DOL appendix E: conformance of UML class diagrams

Till Mossakowski, Alexander Knapp

OntoIOp/DOL, January 26, 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

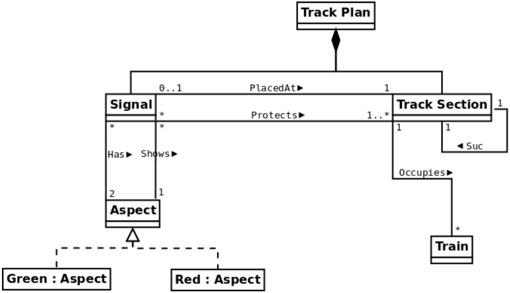
Signatures

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

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

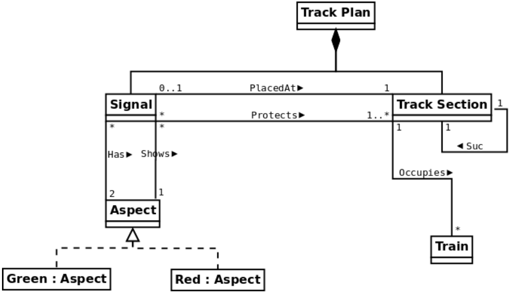
Signature morphisms map these in a compatible way.

Signature Example: Circle DSL (1)



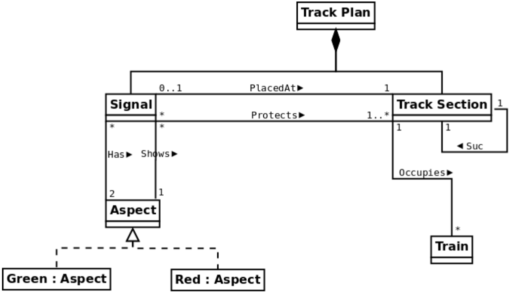
- Classifiers: `TrackPlan`, `Signal`, `TrackSection`, `Aspect`, `Train`

Signature Example: Circle DSL (2)



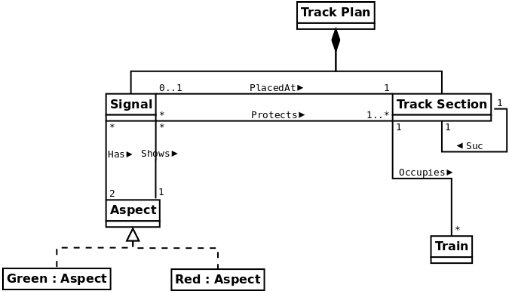
- Compositions:
  $\texttt{TrackPlan}\blacklozenge\texttt{signal} : \texttt{Signal}$,
  $\texttt{TrackPlan}\blacklozenge\texttt{trackSection} : \texttt{TrackSection}$

Signature Example: Circle DSL (3)



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

Signature Example: Circle DSL (4)



- Instance specifications:
  `Green` : `Aspect`, `Red` : `Aspect`

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 $c \in C$, a predicate $\mathrm{CL}(c)$, such that
  - $\mathrm{CL}(\texttt{Boolean}) = \texttt{buml:Boolean}$,
  - $\mathrm{CL}(\texttt{String}) = \texttt{buml:String}$,
  - $\mathrm{CL}(\texttt{Integer}) = \texttt{buml:Integer}$,
  - $\mathrm{CL}(\texttt{UnlimitedNatural}) = \texttt{form:NaturalNumber}$,
  - $\mathrm{CL}(\texttt{Real}) = \texttt{buml:Real}$,
  - $\mathrm{CL}(\texttt{List}[c]) = \texttt{form:Sequence}$ — typed sequences???
  - $\mathrm{CL}(\texttt{Set}[c]) =$???
  - $\mathrm{CL}(\texttt{Bag}[c]) =$???
  - $\mathrm{CL}(\texttt{Pair}[c_1, c_2]) =$???
  - $\mathrm{CL}(\texttt{Enumeration}[v_1, v_2, \ldots, v_n]) =$???
- for each subclass relation $c_1 \leq_C c_2$, an axiom
  `(forall (x) (if (C1 x) (C2 x)))`,
  where C1$=$ $\mathrm{CL}(c_1)$, C2$=$ $\mathrm{CL}(c_2)$,

Models via translation to Common Logic (cont'd)

- CL maps each instance specification declaration $k : c \in K$ to constant $\mathrm{CL}(k)$ and an axiom `(c k)`, where by abuse of notation, we identify c with $\mathrm{CL}(c)$, and k with $(\mathrm{CL}(k))$ (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);
- CL maps each property declaration $c.p(x_1 : c_1, \ldots, x_n : c_n) : c' \in P$ to a predicate $\mathrm{CL}(c.p)$ and axioms
  - (forall (x_1 x_2 $\cdots$ x_n x) (if (c.p x_1 x_2 $\cdots$ x_n x) (c_i x_i))) for each $i = 1 \ldots n$,[1]
  - (forall (x_1 x_2 $\cdots$ x_n x) (if (c.p x_1 x_2 $\cdots$ x_n x) (c' x)))
  - (forall (x_1 x_2 $\cdots$ x_n x y) (if (and (c.p x_1 x_2 $\cdots$ x_n x) (c.p x_1 x_2 $\cdots$ x_n x)) (= x y)))

[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 $\mathrm{CL}(r)$ and an axiom
  `(forall (x y) (if (r x y) (and (c x) (c' y))))`
- 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) (if (and (r_1 x_1 y) (r_2 x_2 y)) (= x_1 x_2)))
- CL maps each association declaration $a(r_1 : c_1, \ldots, r_n : c_n) \in A$ to a predicate $\mathrm{CL}(a)$ 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)

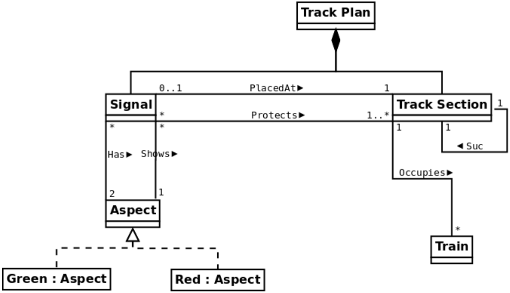
Cardinality expressions e

- ... of compositions
  - How many owned instances? $\#(c \blacklozenge r : c')$
    - For each $o \in c^{\mathcal{O}}$, cardinality of $(c \blacklozenge r : c')^{\mathcal{O}}(o)$
  - How many owners? $\#(c \blacklozenge r : c')[\texttt{owner}]$
- ... 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

Sentences Example: Circle DSL



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