SNAPSHOTS OF SYSTEMS ENGINEERING RESEARCH AT UMCP

Mark Austin, Department of Civil and Environmental Engineering and ISR, University of Maryland, College Park, MD 20742, USA.

Model-Based Systems Engineering Workshop
Goddard Space Flight Center
Greenbelt, Maryland, February 18, 2016
OVERVIEW

Model-Based Systems Engineering at Maryland
• Model-Based Systems Engineering (MBSE) at ISR

Snapshots of Research
• Working with GPM: PaladinRM Software Tool.
• Working with Semantic Web Technologies.
• Integration of NLP with Ontologies and Textual Requirements

Acknowledgements / Co-Workers
• At NASA: David Everett, Jessica Knizhnik, Craig Carignan.
Definition and Scope

- Formalizes the development of systems through the use of models.
- Broad in scope, across multiple stages of system development and multiple physics.

Benefits of MBSE

- Allows for the development of virtual prototypes.
- Facilitates communication among disciplines in team-based development.
- Enables semi-formal and formal approaches to system assessment.
- Management of system complexity.
Tenet 1: Create Big-Picture View and Emphasize Model-Based Systems Engineering. The mathematics needed for formal approaches to systems engineering is foreign to many engineers.
Tenet 2: Emphasize Disciplined Approaches to Design. Techniques include decomposition, abstraction, and formal analysis.
Tenet 3: To keep the complexity of design activities in check, we need to employ mixtures of semi-formal and formal approaches to system development.
MODEL-BASED SYSTEMS ENGINEERING

Motivating Application Area 1: Buildings!

Pearl River Tower Complex

Green Technology Tower — Architectural Proposal for Chicago
Motivating Application Area 2: Platforms for Biomedical Experimental Research

Source: Mosteller et al., 2012
PART 1

WORKING WITH GPM (2002-2003)
WORKING WITH GPM

Requirements are organized into layers for team development.

Compaction of the tree representation into a graph.
WORKING WITH GPM
WORKING WITH GPM

Paladin RM Graphical User Interface.

Print hardcopy … Hardcopy of Requirements …..
PART 2

WORKING WITH SEMANTIC WEB TECHNOLOGIES
WORKING WITH SEMANTIC WEB TECHNOLOGIES

Layers of Abstraction

- Applications
  - Ontology and reasoning layers
    - Data layers
      - Representation / syntax layers

Semantic Web Technology Stack

- Applications and Interfaces
  - Trust
    - Proof
      - Unifying Logic
        - Ontology: OWL
        - Rules: RIF
          - RDFS
            - Data Interchange: RDF
              - XML
                - Unicode
                  - URI

©2010 University of Maryland
MBSE Capability 2020 -- 2025

- Reduced cycle times
- System of systems interoperability
- Design optimization across broad trade space
- Cross domain effects based analysis

DRAFT
June 24, 2007

- Distributed & secure model repositories crossing multiple domains
- Defined MBSE theory, ontology, and formalisms
- Architecture model integrated with Simulation, Analysis, and Visualization
- Matured MBSE methods and metrics, Integrated System/HW/SW models
- Emerging MBSE standards

- Planning & Support
- Research
- Standards Development
- Processes, Practices, & Methods
- Tools & Technology Enhancements
- Outreach, Training & Education

Institutionalized MBSE across Academia/Industry
Well Defined MBSE
Ad Hoc MBSE Document Centric

2010
2020
2025
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.
WORKING WITH SEMANTIC WEB TECHNOLOGIES

Source: Scott Selberg, MSSE Graduate Student
Design and Trade-Off Analysis with RDF Graphs

Source: Nefretiti Nassar, MSSE Graduate Student.
New idea (2005): Ontology-enabled Traceability Mechanisms

State-of-the-Art Traceability

Proposed Model for Traceability

Approach: Requirements are satisfied through implementation of design concepts. Now traceability pathways are threaded through design concepts.

Key Benefit: Rule checking can be attached to “design concepts” (ontology), therefore, we have a pathway for early validation.
Requirement level (textual representation)
The metro system will start working at 5 am.

Rule level (SWRL)
scheduler(?s)^ hasTime(?s,?t)^ swrlb:greaterThan(?t,5)^ train(?tr)^
^ isAvailable(?tr,true)=›sendTrain(?s,?tr)

Guard Statement
The transition from idle to active is conditional on “[t == 5 am.]” evaluation results.

Expected Behavior
• The scheduler statechart will transition from idle to active at 5:00 am.
• The statechart of at least one train will transition to the “At Station” state.

System structures are modeled as networks and composite hierarchies of components.

Behaviors will be associated with components.

Discrete behavior will be modeled with finite state machines.

Continuous behavior will be represented by partial differential equations.
Requirement: Cooling coil will be locked out for winter operation (55 F)

Rule: (?cc RDF:type Cooling) (?cc ont:isLocked? ?l) (?out_temp ont:hasValue ?v) lessThan(?v,55) ->(?l, true)
WORKING WITH SEMANTIC WEB TECHNOLOGIES

Behavior Modeling and Control
- Actuators:
  - switch
  - damper
  - valve
- Indoor Sensors:
  - thermal
  - humidity
  - luminance
  - occupancy
  - flow
- Components:
  - pipe
  - fan
  - pump
  - b/c coil
- Control Algorithms:
  - Open heating coil valve 30%
  - Closed heating coil valve

Input
- Push data
- Push data
- Push data

Output
- Ontology + Rules + Instances
- Synthesis

Building Floorplan / Architecture
- Automated synthesis of building simulations.
  -- Time history simulation.
  -- Performance assessment.

©2010 University of Maryland
PART 3

INTEGRATION OF NLP WITH ONTOLOGIES AND TEXTUAL REQUIREMENTS
State-of-the-Art Practice: Transforming Textual Requirements into Models

Problem size: 100’s – 1000’s of pages.
Difficulties: slow, error prone, no support for validation.

Our Idea: Integration of NLP with Ontologies and Textual Requirements
Working with the Natural Language Processing Toolkit (NLTK)

Working with the Natural Language Processing Toolkit (NLTK)

Tokenization: Identify and provide access to individual words in the text.

```
text = "These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule."
tokens = nltk.word_tokenize(text)
print tokens
=>
```
Part of Speech Tagging: Identify the role each word plays in the sentence.

```python
my_string = "When I work as a senior systems engineer, I truly enjoy my work."
tokens = nltk.word_tokenize(my_string)
print tokens

tagged_tokens = nltk.pos_tag(tokens)
print tagged_tokens
=>
[('When', 'WRB'), ('I', 'PRP'), ('work', 'VBP'),
 ('as', 'RB'), ('a', 'DT'), ('senior', 'JJ'),
 ('systems', 'NNS'), ('engineer', 'NN'), (',', ',', ','),
 ('I', 'PRP'), ('truly', 'RB'), ('enjoy', 'VB'),
 ('my', 'PRP$'), ('work', 'NN'), (',', ',', '.')]  
```

Legend: WRB = Wh-verb (e.g., How, where, why), PRP = Personal pronoun (e.g., I); RB = Adverb; JJ = Adjective, VBP = Present verb tense, etc.
Working with the Natural Language Processing Toolkit (NLTK)

**Chunking:** These are patterns of part-of-speech tags that define what kinds of words make up a chunk.

**Chinking:** Patterns for what kinds of words should be excluded from a chunk.
Simple Aircraft Application

**Ontology Model**

+Model: Transportation
+Entity: Aircraft
+Engines
+Wings
+Slides
+Throttle Levels
+Altitude Indicator
Length: 254 meters
Passengers Capacity

**Model**

Name: Transportation
Track
Station
Rail Line
Train
Route
Aircraft

**Entity**

Name: Aircraft
Properties:
- engines
- wings
- slides
- throttle levers
- altitude indicator
- length
- passengers capacity
## System Requirements

<table>
<thead>
<tr>
<th>Id</th>
<th>Title</th>
<th>Description</th>
<th>System</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A plane needs wings</td>
<td>A wing is a type of fin with a surface that produces aerodynamic force for flight or propulsion through the atmosphere</td>
<td>1</td>
<td>Edit</td>
</tr>
<tr>
<td>3</td>
<td>The plane needs throttle levers</td>
<td>Each thrust lever displays the engine number of the engine it controls</td>
<td>1</td>
<td>Edit</td>
</tr>
<tr>
<td>4</td>
<td>The length of the plane</td>
<td>The length of the entire aircraft should be 254 meters</td>
<td>1</td>
<td>Edit</td>
</tr>
<tr>
<td>5</td>
<td>The plane should have engines</td>
<td>An aircraft engine is the component of the propulsion system for an aircraft that generates mechanical power</td>
<td>1</td>
<td>Edit</td>
</tr>
<tr>
<td>6</td>
<td>The capacity is 255 passengers</td>
<td>The aircraft needs to have a passengers capacity of 255</td>
<td>1</td>
<td>Edit</td>
</tr>
</tbody>
</table>
Simple Aircraft Application: Analysis of Requirements

### Basic Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chars</td>
<td>547</td>
</tr>
<tr>
<td>Len tokens</td>
<td>94</td>
</tr>
<tr>
<td>Sentences</td>
<td>1</td>
</tr>
<tr>
<td>Porter stems</td>
<td>94</td>
</tr>
<tr>
<td>Lancaster stems</td>
<td>94</td>
</tr>
<tr>
<td>Wnl stems</td>
<td>94</td>
</tr>
</tbody>
</table>

### Objects

<table>
<thead>
<tr>
<th>NN</th>
<th>aircraft</th>
<th>plane</th>
<th>engine</th>
<th>capacity</th>
<th>length</th>
<th>propulsion</th>
<th>atmosphere</th>
<th>component</th>
<th>fin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>flight</td>
<td>force</td>
<td>lever</td>
<td>number</td>
<td>power</td>
<td>surface</td>
<td>system</td>
<td>throttle</td>
<td>thrust</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>type</td>
</tr>
<tr>
<td>NNS</td>
<td>passengers</td>
<td>displays</td>
<td>engines</td>
<td>generates</td>
<td>levers</td>
<td>meters</td>
<td>wings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>aircraft engine</td>
<td>engine number</td>
<td>generates mechanical power</td>
<td>passengers capacity</td>
<td>propulsion system</td>
<td>throttle levers</td>
<td>thrust lever</td>
<td>displays</td>
<td></td>
</tr>
</tbody>
</table>
## System Validation

<table>
<thead>
<tr>
<th>Verified properties</th>
<th>engines</th>
<th>wings</th>
<th>throttle levers</th>
<th>length</th>
<th>passengers capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unverified properties</td>
<td>slides</td>
<td>altitude indicator</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


• Arellano A., Carney E., and Austin M.A., Natural Language Processing of Textual Requirements, ICONS 2015, Barcelona, Spain, April, 2015. Best Paper Award.
Thank You

E-mail: austin@isr.umd.edu
Web site: http://www.isr.umd.edu/~austin