The Prototype Modeling of a Road Transport Domain
A Domain Engineering Exercise

Dines Bjørner
Fredsvej 11, DK-2840 Holte, Danmark
E–Mail: bjorner@gmail.com, URL: www.imm.dtu.dk/~db

October 10, 2024

Abstract

We present a "worked example" of the steps and stages taken during the development of a prototype description of a road transport domain. That example records the thoughts and considerations that a domain analyzer cum describer could typically make. That is, we record, "on the side", the prompts of the domain analysis and description language used in this development.

Contents

1 Introduction

By a prototype domain description we shall here understand a "trial" description which captures essential aspects of a domain, but which abstracts from aspects considered non-essential. The idea is the following: A team of domain analysis & description engineers charged with developing a proper domain description for a seemingly "large scale" domain, see Sect. 3, is well advised in first developing a prototype description.

This paper follows the method for analyzing and describing domains as covered in [1, 4]. That is: we follow, rather strictly the domain analysis and description procedure as "defined" by the modeling ontology of Fig. 1 on the facing page.

We shall adhere to "that" modeling procedure, as said, "rather strictly" ! This means that we can already, before any analysis & description, determine the main project structure:

A Project Structure

1 The Universe of Discourse

2 The Endurants

2.1 External Qualities

2.1.1 Parts

2.1.2 The Endurant State

2.2 Internal Qualities

2.2.1 Unique Identifiers

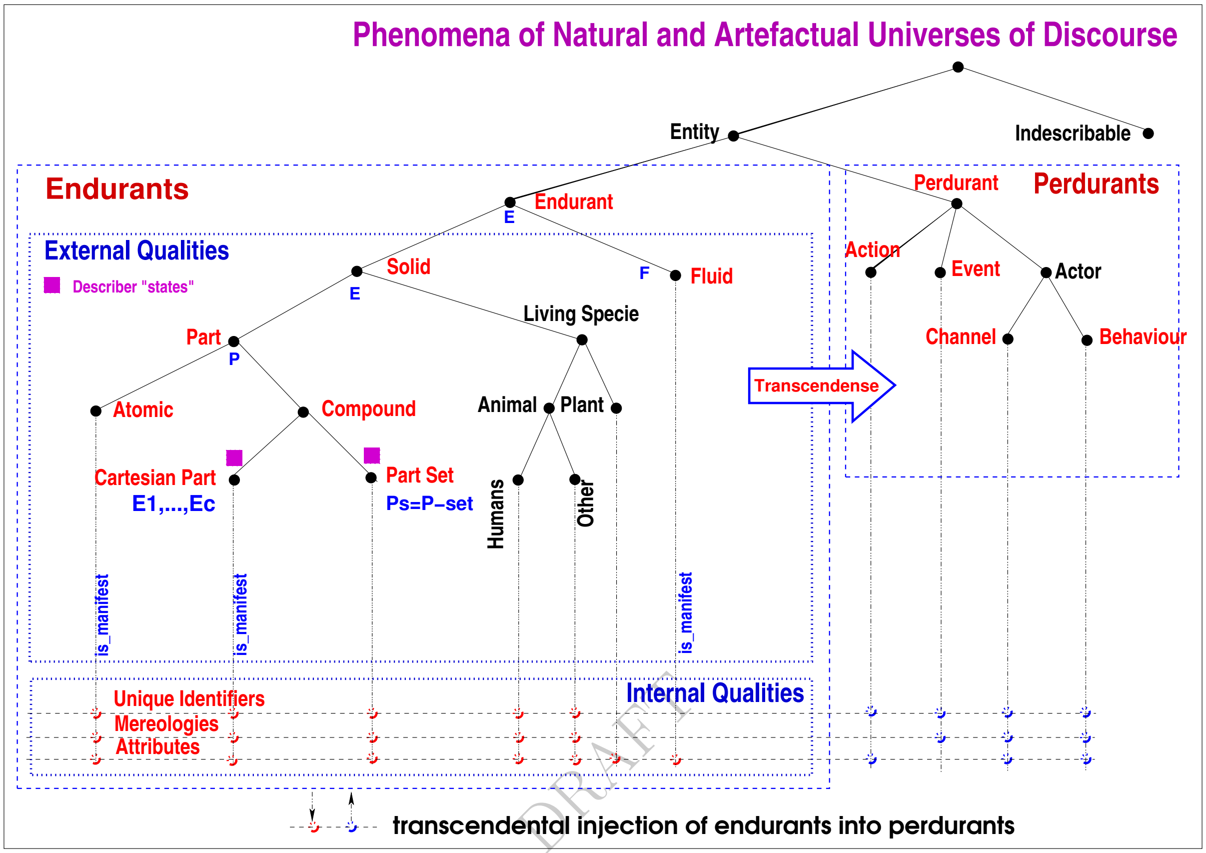

Figure 1: A Domain Analysis & Description Ontology

You can see that that project structure is also reflected in the table-of-contents of the present paper.

2 The Example Development

Our example is that of modeling a **road transport domain:** road transport domain. Not "the road transport domain", but a fair abstraction.

2.1 The Universe of Discourse

Project Start. It is suggested that any domain modeling project start with determining the formal domain type name, as well as a brief narrative.

2.1. **Narrative:**
2.1. The domain is that **of** roads and automobiles:
2.1. Streets with intersections and cars moving along the roads.
2.1. **Formalization:**
2.1. **type** UoD

Choice of Domain name. One may choose, freely, the type name. For example, for this exercise, RTD might be a good choice.
Extent of narrative. One might elaborate further in this narrative. But subsequent narratives and their formalization will do!
Abstractness. One cannot, of course, describe, in all its details, a generic road transport domain. And, similarly, one cannot describe, in all its details, a any concrete road transport domain. There are simply no end to which properties of a road transport domain one might wish to consider. Those that are considered for inclusion in a chosen abstraction are those which the modeler considers important to convey. Important in an example development, say for pedagogic purposes, or important for a concrete, say commercial product development, as a first step of that development.

2.2 The Endurants

2.2.1 External Qualities

Team Formation. The project team are, of course, as is the reader, assumed to be well familiar with the domain analysis and description method – as covered in [1, 4] and as summarized in Fig. 1 on the previous page.

The team leader, in consequence, can therefore, from the very start, lay out a project structure, much like the table-of-contents for Sect. 2, cf. Fig., 1 on the preceding page.

At the first "real" project meeting the domain, here the road transport domain, is given its – ever-so-briefly narrated – universe of discourse type name. At that meeting identification of the external qualities is given "a first try". After some iterations, one at least, the team (the project [staff] manager and colleagues) agree on a first decomposition of UoD into constituent endurants. Based on this decomposition the team members are assigned individual tasks. See further below.
Domain Study. While determining a domain [type] name the project team also "studies" the, or a real, domain[1]. Go out there, with Your colleagues, on the streets. Observe street intersections – which we shall here choose to call *hubs* – and the street stretches between

[1] We have chosen a rather well known domain. Let that not deceive Your analysis.

two immediately neighbouring hubs – we call call these stretches *links*. View some road (or subway) maps:

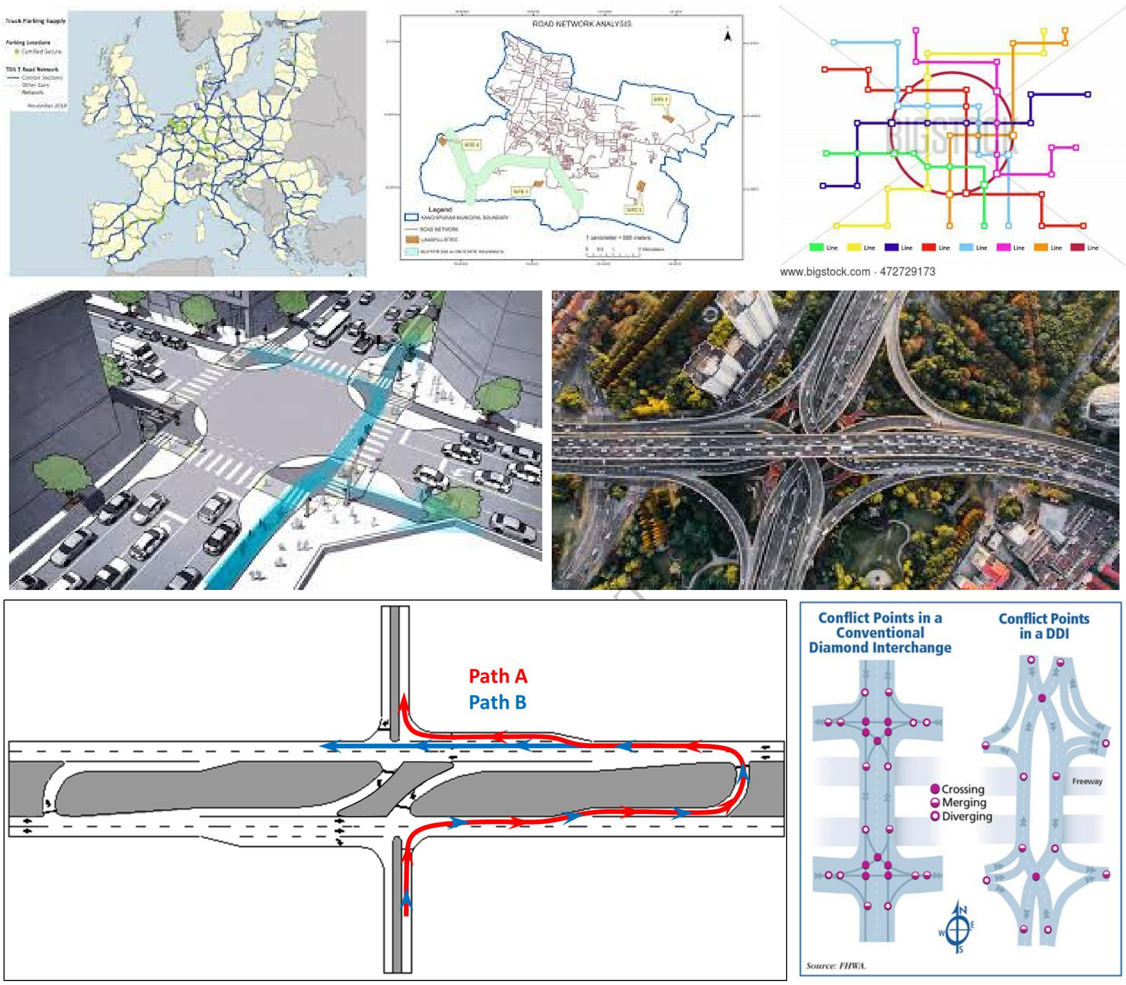

Figure 2: Road Nets, I

2.2.1.1 Parts

2.2.1.1.1 The Road Net We first comment on aspects of the road transport domain analysis. *There You are: the team which is to analyze & describe "the" domain. You follow the prescriptions of the domain analysis & description ontology, cf. Fig. 1 on page 3. Is it describable, i.e., is it an entity? (`is_entity(uod)`).* Yes – yes since you can talk rationally, meaningfully, about roads and automobiles. You then proceed by considering [only] the endurant "side", query: *are the domain entities*, that You consider, *solid or fluid*. They are solid, `is_solid(uod)`. So You go on: *are they parts or are they living species*. They are parts, `is_part(uod)`. *Is the universe of discourse atomic or compound. Well, the universe seems to contain roads and automobiles*. So it `is_compound(uod)`. From analyses like this we can then describe the "upper most part".

Figure 3: Road Nets, II

Narration:

1. We decide to consider the road transport domain to embody (at least these) two components: a Road Net, RN, and a “collection”, CA of automobiles.
2. From a road transport domain, UoD we can [therefore] observe a road net, and
3. a collection of automobiles.

Fomalisation:

type
1. RN, CA
value

2. obs_RN[2]: UoD → RN
3. obs_CA: UoD → CA

Before this first step of domain analysis & description the domain taxonomy shown in Fig. 4 on page 9 consisted of just the root box. After this first step that taxonomy has been augmented by the two box just below the root and the lines from the root to these.

Links and Hubs. We decide to decompose streets into links and hubs. Links are the stretches of streets between two immediately neighbouring street intersections. Hubs are then the street intersections.

Narration:

4. We decide to consider the road net to consist of a set of links, LS, and a set of hubs, SH.
5. From a set of links, LS, one can, indeed, observe a set of these!
6. Links are here considered atomic.
7. And from a set of hubs, HS, one can, indeed, observe a set of those!
8. Hubs are here considered atomic.

   Certain constraints apply to how links and hubs are arranged wrt. one another. These will be taken care of in Sect. 2.2.2.1 on page 10.

Fomalisation:

type
4. LS, HS
5. LS = L-**set**
6. L
7. HS = H-**set**
8. H
value
5. obs_LS: RN → LS
7. obs_HS: RN → HS
axiom
8. [See Sect. 2.2.2.1 on page 10.]

That is it! No consideration in this exercise of what is along the roads: houses, bus stops, etc.. The taxonomy of Fig. 4 on page 9 has now been augmented by the two leftmost sub-trees..

[2]Recall: obs_P, for any type identifier P, names a unique identifier observer. It cannot be formally defined – other than, as here, stating its signature. It "applies" to solid endurants, e:E — where the "application" takes the form of Your inspection – and ("miraculously") yields [what You also observe] a "contained/embodied" part p:P.

2.2.1.1.2 Automobiles

9. From the collection, CA, of automobiles one can observe the set of such.
10. We consider automobiles to be atomic.

type
9. SA = A-**set**
10. A
value
9. obs_SA: CA → SA

That is it ! Automobiles (cars, vehicles) are here simply atomic – and there is no mention of the living species *humans who (may) drive them.*
The taxonomy of Fig. 4 on the facing page has now been augmented by the rightmost sub-tree..

2.2.1.2 The Endurant State For use in later analysis and description we define a/the endurant state of road transport domains.

11. The endurant state is here considered the set of all atomic parts, i.e., the set of all links,hubs and automobiles.[3]

type
11. Σ = (L|H|A)-**set**
variable
11. σ := part_state(uod)
values
11. part_state: (UoD|RN|CA) → Σ
11. part_state(p) ≡
11. is_UoD(p) → part_state(obs_obs_RN(p)) ∪ part_state(obs_obs_CA(p)),
11. is_RN(p) → obs_LS(p) ∪ obs_HS(p),
11. is_CA(p) → obs_SA(p)

2.2.1.3 State Functions In this example we introduce two functions on part states: the retrieval and the removal of parts, whether links, hubs or automobiles.

12. We introduce a common type for links, hubs and parts – and their unique identifiers.
13. To retrieve a part, given its unique identifier, is to "read" from the part state the part with that unique identifier.
14. The remove_A(ai) function
15. removes the automobile designated by ai from the state σ
16. and the unique identifier ai from σ_{uid}.

[3]In subsequent, other, domain models we may include the [compound] road net and the collection of automobiles to be state components. See also Sect. 2.2.1.5 on the next page.

type
14. P = L|H|A, PI = LI|HI|AI
value
15. get_P: **Unit** → PI → P
15. get_P()(pi) ≡ ι p:P • p ∈ σ ∧ uid_P(p)=pi

2.2.1.4 Comments We do not describe the external qualities of the road transport domain further.
Abstraction. Indeed we have abstracted ! No mentioning of traffic lights. (They will be analyzed and described, as attributes, in Sect. 2.2.2.3 on page 11.)) No mentioning of "border" of road net !
Strictness of Development Procedure. The project team are, of course, as is the reader, assumed to be well familiar with the domain analysis and description method – as covered in [1, 4] and as summarized in Fig. 1 on page 3. The method procedure is that the sequence of work items as displayed in the table-of-contents (for Sect. 2) must be followed strictly.

2.2.1.5 A Domain Taxonomy *From the observer and type definitions of the domain parts we can derive the domain taxonomy. Figure 4 diagrams an abstraction. The taxonomy reflects all the endurants formalized above: the UoD, RN, CA, LS, HS, L, H and A. We can distinguish between manifest and conceptual parts. The manifest parts are those to which we ascribe internal qualities. The conceptual "parts" are there to help structure the presentation.*

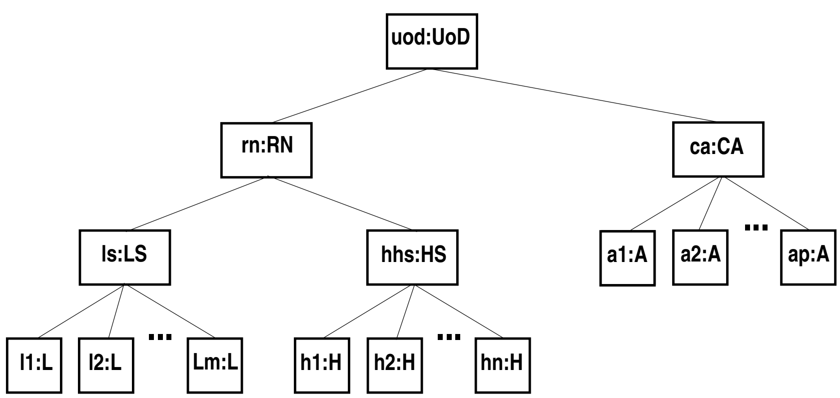

Figure 4: A Road Transport Domain Taxonomy

2.2.2 Internal Qualities

Whither Manifest or Not ? The project team decides which compound parts are manifest, which are "structural" [conceptual] in the sense of structuring the domain taxonomy. We decide in this example to only ascribe internal qualities to atomic parts.
Team members can now be assigned to the analysis & description of distinct manifest endurant parts. That is: the unique identifier, mereology and attribute modeling can "go on in parallel".

2.2.2.1 Unique Identifiers

17. Links, hubs and automobiles have unique identifiers.

type
17. LI, HI, AI
value
17. uid_L: L→LI, uid_H: H→HI, uid_L: L → LI

uid_X, for any type identifier X, names a unique identifier observer. It cannot be formally defined – other than, as here, stating its signature. It applies to solid endurants and ("miraculously") yields their unique identifier..

2.2.2.1.1 The Unique Identifier State

18. There is a variable of type σ_uid.

19. It contains the unique identifiers

20. of all [in this case, atomic] parts [of the domain being analyzed and described.

type
18. Σ_{uid}= (LI|HI|AI)-**set**
variable
19. σ_{uid} := part_uids(σ)
value
20. part_uids: **Unit** → Σ_{uid}
20. part_uids() ≡
20. { uid_L(l) | l:L **in** σ }
20. ∪ { uid_H(h) | h:H **in** σ }
20. ∪ { uid_A(a) | a:A **in** σ }

2.2.2.1.2 The Uniqueness of Parts

21. Uniqueness of parts means that there are as many [atomic] parts as there are their unique identifiers.

axiom
21. **card** σ = **card** σ_{uid}

2.2.2.2 Mereology

22. The mereology of a link is a 1 or a 2 element set of hub identifiers: one if the link loops around the identified hub, two when it is incident upon / emanates from those two identified hubs –

23. such that the hub identifiers are hub identifiers of the road net.

24. The mereology of a hub is a (the) set of link identifiers of the links emanating from the hub –
25. such that the link identifiers are link identifiers of the road net.
26. The mereology of an automobile is the set of all the link and hub identifiers of links and hubs that the automobile is allowed to enter –
27. such that these link and hub identifiers are link and hub identifiers of the road net.

type
22. LM = HI-**set**
24. HM = LI-**set**
26. AM = (LI|HI)-**set**
value
22. merero_L: L → LM
24. merero_H: H → HM
26. merero_A: L → AM
axiom
23. $\forall$ l:L • l $\in \sigma \Rightarrow$ **card** mereo_L(l)$\in${1,2} $\wedge$ mereo_L(l)$\subseteq\sigma_{uid}$
25. $\forall$ h:H • h $\in \sigma \Rightarrow$ mereo_H(h)$\subseteq\sigma_{uid}$
27. $\forall$ a:A • a $\in \sigma \Rightarrow$ mereo_A(a)$\subseteq\sigma_{uid}$

We remind the reader: The present mereology emphasizes the topological layout of road nets and its use by automobiles.

2.2.2.3 Attributes

2.2.2.3.1 Link Attributes

28. Links have lengths.
29. With links one can observe their automobile traffic – as a sequence of "snapshots", each a pair of time stamps and automobile identifiers, together designating the entering and leaving of identified automobiles on that link.
30. Links have states. A link state indicates the direction in which automobiles may move along the link. One-way or two way. Hence a link state is modeled as a set of pairs of the hub identifiers of the hub the link connects.
31. Links can be controlled to be in either of a number of states. That is: links have state spaces: sets of states [that it may attain].
32. Links may be one lane, two lane or more[4].
33. Links have wear-and-tear[5].

[4] Exercise: Redefine to Lane to allow a link to be composed from stretches of [sub-]links with changing number of lanes.
[5] – left further unspecified !

34. Et cetera

type
28. LEN, static[6]
29. LHist = ($\mathbb{TIME} \times$ AI)*, programmable
30. LΣ $-$ (HI$\times$HI)**-set**, programmable
31. LΩ = LΣ**-set**, static
32. Lane = **Nat**, static
33. Wear_and_Tear, monitorable
35. ...
values
28. attr_LEN: L $\rightarrow$ LEN
29. attr_LHist: L $\rightarrow$ LHist
30. attr_LΣ: L $\rightarrow$ LΣ
31. attr_LΩ: L $\rightarrow$ LΩ
32. attr_Lane: L $\rightarrow$ Lane
33. attr_Lane: L $\rightarrow$ Wear_and_Tear
34. ...
axiom
$\forall$ l:L $\bullet$ l $\in \sigma \Rightarrow$
29. **let** his = attr_LΣ(l) **in** $\forall$ (hij,hik)$\in$ his $\Rightarrow$ {hij,hik}$\subseteq$mereo_L(l) $\wedge$ his $\in$ attr_Ω(l) **end**
30. **let** ... **in** ... **end**
31. **let** ... **in** ... **end**

That's all we show in this example. Such attributes as to geographical/spatial location, positions of the two ends of links, their curvature, modeled, for example, by means of [three dimensional] *Bèzier Curves*[7], is left out. So are wear-and-tear of link; whither one, two or more lanes; whither road construction material is macadamized, concrete, gravel, or other; etc.

2.2.2.3.2 Hub Attributes In this example we illustrate that street intersection have traffic lights.

35. Hubs have states: HΣ. A hub state is a subset of the set of pairs of the unique identifiers of the links incident upon/emanating from the link. A air, viz. (li,lj), shall indicate that the hub is open for automobile traversal in direction from the link identified by li to the link identified by lj – and these link identifiers must be those of the hub.

36. Hubs have state spaces: HΩ. The A hub state space designates the hub states that a hub may be in. The current hub state must be n the hub state space.

37. With hubs one can observe their automobile traffic – as a sequence of "snapshots", each a pair of time stamps and automobile identifiers, together designating the entering and leaving of identified automobiles on that hub.

type

[6] We annotate attribute type definitions with reference to Michael A. Jackson's attribute category [5].
[7] https://en.wikipedia.org/wiki/B%C3%A9zier_curve

35. $H\Sigma = (LI \times LI)$**-set**, programmable
36. $H\Omega = H\Sigma$**-set**, static
37. $HHist = (\mathbb{TIME} \times AI)^*$, programmable

type

35. attr_HΣ: $H \to H\Sigma$
36. attr_HΩ: $H \to H\Omega$
37. attr_HHist: $H \to HHist$

axiom

35. $\forall$ h:H $-$ h $\in \sigma \Rightarrow$ **let** hσ = attr_HΣ(h) **in** $\forall$ (li,lj)$\in \sigma \Rightarrow \{li,lj\} \subseteq \sigma_(uid)$ **end**
36. ...
37. ...

Again we do not exemplify many other possible attributes.

2.2.2.3.3 Automobile Attributes

38. Automobiles have position of links and in hubs.
    (a) A link position designates the link, from which hub, to which hub and the fraction along the link the automobile has progressed.
    (b) a hub position designates the hub, and from which link the automobile has entered the hub and the link which link the automobile is aimed.
39. One can speak of the road net history of an automobile: a time-stamped sequence of the links and hubs it has traveled.
40. Automobiles have speed.
41. Automobiles have acceleration.
42. Automobiles are either diesel, gasoline, electrically, or hybrid motors.
43. Automobiles have
44. Automobiles have

type

38. APos == onLink | atHub, programmable
38a. onLink :: $LI \times HI \times Frac \times HI$
38a. Frac = **Real**
38b. atHub :: $HI \times LI \times LI$
39. $AHist = (\mathbb{TIME} \times (LI|HI))^*$, programmable
40. Speed, monitorable
41. Acceleration, monitorable
42. Motor = ″`Diesel`″ | ″`Gasoline`″ | ″`Electric`″ | ″`Hybrid`″, static

value

38. attr_APos: $A \to APos$
39. attr_AHist: $A \to AHist$
40. attr_Speed: $A \to Speed$

41. attr_Acceleration: A → Acceleration
42. attr_Motor: A → Motor
axiom
38a. ∀ mkonLink(li,hj,f,hk):onLink • {li,hj,hk}⊆σ_{uif} ∧ 0<f<1
38b. ∀ mkatHub(hi,lj,lk) • {hi,lj,lk}⊆σ_{uif}
39. ...

Other attributes are: automobile maker, model, registration/license number, owner, kind: bus, truck, sedan, driver, ...

2.2.3 Intentional Pull

The road transport domain has intent: to convey automobiles along roads.

45. If an automobile a is recorded to have been on a specific link ℓ at time τ then that link records that that automobile was there;

    vice versa:

46. if a link ℓ records that an automobile a is on that specific link at time τ then that automobile records that that it was there.

To express the above we introduce some auxiliary notions:

47. a map lh from all link identifiers of the road net to the history of the designated link,
48. a map hh from all hub identifiers of the road net to the history of the designated hub,
49. a map ah from all automobile identifiers of the road net to the history of the designated automobile, and
50. the union map of lh and hh.

value
47. **let** lh = [uid_L(l) ↦ attr_LHist(l) | l:L • l ∈ σ],
48. hh = [uid_H(h) ↦ attr_HHist(h) | h:H • h ∈ σ],
49. ah = [uid_A(a) ↦ attr_AHist(a) | a:A • a ∈ σ] **in**
50. **let** rh = lh ∪ hh **in**
45. ∀ ri:(LI|HI) • ri ∈ σ_{uid} ∧ ri ∈ **dom** rh ⇒
45. **let** rhist = rh(ri) **in**
45. ∀ i:**Nat** • i ∈ **inds** rhist ⇒
45. **let** (τ',ai) = rhist(i) **in**
45. **let** ahist = ah(ai) **in**
45. ∃ ! j:**Nat** • j ∈ **inds** ahist ∧
45. **let** (tau″,ri′) = ahist(j) **in** τ'=τ''∧ri=ri ′ **end end end**
45. ≡
46. ∀ ai:AI • ai ∈ σ_{uid} ∧ ai ∈ **dom** ah ⇒
46. **let** ahist = ah(ai) **in**
46. ∀ i:**Nat** • i ∈ **inds** ahist ⇒
46. **let** (τ',ri′) = ahist(i) **in**

46. $\exists$! j:**Nat** • j $\in$ **inds** ahist $\wedge$
46. **let** (tau″,ri″) = (lh $\cup$ hh)(ri)(j) **in** τ'=$\tau''$$\wedge$ri=ri ′ **end end end**
46. **end**
45. **end end**

2.3 The Perdurants

From parts we transcendentally deduce behaviours. In this example, we have chosen, in Sect. 2.2, only the atomic parts of the domain to be manifest. So we shall only deal with the link, hub and automobile behaviours.

2.3.1 Channels

51. Automobiles interact, i.s., CSP communicate, with links and hubs. So the channel indices are sets of two unique identifiers: those of the communicating automobile and those link or hub. The type definition of messages, M, will emerge from our [later] definition of the automobile functionality definition.

51. **channel** { ch[{ai,ri}] | ai:AI,ri:(LI|HI) • {ai,ri}$\subseteq\sigma_{uid}$ } M

2.3.2 Behaviour Signatures

Recall that we conventionally structure the textual definition of behaviour signatures as follows:

value behaviour: BI → Mereo → Static_Attrs → Monitor_Attrs → Program_Attrs **Unit**

52. By ... we omit consideration of static and monitorable attributes.
53. Etc.
54. Etc.

52. link: LI → LM → (LEN×...) → ... → (LHist×LΣ×LΩ) **Unit**
53. hub: HI → HM → ... → ... → (HHist×HΣ×HΩ) **Unit**
54. automobile: AI → AM → ... → ... → (APos×AHist) **Unit**

2.3.3 Behaviour Definitions

When defining the functionality of the domain behaviours we must consider their possible actions. Link and hub behaviours are, in this respect "passive", i.e., do not, in this example, initiate actions on their own initiative. Automobile behaviours, in this example, are the only such which do. Automobiles

- drive along links, cf. Items 56a on the next page, 58 on the following page

- make temporary halts along links, cf. Items 56b, 59 on the facing page
- stops altogether: leaves the road net link, cf. Items 56c, 60 on the facing page
- exit links while entering hubs, cf. Items 57, 61 on the facing page
- drive through hubs, cf. Items 62(a)i on page 18, 63 on page 18,
- stops altogether: leaves the road net hub, cf. Items 62(a)ii on page 18, 64 on page 18
- exit hubs while entering links, cf. Items 62(a)iii on page 18, 65 on page 18

2.3.3.1 Automobile on a Link

55. Automobile behaviours are either on a link or at a hub.
56. On a link, if not having reached the end of the link, an automobile internal non-deterministically, $\sqcap$
    (a) either proceeds along the link,
    (b) or makes a temporary halt,
    (c) or the automobile leaves the road net,
57. or, deterministically, the automobile leaves the link and enters the next hub.

value
56. automobile(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist) $\equiv$
56. **if** $0 \leq f < 1$ **then**
56a. (auto_proceeds_on_link(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist)
56b. $\sqcap$ auto_halts_on_link(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist)
56c. $\sqcap$ auto_stops_at_Link(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist))
57. **else** auto_leaves_Link_enters_hub(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist) **end**

58. An automobile on a link
    (a) proceeds by moving along "a bit", δ, [maybe reaching the end of the link],
    (b) notifies the link of its presence,
    (c) and resumes being an automobile on that link with an augmented history.

value
58. auto_proceeds_on_link(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist)
58a. **let** $f' = f+\delta$ **axiom** $0<\delta<(1-f)$, τ = **record**_$\mathbb{TIME}$() **in**
58b. ch[{ai,li}] ! (τ,ai) ;
58c. automobile(ai)(am)(...)(...)(onLink(li,fhi,f+δ,thi),$\langle(\tau$,li)$\rangle\widehat{\ }$ahist) **end**
type
51. M = ($\mathbb{TIME}\times$LI) | ...

59. An automobile on a link
    (a) makes a temporary halt till some time
    (b) notifies the link of its presence,
    (c) and resumes being an automobile on that link with an augmented history.

value
59. auto_halts_on_link(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist)
59a. **let** τ = **record_**$\mathbb{TIME}$() **in**
59b. ch[{ai,li}] ! (τ,ai) ;
59c. automobile(ai)(am)(...)(...)(onLink(li,hi,f,tli),$\langle(\tau$,li)$\rangle$^ahist) **end**

60. An automobile on a link
    (a) at some time,
    (b) after having notified the link of its (last) presence,
    (c) decides to leave the road net
    (d) and stops!

60. auto_stops_at_link(ai)(am)(...)(...)(onLink(li,fhi,f,thi),ahist) $\equiv$
60a. **let** τ = **record_**$\mathbb{TIME}$() **in**
60b. ch[{ai,li}] ! (τ,ai) ;
60d. **stop end**

61. An automobile on a link
    (a) leaves that link when it has reached its end
    (b) by [internal non-determinism] selecting a next link on to which to move ["after" the hub]
    (c) notifies the link (being left) and the hub being entered of its [last, resp. first] presence,
    (d) and proceeds to be an automobile at the hub.

value
61. auto_leaves_link_enters_hub(ai)(am)(...)(...)(onLink(li,fhi,1,thi),ahist) $\equiv$
61a. **let** lis = mereo_H(get_P(thi)) **in**
61b. **let** li:LI • li $\in$ lis, τ = **record_**$\mathbb{TIME}$() **in**
61c. ch[{ai,li}] ! (τ,ai) ‖ ch[{ai,thi}] ! (τ,ai) ;
61d. automobile(ai)(am)(...)(...)(atHub(),$\langle\rangle$^ahist)
61b. **end end**

2.3.3.2 Automobile at a Hub

62. At a hub an automobile
    (a) internal non-deterministically, ⌈⌉,
        i. either proceeds through/around the hub, cf. Items 62(a)i, 63
        ii. or makes a final stop, cf. Items 62(a)ii, 64
        iii. or the automobile leaves the hub and proceeds to the target link, cf. Items 62(a)iii, 65.

```
value
62. automobile(ai)(am)(...)(...)(atHub(hi,fli,tli),ahist) ≡
62(a)i.    ( auto_proceeds_at_hub(ai)(am)(...)(...)(atHub(hi,fli,tli),ahist)
62(a)ii.   ⌈⌉ auto_stops_at_a_hub(ai)(am)(...)(...)(atHub(hi,fli,tli),ahist)
62(a)iii.  ⌈⌉ auto_leaves_hub_enters_link(ai)(am)(...)(...)(atHub(hi,fli,tli),ahist) )
```

63. An automobile at a hub
    (a) proceeds by so notifying that hub
    (b) and resumes being an automobile at that hub with an updated history.

```
63.   auto_proceeds_at_hub(ai)(am)(...)(...)(atHub(hi,fli,tli),ahist) ≡
63.      let τ = record_TIME() in
63a.      ch[{ai,hi}] ! (`tai,ai) ;
63b.      auto_proceeds_at_hub(ai)(am)(...)(...)(atHub(hi,fli,tli),⟨(τ,hi)⟩^ahist) end
```


64. An automobile at a hub
    (a) at some time,
    (b) after having notified the hub of its (last) presence,
    (c) decides to leave the road net
    (d) and stops!

```
64.   auto_stops_at_hub(ai)(am)(...)(...)(atHub(hi,fli,tli),ahist) ≡
64a.      let τ = record_TIME() in
64b.      ch[{ai,hi}] ! (τ,ai) ;
64d.      stop end

64c.      remove_P(ai) ;
```

65. An automobile at a hub
    (a) at some time decides to leave the hub and enter its target link,
    (b)
    (c) so notifies the hub and that target link, tli,

(d) and resumes being an automobile, but now on that target link.

value
65. auto_leaves_hub_enters_link(ai)(am)(...)(...)(atHub(hi,fli,tli),ahist) ≡
65a. **let** τ = **record_TIME**() **in**
65c. (ch[{ai,hi}] ! (τ,ai) ∥ ch[{ai,tli}] ! (τ,ai));
65d. automobile()(ai)(am)(...)(...)(get_onLink(hi,tli),⟨(τ,hi),(τ,tli)⟩^ahist) **end**

65b. get_onLink_Info: HI×LI → **Unit** → LI → onLink
65b. get_onLink_Info(hi,li) ≡
65b. **let** his = mereo_L(get_P(li)) **in**
65b. **let** (fhi′,thi):(LI×LI) • {fhi′,thi}⊆his ∧ hi=fhi′ **in**
65b. onLink(li,hi,0,thi) **end end**

2.3.3.3 Link Behaviour

66. Links
    (a) external non-deterministically, ⌷, awaits notification from any of the road net automobiles as to their timed presence on the link
    (b) where-after links resumes being links with an augmented history.

value
66. link: LI → LM → (LEN × LΩ × ...) → ... → (LΣ × LHist) **Unit**
66. link(li)(lm)(...)(...)(...,lhist) ≡
66a. **let** (τ,ai) = ⌷ { ch[{li,ai}] ? | ai:AI•ai ∈ σ_{uid} } **in**
66b. link(li)(hm)(...)(...)(...,⟨(τ,ai)⟩^lhist) **end**

2.3.3.4 Hub Behaviour

67. Hubs
    (a) external non-deterministically, ⌷, awaits notification from any of the road net automobiles as to their timed presence in the hub
    (b) where-after hubs resumes being hub with an augmented history.

value
67. hub: HI → HM → (HΩ × ...) → ... → (HΣ × HHist) **Unit**
67. hub(hi)(hm)(...)(...)(...,hhist) ≡
67a. **let** (τ,ai) = ⌷ { ch[{hi,ai}] ? | ai:AI•ai ∈ σ_{uid} } **in**
67b. hub(hi)(hm)(...)(...)(...,⟨(τ,ai)⟩^hhist) **end**

2.3.4 Domain Initialization

An initialization of the road transport domain is the

68. parallel composition of all automobile behaviours
69. in parallel with the
70. parallel composition of all link behaviours
71. in parallel with the
72. parallel composition of all hub behaviours.

```
68. ‖ { automobile(uid_A(a))
68.         (mereo_A(a))
68.             (...)
68.                 (...)
68.                     (attr_APos(a),attr_AHist(a)) | a:A • a ∈ σ }
69. ‖
70. ‖ { link(uid_L(a))
70.         (mereo_L(l))
70.             (attr_LEN(l),attr_LΩ(l)...)
70.                 (...)
70.                     (attr_LΣ(l),attr_LHist(l)) | l:L • l ∈ σ }
71. ‖
72. ‖ { hub(uid_H(a))
72.           (mereo_H(h))
72.               (attr_HΩ(l),...)
72.                   (...)
72.                       (attr_HΣ(l),attr_HHist(l)) | h:H • h ∈ σ }
```

– where we omit most static and all monitorable arguments!

3 Variations on a Theme

The development of the prototype domain description of Sect. 2 now serves as a pre-requisite basis for developing a proper description of a realistic road transport domains. Some of these will be exemplified in this section.

3.1 Road Net Management

3.1.1 Road Building

3.1.2 Road Maintenance

3.1.3 Road Pricing

3.2 Automobile Management

3.2.1 Dept. of Vehicles

3.2.2 Automobile Club

3.2.3 Traffic Information

4 Conclusion

5 Bibliography

References

[1] Dines Bjørner. *Domain Science & Engineering – A Foundation for Software Development*. EATCS Monographs in Theoretical Computer Science. Springer, Heidelberg, Germany, 2021. A revised version of this book is [3].

[2] Dines Bjørner. Domain Modelling – A Primer. A short version of [3]. xii+202 pages[8], May 2023.

[3] Dines Bjørner. Domain Science & Engineering – A Foundation for Software Development. Revised edition of [1]. xii+346 pages[9], January 2023.

[4] Dines Bjørner. Domain Modelling. *To be submitted*, pages 1–32, September 2024. Institute of Mathematics and Computer Science. Technical University of Denmark.

[5] Michael A. Jackson. *Software Requirements & Specifications: a lexicon of practice, principles and prejudices*. ACM Press. Addison-Wesley, Reading, England, 1995.

[8] This book is currently being translated into Chinese by Dr. Yang ShaoFa, IoS/CAS (Institute of Software, Chinese Academy of Sciences), Beijing and into Russian by Dr. Mikhail Chupilko and his colleagues, ISP/RAS (Institute of Systems Programming, Russian Academy of Sciences), Moscow

[9] Due to copyright reasons no URL is given to this document's possible Internet location. A primer version, omitting certain chapters, is [2]