

### 3 An Ontology of Domain Facets

53

**Definition: Domain.** *An area of activity which some software<sup>[685]</sup> is to support (or supports) or partially or fully automate (resp. automates).*

The term ‘application domain’ is considered synonymous with the term ‘domain’.

**Definition: Domain Description.** *A textual, informal or formal document which describes a domain as it is.*

Usually a domain description is a set of documents with many parts recording many facets of the domain: The *business process<sup>[99]</sup>es*, *intrinsic<sup>[399]</sup>s*, *support technology<sup>[725]</sup>*, *rules and regulations<sup>[640]</sup>*, *management and organisation<sup>[445]</sup>*, and the *human behaviour<sup>[345]</sup>s*.

**Definition: Domain Engineering.** *The engineering of the development of a domain description<sup>[243]</sup>, from identification of domain<sup>[239]</sup> stakeholder<sup>[703]</sup>s, via domain acquisition<sup>[240]</sup>, domain analysis<sup>[241]</sup>, terminologisation, and domain description<sup>[243]</sup> to domain validation<sup>[256]</sup> and domain verification<sup>[257]</sup>.*

**Definition: Domain Facet.** *By a domain facet we understand one amongst a finite set of generic ways of analysing a domain: A view of the domain, such that the different facets cover conceptually different views, and such that these views together cover the domain.*

We consider here the following domain facets: *business process<sup>[99]</sup>es*, *intrinsic<sup>[399]</sup>s*, *support technology<sup>[725]</sup>*, *rules and regulations<sup>[640]</sup>*, *management and organisation<sup>[445]</sup>*, and *human behaviour<sup>[345]</sup>*.

#### 3.1 What Can Be Observed

57

- “Whether you can observe a thing or not depends on the theory which you use. It is the theory which decides what can be observed.”
- Albert Einstein objecting to the placing of observables at the heart of the new quantum mechanics, during Heisenberg’s 1926 lecture at Berlin; related by Heisenberg, quoted in *Unification of Fundamental Forces* (1990) by Abdus Salam ISBN 0521371406.

#### 3.2 Intrinsic<sup>s</sup>

58

**Definition: Intrinsic<sup>s</sup>.** *By the intrinsic<sup>s</sup> of a domain<sup>[239]</sup> we shall understand those phenomena and concepts of a domain which are basic to any of the other facets, with such a domain intrinsic<sup>s</sup> initially covering at least one stakeholder<sup>[703]</sup> view.*

##### 3.2.1 Net Topology Descriptors

59

Instead of dealing with the entire phenomenon of a net, that is, the real, physical, geographic “thing”, we can describe essentials of a net, for example how its hub and links are connected.

52. One way of abstractly modelling a net descriptor is as a map, **nd**, from hub identifiers to simple maps, **lihi**, from link identifiers to hub identifiers,
53. such that
- for all **hi** in (the definition set of) **nd** it is the case that
  - if **hi** maps to **lihi**,
  - and in that link identifier to hub identifier map, **li** maps to **hi'**,
  - then **hi'** is different from **hi** and
  - hi'** maps to an **lihi'** in which **li** is defined and maps to **hi**.
  - And there are only such pairings.

60

**type**52.  $ND' = HI \xrightarrow{m'} (LI \xrightarrow{m'} HI)$ 52.  $ND = \{ |nd':ND \bullet wf\_ND(nd') | \}$ **value**53.  $wf\_ND: ND' \rightarrow \mathbf{Bool}$ 53.  $wf\_ND(nd) \equiv$ 53a.  $\forall hi:HI \bullet hi \in \mathbf{dom} \ nd \Rightarrow$ 53b.  $\mathbf{let} \ lihi = nd(hi) \ \mathbf{in}$ 53c.  $\forall li:LI \bullet li \in \mathbf{dom} \ lihi \Rightarrow$ 53c.  $\mathbf{let} \ hi' = (nd(hi))(li) \ \mathbf{in}$ 53d.  $hi \neq hi' \wedge$ 53e.  $hi' \in \mathbf{dom} \ nd \wedge li \in \mathbf{dom} \ (nd(hi')) \wedge hi = (nd(hi'))(li)$ 53f.  $\mathbf{end} \ \mathbf{end}$ 

61

From a net one can construct its net descriptor:

**value** $conND: N \rightarrow ND$  $conND(n) \equiv$ 

$$[ hi \mapsto [ li \mapsto hi' | li:LI, hi':HI \bullet li \in \mathit{obs\_LIs}(getH(hi,n)) \wedge \{hi, hi'\} = \mathit{obs\_HIs}(getL(li,n)) ] |$$

$$hi:HI \bullet hi \in \mathit{xtrHIs}(n) ]$$

### 3.2.2 Link States and Link State Spaces

62

Links are (one of the) means of transport<sup>6</sup>. Hubs allow movement along one (hub-connected) link to be diverted onto another (hub-connected) link.

<sup>6</sup>Other means are vehicles moving along links and crossing hubs and the locomotive force that drives the vehicles. Freight, including people, are what is being transported.

We introduce the notions of the state of a link, the state of a hub, the state space of a link and the state space of a hub. States abstract directions of movement.

Links are, by our previous definitions, bi-directional: from one of the connected hubs to the other, and vice versa. And hubs are multi-directional: from potentially any link via the hub to potentially any link.

Let the observed hub identifiers of a link  $\ell$  be  $\{h_j, h_k\}$ , then link  $\ell$  can potentially be in any one of the four link states:  $\{(h_j, h_k), (h_k, h_j)\}$ ,  $\{(h_j, h_k)\}$ ,  $\{(h_k, h_j)\}$  and  $\{\{\}\}$ . Any one particular link may always remain in one and the same state, or it may from time to time undergo transitions between any subset of the potential link state space.

54. Link states,  $l\sigma:L\Sigma$ , are set of pairs of hub identifiers.

55. Link state spaces are set of link states.

56. From a link one can generate the link state space of all potential link states.

57. From a link one can observe the current link state  $l\sigma:L\Sigma$ .

58. From a link one can observe the link state space  $l\omega:L\Omega$ .

#### type

54.  $L\Sigma = (\text{HI} \times \text{HI})\text{-set}$

55.  $L\Omega = L\Sigma\text{-set}$

#### value

56.  $\text{generate\_full\_L}\Sigma: L \rightarrow L\Sigma$

56.  $\text{generate\_full\_L}\Sigma(l) \equiv$

56.  $\{\}\cup\{(hi',hi'')\mid hi',hi'':\text{HI}\cdot hi' \neq hi'' \wedge \{hi',hi''\} = \text{obs\_HIs}(l)\}$

56.  $\text{generate\_L}\Omega: L \rightarrow L\Omega$

56. **let** fullL $\sigma = \text{generate\_full\_L}\Sigma(l)$  **in**

56.  $\{\{\}, \cup\{\sigma \mid \sigma:L\Sigma \cdot \sigma \subseteq \text{fullL}\sigma\}\}$  **end**

57.  $\text{obs\_L}\Sigma: L \rightarrow L\Sigma$

58.  $\text{obs\_L}\Omega: L \rightarrow L\Sigma\text{-set}$

### 3.2.3 Hub States and Hub State Spaces

66

59. Hub states,  $h\sigma:H\Sigma$ , are sets of pairs of link identifiers  $((l_i, l_k))$ , designating that if  $(l_i, l_k)$  is in the current hub state then movement can take place from the link designated by  $l_i$  (via hub  $h$ ) to the link designated by  $l_k$ .

60. Hub state spaces are set of hub states.

61. From a hub one can generate the hub state space of all potential hub states.

62. From a hub one can observe the current hub state  $h\sigma:H\Sigma$ .

63. From a hub one can observe the hub state space  $h\omega:H\Omega$ .

67

**type**

59.  $H\Sigma = (LI \times LI)\text{-set}$

60.  $H\Omega = H\Sigma\text{-set}$

**value**

61.  $\text{generate\_full\_H}\Sigma: H \rightarrow H\Sigma$

61.  $\text{generate\_full\_H}\Sigma(h) \equiv$

61.  $\{\} \cup \{(li', li'') \mid li', li'': LI \bullet \{li', li''\} \subseteq \text{obs\_LIs}(h)\}$

56.  $\text{generate\_H}\Omega: H \rightarrow H\Omega$

56. **let**  $\text{fullH}\sigma = \text{generate\_full\_H}\Sigma(h)$  **in**

56.  $\{\} \cup \{\sigma \mid \sigma: H\Sigma \bullet \sigma \subseteq \text{fullH}\sigma\}$  **end**

62.  $\text{obs\_H}\Sigma: H \rightarrow H\Sigma$

62.  $\text{obs\_H}\Omega: H \rightarrow H\Sigma\text{-set}$

**3.2.4 State and State Space Wellformedness**

68

64. States must be in appropriate state spaces.

65. State spaces must be subsets of all potential appropriate states.

**axiom**

$\forall n:N, l:L, h:H \bullet l \in \text{obs\_Ls}(n) \wedge h \in \text{obs\_Hs}(n) \Rightarrow$

54.  $\text{obs\_L}\Sigma(l) \in \text{obs\_L}\Omega(l) \wedge$

55.  $\text{obs\_L}\Omega(l) \subseteq \text{generate\_full\_L}\Sigma(l) \wedge$

54.  $\text{obs\_H}\Sigma(h) \in \text{obs\_H}\Omega(h) \wedge$

55.  $\text{obs\_H}\Omega(h) \subseteq \text{generate\_full\_H}\Sigma(h)$

**theorems:**

$\forall n:N, l:L, h:H \bullet l \in \text{obs\_Ls}(n) \wedge h \in \text{obs\_Hs}(n) \Rightarrow$

$\text{obs\_L}\Sigma(l) \subseteq \{(hi', hi'') \mid hi', hi'': H \bullet \{hi', hi''\} \subseteq \text{obs\_HIs}(l)\} \wedge$

$\text{obs\_H}\Sigma(h) \subseteq \{(li', li'') \mid li', li'': L \bullet \{li', li''\} \subseteq \text{obs\_LIs}(h)\}$

### 3.2.5 Concrete Types for Simple Entities

69

As an alternative for, or as a step of refinement from the earlier sorts of nets, hubs and links one can simplify matters by concrete types for these simple entities.

- 66. Nets are Cartesians of sets of hubs and links.
- 67. A link is a Cartesian of a link identifier, a set of exactly two hub identifiers, a link state, a link state space, and a number of presently further unspecified link attributes.
- 68. A hub is a Cartesian of a hub identifier, a set of zero, one or more link identifiers, a hub state, a hub state space, and a number of presently further unspecified hub attributes.

#### type

- 66.  $N = \mathbf{H\text{-set}} \times \mathbf{L\text{-set}}$
- 67.  $L :: \text{obs\_LI:LI} \times \text{obs\_HIs:HI-set} \times L\Sigma \times L\Omega \times L\text{Atrs}$
- 68.  $H :: \text{obs\_HI:HI} \times \text{obs\_LIs:LI-set} \times H\Sigma \times H\Omega \times H\text{Atrs}$

We leave it to the reader to narrate the wellformedness constraints.

#### axiom

- $\forall (hs,ls):N \cdot ls \neq \{\} \Rightarrow \mathbf{card} \ hs \geq 2 \wedge$
- $\forall l',l'':L \cdot \{l',l''\} \subseteq ls \wedge l' \neq l'' \Rightarrow \text{obs\_LI}(l') \neq \text{obs\_LI}(l'') \wedge$
- $\forall h',h'':H \cdot \{h',h''\} \subseteq hs \wedge h' \neq h'' \Rightarrow \text{obs\_HI}(h') \neq \text{obs\_HI}(h'') \wedge$
- $\forall l:(li,his,l\sigma,l\omega,l\text{atrs}):L \cdot l \in ls \Rightarrow$
- $\quad \mathbf{card} \ his = 2 \wedge his \subseteq \{\text{obs\_HI}(h'') | h'' : H \cdot h'' \in hs\} \wedge$
- $\quad l\sigma \in \text{generate\_full\_L}\Sigma(l) \wedge$
- $\quad l\omega \in \omega \subseteq \text{generate\_full\_L}\Sigma(l) \wedge$
- $\forall h:(hi,lis,h\sigma,h\omega,h\text{atrs}):H \cdot h \in hs \Rightarrow$
- $\quad lis \subseteq \{\text{obs\_LI}(l''') | l''' : L \cdot l''' \in ls\} \wedge$
- $\quad h\sigma \in \text{generate\_full\_H}\Sigma(h) \wedge$
- $\quad h\omega \in \omega \subseteq \text{generate\_full\_H}\Sigma(h)$

### 3.2.6 Example Hub Crossings

72

Figure 1 shows four hub/partial link corner diagrams (1.–4.). These are intended to show four distinct hub states. Let the center diagram (5.) of Fig. 1 indicate the link identifiers of the four partial links of each of the four hub/partial link diagrams.

The top left hub/link diagram (1.) thus can be claimed to depict hub state  $\{(A, B), (A, C), (A, D), (B, C), (C, D), (D, A)\}$ .

Photo 2 on page 32 shows a semaphore which seems to be able to display all kinds of states.

The point of this example is to show that a hub may take on many states, that not all hub states may be desirable (viz., lead to crossing traffic if so interpreted), and that to reach from one hub state to another one must change the state.

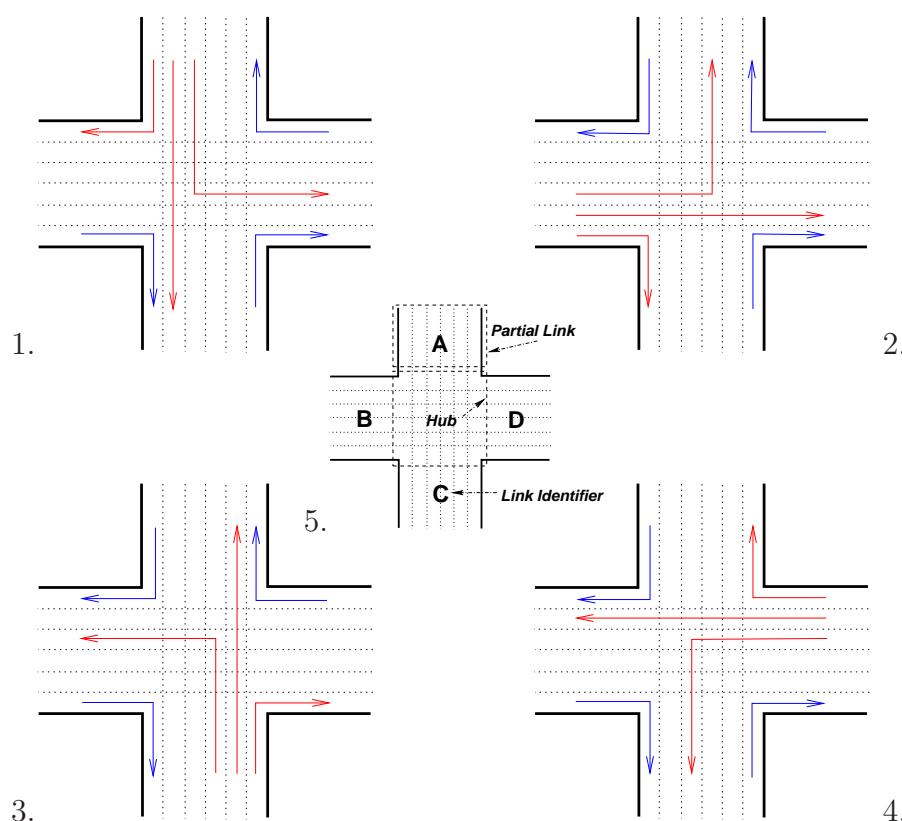


Figure 1: Four “Safe” Flows

### 3.2.7 Actions Continued

74

69. The action  $\text{change\_H}\Sigma$  takes a hub,  $h$ , in some state, and a desired next state,  $h\sigma'$ , and results in a hub,  $h'$ , which

- a) has the same hub identifier as  $h$ ,  
is connected to the same links as  $h$ ,  
has the same hub state space as  $h$ ,  
has the same attributes (names and values) as  $h$ ,
- b) but whose state may have changed.

69b. The new state of  $h'$  ought be  $h\sigma'$ , but electro-mechanical or other failures in setting the state may set the new state to any state of the potential states of  $h$  (i.e.,  $h'$ ), not just to any state in the hub state space of  $h$ .

75

value

69.  $\text{change\_H}\Sigma: H \times H\Sigma \rightarrow H$



Figure 2: A General Purpose Traffic Light

69.  $\text{change\_H}\Sigma((\text{hi}, \text{lis}, \text{h}\sigma, \text{h}\omega, \text{hatrs}), \text{h}\sigma') \equiv$   
 69b. **let**  $\text{h}\sigma''' \in \text{generate\_full\_H}\Sigma\text{s}$  **in**  
 69a.  $(\text{hi}, \text{lis}, \text{h}\sigma''', \text{h}\omega, \text{hatrs})$  **end**

Had we specified that the resulting state must be  $\text{h}\sigma'$  then we had prescribed a requirements to a change operation. As it is now we have described a domain phenomenon, namely that operations may fail.

### 3.3 Support Technologies

76

**Definition: Support Technology.** *By a support technology we understand a facet<sup>[285]</sup> of a domain<sup>[239]</sup>, one which reflects its (current) dependency on human, mechanical, electro-mechanical, electronic and/or other technologies (i.e., tools) in order to carry out its business process<sup>[99]</sup>es.*

#### 3.3.1 Traffic Signals

77

A traffic signal represents a technology in support of visualising hub states and in effecting state changes.

70. A hub state is now modelled as a triple: the link identifier  $l_i$  (“coming from”), a colour (**red**, **yellow**, and **green**), and another the link identifier  $l_j$  (“going to”).

71. Signalling is now a sequence of one or more pairs of next hub states and time intervals:

$$\langle (h\sigma_1, ti_1), (h\sigma_2, ti_2), \dots, (h\sigma_{n-1}, ti_{n-1}), (h\sigma_n, ti_n) \rangle, n > 0$$

78

The idea of a signalling is to first change the designated hub to state  $h\sigma_1$ , then wait  $ti_1$  time units, then set the designated hub to state  $h\sigma_2$ , then wait  $ti_2$  time units, etcetera, ending with final state  $\sigma_n$  and a (supposedly) long time interval  $ti_n$  before any decisions are to be made as to another signalling.

The set of hub states  $\{h\sigma_1, h\sigma_2, \dots, h\sigma_{n-1}\}$  of

$$\langle (h\sigma_1, ti_1), (h\sigma_2, ti_2), \dots, (h\sigma_{n-1}, ti_{n-1}), (h\sigma_n, ti_n) \rangle, n > 0$$

are called intermediate states.

Their purpose is to secure an orderly phase out of green via yellow to red and phase in of red via yellow to green in some order for the various directions.

We leave it to the reader to devise proper wellformedness conditions for signaling sequences as they depend on the hub topology.

79

72. A street signal (a semaphore) is now abstracted as a map from pairs of hub states to signalling sequences.

The idea is that given a hub one can observe its semaphore, and given the state,  $h\sigma$  (not in the above set), of the hub “to be signalled” and the state  $h\sigma_n$  into which that hub is to be signalled “one looks up” under that pair in the semaphore and obtains the desired signalling.

### type

70.  $H\Sigma = LI \times \text{Colour} \times LI$

70.  $\text{Colour} == \text{red} \mid \text{yellow} \mid \text{green}$

71.  $\text{Signalling} = (H\Sigma \times \text{TI})^*$

71.  $\text{TI}$

72.  $\text{Semaphore} = (H\Sigma \times H\Sigma) \xrightarrow{m} \text{Signalling}$

### value

72.  $\text{obs\_Semaphore}: H \rightarrow \text{Semaphore}$

80

73. A hub semaphore, **sema**, contains only such hub states as are observed in the hub state space.

a) Let **hsps** be the set of “from/to” hub state pairs in semaphore **sema**.

b) Then **hs** is the set of all hub states mentioned in **hsps**.

c) To **hs** join all the hub states mentioned in any signalling, **sg**, of **sema**.



```

73. hub_state_space: Sempahore  $\rightarrow$   $H\Sigma$ -set
73. hub_state_space(sema)  $\equiv$ 
73a. let hsps= $\{hsp|hsp:(H\Sigma \times H\Sigma) \cdot hsp \in \mathbf{dom\ sema}\}$  in
73b. let hs= $\{h\sigma', h\sigma'' | h\sigma', h\sigma'': H\Sigma \cdot (h\sigma', h\sigma'') \in \text{hsps}\}$  in
73c.  $hs \cup \cup \{\{h\sigma | (h\sigma, ti) : (H\Sigma \times TI) \cdot (h\sigma, ti) \in \mathbf{elems\ sg}\} | sg : \text{Signalling} \cdot sg \in \mathbf{rng\ sema}\}$ 
73. end end
axiom
73.  $\forall h : H \cdot \cup \text{obs\_H}\Omega(h) = \text{hub\_state\_space}(\text{obs\_Semaphore}(h))$ 

```

### 3.3.2 Traffic “Control”

81

74. Given two hub states,  $h\sigma_{\text{init}}$  and  $h\sigma_{\text{end}}$ , where  $h\sigma_{\text{init}}$  designates a present hub state and  $h\sigma_{\text{end}}$  designates a desired next hub state after signalling.
75. Now signalling is a sequence of one or more successful hub state changes.

**value**

```

74. signalling:  $H\Sigma \times H\Sigma \rightarrow H \rightarrow H$ 
75. signalling( $h\sigma_{\text{init}}, h\sigma_{\text{end}}$ )(h)  $\equiv$ 
75. let sema = obs_Semaphore(h) in
75. let sg = sema( $h\sigma_{\text{init}}, h\sigma_{\text{end}}$ ) in
75. signal_sequence(sg)(h) end end
75. pre ( $h\sigma_{\text{init}}, h\sigma_{\text{end}} \in \mathbf{dom\ obs\_Semaphore}(h)$ )

```

```

75. signal_sequence( $\langle \rangle$ )(h)  $\equiv$  h
75. signal_sequence( $\langle (h\sigma, ti) \rangle^{\wedge} sg$ )(h)  $\equiv$ 
75. let  $h\sigma' = \text{change\_H}\Sigma(h)(h\sigma)$  in
75. if  $h\sigma' \neq h\sigma$  then chaos
75. else wait(ti); signal_sequence(sg)(h) end end

```

If a desired hub state change fails (**chaos**) then we do not define the outcome of signalling.

## 3.4 Rules and Regulations

83

**Definition: Rule.** A rule stipulates a regulating principle. In the context of modelling domain rules we shall understand a domain rule as some text whose meaning is a predicate<sup>[536]</sup> over a pair of suitably chosen domain state<sup>[705]</sup>s. We may assume that a domain action<sup>[12]</sup> or a domain event<sup>[281]</sup> takes place in the first of these states and results in the second of these states. If the predicate is true then we say that the rule has been obeyed, otherwise that it has been violated.

Usually a domain rule is paired with a possibly remedying regulation.

**Definition: Regulation.** A regulation stipulates that an *action*<sup>[12]</sup> be taken in order to remedy a previous action which violated a *rule*<sup>[638]</sup>. That is, a regulation is some *text* which designates a possibly composite *action*<sup>[12]</sup>, that is, a *state-to-state change* which ostensibly results in a state in which the *rule*, “attached” to the regulation, now holds.

### 3.4.1 Vehicles

85

In preparation for examples of transportation rules and regulations we introduce vehicles.

76. Vehicles are further undefined quantities except that

- a) vehicles have unique identifiers,
- b) vehicles are either positioned
  - i. at/in hubs
  - ii. or on links, in some fractional (non-zero) distance from a hub toward the connecting hub.

77. From a net (sort) one can observe all the vehicles of the net.<sup>7</sup>

78. No two vehicles so observed have the same identifier.

86

#### type

76. V

76a. VI

76b. VP = HP | LP

76(b)i. HP == atH(hi:HI)

76(b)ii. LP == onL(li:LI,fhi:HI,f:F,thi:HI)

76(b)ii. F = {f:F•0<f<1}

#### value

76a. obs\_VI: V → VI

76b. obs\_VP: V → VP

77. obs\_Vs: N → V-set

#### axiom

78.  $\forall v:V \cdot v \in \text{obs\_Vs}(n) \Rightarrow$

78.  $\exists \text{onL}(\text{li}, \text{fhi}, \text{f}, \text{thi}): \text{VP} \cdot \text{onL}(\text{li}, \text{fhi}, \text{f}, \text{thi}) = \text{obs\_VP}(v) \Rightarrow$

78.  $\exists l:L \cdot l \in \text{obs\_Ls}(n) \wedge \text{li} = \text{obs\_LI}(l) \wedge \{\text{fhi}, \text{thi}\} = \text{obs\_HIs}(l) \vee$

78.  $\exists \text{atH}(\text{hi}): \text{VP} \cdot \text{atH}(\text{hi}) = \text{obs\_VP}(v) \Rightarrow$

78.  $\exists h:H \cdot h \in \text{obs\_Hs}(n) \wedge \text{hi} = \text{obs\_HI}(h)$

MORE TO COME

<sup>7</sup>Thus a concrete net type, in addition to hubs and links (now) also contains vehicles.

### 3.4.2 Traffic

87

79. By traffic we understand a continuous function from time to a pair of nets and position of vehicles.
80. By time we understand a dense set of points with dense and points being mathematical concepts [57, 222].

**type**

79.  $\text{TF} = \text{T} \rightarrow (\text{sel\_net}:\text{N} \times \text{sel\_veh\_pos}:(\text{V} \xrightarrow{m} \text{VP}))$

80.  $\text{T}$

#### Wellformedness of Traffic

Expressing the wellformedness of traffic is not a simple matter. We shall approach this task in a number of “small steps”.

#### • Static Wellformedness

81. We define a predicate over vehicle positions.
- a) Every vehicle in the traffic has a proper position on the net, either at a hub or along a link.
  - b) No two vehicles of the traffic can occupy exactly the same link position. (That is, the link positions  $\text{onL}(\text{li}, \text{hi}, f, \text{hi}')$  and  $\text{onL}(\text{li}, \text{hi}, f', \text{hi}')$  must have the two fractions  $(f, f')$  differ – be it ever so “minutely”).

We first define two auxiliary functions:<sup>8</sup>

**value**

$\text{obs\_HIs}: \text{N} \rightarrow \text{HI-set}$

$\text{obs\_HIs}(n) \equiv \{\text{obs\_HI}(h) \mid h:\text{H} \bullet h \in \text{obs\_Hs}(n)\}$

$\text{obs\_LIs}: \text{N} \rightarrow \text{LI-set}$

$\text{obs\_LIs}(n) \equiv \{\text{obs\_LI}(h) \mid l:\text{L} \bullet l \in \text{obs\_Ls}(n)\}$

81.  $\text{proper\_vehicle\_positions}: \text{TF} \rightarrow \text{Bool}$

81.  $\text{proper\_vehicle\_positions}(\text{tf}) \equiv$

81.  $\forall t:\text{T} \bullet t \in \text{DOMAIN } \text{tf} \bullet$

81. **let**  $(n, \text{vps}) = \text{tf}(t)$  **in**

81a.  $\forall v:\text{V} \bullet v \in \text{dom } \text{vp} \bullet \text{is\_net\_position}(\text{vps}(v))(n)$

81b.  $\forall v':\text{V} \bullet v' \in \text{dom } \text{vp} \wedge v \neq v' \Rightarrow \text{diff\_net\_pos}(\text{vps}(v), \text{vps}(v'))$

81. **end**

<sup>8</sup>They really ought to have been defined much earlier!

81a.  $\text{is\_net\_position}: \text{VP} \rightarrow \text{N} \rightarrow \mathbf{Bool}$   
81a.  $\text{is\_net\_position}(\text{vp})(\text{n}) \equiv$   
81a.   **case**  $\text{vp}$  **of**  
81a.      $\text{atH}(\text{hi}) \rightarrow \text{hi} \in \text{obs\_HIs}(\text{n}),$   
81a.      $\text{onL}(\text{li}, \text{fhi}, \text{f}, \text{thi}) \rightarrow \text{li} \in \text{obs\_LIs}(\text{n}) \wedge \{\text{fhi}, \text{thi}\} \subseteq \text{obs\_HIs}(\text{n})$   
81a.   **end**  
81b.  $\text{diff\_net\_pos}: \text{VP} \times \text{VP} \rightarrow \mathbf{Bool}$   
81b.  $\text{diff\_net\_pos}(\text{vp}, \text{vp}') \equiv$   
81b.   **case**  $(\text{vp}, \text{vp}')$  **of**  
81b.      $(\text{atH}(\text{hi}), \text{atH}(\text{hi})) \rightarrow \mathbf{true},$   
81b.      $(\text{onL}(\text{li}, \text{fhi}, \text{f}, \text{thi}), \text{onL}(\text{li}, \text{fhi}, \text{f}', \text{thi})) \rightarrow \text{f} \neq \text{f}',$   
81b.      $\_ \rightarrow \mathbf{true}$   
81b.   **end**

90

### • Dynamic Wellformedness

82. Vehicles, when moving, move monotonically, that is,

- a) if a vehicle, at some time,  $t$ , is at a link position  $\text{onL}(\text{li}, \text{hi}, \text{f}, \text{hi}')$  where  $\text{f}$  is not infinitesimally close to 1, then that vehicle will, at some later time  $t'$ , infinitesimally close to  $t$ , be at link position  $\text{onL}(\text{li}, \text{hi}, \text{f}', \text{hi}')$  where  $\text{f}'$  is infinitesimally close to  $\text{f}$ ;
- b) if the vehicle, at some time,  $t$ , is at a link position  $\text{onL}(\text{li}, \text{hi}, \text{f}, \text{hi}')$  where  $\text{f}$  is indeed infinitesimally close to 1, then that vehicle will, at some infinitesimally later time  $t'$ , be at hub position  $\text{atH}(\text{hi}')$ ;
- c) and if the vehicle, at some time,  $t$ , is at a hub position  $\text{atHP}(\text{hi})$  then the vehicle will at some infinitesimally later time  $t'$  either be at hub position  $\text{atHP}(\text{hi})$  or at some link position  $\text{onL}(\text{li}, \text{hi}, \text{f}, \text{hi}')$  where  $\text{f}$  is infinitesimally close to 0.

91

### value

82.  $\text{monotonic}: \text{TF} \rightarrow \mathbf{Bool}$   
82.  $\text{monotonic}(\text{tf}) \equiv$   
82.    $\forall t, t': \text{T} \bullet \{t, t'\} \subseteq \text{DOMAIN } \text{tf} \bullet$   
82.   **let**  $(\text{n}, \text{vps}) = \text{tf}(t), (\text{n}', \text{vps}') = \text{tf}(t')$  **in**  
82.    $\text{INFINITESIMALLY CLOSE } (t, t') \wedge t < t' \Rightarrow$   
82.    $\forall v: \text{V} \bullet v \in \text{dom } \text{vps} \cap \text{dom } \text{vps}' \bullet$   
82.   **case**  $(\text{vps}(v), \text{vps}'(v))$  **of**  
82a.      $(\text{onL}(\text{li}, \text{fhi}, \text{f}, \text{thi}), \text{onL}(\text{li}, \text{fhi}, \text{f}', \text{thi})) \rightarrow$   
82a.          $\text{f} < \text{f}' \wedge \text{INFINITESIMALLY CLOSE } (\text{f}, \text{f}')$ ,  
82b.      $(\text{onL}(\text{li}, \text{fhi}, \text{f}, \text{thi}), \text{atH}(\text{thi})) \rightarrow$   
82b.          $\text{INFINITESIMALLY CLOSE } (\text{f}, 1),$

```

82c.      (atH(hi),atH(hi)) → true,
82c.      (atH(hi),onL(li,hi,f,thi)) →
82c.      INFINITESIMALLY CLOSE (0,f),
82.      _ → true
82.      end end

```

83. If a vehicle is (has been) moving along a link  $l_i$  and is now,

- at time  $t$ , at position  $\text{onL}(l_i, h_j, f, h_k)$ , that is, moving from  $h_j$  to  $h_k$ ,
- then it cannot at a subsequent, infinitesimally close time,  $t'$ , be at a position
- $\text{onL}(l_i, h_k, f', h_j)$ , that is, moving in the opposite direction,  $h_k$  to  $h_j$ .

**value**

```

83. God_does_not_play_dice9: TF → Bool
83. God_does_not_play_dice(tf) ≡
83.  ∀ t,t':T • {t,t'} ⊆ DOMAIN tf ∧ t < t' ∧ INFINITESIMALLY CLOSE (t,t') ⇒
83.    let (n,vps) = tf(t),(n',vps')=tf(t') in
83.      ∀ v:V • v ∈ dom vps ∩ dom vps' ⇒
83.        case (vps(v),vps'(v)) of
83.          (onL(li,fhi,_,thi),onL(li,thi,_,fhi)) → false,
83.          _ → true
83.        end end

```

84. If a vehicle is (has been) moving along and has,

- at time  $t$ , been at some position  $p$ , and
- at time  $t'$ , later than  $t$ , is at some position  $p'$ ,
- then it must at all times  $t''$  between  $t$  and  $t'$  have been somewhere on the net.

**value**

```

84. no_ghost_vehicles: TF → Bool
84. no_ghost_vehicles(tf) ≡
84.  ∀ t,t':T • {t,t'} ⊆ DOMAIN tf ∧ t < t' ⇒
84.    let (n,vps) = tf(t),(n',vps')=tf(t') in
84.      ∀ v:V • v ∈ dom vps ∩ dom vps' ⇒
84.        ∀ t'':T • t < t'' < t' ⇒
84.          let (n'',vps'') = tf(t'') in v ∈ dom vps'' end
84.    end

```

---

<sup>9</sup>Albert Einstein: "I, at any rate, am convinced that He does not throw dice." Letter to Max Born (4 December 1926); The Born-Einstein Letters (translated by Irene Born) (Walker and Company, New York, 1971) ISBN 0-8027-0326-7. Reflects Einstein's view of Quantum Mechanics at the time.

### 3.4.3 Traffic Rules (I of II)

95

85. A vehicle must not move from a hub,  $h_i$ , into a link  $\ell$  (from hub (identified by)  $h_i$  to hub (identified by)  $h_j$ ) which is closed in direction  $(h_i, h_j)$ , that is, where  $(h_i, h_j)$  is not in the current state of link.

**rule:**

```

85.  $\forall \text{tf}:\text{TF}, \text{t}:\text{T} \cdot \text{t} \in \text{DOMAIN}(\text{tf}) \Rightarrow$ 
85.   let (n,tp) = tf(t) in
85.    $\forall \text{v}:\text{V} \cdot \text{v} \in \text{dom tp} \Rightarrow$ 
85.     case tp(v) of
85.       atH(hi)  $\rightarrow$ 
85.         let  $\text{t}':\text{T} \cdot \text{t}' > \text{t} \wedge \text{t}' \in \text{DOMAIN}(\text{tr}') \wedge \text{INFINITESIMALLY\_CLOSE}(\text{t}, \text{t}') \text{ in}$ 
85.         let (n',tp') = tf(t') in
85.          $\exists \text{li}:\text{LI}, \text{hi}':\text{HI}, \text{f}:\text{F}, \text{hi}'':\text{HI} \cdot$ 
85.            $\text{hi}' = \text{hi} \wedge \text{INFINITESIMALLY\_CLOSE}(\text{f}, 0) \wedge$ 
85.            $\text{tp}'(\text{v}) = \text{onL}(\text{li}, \text{hi}', \text{f}, \text{hi}'') \wedge (\text{hi}, \text{hi}'') \notin \text{obs\_L}\Sigma(\text{getL}(\text{li}, \text{n}'))$ 
85.          $\_ \rightarrow \dots$ 
85.     end end end end

```

We shall give another rule after the next section.

### 3.4.4 Another Traffic Regulator

96

We present an abstraction of a more conventional traffic signal than modelled in Items 70 on page 32 to 73 on page 33.

86. A traffic signal now simply shows an entry permit: either **red**, **yellow** or **green** at the hub when “leaving” any link, i.e., at the entry to a hub from any link.

**type**

86.  $\text{EP} == \text{red} \mid \text{yellow} \mid \text{green}$

86.  $\text{H}\Sigma = \text{LI} \xrightarrow{m} \text{EP}$

**axiom**

86.  $\forall \text{h}:\text{H} \cdot \text{obs\_LLs}(\text{h}) = \text{dom obs\_H}\Sigma(\text{h})$

We leave it to the reader to express a constraint over hub state spaces as to how there must be hub states such that entry from any link is possible.

### 3.4.5 Traffic Rules (II of II)

97

87. Vehicles must not enter a hub if entry permission is not **green**.

**rule:**

```

87.  $\forall \text{tf}:\text{TF}, \text{t}:\text{T} : \text{t} \in \text{DOMAIN}(\text{tf}) \Rightarrow$ 
87.   let (n,vps) = tf(t) in
87.    $\forall \text{v}:\text{V} \bullet \text{v} \in \text{dom vps} \Rightarrow$ 
87.     case vps(v) of
87.       onL(li,hi,f,hi')  $\rightarrow$ 
87.         INFINITESIMALLY_CLOSE(f,1)  $\wedge$ 
87.         let h $\sigma$  = obs_H $\Sigma$ (getH(hi',n)),
87.           t':T  $\bullet$  t' > t  $\wedge$  INFINITESIMALLY_CLOSE(t,t') in
87.           let (n',vps') = vps(t') in
87.             h $\sigma$ (li)  $\neq$  green  $\wedge$  vps'(v)  $\neq$  atH(hi') assert: vps'(v) = onL(li,hi,f,hi')
87.           end end
87.       _  $\rightarrow$  ...
87.   end end

```

## 3.5 Scripts

98

**Definition: Scripts.** A script is plan of action. By a domain script we shall, more specifically, understand the structured, almost, if not outright, formally expressed, wording of a set of rules and regulations<sup>[640]</sup>.

See also *license*<sup>[424]</sup> and *contract*<sup>[181]</sup>. Definitions follow.

### 3.5.1 Routes as Scripts

99

#### Paths

88. A path is a triple:

- a) a hub identifier,  $h_i$ , a link identifier,  $l_j$ , and another hub identifier,  $h_k$ , distinct from  $h_i$ ,
- b) such that there is a link  $\ell$  with identifier  $l_j$  in a net  $n$  such that  $\{h_i, h_k\}$  are the hub identifiers that can be observed from  $\ell$ .

**type**

88. Pth = HI  $\times$  LI  $\times$  HI

**axiom**

88a.  $\forall (\text{hi}, \text{li}, \text{hi}'):\text{Pth} \bullet \exists \text{n}:\text{N}, \text{l}:\text{L} \bullet \text{l} \in \text{obs\_Ls}(\text{n}) \Rightarrow$

88b.  $\text{obs\_LI}(\text{l}) = \text{li} \wedge \text{obs\_HIs}(\text{l}) = \{\text{hi}, \text{hi}'\}$

89. From a net one can extract all its paths:

- a) if  $l$  is a link of the net,

- b)  $l_j$  its identifier,
- c)  $\{h_i, h_k\}$  the identifiers of its connected hubs,
- d) then  $(h_i, l_j, h_k)$  and  $(h_k, l_j, h_i)$  are paths of the net.

**value**

89. paths:  $N \rightarrow \text{Pth-set}$

89a.  $\text{paths}(n) \equiv$

89d.  $\{(hi, lj, hk), (hk, lj, hi) \mid l:LI, lj:LI, hi, hk:HI \bullet l \in \text{obs\_Ls}(n) \wedge$

89b.  $lj = \text{obs\_LI}(l) \wedge$

89c.  $\{hi, hk\} = \text{obs\_HIs}(l)\}$

101

90. From a net descriptor one can (likewise) extract all its paths:

- a) Let  $h_i, h_k$  be any two distinct hub identifiers of the net descriptor (definition set),
- b) such that they both map into a link identifier  $l_j$ ,
- c) then  $(h_i, l_j, h_k)$  and  $(h_k, l_j, h_i)$  are paths of the net.

**value**

89. paths:  $ND \rightarrow \text{Pth-set}$

89.  $\text{paths}(nd) \equiv$

90a.  $\{(hi, lj, hk), (hk, lj, hi) \mid hi, hk:HI, lj:LI \bullet hi \neq hk \wedge \{hi, hk\} \subseteq \text{dom } nd \Rightarrow$

90b.  $lj \in \text{dom } nd(hi) \cap \text{dom } nd(hk)\}$

**Routes**

102

91. A route of a net is a sequence of zero, one or more paths such that

- a) all paths of a route are paths of the net and
- b) adjacent paths in the sequence “share” hub identifiers.

**type**

91.  $R = \text{Pth}^*$

**axiom**

91.  $\forall r:R, \exists n:N \bullet$

91a.  $\text{elems } r \subseteq \text{paths}(n) \wedge$

91b.  $\forall i:\text{Nat} \bullet \{i, i+1\} \subseteq \text{inds } r \Rightarrow$

91b.  $\text{let } (\_, \_, hi) = r(i), (hi', \_, \_) = r(i+1) \text{ in } hi = hi' \text{ end}$

103



92. From a net,  $n$ , we can generate the possibly infinite set of finite and possibly infinite routes:

- a)  $\langle \rangle$  is a path (**basis clause 1**);
- b) if  $p$  is a path of  $n$  then  $\langle p \rangle$  is a path of  $n$  (**basis clause 2**);
- c) if  $r$  and  $r'$  are non-empty routes of  $n$ 
  - i. and the last  $h_i$  of  $r$  is the same as the first  $h_j$  of  $r'$
  - ii. then the concatenation of  $r$  and  $r'$  is a route**(induction clause)**.
- d) Only such routes which can be formed by a (finite, respectively infinite) application of basis clauses Items 92a and 92b and induction clause Item 92c are routes (**extremal clause**).

**value**

92. routes:  $N|ND \rightarrow R\text{-infset}$

92. routes(nond)  $\equiv$

92a. **let**  $rs = \{\langle \rangle\} \cup$

92b.  $\{\langle p \rangle \mid p: Pth \bullet p \in paths(nond)\} \cup$

92(c)ii.  $\{r \hat{\ } r' \mid r, r': R \bullet r \in rs \wedge r' \in rs \wedge$

92(c)i.  $\exists h_i, h_i', h_i'', h_i''': H, li: LI \bullet$

92(c)i.  $r = r'' \hat{\ } \langle (h_i, li, h_i') \rangle \wedge r' = \langle (h_i'', li', h_i''') \rangle \hat{\ } r''' \wedge$

92(c)i.  $h_i' = h_i''\}$  **in**

92d.  $rs$  **end**

### 3.5.2 Bus Timetables as Scripts

105

#### Buses

93. Buses are vehicles,

94. with bus identifiers being the same as vehicle identifiers.

**type**

93.  $B$

94.  $BI \subseteq VI$

#### Bus Stops

95. A link bus stop indicates the link (by its identifier), the from and to hub identifiers, and the fraction “down the link” from the from to the to hub identifiers.

**type**

95.  $BS = mkL\_BS(sel\_fhi: HI, sel\_li: LI, sel\_f: F, sel\_thi: HI)$

## Bus Routes

106

96. A bus stop list is a sequence of two or more bus stops,  $bsl$ .
97. A bus route,  $br$ , is a pair of a net route,  $r$ , and a bus stop list,  $bsl$ , such that route  $r$  is a route of  $n$  and such that  $bsl$  is embedded in  $r$ . If
- there exists an index list,  $il$ , of ascending indices of the route  $r$  and of the length of  $bsl$
  - such that the  $i$ th path of  $r$
  - share from and to hub identifiers and link identifier with the  $il(i)$ th bus stop of  $bsl$

then  $bsl$  is embedded in  $r$ .

98. We must allow for two or more stops along a bus route to be adjacent on the same link — in which case the corresponding fractions must likewise be ascending.

107

**value** $n:\mathbb{N}$ **type**96.  $BSL = BS^*$ 97.  $BR = \{(r,bsl):(R \times BSL) \bullet wf\_BR(r,bsl)\}$ **value**97.  $wf\_BR: BR \rightarrow \mathbf{Bool}$ 97.  $wf\_BR(r,bsl) \equiv \exists n:\mathbb{N}, r:R \bullet r \in routes(n) \wedge is\_embedded\_in(r,bsl)$ 97a.  $is\_embedded\_in: BR \rightarrow \mathbf{Bool}$ 97a.  $is\_embedded\_in(r,bsl) \equiv$ 97b.  $\exists il:\mathbf{Nat}^* \bullet len\ il = len\ bsl \wedge inds\ il \subseteq inds\ r \wedge ascending(il) \Rightarrow$ 97c.  $\forall i:\mathbf{Nat} \bullet i \in inds\ il \Rightarrow$ 97c.  $\mathbf{let}\ (hi,lj,hk) = r(il(i)), (hi',lj',f,hk') = bsl(i)\ \mathbf{in}$ 97c.  $hi=hi' \wedge lj=lj' \wedge hk=hk'\ \mathbf{end} \wedge$ 98.  $\forall i:\mathbf{Nat} \bullet \{i,i+1\} \subseteq inds\ il \Rightarrow$ 98.  $\mathbf{let}\ (hi,lj,f,hk)=bsl(i), (hi',lj',f',hk')=bsl(i+1)\ \mathbf{in}$ 98.  $hi=hi' \wedge lj=lj' \wedge hk=hk' \Rightarrow f < f'\ \mathbf{end}$ 

$ascending: \mathbf{Nat}^* \rightarrow \mathbf{Bool}$ ,  $ascending(il) \equiv \forall i:\mathbf{Nat} \bullet \{i,i+1\} \subseteq inds\ il \Rightarrow il(i) \leq il(i+1)$

The  $\leq$  of the  $ascending$  predicate allows for more than one stop along the same route

**Bus Schedule**

108

99. A timed bus stop is a pair of a time and a bus stop.
100. A timed bus stop list is a sequence of timed bus stops.
101. A bus schedule is a pair of a route and a timed bus stop list such that
- there is a net of which the routes is indeed a route,
  - the bus stop list of the timed bus stop list is embedded in the route, and
  - ‘later’ listed bus stops register later times.
102. SimpleBusSchedules remove routes from BusRoutes.

**type**

99. TBS :: sel\_T:T sel\_bs:BS

100. TBSL = TBS\*

101. BusSched =  $\{(r, \text{tbls}) : (R \times \text{TBSL}) \bullet \text{wf\_BusSched}(r, \text{tbls})\}$ **value**101. wf\_BusSched: BusSched  $\rightarrow$  **Bool**101. wf\_BusSched(r, tbls)  $\equiv$ 101.  $\exists n:N \bullet r \in \text{routes}(n)$ 101.  $\wedge \text{let } \text{bsl}:\text{SBS} = \langle \text{sel\_BS}(\text{tbls}(i)) | i:[1..\text{len } \text{tbls}] \rangle \text{ in } \text{is\_embedded\_in}(r, \text{bsl}) \text{ end}$ 101.  $\wedge \forall i:\text{Nat} \bullet \{i, i+1\} \subseteq \text{inds } \text{tbls} \Rightarrow \text{sel\_T}(\text{tbls}(i)) < \text{sel\_T}(\text{tbls}(i+1))$ **type**102. SBS =  $\{|\text{bsl}:\text{BS}^* \bullet \exists n:N, r:R \bullet r \in \text{routes}(n) \wedge \text{is\_embedded\_in}(r, \text{bsl})|\}$ **Timetable**

110

The concept of a bus line captures all those bus schedules which ply the same bus route but at different times. A timetable is made up from distinctly named bus lines.

103. A bus line has a unique bus line name.
104. We say that two bus schedules are the same if they are based on the same route and if they differ only in their times.
105. Each of the different bus routes of a bus line has a unique bus number.
106. A route bus schedule pairs a route with simple bus schedules for each of a number of busses (identified by their bus number).
107. A bus timetable (listing, map) maps bus line names to route bus schedules.
108. A timetable is a pair, a net and a table.

109. A well-formed timetable must satisfy same bus schedules within each bus line  
 110. All bus numbers are distinct across bus lines.

111

**type**103.  $\text{BLN}_m$ **value**104.  $\text{same\_bus\_schedule}: \text{BusSched} \times \text{BusSched} \rightarrow \mathbf{Bool}$ 104.  $\text{same\_bus\_schedule}((r1, \text{btl1}), (r2, \text{btl2})) \equiv$ 104.  $r1 = r2 \wedge \mathbf{len} \text{ btl1} = \mathbf{len} \text{ btl2} \wedge$ 104.  $\langle \text{sel\_BS}(\text{btl1}(i)) | i: [1.. \mathbf{len} \text{ btl1}] \rangle = \langle \text{sel\_BS}(\text{btl2}(i)) | i: [1.. \mathbf{len} \text{ btl2}] \rangle$ **type**105.  $\text{BNo}$ 106.  $\text{RBS} :: \text{sel\_R}:R \text{ sel\_btbl}:(\text{BNo} \xrightarrow{m} \text{SBS})$ 107.  $\text{TBL} = \text{BLN}_m \xrightarrow{m} \text{RBS}$ 108.  $\text{TT}' = \text{ND} \times \text{TBL}$ 109.  $\text{TT} = \{ | \text{tt}: \text{TT}' \bullet \text{wf\_TT}(\text{tt}) | \}$ 

112

**value**109.  $\text{wf\_TT}: \text{TT}' \rightarrow \mathbf{Bool}$ 109.  $\text{wf\_TT}(\_, \text{tbl}) \equiv$ 109.  $\forall \text{bln}: \text{BLN}_m \bullet \text{bln} \in \mathbf{dom} \text{ tbl} \Rightarrow$ 109.  $\forall \text{bno}, \text{bno}': \text{BNo} \bullet \{ \text{bno}, \text{bno}' \} \subseteq \mathbf{dom} \text{ sel\_btbl}(\text{tbl}(\text{bln})) \Rightarrow$ 109.  $\text{same\_bus\_schedule}(\text{sel\_R}(\text{tbl}(\text{bln})), \text{sel\_btbl}(\text{tbl}(\text{bln}))(\text{bno}),$ 109.  $\text{sel\_R}(\text{tbl}(\text{bln})), \text{sel\_btbl}(\text{tbl}(\text{bln}))(\text{bno}')) \wedge$ 110.  $\forall \text{bln}', \text{bln}'': \text{BLN}_m \bullet \{ \text{bln}', \text{bln}'' \} \subseteq \mathbf{dom} \text{ tbl} \wedge \text{bln}' \neq \text{bln}'' \Rightarrow$ 110.  $\mathbf{dom} \text{ sel\_btbl}(\text{tbl}(\text{bln}')) \cap \mathbf{dom} \text{ sel\_btbl}(\text{tbl}(\text{bln}'')) = \{ \}$ **3.5.3 Route and Bus Timetable Denotations**

113

What are routes and bus timetables scripting ?

Routes (list of connected link traversal designations) script that one may transport people or freight along the sequence of designated links.

Bus timetables script (at least) two things: the set of bus traffics on the net which satisfy the bus timetable, and information that potential and actual bus passengers may, within some measure of statistics (and probability), rely upon for their bus transport.

114

Here, we shall not develop the idea of bus timetables denoting certain traffics. Instead we refer to our previously sketched model of traffics (Sect. 3.4.2, Pages 36–39).

Route (designations) and bus timetables script potential and actual route travels, respectively script the dispatch of buses and their travelling.

Bus timetables can also be seen as a form of contracts between the bus operators offering the bus services and potential and actual passengers, with the contract promising timely transport. In the next section, Sect. 3.5.4, we shall sketch a language of bus service contracts and bus service actions implied by such contracts.

### 3.5.4 Licenses and Contracts

115

**Definition: License.** A license is a *script*<sup>[651]</sup> specifically expressing a permission to act; is freedom of action; is a permission granted by competent authority to engage in a business or occupation or in an activity otherwise unlawful; a document, plate, or tag evidencing a license granted; a grant by the holder of a copyright or patent to another of any of the rights embodied in the copyright or patent short of an assignment of all rights.

Licenses appear more to have morally than legally binding power.

**Definition: Contract.** A contract is a *script*<sup>[651]</sup> specifically expressing a legally binding agreement between two or more parties — hence a document describing the conditions of the contract; a contract is business arrangement for the supply of goods or services at fixed prices, times and locations. In software development a contract specifies what is to be developed: (1) a *domain description*<sup>[243]</sup>, (2) a *requirements prescription*<sup>[615]</sup>, or (3) a *software design*<sup>[688]</sup>; or a combination of these (1–2, 2–3, 1–3). A contract further specifies how it might, or must be developed; criteria for acceptance of what has been developed; delivery dates for the developed items; who the “parties” to the contract are: the *client*<sup>[116]</sup> and the *developer*<sup>[227]</sup>, etc.

For a comprehensive treatment of licenses and contracts we refer to [48, Chapter 10, Sect. 10.6 (Pages 309–326) [84]].

We shall illustrate fragments of a language for bus service contracts.

The background for the bus contract language is the following. In many large cities around Europe the city or provincial government secures public transport in the form of bus services operated by many different private companies. Section 3.5.2 illustrated the concept of bus (service) timetables. The bus services implied by such a timetable, for a city area — with surrounding suburbs etc. — need not be implemented by just one company, but can be contracted, by the city government public transport office, to several companies, each taking care of a subset of the timetable. Different bus operators then take care of non-overlapping parts and all take care of the full timetable. It may even be that extra buses need be scheduled, on the fly, in connection with major sports or concert or other events. Bus operators may experience vehicle breakdowns or bus driver shortages and may be forced to subcontract other, even otherwise competing bus operators to “step in” and alleviate the problem.

### Contracts

121 Schematically we may represent a bus contract as follows:

**Contract**  $cn$  between contractee  $ci$  and contractor  $cj$ :

**This contract contracts**  $cj$  **in the period**  $[t, t']$  **to**

**perform the following services with respect to timetable**  $tt$ :

**operate bus lines**  $\{blj_1, blj_2, \dots, blj_n\}$   
**subject to the following occasional exceptions:**  
     **cancellation of bus tours:**  
          $\{(blj_a, \{bno_{a_1}, \dots, bno_{a_m}\}), \dots\}$  **subject to conditions** cbt  
     **insertion of bus tours on lines**  
          $\{blj_\alpha, blj_\beta, \dots, blj_\gamma\}$  **subject to conditions** ibt  
     **subcontracting bus tours on lines**  
          $\{blj_\delta, blj_\phi, \dots, blj_\omega\}$  **subject to conditions** scbt.

122

111. A bus contract has a header with the distinct names of a contractee and a contractor and a time interval.
112. A bus contract presents a timetable.
113. A bus contract presents a set of bus lines (by their identifiers) such that these are in the timetable.
114. And a bus contract may list one or more of three kinds of “exceptions”:
- a) cancellation of one or more named bus tours on one or more bus lines subject to certain (specified) conditions;
  - b) insertion of one or more extra bus tours on one or more bus lines subject to certain (specified) conditions;
  - c) subcontracting one or more unspecified bus tours on one or more bus lines subject to certain (specified) conditions — to further unspecified contractors.

123

We abstract the above quoted “one or more of three kinds of exceptions” as one possibly empty clause for each of these alternatives.

115. A bus contract now contains a header, a timetable, the subject bus lines and the exceptions,
116. such that
- a) line names mentioned in the contract are those of the bus lines of the timetable, and
  - b) bus (tour) numbers are those of the appropriate bus lines in the timetable.
117. The calendar period is for at least one full day, midnight to midnight.
118. A named contract is a pair of a contract name and a contract.

124

**type**

- 111.  $CN_m, CId, D, T, CON$
- 111.  $CH = CId \times CId \times (D \times D)$
- 112.  $CT = TT$
- 113.  $CLs = BLNm\text{-set}$
- 114.  $CE = (CA \times IN \times SC) \times CON$
- 114a.  $CA = BLNm \xrightarrow{m} BNo\text{-set}$
- 114b.  $IN = BLNm \xrightarrow{m} BNo\text{-set}$
- 114c.  $SC = BLNm\text{-set}$
- 115.  $CO' = CH \times CT \times CLs \times CE$
- 116.  $CO = \{ |co:CO' \bullet wf\_CO(co)| \}$
- 118.  $NCO = CN_m \times CO$

125

**value**

- 116.  $wf\_CO: CO' \rightarrow \mathbf{Bool}$
- 116.  $wf\_CO((ce, cr, (d, d')), (nd, tbl), cls, ((blns, blns', bls), con)) \equiv$
- 113.  $ce \neq cr \wedge$
- 116a.  $cls \subseteq \mathbf{dom} \text{ tbl} \wedge$
- 116b.  $\forall bli, bli': BLNm \bullet bli \in \mathbf{dom} \text{ blns} \wedge bli' \in \mathbf{dom} \text{ blns}' \Rightarrow$
- 116a.  $\{bli, bli'\} \subseteq \mathbf{dom} \text{ tbl} \wedge$
- 116b.  $\text{blns}(bli) \cup \text{blns}'(bli') \subseteq \mathbf{dom} \text{ sel\_btbt1}(tbl(bli)) \wedge$
- 116a.  $\text{bls} \subset \mathbf{dom} \text{ tbl} \wedge$
- 117.  $d < d'$

**Contractual Actions**

**126** An bus operator can now perform a number of actions according to a contract. We schematise these:

**For contract  $cn$  commence bus tour, line: bli and bus no.: bno**

**For contract  $cn$  cancel bus tour, line: bli and bus no.: bno**

**For contract  $cn$  insert extra bus tour, line: bli and bus no.: bno**

**Subcontract with respect to contract  $cn$  the following:**

**Contract  $cn'$ : for the calendar period  $[d, d']$  contractee  $ci$  contracts contractor  $cj$  to perform the following services with respect to timetable  $tt$ :**

**operate bus lines  $\{bl_{j_1}, bl_{j_2}, \dots, bl_{j_n}\}$**

**subject to the following occasional exceptions:**

**cancellation of bus tours:**

**$\{(bl_{j_c}, \{bno_{c_1}, \dots, bno_{c_m}\}), \dots\}$  subject to conditions  $cbt$**

**insertion of bus tours on lines**

$\{(\text{blj}_i, \{\text{bno}_{i_1}, \dots, \text{bno}_{i_n}\}), \dots\}$  **subject to conditions** *ibt*  
**subcontracting bus tours on lines**  
 $\{\text{blj}_\delta, \text{blj}_\phi, \dots, \text{blj}_\omega\}$  **subject to conditions** *scbt*.

127

119. A bus operator action is either a **commence**, a **cancellation**, an **insertion** or a **subcontracting** action. All actions refer to the (name of) the **contract** with respect to which the action is contracted.
- a) A **commence** action designator states the bus line concerned and the bus number of that line.
  - b) A **cancellation** action designator states the bus line concerned and the bus number of that line.
  - c) An **insertion** action designator states the bus line concerned and the bus number of that line — for which an extra bus is to be inserted.<sup>10</sup>
  - d) A **subcontracting** action designator, besides the name of the contract with respect to which the subcontract is a subcontract, state a named contract (whose contract name is unique).

128

**type**

119.  $\text{Act} = \text{Com} \mid \text{Can} \mid \text{Ins} \mid \text{Sub}$   
 119a.  $\text{Com} == \text{mkCom}(\text{sel\_cn}:\text{CNm}, \text{sel\_bli}:\text{BLNm}, \text{sel\_bno}:\text{BNo})$   
 119b.  $\text{Can} == \text{mkCan}(\text{sel\_cn}:\text{CNm}, \text{sel\_bli}:\text{BLNm}, \text{sel\_bno}:\text{BNo})$   
 119c.  $\text{Ins} == \text{mkIns}(\text{sel\_cn}:\text{CNm}, \text{sel\_bli}:\text{BLNm}, \text{sel\_bno}:\text{BNo})$   
 119d.  $\text{Sub} == \text{mkSub}(\text{sel\_cn}:\text{CNm}, \text{sel\_con}:\text{NCO})$

**Wellformedness of Contractual Actions**

129

120. In order to express wellformedness conditions, that is, pre-conditions, for the action designators we introduce a **context** which map contract names to contracts.
121. Wellformedness of a contract is now expressed with respect to a context.

**type**

120.  $\text{CTX} = \text{CNm} \xrightarrow{m} \text{CO}$

**value**

121.  $\text{wf\_Act}: \text{Act} \rightarrow \text{CTX} \rightarrow \mathbf{Bool}$

130

<sup>10</sup>The insertion of buses in connection with either unscheduled or extraordinary (sports, concerts, etc.) events can be handled by special, initial contracts.



- Let a defined **cnm** entry in **ctx** be a contract:  $((ce,cr),(nd,tbl),cls,(blns,bls,bls'),(d,d'))$ .

122. If **cmd** is a commence command  $mkCom(cnm,bln,bno)$ , then

- contract name **cnm** must be defined in context **ctx**;
- bus line name **bln** must be defined in the contract, that is, in **cls**, and
- bus number **bno** must be defined in the bus table part of table **tbl**.

122.  $wf\_Act(mkCom(cnm,bln,bno))(ctx) \equiv$

122a.  $cnm \in \mathbf{dom} \ ctx \ \wedge$

122.  $\mathbf{let} \ ((ce,cr),(nd,tbl),cls,(blns,bls,bls'),(d,d')) = ctx(cnm) \ \mathbf{in}$

122b.  $bln \in cls \ \wedge$

122c.  $bno \in \mathbf{dom} \ sel\_btbl(tbl(bln)) \ \mathbf{end}$

131

123. **cancellation** and **insertion** commands have the same static wellformedness conditions as have commence command.

123.  $wf\_Act(mkCan(cnm,bln,bno))(ctx) \equiv wf\_Act(mkCom(cnm,bln,bno))(ctx)$

123.  $wf\_Act(mkIns(cnm,bln,bno))(ctx) \equiv wf\_Act(mkCom(cnm,bln,bno))(ctx)$

132

124. If **cmd** is a subcontract command then

Let the subcontract command and the **cnm** named contract in **ctx** be  $mkSub(cnm,nco:(cnm',(ce',cr',(d'',d''')), (nd',tbl'),cls',(blns',bls'',bls'''))$  respectively  $((ce,cr,(d,d')), (nd,tbl), cls, (blns,bls,bls'))$ .

- contract name **cnm** must be defined in context **ctx**;
- contract name **cnm'** must not be defined in context **ctx**;
- the calendar period of the subcontract must be within that of the contract from which it derives;
- the net descriptors **nd** and **nd'** must be identical;
- the tables **tbl** and **tbl'** and must be identical and
- the set, **cls'**, of bus line names that are the scope of the subcontracting must be a subset of **bls'**.

133

124.  $wf\_Act(mkSub(cnm,nco:(cnm',co:((ce',cr',(d'',d''')), (nd',tbl'),cls',(blns',bls'',bls''')))))(ctx)$

124a.  $cnm \in \mathbf{dom} \ ctx \ \wedge$

124.  $\mathbf{let} \ co' = ((ce,cr,(d,d')), (nd,tbl),cls,(blns,blns',bls')) = ctx(cnm) \ \mathbf{in}$

124b.  $cnm' \notin \mathbf{dom} \ tbl \ \wedge$

124c.  $d \leq d'' \leq d''' \leq d' \ \wedge$

124d.  $nd' = nd \ \wedge$

124e.  $tbl' = tbl \ \wedge$

124f.  $cls' \subseteq bls' \ \mathbf{end}$

Wellformedness of contracts,  $\text{wf\_CO}(\text{co})$  and  $\text{wf\_CO}(\text{co}')$ , secures other constraints.

We do not here bring any narrated or formalised description of the semantics of contracts and actions. First such a description would be rather lengthy. Secondly a specification would be more of a requirements prescription.

### 3.6 Management and Organisation

135

**Definition: Management.** *Management is about resource<sup>[620]</sup>s: their acquisition<sup>[11]</sup>, scheduling<sup>[646]</sup> (over time), allocation<sup>[33]</sup> (over locations), deployment (in performing actions) and disposal (“retirement”). We distinguish between board-directed, strategic, tactical and operational actions. Board-directed actions target mainly financial resources: obtaining new funds through conversion of goodwill into financial resources, acquiring and selling “competing” or “supplementary” business units. Strategic actions (see Item 716 on page 220) convert financial resources into production, service supplies and resources and vice-versa — and in this these actions schedule availability of such resources. Tactical actions (see Item 741 on page 222) mainly allocate resources. Operational actions order, monitor and control the deployment of resources in the performance of actions.*

136

137

**Definition: Organisation.** *Organisation is about the “grand scale”, executive and strategic national, continental or global (world wide) (i) allocation of major resource (e.g., business) units, whether in a hierarchical, in a matrix, or in some other organigram-specified structure, (ii) as well as the clearly defined relations (which information, decisions and actions are transferred) between these units, and (iii) organisational dynamics.*

138

**Definition: Management & Organisation.** *The composite term management and organisation applies in connection with management<sup>[444]</sup> as outlined just above and with organisation<sup>[500]</sup> also outlined above. The term then emphasises the relations between the organisation and management of an enterprise.*

• • •

The borderlines within management actions and across organisation “layouts” are fuzzy.

#### 3.6.1 Transport System Examples

139

We shall only present sketchy examples of management and organisation.<sup>11</sup>

<sup>11</sup>Two remarks: (1) From an albeit superficial study of curricula of a number of business schools it seems, to this author, that the decomposition in *management and organisation* and into *executive, strategic, tactical and operational actions* is not quite the way the *financial, market, sales, product and production* (business administration) aspects of enterprises are looked upon in these schools. (2) We have, in [30], studied issues of management and organisation, and we shall elsewhere study these from the point of view of the signatures of *Executive, Strategic, Tactical and Operational* functions as they apply to and results in one or more of the resource types: Finance, Resource, spatial Location and Temporal notions of “business environments” ( $\rho : ENV$  which binds resource names to SCHEDULEs) and “business states” ( $\sigma : \Sigma$  which binds resource names to resource values) — and where SCHEDULEs binds resource names to time intervals and [al]locations.

- **Executive actions:** Deciding on major re-organisation of a transport net (for example introduction of toll roads or freeways, road pricing, major bridges across wide waters [potentially connecting two hitherto unconnected nets], and their management) are executive actions. So are decisions on merging or splitting transport from or into several transport services. Reorganising an enterprise from one characterised by a “deep” hierarchy of management layers (a hierarchy which may very well exemplify highly centralised both administrative and functional monitoring and control) into a matrix of two “shallow” hierarchies, one which addresses tactical and operational management and one which addresses executive and strategic management — with the former (the operations) being replicated across geographical areas while the latter applies “globally” — such reorganisations reflect executive actions (but are carried out by strategic and tactical management).
- **Strategic actions:** Adding or removing transport links, or major reorganisation of bus timetables are strategic actions. Splitting a(n own) contract into what is still to be operated and subcontracting other parts, for definite, to other bus operators are also strategic actions.
- **Tactical actions:** Insertion and cancellation of bus services are tactical actions. Subcontracting some parts of a timetable demanded service, for a short while, to other bus operators could be considered tactical actions.
- **Operational actions:** Commencing and thus, in general, allocating drivers to and sending these off on bus services are operational actions. So are announcing insertion of new (unscheduled) and cancellation of scheduled routes.

### 3.7 Human Behaviour

142

**Definition: Human Behaviour.** *By human behaviour we shall here understand the way a human follows the enterprise rules and regulations<sup>[640]</sup> as well as interacts with a machine<sup>[436]</sup>: dutifully honouring specified (machine dialogue<sup>[230]</sup> or) protocol<sup>[561]</sup>s, or negligently so, or sloppily not quite so, or even criminally not so! Human behaviour is a facet<sup>[285]</sup> of the domain<sup>[239]</sup>. We shall thus model human behaviour also in terms of it failing to react properly, i.e., humans as non-deterministic agent<sup>[24]</sup>s!*

### 3.8 Towards Theories of Domain Facets

144

#### 3.8.1 A Theory of Intrinsic

145

#### 3.8.2 Theories of Support Technologies

146

**An Example** Traffic (tf:TF), intrinsically, is a total function over some time interval, from time (t:T) to continuously positioned (p:P) vehicles (tn:TN).

Conventional optical sensors sample, at regular intervals, the intrinsic train traffic. The result is a sampled traffic (**stf:sTF**). Hence the collection of all optical sensors, for any given net, is a partial function from intrinsic (**itf**) to sampled train traffics (**stf**).

We need to express quality criteria that any optical sensor technology should satisfy — relative to a necessary and sufficient description of a **closeness** predicate. 147

For all intrinsic traffics, **itf**, and for all optical sensor technologies, **og**, the following must hold: Let **stf** be the traffic sampled by the optical gates. For all time points, **t**, in the sampled traffic, those time points must also be in the intrinsic traffic, and, for all trains, **tn**, in the intrinsic traffic at that time, the train must be observed by the optical gates, and the actual position of the train and the sampled position must somehow be checkable to be close, or identical to one another.

Since hubs change state with time,  $n:\mathbb{N}$ , the net needs to be part of any model of traffic. 148

### type

$T, TN$   
 $P = HP \mid LP$   
 $NetTraffic :: net:\mathbb{N} \times trf:(V \xrightarrow{m} P)$   
 $iTF = T \rightarrow NetTraffic$   
 $sTF = T \xrightarrow{m} NetTraffic$   
 $oG = iTF \xrightarrow{\sim} sTF$

### value

$[close] c: NetTraffic \times TN \times NetTraffic \xrightarrow{\sim} \mathbf{Bool}$

### axiom

$\forall itt:iTF, og:OG \bullet \mathbf{let} stt = og(itt) \mathbf{in}$   
 $\quad \forall t:T \bullet t \in \mathbf{dom} stt \bullet$   
 $\quad t \in \mathbf{DOM} itt \wedge \forall Tn:TN \bullet tn \in \mathbf{dom} trf(itt(t))$   
 $\quad \Rightarrow tn \in \mathbf{dom} trf(stt(t)) \wedge c(itt(t), tn, stt(t)) \mathbf{end}$

**DOM** is not an **RSL** operator. It is a mathematical way of expressing the definition set of a general function. Hence it is not a computable function.

Checkability is an issue of testing the optical sensors when delivered for conformance to the **closeness** predicate, i.e., to the axiom.

### General

**149** The formal requirements can be narrated: Let  $\Theta_i$  and  $\Theta_a$  designate the spaces of intrinsic and actual-world configurations (contexts and states). For each intrinsic configuration model — that we know is support technology assisted — there exists a support technology solution, that is, a total function from all intrinsic configurations to corresponding actual configurations. If we are not convinced that there is such a function then there is little hope that we can trust this technology

**type**

$\Theta_i, \Theta_a$   
 $ST = \Theta_i \rightarrow \Theta_a$

**axiom**

$\forall sts:ST\text{-set}, st:ST \cdot st \in sts \Rightarrow \forall \theta_i:\Theta_i, \exists \theta_a:\Theta_a \cdot st(\theta_i) = \theta_a$

### 3.8.3 A Theory of Rules & Regulations

150

There are, abstractly speaking, usually three kinds of languages involved wrt. (i.e., when expressing) rules and regulations (respectively when invoking actions that are subject to rules and regulations). Two languages, **Rules** and **Reg**, exist for describing rules, respectively regulations; and one, **Stimulus**, exists for describing the form of the [always current] domain action stimuli.

A syntactic stimulus, **sy\_sti**, denotes a function, **se\_sti**:STI:  $\Theta \rightarrow \Theta$ , from any configuration to a next configuration, where configurations are those of the system being subjected to stimulations. A syntactic rule, **sy\_rul**:Rule, stands for, i.e., has as its semantics, its meaning, **rul**:RUL, a predicate over current and next configurations,  $(\Theta \times \Theta) \rightarrow \mathbf{Bool}$ , where these next configurations have been brought about, i.e., caused, by the stimuli. These stimuli express: If the predicate holds then the stimulus will result in a valid next configuration.

**type**

Stimulus, Rule,  $\Theta$   
 $STI = \Theta \rightarrow \Theta$   
 $RUL = (\Theta \times \Theta) \rightarrow \mathbf{Bool}$

**value**

meaning: Stimulus  $\rightarrow$  STI  
 meaning: Rule  $\rightarrow$  RUL

valid: Stimulus  $\times$  Rule  $\rightarrow \Theta \rightarrow \mathbf{Bool}$

$\text{valid}(\text{sy\_sti}, \text{sy\_rul})(\theta) \equiv \text{meaning}(\text{sy\_rul})(\theta, (\text{meaning}(\text{sy\_sti}))(\theta))$

valid: Stimulus  $\times$  RUL  $\rightarrow \Theta \rightarrow \mathbf{Bool}$

$\text{valid}(\text{sy\_sti}, \text{se\_rul})(\theta) \equiv \text{se\_rul}(\theta, (\text{meaning}(\text{sy\_sti}))(\theta))$

A syntactic regulation, **sy\_reg**:Reg (related to a specific rule), stands for, i.e., has as its semantics, its meaning, a semantic regulation, **se\_reg**:REG, which is a pair. This pair consists of a predicate, **pre\_reg**:Pre\_REG, where  $\text{Pre\_REG} = (\Theta \times \Theta) \rightarrow \mathbf{Bool}$ , and a domain configuration-changing function, **act\_reg**:Act\_REG, where  $\text{Act\_REG} = \Theta \rightarrow \Theta$ , that is, both involving current and next domain configurations. The two kinds of functions express: If the predicate holds, then the action can be applied.

The predicate is almost the inverse of the rules functions. The action function serves to undo the stimulus function.

**type**

```

Reg
Rul_and_Reg = Rule × Reg
REG = Pre_REG × Act_REG
Pre_REG =  $\Theta \times \Theta \rightarrow \mathbf{Bool}$ 
Act_REG =  $\Theta \rightarrow \Theta$ 

```

**value**

```

interpret: Reg  $\rightarrow$  REG

```

The idea is now the following: Any action of the system, i.e., the application of any stimulus, may be an action in accordance with the rules, or it may not. Rules therefore express whether stimuli are valid or not in the current configuration. And regulations therefore express whether they should be applied, and, if so, with what effort. 156

More specifically, there is usually, in any current system configuration, given a set of pairs of rules and regulations. Let  $(\text{sy\_rul}, \text{sy\_reg})$  be any such pair. Let  $\text{sy\_sti}$  be any possible stimulus. And let  $\theta$  be the current configuration. Let the stimulus,  $\text{sy\_sti}$ , applied in that configuration result in a next configuration,  $\theta'$ , where  $\theta' = (\text{meaning}(\text{sy\_sti}))(\theta)$ . Let  $\theta'$  violate the rule,  $\sim\text{valid}(\text{sy\_sti}, \text{sy\_rul})(\theta)$ , then if predicate part,  $\text{pre\_reg}$ , of the meaning of the regulation,  $\text{sy\_reg}$ , holds in that violating next configuration,  $\text{pre\_reg}(\theta, (\text{meaning}(\text{sy\_sti}))(\theta))$ , then the action part,  $\text{act\_reg}$ , of the meaning of the regulation,  $\text{sy\_reg}$ , must be applied,  $\text{act\_reg}(\theta)$ , to remedy the situation. 157

**axiom**

```

 $\forall (\text{sy\_rul}, \text{sy\_reg}) : \text{Rul\_and\_Regs} \bullet$ 
  let  $\text{se\_rul} = \text{meaning}(\text{sy\_rul})$ ,
       $(\text{pre\_reg}, \text{act\_reg}) = \text{meaning}(\text{sy\_reg})$  in
   $\forall \text{sy\_sti} : \text{Stimulus}, \theta : \Theta \bullet$ 
     $\sim\text{valid}(\text{sy\_sti}, \text{se\_rul})(\theta)$ 
       $\Rightarrow \text{pre\_reg}(\theta, (\text{meaning}(\text{sy\_sti}))(\theta))$ 
         $\Rightarrow \exists n\theta : \Theta \bullet \text{act\_reg}(\theta) = n\theta \wedge \text{se\_rul}(\theta, n\theta)$ 
  end

```

It may be that the regulation predicate fails to detect applicability of regulations actions. That is, the interpretation of a rule differs, in that respect, from the interpretation of a regulation. Such is life in the domain, i.e., in actual reality 158

**3.8.4 A Theory of Management & Organisation**

160

**3.8.5 A Theory of Human Behaviour**

161

Commensurate with the above, humans interpret rules and regulations differently, and not always “consistently” — in the sense of repeatedly applying the same interpretations.

Our final specification pattern is therefore: 162

**type**

Action =  $\Theta \xrightarrow{\sim} \Theta$ -infset

**value**

hum\_int: Rule  $\rightarrow \Theta \rightarrow$  RUL-infset

action: Stimulus  $\rightarrow \Theta \rightarrow \Theta$

hum\_beha: Stimulus  $\times$  Rules  $\rightarrow$  Action  $\rightarrow \Theta \xrightarrow{\sim} \Theta$ -infset

hum\_beha(sy\_sti,sy\_rul)( $\alpha$ )( $\theta$ ) as  $\theta$ set

**post**

$\theta$ set =  $\alpha(\theta) \wedge$  action(sy\_sti)( $\theta$ )  $\in$   $\theta$ set

$\wedge \forall \theta': \Theta \bullet \theta' \in \theta$ set  $\Rightarrow$

$\exists$  se\_rul: RUL  $\bullet$  se\_rul  $\in$  hum\_int(sy\_rul)( $\theta$ )  $\Rightarrow$  se\_rul( $\theta, \theta'$ )

163

The above is, necessarily, sketchy: There is a possibly infinite variety of ways of interpreting some rules. A human, in carrying out an action, interprets applicable rules and chooses one which that person believes suits some (professional, sloppy, delinquent or criminal) intent. “Suits” means that it satisfies the intent, i.e., yields **true** on the pre/post-configuration pair, when the action is performed — whether as intended by the ones who issued the rules and regulations or not. We do not cover the case of whether an appropriate regulation is applied or not