Towards Integrated Engineering Models: Clarifying the Role of OSLC

Bill Beavin, Technical Fellow
The Boeing Company
OMG Technical Meeting; La Jolla, CA; 7-11 December 2015
OSLC Summit, 9 December 2015
Biography

Bill Beavin, a Software Engineer and Boeing Technical Fellow, has supported the development and use of models and simulations to address real-world issues since 1985. Within those 30+ years he has directly supported many programs in the air, space, sea, undersea, manned, unmanned, and networked systems domains, as well as numerous live, virtual, and constructive simulation technology development efforts. As a member of the Boeing Research and Technology Analytics and Decision Support team, he is currently focused on the application of semantic related technologies towards engineering models and simulations.
Abstract

Many engineers use tools to interact with models that serve as the basis for a product. In the case of a highly complex product this pattern may be instantiated in thousands of engineers, and hundreds of tools and models, for a single product. This increasing product complexity drives the need for engineers to collaborate more effectively, tools to interoperate more seamlessly, and models to integrate intrinsically. OSLC has emerged as a promising technology to help address this need, but its role in doing so is currently unclear. This presentation will describe an attempt to clarify that role.
Approach

- Clarify a key problem a collaboration framework needs to solve
- Define some patterns the collaborative framework needs to support
- Identify and assess candidate collaboration framework solution approaches
- Make some observations
Clarify the Problem: Product Development Pattern

Engineer Uses Tool* Interacts Model Basis Product

*Or tool suite

Collaboration “Co-Labor” Threads

Engineer Instances

Tool Instances

Model Instances

Product Instance

E₁ Uses T₁ Interacts M₁ Basis P

E₂ Uses T₂ Interacts M₂ Basis

E₃ Uses T₃ Interacts M₃ Basis
Traditional Collaboration Approaches

- **Engineer Instances**
  - $E_1$
  - $E_2$
  - $E_3$

- **Tool Instances**
  - $T_1$
  - $T_2$
  - $T_3$

- **Model Instances**
  - $M_1$
  - $M_2$
  - $M_3$

- **Product Instance**
  - $P$

**Std Processes/Train**
- Brittle
- Recurring with change
- Limited Scale
- Limited Scope

**Integrate the Tools**
- Brittle
- Recurring with change
- Limited Scale
- Limited Scope

**Integrate the Models**
- Brittle
- Recurring with change
- Limited Scale
- Limited Scope
Desired Effect Approach: First Time Quality

Engineer Instances

Tool Instances

Model Instances

Product Instance

- $E_1$ Uses $T_1$ Interacts
- $E_2$ Uses $T_2$ Interacts
- $E_3$ Uses $T_3$ Interacts

$M_1$, $M_2$, $M_3$ are models with $MTOW_{M1}$, $MTOW_{M2}$, $MTOW_{M3}$ respectively.

Product Instance

$P$ with $MTOW_p$.

Product Wide Consistency and Correctness

1st Time Quality Needs

Backchain
Desired Effect Approach: First Time Quality

Engineer Instances

\[ E_1 \]

Interacts

\[ T_1 \]

Uses

\[ M_{1} \]

Basis

MTOW_{M1}

Product Instance

\[ P \]

MTOW_{P}

Product Wide Model Consistency with Tool Specific Correctness

Product Wide Model Consistency and Correctness

Needs

Backchain

Product Wide Model Consistency with Expertise Specific Correctness

Uses

\[ E_2 \]

Interacts

\[ T_2 \]

Uses

\[ M_{2} \]

Basis

MTOW_{M2}

Uses

\[ E_3 \]

Interacts

\[ T_3 \]

Uses

\[ M_{3} \]

Basis

MTOW_{M3}

Needs

1st Time Quality

MTOW
A Pattern for Correctness: Consistency

Goal: Product Wide Model Consistency

Model Instances:

- Model\(_1\) \(MTOW_{M1}\)
- Model\(_2\) \(MTOW_{M2}\)
- Model\(_3\) \(MTOW_{M3}\)

Consistency: A realization exists that is conformant to each model.
A Pattern for Correctness: Consistency

Goal: Product Wide Model Consistency

Model Instances

- Model$_1$ with MTOW$_{M1}$
- Model$_2$ with MTOW$_{M2}$
- Model$_3$ with MTOW$_{M3}$

Is consistent with

Realization: Something planned turned into something real

Product Instance

- Product with MTOW$_p$
  
  May be a simulated product

Consistency: A realization exists that is conformant to each model
A Pattern for Correctness: Consistency

**Goal:** Product Wide Model Consistency

**Model Instances**

- Model$_1^{\text{MTOW}}_{M1}$
- Model$_2^{\text{MTOW}}_{M2}$
- Model$_3^{\text{MTOW}}_{M3}$

Each model instance is consistent with the others:

- Model$_1^{\text{MTOW}}_{M1}$ is consistent with Model$_2^{\text{MTOW}}_{M2}$
- Model$_2^{\text{MTOW}}_{M2}$ is consistent with Model$_3^{\text{MTOW}}_{M3}$
- Model$_3^{\text{MTOW}}_{M3}$ is consistent with Model$_1^{\text{MTOW}}_{M1}$

**Realization:** Something planned turned into something real

**Product Instance**

- **MTOW$_p$**
  - May be a simulated product
  - Realizes least requirements
  - Consistency: A realization exists that is conformant to each model
A Pattern for Correctness: Consistency

**Goal**

Product Wide Model Consistency

**Model Instances**

1. Model₁ \( \text{MTOW}_{M1} \)  
   - Is consistent with \( \text{MTOW}_{M2} \)
2. Model₂ \( \text{MTOW}_{M2} \)  
   - Is consistent with \( \text{MTOW}_{M3} \)
3. Model₃ \( \text{MTOW}_{M3} \)  
   - Is consistent with \( \text{MTOW}_{M1} \)

**Product Instance**

Product \( \text{MTOW}_p \)  
- May be a simulated product
- Realizes
  - Is conformant to \( M_1 \)
  - Is conformant to \( M_2 \)
  - Is conformant to \( M_3 \)

**Consistency:** A realization exists that is conformant to each model
Conformance: A realization exists in a state that fulfills a required state defined in a model.

State: The condition of something with respect to circumstances or attributes.
A Pattern for Correctness: Conformance

Conformance: A realization exists in a state that fulfills a required state defined in a model.

Model Instance

Model 1
MTOW_{M1}

Model 2
MTOW_{M2}

Model 3
MTOW_{M3}

Model State

Model 1
Value_{M1}

Model 2
Value_{M2}

Model 3
Value_{M3}

Product Instance

Product
Value_{p}

Product State

MTOW_{p}
Conformance: A realization exists in a state that fulfills a required state defined in a model.

Goal: Product Wide Model Conformance
A Pattern for Correctness: Conformance

Conformance: A realization exists in a state that fulfills a required state defined in a model.

Fulfills Modeled State

Reification: Making something real

Goal: Product Wide Model Conformance
Patterns for Correctness: Conformance

Goal: Product Wide Model Correctness

1) Realization is Conformant
   A realization exists in a state that fulfills a required state

Model Instance

Product Instance

Realization Conformance Pattern

Product MTOW_p

State_p

Value_p

M_1 MTOW_M1

State_M1

Value_M1
Patterns for Correctness: Consistency

Goal: Product Wide Model Correctness

1) Realization is Conformant
   A realization exists in a state that fulfills a required state

2) Models are Consistent
   A realization exists that is conformant to each model

Model Consistency Pattern

- Realization instance
  - $M_1$: MTOW
  - $M_2$: MTOW
  - $M_3$: MTOW

- State instances
  - $State_{M1}$
  - $State_{M2}$
  - $State_{M3}$

- Value instances
  - $Value_{M1}$
  - $Value_{M2}$
  - $Value_{M3}$

- Fulfillment relationships
  - $M_1$ fulfills $State_{M1}$
  - $M_2$ fulfills $State_{M2}$
  - $M_3$ fulfills $State_{M3}$

- Realization relationship
  - Product MTOW instance realizes
    - $M_1$ with $State_{M1}$
    - $M_2$ with $State_{M2}$
    - $M_3$ with $State_{M3}$
Patterns for Correctness: Completeness

**Goal**
Product Wide Model Correctness

1) Realization is Conformant
A realization exists in a state that fulfills a required state

2) Models are Consistent
A realization exists that is conformant to each model

3) Models are Complete
All model parameters have values realizable in a product

Model Completeness Pattern

- $M_1$, MTOW$_{M1}$ realizes State$_{M1}$, Value$_{M1}$
- $M_2$, MTOW$_{M2}$ realizes State$_{M2}$, Value$_{M2}$
- $M_3$, MTOW$_{M3}$ realizes State$_{M3}$, Value$_{M3}$

Product, MTOW$_P$, realized

State$_P$, Value$_P$
Patterns for Correctness: Suitability

**Goal**

Product Wide Model Correctness

1) **Realization is Conformant**
   
   A realization exists in a state that **fulfills a required state**

2) **Models are Consistent**
   
   A realization exists that is conformant to each model

3) **Models are Complete**
   
   All model parameters have values realizable in a product

4) **Realization is Suitable**
   
   Models produce a realization that **fulfills all requirements**

**Realization Suitability Pattern**

- **Product MTOW**
  - **State**
  - **Value**

- **Model Instances**
  - **M1**
    - **MTOW**
    - **Value**

- **Model Instances**
  - **M2**
    - **MTOW**
    - **Value**

- **Model Instances**
  - **M3**
    - **MTOW**
    - **Value**

Each model instance realize states with values that fulfill requirements, ensuring the product wide model correctness.
Can We Assess “Correctness”???

We need the collaboration framework to be able to test the knowledge embedded in the models vs. knowledge patterns.
Collaboration Framework: Building Blocks

Resource State Representation Management

Resource State Representations

Resource States

Resource Definitions

Model_1 OSLC SP
Model_2 OSLC SP
Model_3 OSLC SP
Model_p OSLC SP

Model_1 RDF/OWL
Model_2 RDF/OWL
Model_3 RDF/OWL
Model_p RDF/OWL

State_{M1}
Value_{M1}
State_{M2}
Value_{M2}
State_{M2}
Value_{M2}
State_p
Value_p

Model_1 MTOW_{M1}
Model_2 MTOW_{M2}
Model_3 MTOW_{M3}
Product MTOW_p

RDF/OWL
OSLC SP
Meta-Knowledge

Bill Beavin, 12/8/2015, Clarifying the Role of OSLC.ppt | 22
Collaboration Framework: Conformance Analytics

“An realization exists in a state that fulfills a required state”

State_{M1}
Value_{M1}

Model_1
MTOW_{M1}

Model_2
MTOW_{M2}

Model_3
MTOW_{M3}

State_{M2}
Value_{M2}

Product MTOW_p

State_p
Value_p

Realization Conformance Pattern

Model_1 RDF/OWL
Fulfills
Model_2 RDF/OWL
Conformance Reasoner
Model_3 RDF/OWL
Realization
Model_p RDF/OWL

Model_1 OSLC SP
Fulfills
Model_2 OSLC SP
Model_3 OSLC SP
Model_p OSLC SP
“Models produce a realization that fulfills all requirements”
Example

- Tool₁ is used to define a required value for MTOW and stores the value in Model₁
- Tool₂ serves as the central design manager and “Single Source of Truth” (SSOT), accessing the required MTOW value from Model₁ and presenting it as part of the SSOT in Model₂
- Tool₃ needs to reference the required MTOW value from Model₂ in the SSOT for use in Tool₃ via Model₃
Example Mapped to Collaboration Framework

<table>
<thead>
<tr>
<th>Resource State</th>
<th>Model\textsubscript{1}</th>
<th>Model\textsubscript{2}</th>
<th>Model\textsubscript{3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>MTOW\textsubscript{M1}</td>
<td>MTOW\textsubscript{M2}</td>
<td>MTOW\textsubscript{M3}</td>
</tr>
<tr>
<td>State</td>
<td>\text{State}_{M1} \text{MTOW} = 100,000</td>
<td>\text{State}_{M2} \text{MTOW} = \text{null}</td>
<td>\text{State}_{M3} \text{MTOW} = \text{null}</td>
</tr>
<tr>
<td>Representation</td>
<td>Model\textsubscript{1}</td>
<td>Model\textsubscript{2}</td>
<td>Model\textsubscript{3}</td>
</tr>
<tr>
<td>Representation</td>
<td>model\textsubscript{1}.rdf</td>
<td>model\textsubscript{2}.rdf</td>
<td>model\textsubscript{3}.rdf</td>
</tr>
</tbody>
</table>

Requirements | Design | Analysis
Key Issue: Resource State Synchronization

<table>
<thead>
<tr>
<th>Resources</th>
<th>Requirements</th>
<th>Design</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model (_1)</td>
<td></td>
<td>Model (_2)</td>
<td>Model (_3)</td>
</tr>
<tr>
<td>MTOW(_{M1})</td>
<td></td>
<td>MTOW(_{M2})</td>
<td>MTOW(_{M3})</td>
</tr>
<tr>
<td>MTOW = 100,000</td>
<td>State(_{M1})</td>
<td>State(_{M2})</td>
<td>State(_{M3})</td>
</tr>
<tr>
<td></td>
<td>MTOW = null</td>
<td>MTOW = null</td>
<td></td>
</tr>
<tr>
<td>Model (_1)</td>
<td></td>
<td>Model (_2)</td>
<td>Model (_3)</td>
</tr>
<tr>
<td>model1.rdf</td>
<td></td>
<td>model2.rdf</td>
<td>model3.rdf</td>
</tr>
<tr>
<td>Model (_1)</td>
<td></td>
<td>Model (_2)</td>
<td>Model (_3)</td>
</tr>
<tr>
<td>MTOW(_{M1})</td>
<td></td>
<td>MTOW(_{M2})</td>
<td>MTOW(_{M3})</td>
</tr>
<tr>
<td>MTOW = null</td>
<td>State(_{M2})</td>
<td>State(_{M3})</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MTOW = null</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Approach 1: State Synchronization via OSLC

<table>
<thead>
<tr>
<th>Resource</th>
<th>State</th>
<th>Representation Management</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>State</th>
<th>Representations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resources</th>
<th>Requirements</th>
<th>Design</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Model_1
OSLC SP

Model_1
model1.rdf

Model_2
OSLC SP

Model_2
model2.rdf

Model_3
OSLC SP

Model_3
model3.rdf

State_{M1}
MTOW = 100,000

Model_1
MTOW_{M1}

State_{M2}
MTOW = 100,000

Model_2
MTOW_{M2}

State_{M3}
MTOW = 100,000

Model_3
MTOW_{M3}

“Serialize”

“Deserialize”

“Update”

“Read”

“Update”

“Read”

“Update”

“Read”

“Update”

“Read”

“Update”

“Read”
Approach 2: State Synchronization via DDS

State $M_1$
MTOW = 100,000

State $M_2$
MTOW = 100,000

State $M_3$
MTOW = 100,000

“Publish”
“Subscribe”

DDS

Resource State Representation Management

Resource State Representations

Resource States

Resources

Requirements

Design

Analysis
Approach 3: OSLC and DDS

- Initiate and manage representation changes via OSLC
- Propagate state changes via DDS, etc.
- Consistency Reasoner tests state propagation

Note that these patterns all include a representation of the Product
Example – Reshaped with Product

```
<table>
<thead>
<tr>
<th>Requirements</th>
<th>Design</th>
<th>Analysis</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool₁</td>
<td>Tool₂</td>
<td>Tool₃</td>
<td></td>
</tr>
<tr>
<td>Model₁</td>
<td>Model₂</td>
<td>Model₃</td>
<td></td>
</tr>
<tr>
<td>MTOW₉₉₁</td>
<td>MTOW₉₉₂</td>
<td>MTOW₉₉₃</td>
<td>MTOW₉₉₉</td>
</tr>
<tr>
<td>State₉₉₁</td>
<td>State₉₉₂</td>
<td>State₉₉₃</td>
<td>State₉₉₉</td>
</tr>
<tr>
<td>MTOW = 100,000</td>
<td>MTOW = null</td>
<td>MTOW = null</td>
<td>MTOW = null</td>
</tr>
<tr>
<td></td>
<td>State₉₉₉</td>
<td>State₉₉₉</td>
<td>State₉₉₉</td>
</tr>
<tr>
<td></td>
<td>MTOW = 100,000</td>
<td>MTOW = 100,000</td>
<td>MTOW = 100,000</td>
</tr>
</tbody>
</table>

"Product" is not specifically called out in the example but is needed for consistency and correctness checking throughout the lifecycle.
Framework Example – Consistency Check

Resource Definitions
- Model\textsubscript{1} MTOW\textsubscript{M1}
- Model\textsubscript{2} MTOW\textsubscript{M2}
- Model\textsubscript{3} MTOW\textsubscript{M3}
- Product MTOW\textsubscript{P}

Resource State Representations
- State\textsubscript{M1} 100,000
- State\textsubscript{M2} 100,000
- State\textsubscript{M3} 100,000
- State\textsubscript{P} 100,000

Resource State Mgmt
- Model\textsubscript{1} OSLC SP
- Model\textsubscript{2} OSLC SP
- Model\textsubscript{3} OSLC SP
- Model\textsubscript{P} OSLC SP

Model Consistency Pattern
- Model\textsubscript{1} RDF/OWL
- Model\textsubscript{2} RDF/OWL
- Model\textsubscript{3} RDF/OWL
- Model\textsubscript{P} RDF/OWL

Consistency Reasoner
- Realization
- Fulfills
- Realizes
- Product Instance

DDS

State M1
100,000

State M2
100,000

State M3
100,000

State P
100,000
So… Where are the Tools/Suites in this Picture?

Model₁ OSLC SP

Model₂ RDF/OWL

Model₃ RDF/OWL

Modelₚ RDF/OWL

DDS

Consistency Reasoner

Resource State Representations

Stateₘ₁ 100,000

Stateₘ₂ 100,000

Stateₘ₃ 100,000

Stateₚ 100,000

Model Definitions

Model₁ MTOWₘ₁

Model₂ MTOWₘ₂

Model₃ MTOWₘ₃

Product MTOWₚ

Model Instances

Model Consistency Pattern

Product Instance

| M₁ | MTOWₘ₁ | Stateₘ₁ | Valueₘ₁ |
| M₂ | MTOWₘ₂ | Stateₘ₂ | Valueₘ₂ |
| Mₚ | MTOWₘ₃ | Stateₘ₃ | Valueₘ₃ |

Consistency

Realizes

Fulfills

Fulfills

Fulfills
So... Where are the Tools/Suites in this Picture?
How do the Tools/Suites Communicate?

OSLC Service Provider Related Standards
Catalog, Resource Shape, Query, Creation, Oauth, etc.

Linked Data Related Standards
RDF, OWL, JSON, SPARQL, etc.

State Exchange Related Standards
DDS, DIS, HLA, CORBA, etc.

Resource Related Standards
Mechanical (AP203), Electrical (AP210), PLCS(AP239), etc.

Zoom in for a “Tool$_1$ to Tool$_N$” pattern

Resource State Representation Management

Resource State Representations

Resource States

Resources

Model Consistency Pattern

Consistency Reasoner

MTOW

Modeling Tool/Suite

Simulation and/or Prototyping Tools/Suites

Analytics Tools/Suites

OSLC SP

Modeling Tool/Suite

Modeling Tool/Suite

Modeling Tool/Suite

Modeling Tool/Suite

Resource State Management

Resource Definitions

Resource State Definitions

Resource State Representations

Resource State Representation Management
The “N-Dimensional Semantic Zipper”

- **Resource States**
  - Related Standards
  - Linked Data
  - OSLC Service Provider
  - State Exchange
  - Resource Definitions
  - Only focusing here is like “Semantic Duct Tape”

- **Tool/Suite**
  - 1
  - n

- **OSLC Service Provider Related Standards**
  - Catalog, Resource Shape, Query, Creation, OAuth, etc.

- **Linked Data Related Standards**
  - RDF, OWL, JSON, SPARQL, etc.

- **State Related Exchange Standards**
  - DDS, DIS, HLA, CORBA, etc.

- **Resource Related Standards**
  - Mechanical (AP203), Electrical (AP210), PLCS (AP239), etc.
Observations: Key to Interoperability

The

Resource ->

Resource State ->

Resource State Representation ->

Resource State Representation Management

“Meta-Knowledge” pattern is key to multi-layer interoperability
Observations: What OSLC Adds

OSLC adds capability at the

“Resource State Representation Management” Level
Contact Info

Bill Beavin  william.c.beavin@boeing.com