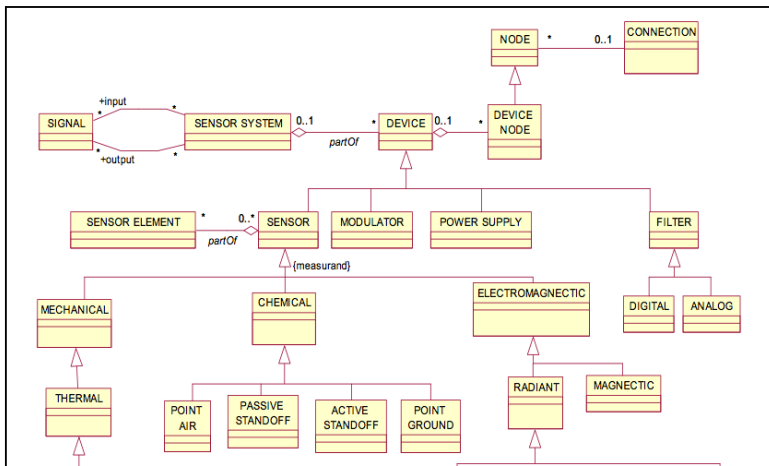


High-Level Sensor Ontology



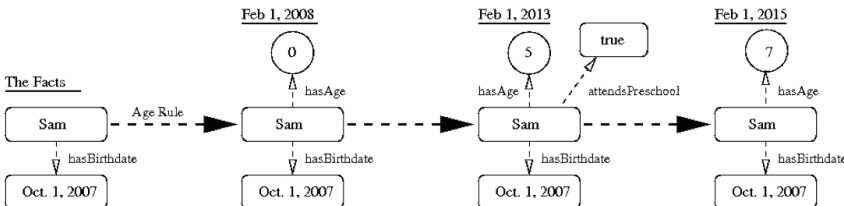
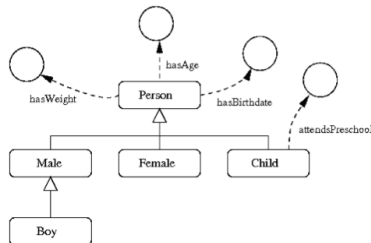
Example 1. A Simple Family Model

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

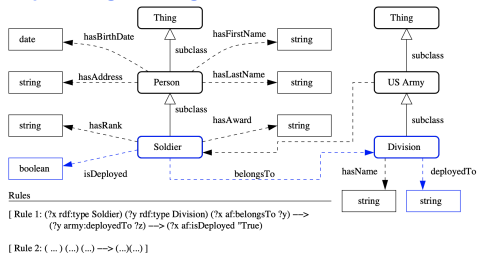
Statement[4] Subject : http://austin.org/family#Sam
              Predicate: http://austin.org/family#hasWeight
              Object   : "35.0~http://www.w3.org/2001/... #double"

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



Step 2: Add Data (1944)

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

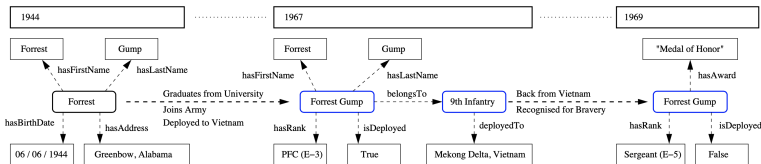
Military Deployment Data (1967)

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:

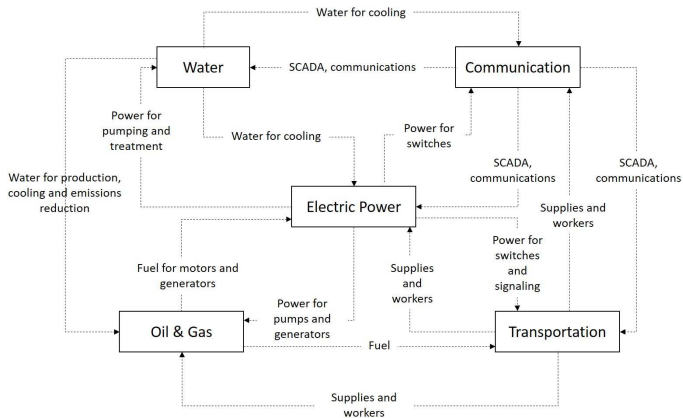
- Cities are **system of systems**.
- City subsystems may have a preference to operating as independently as possible from the other subsystems.
- Strategic **collaboration** among subsystems is often **needed** to either **avoid cascading failures** across systems and/or **recover from a loss of functionality**.

Systems-of-systems need not be complicated:



Motivation

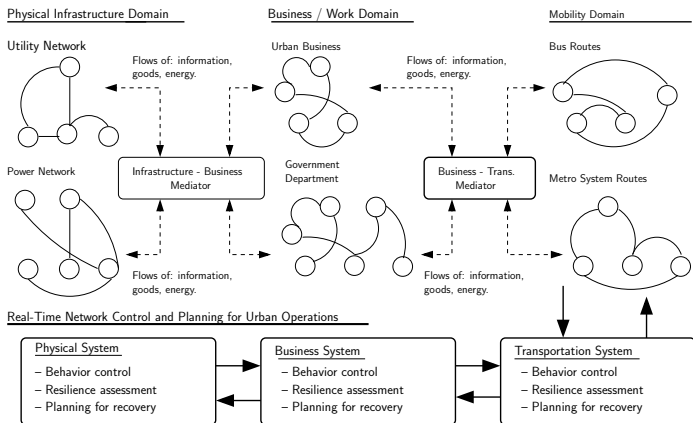
Dependency Relationships Among Infrastructures



Source: Gao et al., 2015.

Motivation

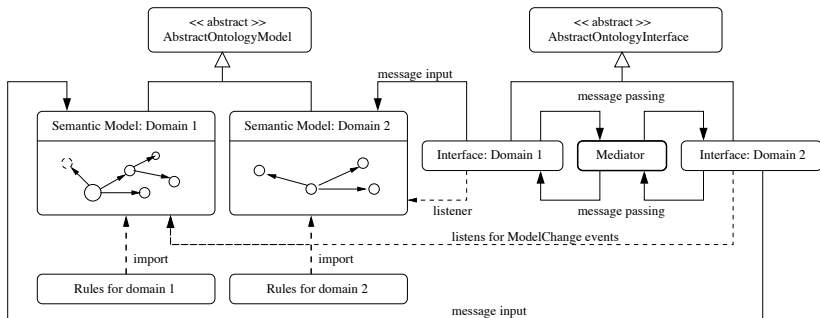
Architecture for Multi-domain Behavior Modeling



Source: Coelho, Austin, and Blackburn, 2017.

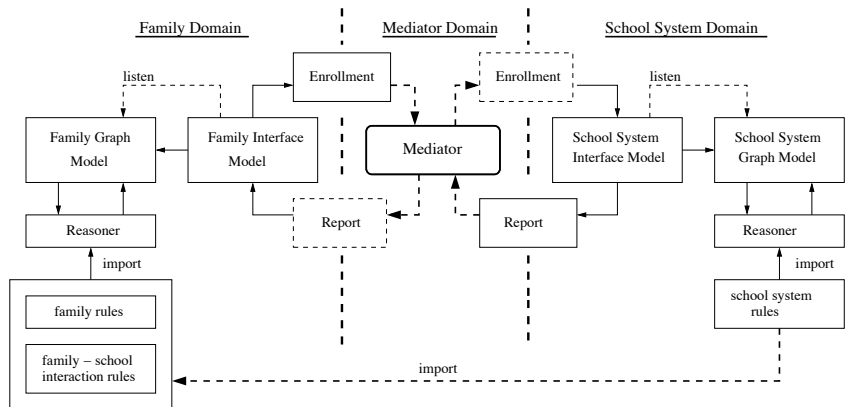
Distributed Behavior Modeling (Initial)

Initial Idea (2014) Use **semantic graphs** to model **behavior** of **individual entities** (e.g., a family).



Wrap entities with **interfaces**. Enable **communication** among entities with **message passing**.

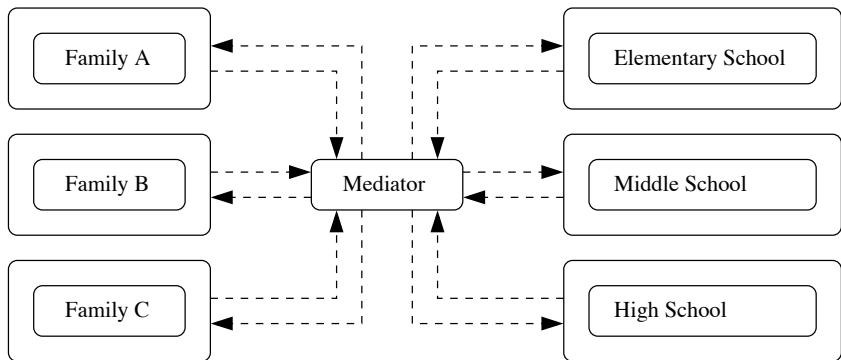
Example 3. Family-School System Dynamics



Example 3. Framework for Communication

Family Domain

School Domain



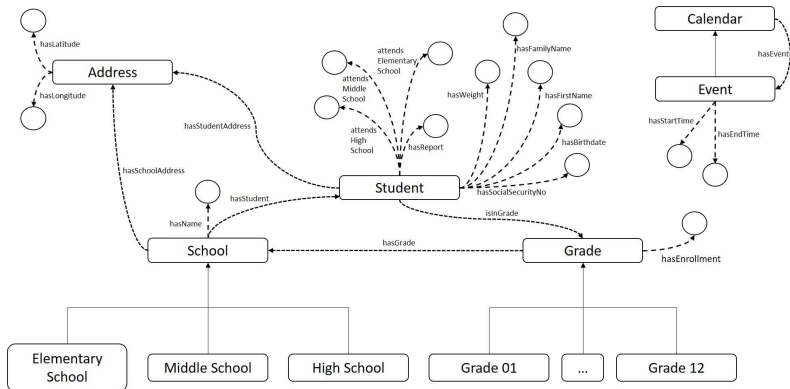
Example 3. Family Datafile (XML)

```
<?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)

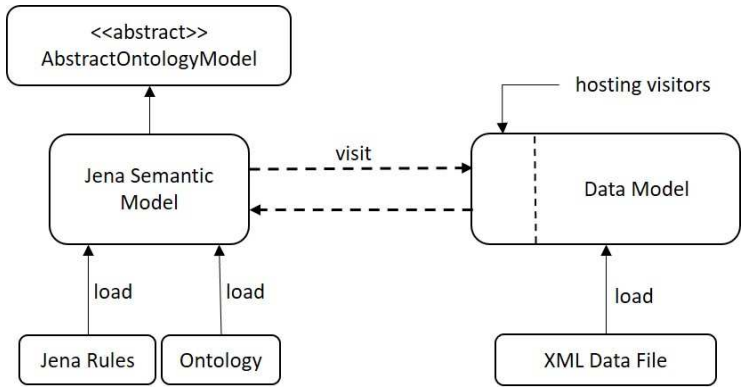
```
<?xml version="1.0" encoding="UTF-8"?>
<SchoolSystemModel author="Maria Coelho" date="2017" source="UMD">
  <School>
    <attribute text="Type" value="High School"/>
    <attribute text="Name" value="River Hill High School"/>
    <attribute text="Grade" value="Grade09"/>
    <attribute text="Grade" 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

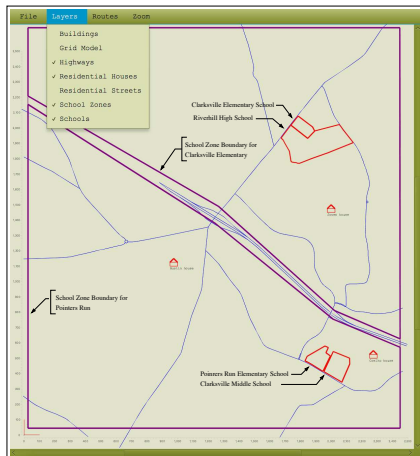
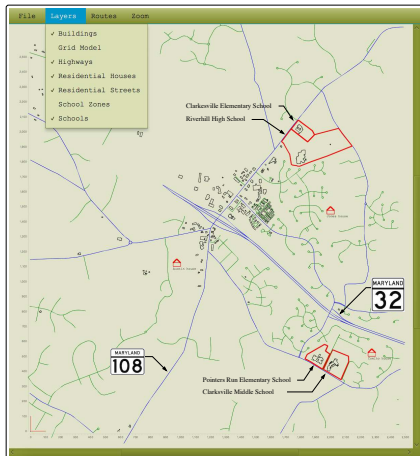


Source: Maria Coelho, MS Thesis, 2017.

Example 3. Populating Models with Data



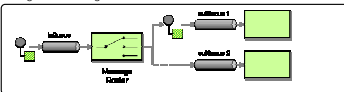
Example 4. Family-School-Urban-Geography Dynamics



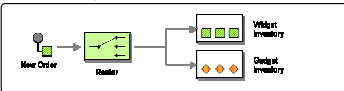
Future Work. Smart Messages with Apache Camel

Mechanisms for Message Transmission and Processing in Apache Camel.

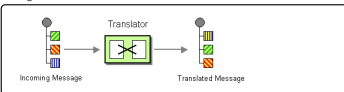
Message-based Routing



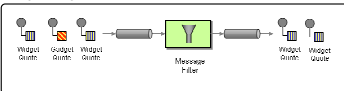
Content-based Routing



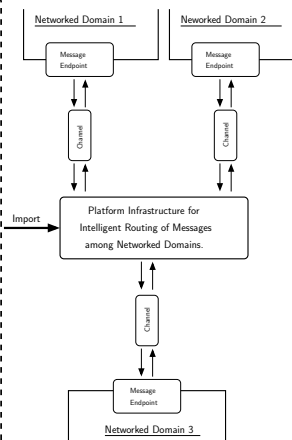
Message-based Translation



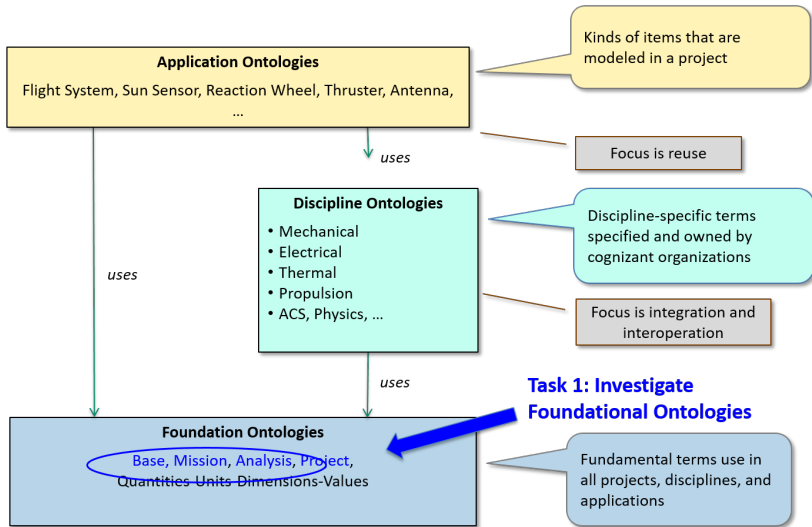
Message Filtering



Distributed System Behavior Modeling

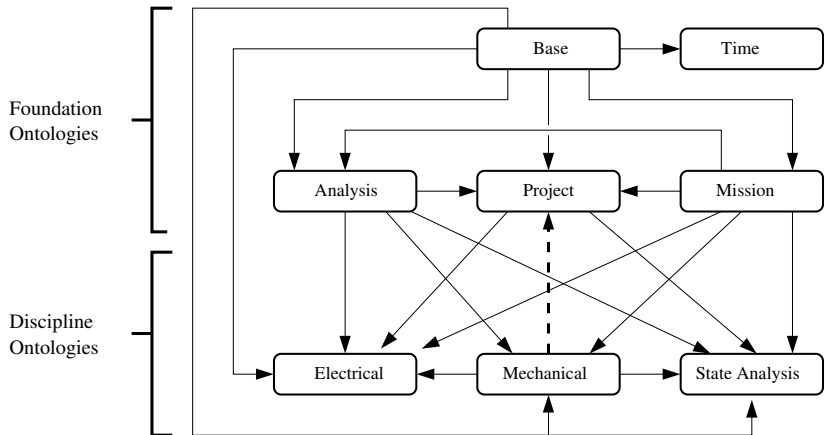


Side-by-Side: Semantic/SysML Modeling at JPL



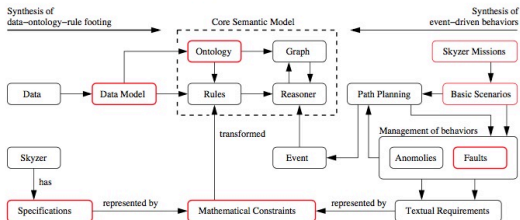
Concern 1: Dependencies Among Ontologies

What happened to notions of modularity?

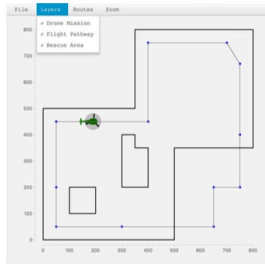


Data-Driven Approach (Synthesis of UAV Operations)

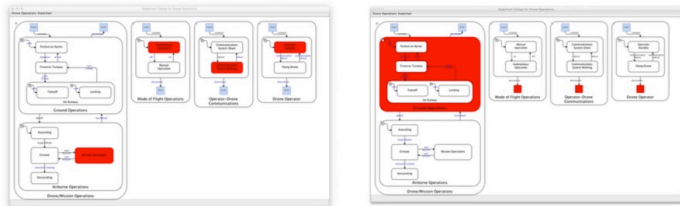
Synthesis of data-ontology-rule footing + event-driven behaviors.



Simulation in Whistle ...



Visualization of subsystem behaviors ...



Template for Semantic Modeling + Processing of Events

