DOL appendix E: conformance of UML class diagrams Second version

Till Mossakowski, Alexander Knapp

New stuff displayed in red

OntoIOp/DOL, February 9, 2015

Overview

Conformance of UML class diagrams to DOL means to provide an institution, i.e.

- Signatures (compact representation of the non-logical symbols occurring in the abstract syntax tree)
- Models (here, we do this via a translation to Common Logic)
- Sentences (compact representation of the constraints occurring in the abstract syntax tree)
- Satisfaction (of a sentence in a model)
- Signature morphisms (transcends proper UML) and associated model reducts and sentence translations

Signatures

Signatures $\Sigma \in |\mathrm{Sig}|$ comprise

- Classifier hierarchy $(C, \leq)$
  e.g. `Flight` $<$ `Travel`
- Instance specifications $k : c$
  e.g. `LH123:Flight`
  (OK, this is for object diagrams...)
- Property declarations $c.p(x_1 : c_1, \ldots, x_n : c_n) : \tau[c']$
  e.g. `Flight.number:Int`
- Composition declarations $c{\blacklozenge}r : \tau[c']$
  e.g. `TrackPlan`◆`signal` : `Signal`
- Association declarations $a(r_1 : c_1, \ldots, r_n : c_n)$ e.g.
  `PlacedAt(signal` : `Signal`, `trackSection` :
  `TrackSection)`

Signature morphisms map these in a compatible way.

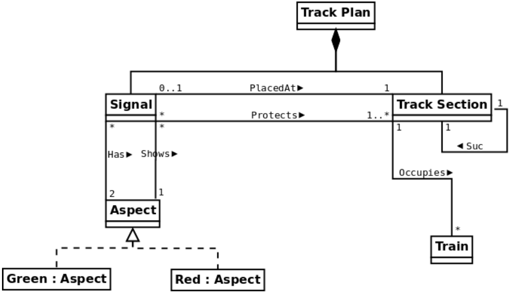
Classifier annotations

For each classifier $c \in C$ of a classifier hierarchy $(C, \leq_C)$ we use the classifier annotations OrderedSet, Set, Sequence, and Bag representing the meta-properties "ordered" and "unique" according to the following table:

	ordered	*not ordered*
unique	OrderedSet	Set
not unique	Sequence	Bag

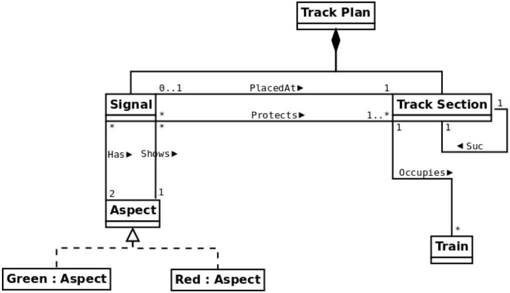
We write $\tau[c]$ for an annotated classifier for $\tau \in \{\text{OrderedSet}, \text{Set}, \text{Sequence}, \text{Bag}\}$. The default is "not ordered" and "unique" (UML Superstructure Specification 2.4.1, p. 96).

Signature Example: Circle DSL (1)



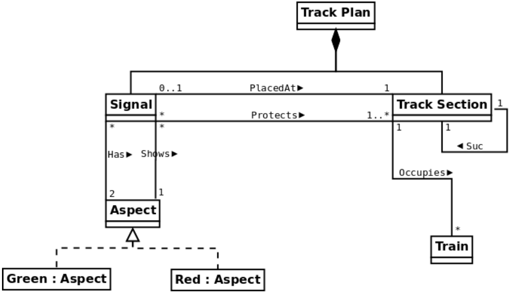
- Classifiers:
  `TrackPlan`, `Signal`, `TrackSection`, `Aspect`, `Train`

Signature Example: Circle DSL (2)



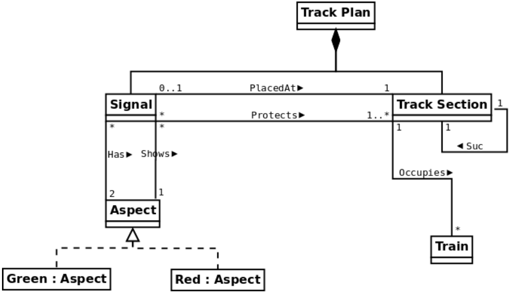
- Compositions:
  TrackPlan◆signal : Set[Signal],
  TrackPlan◆trackSection : Set[TrackSection]

Signature Example: Circle DSL (3)



- Associations, e.g.:
  PlacedAt({signal : Signal, trackSection : TrackSection}),
  Protects({signal : Signal, trackSection : TrackSection}),
  Suc({trackSection1 : TrackSection, trackSection2 : TrackSection})

Signature Example: Circle DSL (4)



- Instance specifications:
  Green : Aspect, Red : Aspect

Enumerations in Common Logic

```
(iff (exhaustive c ...)
     (forall (x) (if (c x) (oneof x ...))))
     // does ... exhaust the extension of c?
(not (oneof x)) // is x among the remaining arguments?
(iff (oneof x y ...) (or (= x y) (oneof x ...)))
(iff (enumeration c ...)
     (and (exhaustive c ...) (distinct ...)))
      // c is an enumeration type with values ...
```

Models via translation to Common Logic

Models are inherited from Common Logic, via a translation:
For a class net $\Sigma = ((C, \leq_C), K, P, M, A)$, we define a Common Logic theory $CL(\Sigma)$ consisting of:

- for each class or datatype $c \in C$, a predicate CL(c), such that
  - CL(`Boolean`) = `buml:Boolean`,
  - CL(`String`) = `buml:String`,
  - CL(`Integer`) = `buml:Integer`,
  - CL(`UnlimitedNatural`) = `form:NaturalNumber`,
  - CL(`Real`) = `buml:Real`,
  - CL(`List`[c]) = `form:Sequence` — only untyped sequences!
  - CL(`Set`[c]) = `form:Set` — to be specified
  - CL(`OrderedSet`[c]) = `form:OrderedSet` — to be specified
  - CL(`Bag`[c]) = `form:Bag` — to be specified
  - CL(c) = c, if c is an enumeration type with values $k_1, \ldots, k_n$. Additionally, the Common Logic theory is augmented by (`enumeration` c $k_1 \cdots k_n$)
- for each subclass relation $c_1 \leq_C c_2$, an axiom `(forall (x) (if (C1 x) (C2 x)))`, where C1= CL(c_1), C2= CL(c_2).

Models via translation to Common Logic (cont'd)

- CL maps each instance specification declaration $k : c \in K$ to constant CL(k) and an axiom `(c k)`, where by abuse of notation, we identify c with CL(c), and k with (CL(k)) (this abuse of notation will also be used in the sequel);
- for two instance specifications $k_1 : c$ and $k_2 : c$ with $k_1 \neq k_2$, an axiom `(not (=` k_1 k_2`))` (the unique name assumption);

Models via translation to Common Logic (cont'd)

- CL maps each property declaration $c.p(x_1 : c_1, \ldots, x_n : c_n) : \tau[c'] \in P$ to a predicate $\mathrm{CL}(c.p)$ and axioms
  - (forall (x x_1 x_2 $\cdots$ x_n y)
    (if (c.p x x_1 x_2 $\cdots$ x_n y) (c x)))
  - (forall (x x_1 x_2 $\cdots$ x_n y)
    (if (c.p x x_1 x_2 $\cdots$ x_n y) (c_i x_i)))
    for each $i = 1 \ldots n$,[1]
  - (forall (x x_1 x_2 $\cdots$ x_n y)
    (if (c.p x x_1 x_2 $\cdots$ x_n y) ($\tau[c']$ y)))
  - (forall (x x_1 x_2 $\cdots$ x_n y m)
    (if (and (c.p x x_1 x_2 $\cdots$ x_n y) (member m y))
    (c' m)))
  - (forall (x x_1 x_2 $\cdots$ x_n)
    (exists (y) (c.p x x_1 x_2 $\cdots$ x_n y)))
  - (forall (x x_1 x_2 $\cdots$ x_n y z)
    (if (and (c.p x x_1 x_2 $\cdots$ x_n y)
    (c.p x x_1 x_2 $\cdots$ x_n z)) (= y z)))

[1] Note that the $\cdots$ here is meta notation, not a sequence marker!

Models via translation to Common Logic (cont'd)

- CL maps each composition declaration $c \blacklozenge r : c' \in M$ to a predicate CL(r) and an axiom
  (forall (x y) (if (r x y) (and (c x) ($\tau[c']$ y))))
  (forall (x) (if (c x)
      (exists (y) (and (r x y) (τ[c'] y)))))
  (forall (x y z) (if (and (r x y) (r x z))
      (= y z)))
- for any pair of composition declarations $c_1 \blacklozenge r_1 : c_1'$ and $c_2 \blacklozenge r_2 : c_2'$, an axiom stating "each instance has at most one owner":
  (forall (x_1 x_2 y_1 y_2)
      (if (and (r_1 x_1 y_1) (r_2 x_2 y_2))
- CL maps each association declaration
  $a(r_1 : c_1, \ldots, r_n : c_n) \in A$ to a predicate CL(a) and an axiom
  (forall (x_1 x_2 $\cdots$ x_n) (if (a x_1 x_2 $\cdots$ x_n)
      (and (c_1 x_1) $\cdots$ (c_n x_n)))))

Sentences and Satisfaction

Sentences $\varphi \in |Sen(\Sigma)|$ capture *multiplicities*

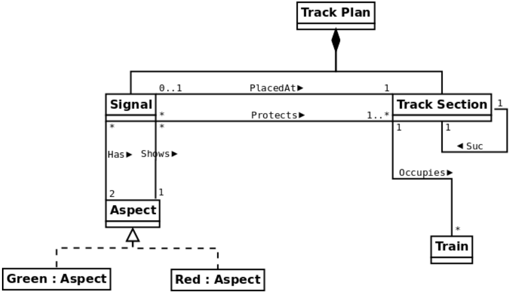
- Comparing carnality's $e \leq \ell$, $e \geq \ell$ (ℓ a natural number)

Cardinality expressions e

- ... of compositions
  - How many owned instances? $\#(c \blacklozenge r : c')$
  - How many owners? $\#(c \blacklozenge r : c')[\mathrm{owner}]$
- ... of associations
  - How many tuples, when fixing a subset of roles? $\#z(\{r_1 : c_1, \ldots, r_n : c_n\})[r_{i_1}, \ldots, r_{i_m}]$

Satisfaction relation $\mathcal{M} \models_\Sigma e \leq \ell$, $\mathcal{M} \models_\Sigma \ell \leq e$

- Compare each cardinality resulting from evaluating e to ℓ



- Cardinality of associations, e.g. : $\#\mathtt{PlacedAt}(\{\mathtt{signal} : \mathtt{Signal}, \mathtt{trackSection} : \mathtt{TrackSection}\})[\mathtt{signal}] \leq 1$ $\#\mathtt{PlacedAt}(\{\mathtt{signal} : \mathtt{Signal}, \mathtt{trackSection} : \mathtt{TrackSection}\})[\mathtt{signal}] \geq 1$

Satisfaction relation.

Inherited from CL via translation CL(_):

- $\mathrm{CL}(\ell \leq \#c{\blacklozenge}r : c') =$
```
(forall (x y n)
     (if (and (r x y)
                   (form:sequence-length y n))
          (leq [[ℓ]] n))
```
- $\mathrm{CL}(\ell \leq \#c{\blacklozenge}r : c') =$
```
(forall (x y n)
     (if (and (r x y)
                   (form:sequence-length y n))
          (geq [[ℓ]] n))
```
- $\mathrm{CL}(c{\blacklozenge}r : c'!) =$
```
(forall (y)
     (if (c' y) (exists (x)
                       (and (c x) (r x y)))))
```

where $[\![-]\!] : \mathit{NumLit} \to \mathbb{Z}$ maps a numerical literal to an integer.

Satisfaction relation (cont'd)

- $\mathrm{CL}(\ell \leq \#a(r_1 : c_1, \ldots, r_n : c_n)[r_{i_1}, \ldots, r_{i_m}] =$
  (forall (x_{i_1} $\cdots$ x_{i_m})
      (if (and (c_{i_1} x_{i_1}) $\cdots$ (c_{i_m} x_{i_m}))
          ((min-card-tuple a x_{i_1} $\cdots$ x_{i_m}) sel_{i_1} $\cdots$ sel_{i_n})))
- $\mathrm{CL}(\#a(r_1 : c_1, \ldots, r_n : c_n)[r_{i_1}, \ldots, r_{i_m}] \leq \ell =$
  (forall (x_{i_1} $\cdots$ x_{i_m})
      (if (and (c_{i_1} x_{i_1}) $\cdots$ (c_{i_m} x_{i_m}))
          ((max-card-tuple a x_{i_1} $\cdots$ x_{i_m}) sel_{i_1} $\cdots$ sel_{i_n})))