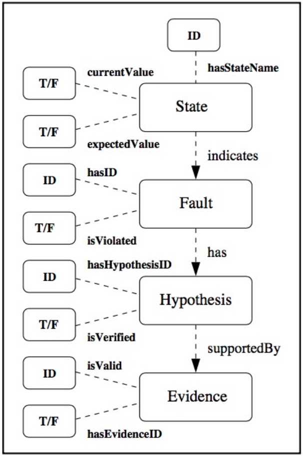
Legend: data = building geometry; context = occupant behavior; algorithms = reasoning.

Multi-Domain Building Semantics

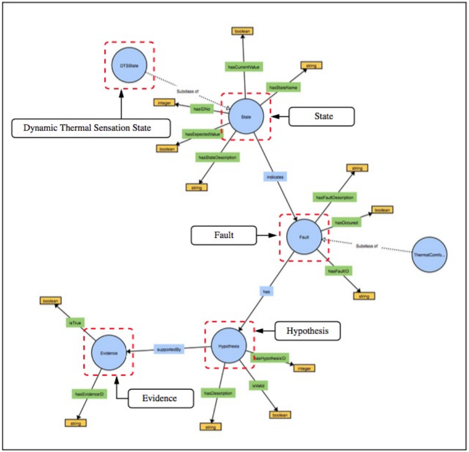


Multi-Domain Rule-based Reasoning

Flowchart for Processing of Faults

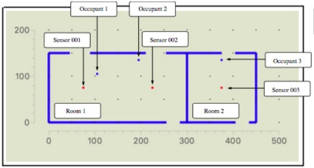


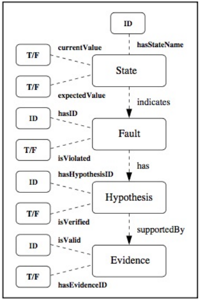
Fault Detection and Diagnostic Analysis Ontology



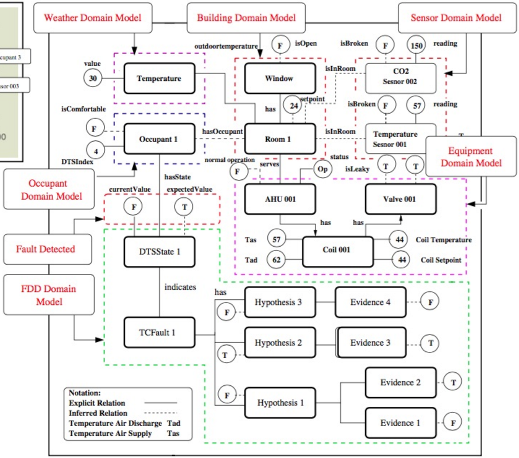
Multi-Domain Rule-based Reasoning

Case Study Problem



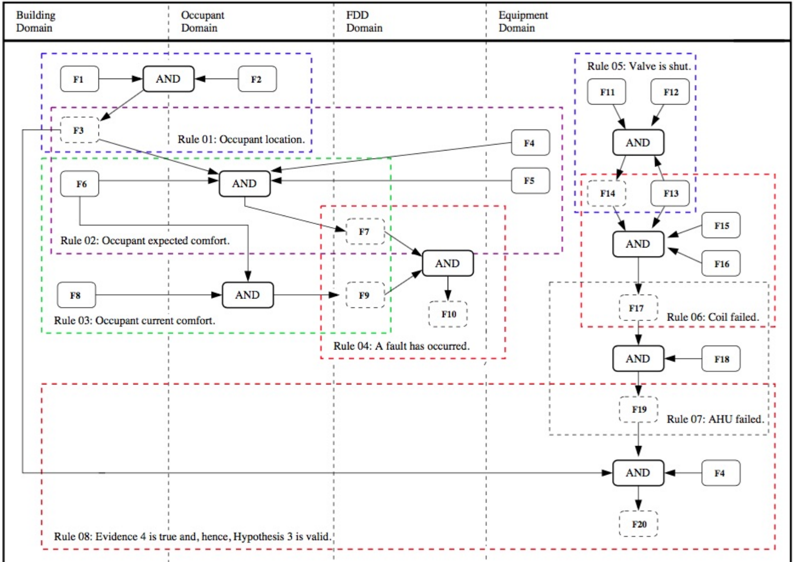


Snapshot of Fully Assembled Semantic Graph



Multi-Domain Rule-based Reasoning

Snapshot of Multi-Domain Evaluation and Forward Chaining of Rules



References

- Abraham J., Semantic Foundations for Formalizing Brain Cancer Profiles, MS Thesis in Systems Engineering, University of Maryland, April 2019.
- Austin M.A., Delgoshaei P. and Nguyen A., Distributed Systems Behavior Modeling with Ontologies, Rules, and Message Passing Mechanisms, Procedia Computer Science, vol. 44, pp. 373–382, 2015.
- Austin M.A., Delgoshaei P., Coelho M. and Heidarinejad M., Architecting Smart City Digital Twins: Combined Semantic Model and Machine Learning Approach, Journal of Management in Engineering, ASCE, Volume 36, Issue 4, July, 2020.
- Coelho M., Distributed Behavior Modeling of Urban Systems with Ontologies, Rules and Message Passing Mechanisms, *M.S. Thesis* (Available on UMD DRUM), M.S. in Civil Systems, April 2017,
- Coelho M., Austin M.A., and Blackburn M.R., Semantic Behavior Modeling and Event-Driven Reasoning for Urban System of Systems, International Journal on Advances in Intelligent Systems, Vol. 10, No 3 and 4, December 2017, pp. 365-382.
- Delgoshaei P. and Austin M.A., Framework for Knowledge-Based Fault Detection and Diagnostics in Multi-Domain Systems: Application to Heating Ventilation and Air Conditioning Systems, International Journal on Advances in Intelligent Systems, Vol. 10, No 3 and 4, December 2017, pp. 393-409.
- Delgoshaei P., Heidarinejad M., and Austin M.A. , Combined Ontology-Driven and Machine Learning Approach to Management of Building Energy Consumption, 2018 Building Performance Analysis Conference and SimBuild, Chicago, September 26-28, 2018.
- Gao J., Liu X., Li D., and Havlin S., Recent Progress on the Resilience of Complex Networks, Energies, Vol. 8, 2015, pp. 12187–12210.
- Wagner et al., An Ontology for State Analysis: Formalizing the Mapping to SysML, Proceedings of 2012 IEEE Aerospace Conference, Big Sky, Montana, March, 2012.