(1. 1.7.)

Start of Lecture 2: A SPECIFICATION ONTOLOGY

2. A Specification Ontology

- In order to describe domains we postulate the following related specification components:
  - *entities*,
  - *actions*,
  - *events* and
  - *behaviours*.

(2. A Specification Ontology)

2.1. Entities

- By an entity we shall understand
  - a phenomenon we can point to in the domain
  - or a concept formed from such phenomena.

(2. A Specification Ontology 2.1. Entities)

Example 1 – Entities

- The example is that of aspects of a transportation net.
- You may think of such a net as being either a road net, a rail net, a shipping net or an air traffic net.
- Hubs are then street intersections, train stations, harbours, respectively airports.
- Links are then street segments between immediately adjacent intersections, rail tracks between train stations, sea lanes between harbours, respectively air lanes between airports.

1 There are hubs and links.

2 There are nets, and a net consists of a set of two or more hubs and one or more links.

3 There are hub and link identifiers.

4 Each hub (and each link) has an own, unique hub (respectively link) identifier (which can be observed (ω) from the hub [respectively link]).

type
[1] H, L,
[2] N = H-**set** × L-**set**
axiom [nets−hubs−links−1]
[2] ∀ (hs,ls):N · **card** hs≥2 ∧ **card** ks≥1
type
[3] HI, LI
value
[4] ωHI: H → HI, ωLI: L → LI
axiom [nets−hubs−links−2]
[4] ∀ h,h′:H, l,l′:L · h≠h′ ⇒ ωHI(h)≠ωHI(h′) ∧ l≠l′⇒ωLI(l)≠ωLI(l′)

- In order to model the physical (i.e., domain) fact
  - that links are delimited by two hubs and
  - that one or more links emanate from and are, at the same time, incident upon a hub
- we express the following:

5 From any link of a net one can observe the two hubs to which the link is connected. We take this ‘observing’ to mean the following: from any link of a net one can observe the two distinct identifiers of these hubs.

6 From any hub of a net one can observe the identifiers of one or more links which are connected to the hub.

7 Extending Item [5]: the observed hub identifiers must be identifiers of hubs of the net to which the link belongs.

8 Extending Item [6]: the observed link identifiers must be identifiers of links of the net to which the hub belongs.

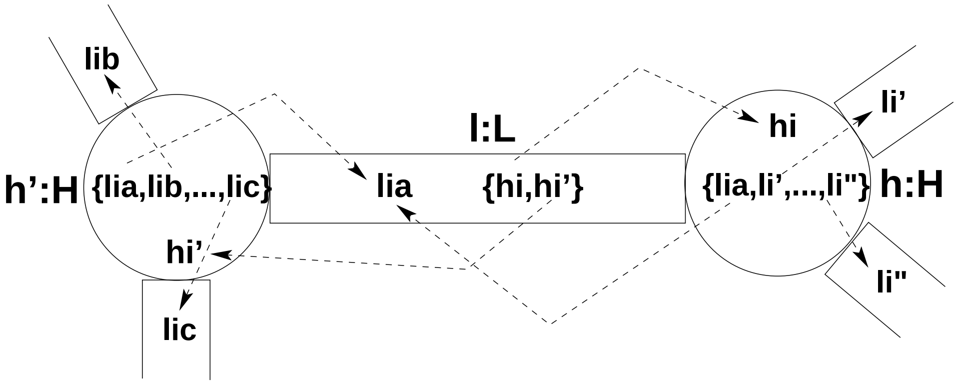

Figure 1: Connected links and hubs with observable identifiers

value
[5] ωHIs: L → HI-**set**,
[6] ωLIs: H → LI-**set**,
axiom [net–hub–link–identifiers–1]
[5] ∀ l:L • **card** ωHIs(l)=2 ∧
[6] ∀ h:H • **card** ωLIs(h)≥1 ∧
∀ (hs,ls):N •
[5] ∀ h:H • h ∈ hs ⇒ ∀ li:LI • li ∈ ωLIs(h)
⇒ ∃ l′:L • l′ ∈ ls ∧ li=ωLI(l′) ∧ ωHI(h) ∈ ωHIs(l′) ∧
[6] ∀ l:L • l ∈ ls ⇒ ∃ h′,h″:H • {h′,h″}⊆hs ∧ ωHIs(l)={ωHI(h′),ωHI(h″)}
[7] ∀ h:H • h ∈ hs ⇒ ωLIs(h) ⊆ iols(ls)
[8] ∀ l:L • l ∈ ls ⇒ ωHIs(h) ⊆ iohs(hs)
value
iohs: H-**set** → HI-**set**, iols: L-**set** → LI-**set**
iohs(hs) ≡ {ωHI(h)|h:H•h ∈ hs}
iols(ls) ≡ {ωLI(l)|l:L•l ∈ ls}

- In the above extensive example we have focused on just five entities: nets, hubs, links and their identifiers.
- The nets, hubs and links can be seen as separable phenomena.
- The hub and link identifiers are conceptual models of the fact that hubs and links are connected
  - so the identifiers are abstract models of 'connection',
  - i.e., the mereology of nets, that is, of how nets are composed.
- These identifiers are attributes of entities.
- Links and hubs have been modelled to possess link and hub identifiers.
  - A link's "own" link identifier enables us to refer to the link,
  - A link's two hub identifiers enables us to refer to the connected hubs.
  - Similarly for the hub and link identifiers of hubs and links.

9 A hub, h_i, state, $h\sigma$, is a set of hub traversals.

10 A hub traversal is a triple of link, hub and link identifiers $(l_{i_{in}}, h_{i_i}, l_{i_{out}})$ such that $l_{i_{in}}$ and $l_{i_{out}}$ can be observed from hub h_i and such that h_{i_i} is the identifier of hub h_i.

11 A hub state space is a set of hub states such that all hub states concern the same hub.

type

[9] $\mathsf{HT} = (\mathsf{LI}\times\mathsf{HI}\times\mathsf{LI})$

[10] $\mathsf{H\Sigma} = \mathsf{HT}\textbf{-set}$

[11] $\mathsf{H\Omega} = \mathsf{H\Sigma}\textbf{-set}$

value

[10] $\omega\mathsf{H\Sigma}\text{: } \mathsf{H} \to \mathsf{H\Sigma}$

[11] $\omega\mathsf{H\Omega}\text{: } \mathsf{H} \to \mathsf{H\Omega}$

axiom [hub−states]

$\forall$ n:N,h:H•h $\in \omega$Hs(n)$\Rightarrow$wf_HΣ(ωHΣ(h))$\wedge$wf_HΩ(h,ωHΩ(h))

value

wf_HΣ: H$\Sigma \to$ **Bool**, wf_HΩ: H$\times$H$\Omega \to$ **Bool**

wf_HΣ(hσ) $\equiv \forall$ (li,hi,li′),(_,hi′,_):HT•(li,hi,li′)$\in$ h$\sigma \Rightarrow$ {li,li′}$\subseteq\omega$LIs(h)$\wedge$hi=ωHI(h)$\wedge$hi′=hi

wf_HΩ(h,hω) $\equiv \forall$ hσ:HΣ•h$\sigma \in$ h$\omega\Rightarrow$wf_HΣ(hσ)$\wedge$h$\sigma\neq${} $\Rightarrow$

let (li,hi,li′):HT•(li,hi,li′)$\in$ hσ **in** hi=ωHI(h) **end**

■ End of Example 1

2.2. Actions

- A set of entities form a domain state.
- It is the domain engineer which decides on such states.
- A function is an action if,
  - when applied
    - to zero, one or more arguments
    - and a state,
  - it then results in a state change.
- (Arguments could be other entities or just values of entity attributes.)

Example 2 – Deterministic Hub State Setting

12 Our example action is that of setting the state of hub.

13 The setting applies to a hub

14 and a hub state in the hub state space

13 and yields a “new” hub.

15 The before and after hub identifier remains the same.

16 The before and after hub state space remains the same.

17 The result hub state is that being set (i.e., the argument hib state).

value

[12] set_HΣ: H $\times$ H$\Sigma \to$ H

[13] set_HΣ(h,hσ) **as** h′

[14] **pre** h$\sigma \in \omega$HΩ(h)

[15] **post** ωHI(h)=ωHI(h′)$\wedge$

[16] ωHΩ(h)=ωHΩ(h′)$\wedge$

[17] ωHΣ(h′)=hσ

■ End of Example 2

- Example 2 illustrated a deterministic action:
  - one that always succeeded
  - in carrying out the prescribed operation.
- But, as we shall see later,
  - the domain technology may be faulty and
  - an action, as carried out by such a technology,
  - may fail to have the desired effect.

Example 3 – Non-Deterministic Hub State Setting

17 The result hub state is the that of the argument.

value

[12] set_HΣ: H × HΣ → H
[13] set_HΣ(h,hσ) **as** h′
[14] **pre** hσ ∈ ωHΩ(h)
[15] **post** ωHI(h)=ωHI(h′)∧
[16] ωHΩ(h)=ωHΩ(h′)∧
[17] ωHΣ(h′)=hσ

2.3. Events

- Any domain state change is an event.
- A situation
  - in which a (specific) state change was expected
  - but none (or another) occurred is an event.
- Some events are more "interesting" than other events.
- Not all state changes are caused by actions of the domain.

Example 4 – Events: Failure State Transitions

18 A hub is in some state, $h\sigma$.

19 An action directs it to change to state $h\sigma'$ where $h\sigma' \neq h\sigma$.

20 But after that action the hub remains either in state $h\sigma$ or is possibly in a third state, $h\sigma''$ where $h\sigma'' \notin \{h\sigma, h\sigma'\}$.

21 Thus an "interesting event" has occurred !

```
∃ n:N,h:H,hσ,hσ′:HΣ·h ∈ ωHs(n)∧
  [19,20]  {hσ,hσ′}⊆ωHΩ(h)∧card{hσ,hσ′}=2 ∧
  [18]        ωHΣ(h)=hσ ;
  [19]        let h′ = set_HΣ(h,hσ′) in
  [20]        ωHΣ(h′)∈ ωHΣ(h′)\{hσ′} ⇒
  [21]        "interesting event" end
```

- It only makes sense to change hub states if there are more than just one single such state.

■ End of Example 4

2.4. Behaviours

- A behaviour is a set of
  - zero, one or more sequences of sets of
    - ∗ actions
    - ∗ or behaviours,
    - ∗ including events.

Example 5 – Behaviours: Blinking Semaphores

22 Let h be a hub of a net n.

23 Let $h\sigma$ and $h\sigma'$ be two distinct states of h.

24 Let $ti : TI$ be some time interval.

25 Let h start in an initial state $h\sigma$.

26 Now let hub h undergo an ongoing sequence of n changes

26a from $h\sigma$ to $h\sigma'$ and

26b then, after a wait of ti seconds,

26c and then back to $h\sigma$.

26d After n blinks a pause, $tp : TI$, is made and blinking restarts.


```
type
  TI
value
  ti,tj:TI [axiom tj¿¿ti]
  n:Nat,
  [26]  blinking: H × HΣ × HΣ → Unit
  [26]  blinking(h,hσ,hσ′,m) in
  [25]     let h′ = set_HΣ(h,hσ) in
  [26c]    wait ti ;
  [26a]    let h″ = set_HΣ(h′,hσ′) in
  [26c]    wait ti ;
  [26]     if m=1
  [26]        then skip
  [26]        else blinking(h,hσ,hσ′,m−1) end end end
  [26]      wait tj ;
  [26d]     blinking(h,hσ,hσ′,m)
  [23]     pre {hσ,hσ′}⊆ωHΩ(h)∧hσ≠hσ′
  [26]          ∧ initial m=100
```

■ End of Example 5

End of Lecture 2: A SPECIFICATION ONTOLOGY