Rule Models as Semantic Models for Command and Control

Francisco Loaiza
Steven Wartik
Institute for Defense Analyses

TOC

• Background
• JC3IEDM
• Business Rules In C2
• Beyond OCL
• Conclusions
Background

State of the Art

- Most information modeling languages used to develop databases, e.g., IDEF1-X, UML, provide only partial graphical depiction capabilities when it comes to expressing constraints and applicable business rules controlling the creation, use and maintenance of the data that is being modeled.
- UML extends its modeling capabilities for constraints and business rules via the Object Constraint Language (OCL).
Consequences for C2

• To take advantage of OCL the models must be recast in UML
• Our Approach
  - Convert our test C2 Model from IDEF1-X to UML
  - Rewrite ‘constraints’ and Business Rules as OCL Statements
  - Assess the applicability of more powerful ‘rule languages’ (e.g., SBVR)

What is the JC3IEDM?
The Joint Consultation, Command, and Control Information Exchange Data Model

- Defines the objects in the universe of discourse (Facilities, Features, Materiel, Organizations, Persons)
- Describes the state of the universe: past, present, and future
- Records observed events
- Plans to use what you have to achieve objectives
- Monitors the execution of planned activity

JC3IEDM Information Exchange

Effective C2 For Operations

Shared Picture

Information Transfer

Local Implementation

Common Interface

System Database

JC3IEDM

Common understanding
Business Rules in C2
**Taxonomy of JC3IEDM Business Rules**

- **Business Rules**
  - **Data Integrity/Quality Rules**
  - **Model Use Rules**
  - **Doctrinal Rules**

Explicitly documented:
- JC3IDEM Annex G1
- JC3IDEM Annex G2

From Engineering WG Documentation:
- Intra-Class BRs
- Inter-Class Subtyping Consistency BRs
- Other Inter-Class Business Rules
- Textual BRs

Future work:
- Intra-Class BRs
- Inter-Class Subtyping Consistency BRs
- Other Inter-Class Business Rules
- Textual BRs

**Intra-Class Business Rules**

When AircraftType.categoryCode = 'Fixed wing' the AircraftType.airframeDesignCode must be a value in the set {'Bomber', 'Fighter', 'Glider', 'Transport', 'Not known', 'Not otherwise specified'} or be NULL

```
context AircraftType
inv:
  categoryCode='Fixed wing' implies
  Set {'Bomber', 'Fighter', 'Glider', 'Transport', 'Not known', 'Not otherwise specified'} ->includes(airframeDesignCode)
  or airframeDesignCode.oclIsUndefined()
```
Inter-Class Subtyping Consistency

Business Rules

When instances of DryDock, a specialization of ObjectItem, are created, the appropriate corresponding instance of FacilityType, a specialization of ObjectType, must be instantiated with the value of categoryCode set to ‘Dry-dock’.

```ocl
context ObjectItem
inv: self.oclIsKindOf(DryDock)
  implies is_classified_as->forAll(ot: ObjectType |
      ot.oclIsKindOf(FacilityType) 
      and ot.oclAsType(FacilityType).categoryCode = 'Dry-dock')
```

Other Inter-Class Business Rules

Instances of ControlFeature that constitute elements in the definition of Q-routes entail restrictive associations to conform to the concept of Q-routes.

<table>
<thead>
<tr>
<th>Subject ControlFeature Typed As</th>
<th>Object ControlFeature Typed As</th>
<th>ObjectItemAssociation::categoryCode Value</th>
<th>Number of Permissible Associations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-route</td>
<td>Q-zone</td>
<td>Is part of</td>
<td>1</td>
</tr>
<tr>
<td>Way point</td>
<td>Q-route</td>
<td>Is part of</td>
<td>2 or more</td>
</tr>
</tbody>
</table>

```ocl
context ObjectItemAssociation
inv: is_the_subject_of.oclIsKindOf(ControlFeature) and is_the_subject_of.is_classified_as->forall(oclIsKindOf(RouteType) and oclAsType(RouteType).categoryCode = 'QROUTE') 
  and is_the_object_of.is_classified_as->forAll(oclIsKindOf(ControlFeatureType) 
      and oclAsType(ControlFeatureType).categoryCode = 'QZONE') 
  implies categoryCode = 'ISPART' and is_the_subject_of.is_the_object_of->size() = 1
```
For the instances where the Minefield is a MinefieldLand, then the destructionDatetime is filled only where persistenceCode is “Remote activated destruction” or “timed automatic destruction”.

**OCL**

```
context MinefieldLand
inv: Set {'Remote activated destruction', 'Timed automatic destruction'} ->includes(persistenceCode)
  implies not destructionDatetime.oclIsUndefined()

inv: not Set {'Remote activated destruction', 'Timed automatic destruction'} ->includes(persistenceCode)
  implies destructionDatetime.oclIsUndefined()
```

**Assessment of OCL Capability**

- Almost all the use and data integrity rules of the JC3IEDM can be expressed in OCL
- The only type of rule that is not amenable to capture via OCL is the one that requires the use of operators not available in OCL. In the JC3IEDM this means those rules that require the use of trigonometric functions
- OCL rules are formally equivalent to first order logic (FOL) statements that concern either the behavior of sets produced by set-traversal operators; the values of class properties within a given class; or the values of class properties from different classes in the form of if-then implications.
Beyond OCL

Can We Do Better?

- Shortcomings of OCL
  - OCL rules are always written against the classes defined in a specific UML model
  - OCL does not support mathematical operators
  - Not suited to technophobes:
    - Syntax is non-intuitive
    - Somewhat cumbersome
SBVR

- Semantics of Business Vocabulary and Business Rules
- Rules can be written in Structured English
- Language supports:
  - Quantification operators, e.g., each, some, at least one
  - Logical operators, e.g., not, and, or, if-then, nand, nor, whether-or-not
  - Modal operators, e.g., is obligatory that, is prohibited that, is necessary that, is impossible that, it is permitted that
- Models written in SBVR also support MDA approach
- SBVR rules are FOL statements

SBVR as a Superset of OCL(1)

context GeographicFeature
inv: has_affiliation->size() = 0

classic MeteorologicFeature
inv: has_affiliation->size() = 0

Each «GeographicFeature» must not have an «Affiliation»
Each «MeteorologicFeature» must not have an «Affiliation»
SBVR as a Superset of OCL(2)

When side has meaning for a line, the left-hand side is interpreted according to the direction of the line as determined from an ascending numeration of the points of the line….

If a Line has South-to-North direction then a user must interpret the left hand side of the Line as West.

SBVR explicitly mentions user and ties his behavior to operative intent. This capability is not supported in OCL.

Normative Interactions Specification (NIS)

- A complete, precise, and verifiable documentation of the directives or rules that prescribe the expected characteristics and values of the relationships that are binding upon the objects that participate in the interactions
- In that respect the norms serve to guide, control, or regulate proper and acceptable behavior
NIS Completeness

Each a **Action** must have a **name**
The **name** of an **Action** must be written using ISO-93884 encoding
The **name** of an **Action** cannot exceed 50 characters
Each a **Action** must have a **categoryCode**
The **categoryCode** of an **Action** cannot exceed 6 characters

```
NIS as a PIM
```

```
logical

pr

physical

<table>
<thead>
<tr>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ categoryCode: cha(8)</td>
</tr>
<tr>
<td>+ nameText: char(50)</td>
</tr>
</tbody>
</table>
```

Automated Consistency Checking

```
Prover9 Input
formulas(assumptions).

GeographicFeature(G) -> exists A has_affiliation(G,A).
GeographicFeature(G) -> -(exists A has_affiliation(G,A)).
(exists G GeographicFeature(G)).

end_of_list.

(1) GeographicFeature(G) -> exists A has_affiliation(G,A)
(2) GeographicFeature(G) -> ~(exists A has_affiliation(G,A))
(3) (exists G GeographicFeature(G))
(4) GeographicFeature(c1)
(5) ~GeographicFeature(A) | has_affiliation(A,f1(A))
(6) ~GeographicFeature(A) | ~has_affiliation(A,B)
(7) ~has_affiliation(c1, A)
(8) has_affiliation(c1, f1(c1))
(9) False
```
Conclusions

• The recent work in MIP where the JC3IxEDM specifications have been migrated to UML has opened the door to the formalization via OCL of all the current rules controlling the use of the model and the integrity of the data sets

• The development of a more expressive language for capturing business rules, namely, SBVR, suggests that at a minimum the OCL formulation of the rules should be transformed into SBVR structured English, and that potentially all C2 information interactions could be also formally captured to provide a more robust and stable specification from which one can create through appropriate transformations the required PSMs

• As a bonus, with a NIS written in structured English one could also take advantage of some recent development in automated theorem proofing, many of which accept as input FOL statements