A Philosophy of Domain Science & Engineering

An Interpretation of Kai Sørlander's Philosophy*

Dines Bjørner

Technical University of Denmark, Kgs.Lyngby, Danmark
Fredsvej 11, DK-2840 Holte, Danmark
E–Mail: bjorner@gmail.com, URL: www.imm.dtu.dk/~db

April 23, 2018: 13:48

Abstract

We overview some relations between *domain analysis & description* and *Kai Sørlander's Philosophy*.

1 Introduction

This paper is based on Sørlander's Philosophy [1, 2, 3, 4] and my recent papers [5, 6]. After a brief introduction we first bring a narrated and formalised domain description example.

1.1 What do we mean by Domain ?

By a *domain* we shall understand a **logically describable** segment of a **human assisted** reality, i.e., of the world, its **natural parts** as well as **man-made artifacts:** *endurants* ("still"), existing in space, as well as *perdurants* ("alive"), existing also in time, and where an emphasis is placed on *"human-assistedness"*, that is, that there is *at least one man-made artifact* and that *humans* are a primary cause for change of endurant *states* as well as perdurant *behaviours* "by means" of the man-made artifacts ■

*This document is the paper version of a talk given at the Viktor Pertrovich Ivannikov Memorial Workshop, Yerevan, Armenia, 3 May 2018, see http://www.imm.dtu.dk/~dibj/2018/yerevan/Marandjian.pdf. Both are based on an extensive research report which you can also find on the Internet: http://www.imm.dtu.dk/~dibj/2018/philosophy/filo.pdf

1.2 Examples of Domains

Over the years the *domain analysis & description method* hase been developed and tested through a number of experimental case studies some of which are referred to in this list:

- **railways** [7, 8, 9, 10, 11],
- **container shipping** [12],
- **stock exchange** [13],
- **document systems** [14],
- **oil pipelines** [15],
- **"The Market"** [16],
- **weather information** [17],
- **credit card systems** [18],
- **urban planning** [19],
- **swarms of drones** [20],
- et cetera, et cetera!

1.3 Domains – in Contrast to other "Fields"

Thus **domain science & engineering** is different from **automation** and **cybernetics**: their emphasis is on basing computer applications on mathematics and physics. **Domain science & engineering**, is also different from **optimisation** and **operations research**: their emphasis is on mathematical models of resource scheduling, but not the operational monitoring and control. **Domain science & engineering** is a new field as you might learn from this paper — **all it takes is an open mind!**

1.4 A Triptych of Software Development

Before **software** can be **designed** we must understand what is **required** and what is expected. Before **requirements & expectations** (goals) can be **prescribed** we must understand the **domain** – which hence must be **described**.

1.4.1 Three Phases of Software Development

So there are three phases to software development: **domain engineering**, concerned with *domain analysis & description*, **requirement**s **engineering**, concerned with *requirements analysis & prescription* based on the domain analysis & description, and **software design**, based on domain descriptions and requirements prescriptions for verifying that the software meets expectations and satisfies requirements. In [21, 22] we show how to *"refine & extend"* a *domain description* into a *requirements prescription*. *Domain science & engineering* is only now emerging!

1.4.2 But, really, there is Little new here!?

From the 1970s till today: the study of **programming language semantics** [23] lead to **provably correct programs** and **compilers**. The study of **domain science & engineering** is the study of the **languages** spoken by the *stakeholders* in the **domains** of concern to *domain science & engineering* and is necessary in order to achieve improved confidence in large software systems. In [24, 25, 26, 27, 28, 29, 30, 1977-1984] it is shown how develop formal semantics of programming languages and their interpreters and compilers (for `CHILL` and `Ada`). *But it seems to take some time to "sink in"!*

1.5 So what is the problem?

Well, we wish to make sure that our **domain analysis & description method** rests on a secure foundation, that is, (1) that the **composition** of descriptions "is right", (2) that **elements** of descriptions are **logically founded**, and **(3)** that **the descriptions cannot be otherwise expressed**. For that, (1 2, **3**), we turn, after an example, to **philosophy**. **Can it give us advice?**

2 The Example

We refer to Fig. **??** [pp. ??] for three "renditions" or road nets.

2.1 Endurants

2.1.1 Structures

1 There is the *universe of discourse*, UoD.

From that universe we can **obs**erve:

2 a *road net*, RN, a structure, and

3 a *fleet of automobiles*, FA, a structure.

type
1 UoD **axiom** ∀ uod:UoD • is_structure(uod).
2 RN **axiom** ∀ rn:RN • is_strucure(rn).
3 FA **axiom** ∀ fa:FA • is_structure(fa).
value
2 obs_RN: UoD → RN
3 obs_FA: UoD → FA

4 The road net consists of

a a structure, SH, of hubs and

b a structure, SL, of links.

5 The fleet of automobiles consists of

a a set, As of automobiles.

type
4a SH **axiom** ∀ sh:SH • is_structure(sh)
4b SL **axiom** ∀ sl:SL • is_structure(sl)
5a As = A-set
value
4a obs_SH: RN → SH
4b obs_SL: RN → SL
5a obs_As: FA → As

2.1.2 Parts

.

6 The structure of hubs is a set, sH, of atomic hubs, H.

7 The structure of links is a set, sL, of atomic links, L.

8 The structure of automobiles is a set, sA, of atomic automobiles, A.

type
6 H, sH = **H-set** **axiom** ∀ h:H • is_atomic(h)
7 L, sL = **L-set** **axiom** ∀ l:L • is_atomic(l)
8 A, sA = **A-set** **axiom** ∀ a:A • is_atomic(a)
value
6 obs_sH: SH → sH
7 obs_sL: SL → sL
8 obs_sA: SA → sA

2.1.3 Components

To illustrate the concept of components we describe timber yards, waste disposal areas, road material storage yards, automobile scrap yards, end the like as special "cul de sac" hubs with components. Here we describe road material storage yards.

9 Hubs may contain components, but only if the hub is connected to exactly one link.

10 These "cul-de-sac" hub components may be such things as Sand, Gravel, Cobble Stones, Asphalt, Cement or other.

value
9 has_components: H → **Bool**
type
10 Sand, Gravel, CobbleStones, Asphalt, Cement, ...
10 KS = (Sand|Gravel|CobbleStones|Asphalt|Cement|...)-**set**
value
9 obs_components_H: H → KS
9 **pre**: obs_components_H(h) ≡ **card** mereo(h) = 1

2.1.4 Materials

To illustrate the concept of materials we describe waterways (river, canals, lakes, the open sea) along links as links with material of type water.

11 Links may contain material.

12 That material is water, W.

type
12 W
value
11 obs_material: L → W
11 **pre**: obs_material(l) ≡ has_material(h)

2.1.5 States

13 Let there be given a universe of discourse, *rts*, a state.

From that state we can calculate other states.

14 The set of all hubs, *hs*.

15 The set of all links, *ls*.

16 The set of all hubs and links, *hls*.

17 The set of all automobiles, *as*.

18 The set of all parts, *ps*.

value
13 *rts*:UoD
14 *hs*:H-**set** ≡ obs_sH(obs_SH(obs_RN(*rts*)))
15 *ls*:L-**set** ≡ obs_sL(obs_SL(obs_RN(*rts*)))
16 *hls*:(H|L)-**set** ≡ *hs*∪*ls*
17 *as*:A-**set** ≡ obs_As(obs_FV(*rts*))
18 *ps*:(H|L|BC|B|A)-**set** ≡ *hls*∪*bcs*∪*bs*∪*as*

2.1.6 Unique Identifiers

19 We assign unique identifiers to all parts.

20 By a road identifier we shall mean a link or a hub identifier.

21 Unique identifiers uniquely identify all parts.

a All hubs have distinct [unique] identifiers.

b All links have distinct identifiers.

c All automobiles have distinct identifiers.

type
19 H_UI, L_UI, A_UI
20 R_UI = H_UI | L_UI
value
21a uid_H: H → H_UI
21b uid_L: L → L_UI
21c uid_A: A → A_UI

2.1.7 Mereologies

Mereology is the study and knowledge of parts and part relations. The parts here are the hubs, the links and the automobiles.

22 The mereology of a hub is a pair: (i) the set of all automobile identifiers that may use the hub and (ii) the set of unique identifiers of the links that it is connected to.

type
22 H_Mer = A_UI-**set**×L_UI-**set**
value
22 mereo_H: H → H_Mer

23 The mereology of a link is a pair: (i) the set of identifiers all automobiles that may use the link, (ii) the set of identifiers of the two distinct hubs it is connected to.

type
23 L_Mer = A_UI-**set**×H_UI-**set**
value
23 mereo_L: L → L_Mer

24 The mereology of an automobile is: the set of the unique identifiers of all hubs and links on which they may travel.

type
24 A_Mer = (H_UI|L_UI)-**set**
value
24 mereo_A: A → A_Mer

2.1.8 Attributes

Hubs: We show just one attribute:

25 Hub traffic history. Since we can think rationally about it, it can be described. We model hub traffic history as an attribute: the recording, per unique automobile identifier, of the time ordered presence, APos, in the hub of these automobiles.

```
type
25  H_Traffic = A_UI →m (𝒯 × APos)*
axiom
25  ∀ ht:H_Traffic,ui:A_UI • ui ∈ dom ht ⇒ time_ordered(ht(ui))
value
25  attr_H_Traffic: : → H_Traffic
```

Links: We show just one attribute:

26 Link traffic history: Since we can think rationally about it, it can be described. We model link traffic history as an attribute: the recording, per unique automobile identifier, of the time ordered positions, APos (along the link (from one hub to the next)), of these automobiles.

```
26  L_Traffic = A_UI →m (𝒯×APos)*
axiom
26  ∀ lt:L_Traffic,ui:A_UI • ui ∈ dom lt ⇒ time_ordered(lt(ui))
value
26  attr_L_Traffic: : → L_Traffic
```

Automobiles: We show just a few attributes: We illustrate but a few attributes:

27 Automobiles have a time attribute,

28 Automobiles have dynamic positions on the road net:

- a either *at a hub* identified by some h_ui,
- b or *on a link*, some *fraction, frac:Fract* down an *identified link, l_ui*, from one of its *identified connecting hubs*, fh_ui, in the direction of the other *identified hub*, th_ui.
- c Automobiles, like elephants, never forget: they remember their timed positions of the past,
- d and the current position is the first element of this past !

```
type
27    𝒯
28    APos    == atHub | onLink
28a  atHub    :: h_ui:H_UI
28b  onLink   :: fh_ui:H_UI×l_ui:L_UI×frac:Fract×th_ui:H_UI
28b  Fract    = Real
axiom
28b  frac:Fract • 0<frac≪1
type
28c  A_Hist  = (T × APos)*
value
27    attr_T: A → 𝒯
28    attr_APos: A → APos
28c  attr_A_Hist: A → A_Hist
axiom
28d  □ ∀ a:A •
28d     let (_,apos) = hd(attr_A_Hist(a)) in
28d     apos = attr_APos(a) end
```

2.1.9 Summary of Endurants

We have illustrated the description of *external qualities* of a domain: *structures*, *parts: composite and atomic*, *components* and *materials*; and *internal qualities* of that domain: *unique identification*, *mereology* and *attributes*.

2.2 Transcendentality

Example 1 **A Case of Transcendentality**: We refer to the following example: We can speak of an automobile in at least three *senses:*

- The automobile as it is being maintained, serviced, refueled;
- the automobile as it "speeds" down its route; and
- the automobile as it "appears" (listed) in car registries or advertisements.

The three *senses* are:

- as a part,
- as a behaviour, and
- as an attribute[1] ■

Section 2.3 *transcendentally interprets endurant automobiles* as *perdurant automobiles*. Details of the translation is given in [5].

2.3 Perdurants

2.3.1 Signatures

29 $\text{automobile}_{a_{ui}}$:

- a there is the usual "triplet" of arguments: unique identifier, mereology and static attributes;
- b then there is the one programmable attribute;
- c and finally there are the input/output channel references: first the input time channel,
- d then the input/output allowing communication between the automobile and the hub and link behaviours.

```
value
29    automobile_{a_ui}:
29a    a_ui:A_UI×(__,__,ruis):A_Mer×rn:RegNo
29b      → apos:APos
29c      → in attr_T_ch
29d         in,out {a_r_ch[a_ui,r_ui]
29d                | r_ui:(H_UI|L_UI)•r_ui∈ruis} Unit
29a      pre: ruis = r_ui s ∧ a_ui ∈ a_ui s
```

2.3.2 Behaviours

We define the behaviours in a different order than the treatment of their signatures. We "split" definition of the automobile behaviour into the behaviour of automobiles when positioned at a hub, and into the behaviour automobiles when positioned at on a link. In both cases the behaviours include the "idling" of the automobile, i.e., its "not moving", standing still.

2.3.3 Automobiles:

30 We abstract automobile behaviour at a Hub (hui).

31 The automobile remains at that hub, "idling",

32 informing the hub behaviour,

33 or, internally non-deterministically,

- a moves onto a link, tli, whose "next" hub, identified by th_ui, is obtained from the mereology of the link identified by tl_ui;
- b informs the hub it is leaving and the link it is entering of its initial link position,
- c whereupon the automobile resumes the automobile behaviour positioned at the very beginning (0) of that link,

34 or, again internally non-deterministically,

35 the automobile "disappears — off the radar" !

```
30  automobile_{a_ui}(a_ui,({},(ruis,auis),{}),rn)
30           (apos:atH(fl_ui,h_ui,tl_ui)) ≡
31     (ba_r_ch[a_ui,h_ui] ! (attr_T_ch?,atH(fl_ui,h_ui,tl_ui));
32      automobile_{a_ui}(a_ui,({},(ruis,auis),{}),rn)(apos))
33     ⌈⌉
33a    (let ({fh_ui,th_ui},ruis')=mereo_L(℘(tl_ui)) in
33a          assert: fh_ui=h_ui ∧ ruis=ruis'
30      let onl = (tl_ui,h_ui,0,th_ui) in
33b     (ba_r_ch[a_ui,h_ui] ! (attr_T_ch?,onL(onl)) ||
33b      ba_r_ch[a_ui,tl_ui] ! (attr_T_ch?,onL(onl))) ;
33c      automobile_{a_ui}(a_ui,({},(ruis,auis),{}),rn)
33c            (onL(onl)) end end)
34     ⌈⌉
35      stop
```

36 We abstract automobile behaviour on a Link.

- a Internally non-deterministically, either
  - i the automobile remains, "idling", i.e., not moving, on the link,
  - ii however, first informing the link of its position,
- b or
  - i **if** if the automobile's position on the link *has not yet reached the hub*, **then**
    - A then the automobile moves an arbitrary small, positive **Real**-valued *increment* along the link
    - B informing the hub of this new position,
    - C while resuming being an automobile at the new position, or
  - ii **else**,

[1] in this case rather: as a fragment of an attribute

A while obtaining a "next link" from the mereology of the hub (where that next link could very well be the same as the link the automobile is about to leave),

B the vehicle informs both the link and the imminent hub that it is now at that hub, identified by th_ui,

C whereupon the automobile resumes the vehicle behaviour positioned at that hub;

c or

d the automobile "disappears — off the radar" !

```
36      automobile_{a_ui}(a_ui,({},ruis,{}),rno)
36                    (vp:onL(fh_ui,l_ui,f,th_ui)) ≡
36(a)ii   (ba_r_ch[ thui,aui ]!atH(lui,thui,nxt_lui) ;
36(a)i     automobile_{a_ui}(a_ui,({},ruis,{}),rno)(vp))
36b      ⊓
36(b)i    (if not_yet_at_hub(f)
36(b)i      then
36(b)iA       (let incr = increment(f) in
30             let onl = (tl_ui,h_ui,incr,th_ui) in
36(b)iB         a−r_ch[ l_ui,a_ui ] ! onL(onl) ;
36(b)iC         automobile_{a_ui}(a_ui,({},ruis,{}),rno)
36(b)iC                          (onL(onl))
36(b)i          end end)
36(b)ii     else
36(b)iiA      (let nxt_lui:L_UI•nxt_lui ∈ mereo_H(℘(th_ui)) in
36(b)iiB       a_r_ch[ thui,aui ]!atH(l_ui,th_ui,nxt_lui) ;
36(b)iiC       automobile_{a_ui}(a_ui,({},ruis,{}),rno)
36(b)iiC                        (atH(l_ui,th_ui,nxt_lui)) end)
36(b)i     end)
36c      ⊓
36d        stop
36(b)iA    increment: Fract → Fract
```

3 A Preview of Description Composition and Elements

3.1 "Standard" Domains

Figure 2 Pg. 8 illustrates the generic **composition** of descriptions – the various "branches" of the diagram, and their **elements** – the nodes of the diagram. Figure 2 Pg. 8 intends to show that domains consists of **endurants** ($\mathbf{E}_i$) and **perdurants**; that endurants are either **discrete** or **continuous**; and that discrete endurants are either **structures**, **parts**, or **compoments**; That is: that domains possibly contain all these kinds of elements.

3.2 Influences from Studies of Philosophy, I

Our *study of philosophy unmistakably mandates us to express (— something that all sensible people know —) but only rational, philosophical reasoning can mandate* that besides the *discrete endurants* of *structures*, *parts* and *components*, (already shown) there are also **living species: plants** and **animals**!

3.3 Domain Science & Engineering is Different

As you might now see, the concerns of **domain science & engineering** are different from those of **automation** and **cybernetics**, **optimisation** and **operations research** the sciences & engineering of **electricity**, the sciences & engineering of **electronics**, the sciences & engineering of **chemistry**, the sciences & engineering of **mechanics**, the sciences & engineering of **aerodynamics**, et cetera

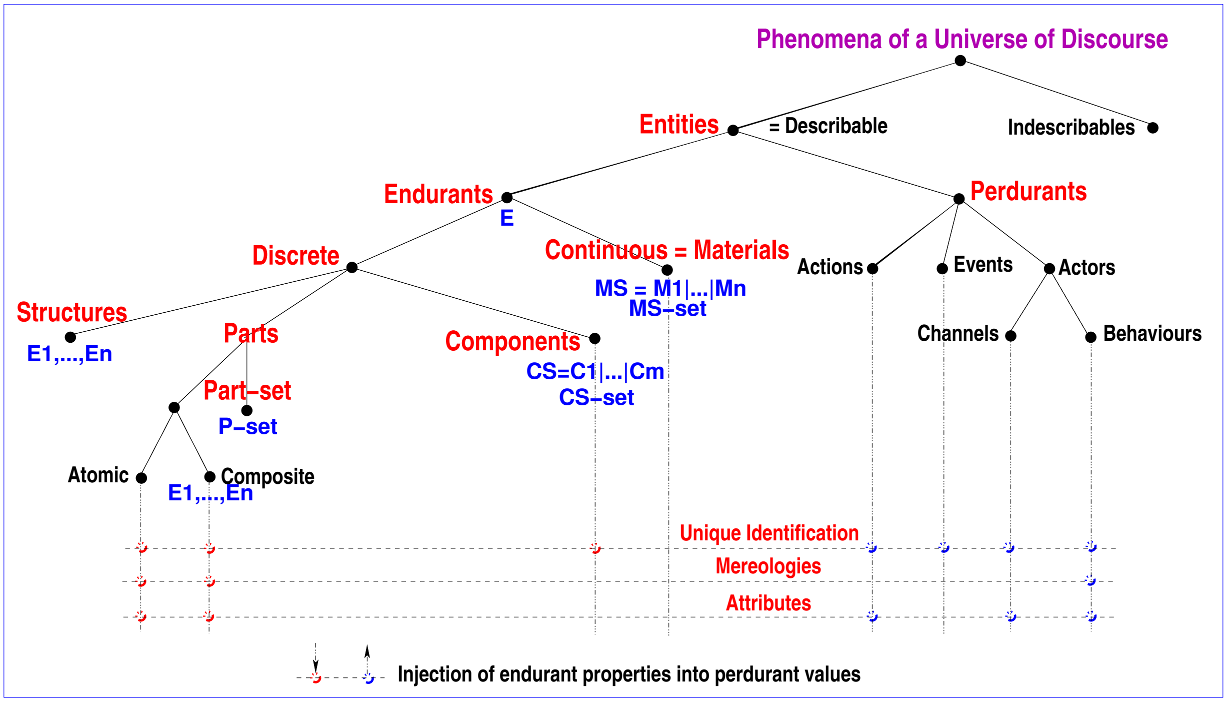

Figure 4: An Initial Upper Ontology for Domains

4 Endurant Qualities: External and Internal

4.1 External Qualities

By *external qualities* of *endurants* we man whether they are *discrete* or *continuous* and, if discrete, whether they are *structures*, *physical* parts *artifacts* or *components*; and if physical parts or artifacts whether they are *atomic* or *composite*. We refer to Fig. 2, and to Items 1 [pp. 3] – 8 [pp. 3], of Sects. 2.1.1–2.1.4. All of these external qualities are observable but can be justified from a point of view of Philosophy.

4.2 Internal Qualities

Usually internal qualities are not observable.

4.2.1 Unique Identification

We can (abstractly) speak of discrete endurants having unique identifies. From the point of view of philosophy uniqueness of discrete endurants follows from our ability to express one predicate of one discrete endurant and a therefrom different predicate of another discrete endurant. The two discrete endurants must therefore have distinct identification. We refer to Items 19– 21c [pp. 4], Sect. 2.1.6.

4.2.2 Mereology

Mereology is the study and knowledge of parts and part relations. Mereology, as a logical/philosophical discipline, can perhaps best be attributed to the Polish mathematician/logician Stanisław Leśniewski [31]. Part relations span from topological to conceptual. Two physical parts that are spatially adjacent, connected, i.e., "touch one another", are related. Two artifacts, of which at least one is mobile, that, at one time or another are spatially adjacent, connected, i.e., "touch one another", are related. We refer to Items 22– 24 [pp. 4], Sect. **2.1.7**.

4.2.3 Attributes

To recall: there are three sets of **internal qualities**: unique part identifiers, part mereology and attributes. Unique part identifiers and part mereology are rather definite kinds of internal endurant qualities. Part attributes form a more "free-wheeling" sets of **internal qualities**.

Possessing attributes types and values form a main basis for expressing propositions about endurants and are thus central to our study of domain science & engineering. We refer to Items 25 [pp. 5]– 28d [pp. 5], Sect. **2.1.8**.

5 Preview: First Lessons of Philosophy for Domain Science & Engineering

We show how the domain analysis & description calculi of [5] satisfy the Philosophy of Kai Sørlander, but also that Sørlander's Philosophy mandates consistent extensions to the calculi (of [5]) in order to form a more complete "whole". Where discrete parts were just that, we must now distinguish between three kinds of parts: (i) **physical parts**, (ii) **living species parts**, and (iii) **artifacts**.

5.1 Physical Parts

(i) **Physical parts** are parts that are not made by man, but are in *space* and *time*, given in *nature*, parts that are subject to the *laws* of physics as formulated by for example *Newton* and *Einstein*, and also subject to the *principle of causality* and *gravitational pull*. They are the parts we treated in [5], without, however, referring to them as such physical parts.

5.2 Living Species

(ii) The **living species parts**, **plants** and **animals**; still subject to the laws and principles of physics, but additionally **unavoidably** endowed with such properties as **causality of purpose**, **Animals additionally** have **sensory organs**, **means of motion**, **instincts**, **incentives** and **feelings**. *We can speak of these [***red***] "things", but maybe we cannot measure them !*

5.3 Humans

Among animals we single out **humans** as parts that are further characterisable: possessing **language**, **learning skills**, being **consciousness**, and having **knowledge**. These aspects were somehow, by us, subsumed in our analysis & description by partially endowing *artifacts* with such properties. We refer to

5.4 Artifacts

(iii) **Artifacts** are the parts made by humans. *Artifacts* have a usual set of attributes of the kind *physical parts* can have; but in addition they have a *distinguished attribute:* **attr_Intent** – expressed as a set of intents by the *humans* who constructed them according to some *purpose*. This more-or-less "standard" *property of intents* determines a form of **counterpart** to the *gravitational pull* of *physical parts* namely, what we shall refer to as **intentional "pull"**.

5.5 A Final Upper Ontology

Figure 3 "merges" the *living species* and *artifact* parts into the upper ontology of Fig. 2 [pp. 8].

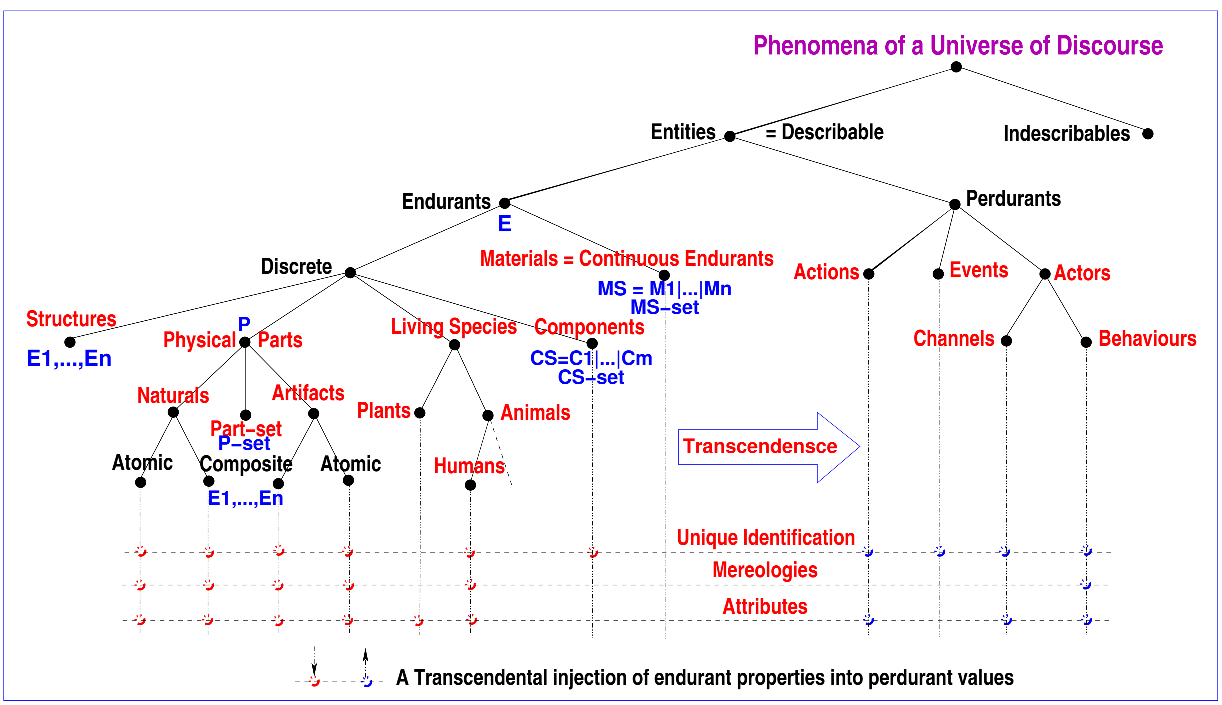

Figure 5: A Final Upper Ontology for Domains

5.6 Influences from Studies of Philosophy, III

5.6.1 Transcendental Deductions

A transcendental argument is a deductive philosophical argument which takes a manifest feature of experience as granted, and articulates which must be the case so that experience as such is possible. Transcendental deductions we introduced into philosophy by **Immanuel Kant** – around 1772.

5.6.2 An Example

- The **bus** standing there is an **endurant**.
- The **bus** "speeding down" its route is a **perdurant**.
- The **bus** as it is listed in the time-table is an **attribute**.

When we claim that the *endurant* (bus) is the "same" as the *perdurant* (bus) then our "claim" is a *transcendental deduction!*

5.6.3 Another Example

We speak of **syntax:** f.ex.: of **programs** in a programming language, and of **semantics:** f.ex.: the **compiled code** of a (the) program. The latter can only by claimed so by a *transcendental deduction!* Thus all *abstract interpretations* of computer program texts: static analysis, model checks, program verification, execution, et cetera are *transcendental deductions!*

6 The Kai Sørlander's Philosophy

6.1 Basic Issues

We present an account of how the Kai Sørlander's Philosophy is argued.

The question is **'what are the necessary characteristics of each and every possible world and our situation in it'.**

To carry out his reasoning Sørlander establishes a number of criteria.

6.1.1 The Inescapable Meaning Assignment

The Inescapable Meaning Assignment

- The *The Inescapable Meaning Assignment*
  is the recognition of the mutual dependency between

- *the meaning of designations and*
- *the consistency relations between propositions.*

6.1.2 An Example: Stacks

Meaning of Designations: Narrative

37 Stacks, s:S, have elements, e:E;

38 the empty_S operation takes no arguments and yields a result stack;

39 the is_empty_S operation takes an argument stack and yields a Boolean value result.

40 the stack operation takes two arguments: an element and a stack and yields a result stack.

41 the unstack operation takes an non-empty argument stack and yields a stack result.

42 the top operation takes an non-empty argument stack and yields an element result.

Consistency Relations: Narrative

43 an empty_S stack is_empty, and a stack with at least one element is not;

44 unstacking an argument stack, stack(e,s), results in the stack s; and

45 inquiring as to the top of a non-empty argument stack, stack(e,s), yields e.

Meaning of Designations: Formal

type
37. E, S
value
38. empty_S: **Unit** $\rightarrow$ S
39. is_empty_S: S $\rightarrow$ **Bool**
40. stack: E $\times$ S $\rightarrow$ S
41. unstack: S $\xrightarrow{\sim}$ S
42. top: S $\xrightarrow{\sim}$ E

Consistency Relations: Formal

43. is_empty(empty_S()) = **true**
43. is_empty(stack(e,s)) = **false**
44. unstack(stack(e,s)) = s
45. top(stack(e,s)) = e

That *the inescapable meaning assignment* is required in order to answer the question of how the world must necessarily be can be seen from the following It makes it possible to distinguish between necessary and empirical propositions **A proposition is necessary** if its truth value depends only on the meaning of the designators by means of which it is expressed

Example 2 **A Proposition which is Necessary**: *The link (i.e. the street segment) is 100 meters long* ■

A proposition is empirical if its truth value does not so depend. An empirical proposition must therefore refer to something which exists independently of its designators, and it must predicate something about the thing to which it refers

Example 3 **A Proposition which is Empirical**: *The link (i.e. the street segment) is the longest link in the road net* ■

The definition *"the world is all that is the case; all that can be described in true propositions"* satisfies *the inescapable meaning assignment*. That which is described in `necessary` propositions is that which is common to [all] possible worlds. A concrete world is all that can be described in true `empirical` propositions

6.1.3 Primary Objects

An empirical proposition must refer to an independently existing thing and must predicate something about that thing. On that basis it is then possible to deduce how those objects that can be directly referred to in simple empirical propositions must necessarily be. Those things are referred to as `primary objects`. A deduction of the `inevitable characteristics` of a possible world is thus identical to a deduction of how primary objects must necessarily be.

6.1.4 Two Requirements to the Philosophical Basis

Two demands have been put to the philosophical basis for our quest. It must not contain empirical preconditions; and the foundation must not consistently be refuted. It must not consistently be false. **The inescapable meaning assignment** satisfies this basis.

6.1.5 The Possibility of Truth

Where Kant builds on the *contradictory* dichotomy of *Das Ding an sich* and *Das Ding für uns*, that is, the possibility of *self-awareness*, Kai Sørlander builds on the *possibility of truth*: Since the possibility of truth cannot in a consistent manner be denied we can hence assume the **contradiction principle:** 'a proposition and its negation cannot both be true'. We assume that the contradiction principle is a *necessary truth.*

6.1.6 The Logical Connectives

Sørlander now deduces the logical connectives: *conjunction* ('and' $\wedge$), *disjunction* ('or', $\vee$), and *implication* ($\Rightarrow$ or $\supset$). That is, they are not taken for granted: They can be deduced!

6.1.7 Necessity and Possibility

A *proposition* is *necessarily true*, **if** its truth follows from the definition of of the designations by means of which it is expressed; **then** it must be true under all circumstances. A *proposition* is *possibly true*, if its *negation* is not *necessarily true*.

6.1.8 Empirical Propositions

An *empirical proposition* refers to an independently existing entities and predicates something that can be either true or false about the referenced entity.

6.2 The Logical Conditions for Describing Physical Worlds

So which are the logical conditions of descriptions of any world ? In [1] and [4] Kai Sørlander, through a series of transcendental deductions "unravels" the following **logical conditions:**

- *symmetry* and *asymmetry*
- *transitivity* and *intransitivity*,
- **space:** *direction, distance*, etc.,
- **time:** *before, after*, in-between etc.,
- **states** and **causality**,
- **kinematics, dynamics**, etc., and
- **Newton's laws**, et cetera.

We shall summarise Sørlander's deductions. To remind the reader: the issue is that of deducing how the *primary entities* must necessarily be.

6.2.1 Symmetry and Asymmetry

There can be **different** *primary entities*. Entity A is *different* from entity B if A can be ascribed a predicate in-commensurable with a predicate ascribed to B. *Different from* is a *symmetric predicate*. If entity A is *identical* to entity B then A cannot be ascribed a predicate which is in-commensurable with any predicate that can be ascribed to B; and then B is identical to A. *Equal to* is a **symmetric predicate**.

6.2.2 Transitivity and Intransitivity

If A is identical to B and B is identical to C then A is identical to C with *identity* then being a *transitive relation*. The relation *different from* is not transitive it is an **transitive relation**.

6.2.3 Space

The two relations *asymmetric* and *symmetric*, by a transcendental deduction, can be given an interpretation: The relation (spatial) *direction* is asymmetric; and the relation (spatial) *distance* is symmetric. From these relations are derived the relation *in-between*. Direction, distance and in-between can, by a transcendental argument, be understood as spatial relations. Hence we must conclude that *primary entities exist in space*. **Space is therefore an unavoidable characteristic of any possible world**. From the direction and distance relations one can derive *Euclidean Geometry*.

6.2.4 States

We must assume that primary entities may be ascribed predicates which are not logically required. That is, they may be ascribed predicates incompatible with predicates which they actually satisfy — in order for it to be logically possible, that one-and-the-same *primary entity* can be ascribed incompatible predicates, if any primary entity can exist in different *state*s. A *primary entity* may be in *one state* where it can be ascribed one predicate, and in *another state* where it can be ascribed another incompatible predicate. **Any entity in every possible world may attain different states.**

6.2.5 Time

Two such different states must necessarily be ascribed different incompatible predicates. But how can we ensure so? Only if states stand in an asymmetric relation to one another. This state relation is also transitive. So that is an indispensable property of any world. By a transcendental deduction we say that *primary entities exist in time*. **So every possible world must exist in time**.

6.2.6 Causality

States are related by the *time relation*s "before" and "after". These are asymmetric and transitive relations. But how can it be so? Propositions about primary entities at different times must necessarily be logically independent of one another. This follows from the possibility that a primary entity necessarily be ascribed different, incompatible predicates at different times. It is therefore logically **impossible** from the primary entities alone to deduce how a primary entity is at on time point to how it is at another time point. How, therefore, can these predicates supposedly of one and the same entity at different time points be about the *same entity?* There can be no logical implication about this! Transcendentally therefore there must be a *non-logical implicative* between propositions about properties of a primary entity at different times. Such an *non-logical implicative* must depend on *empirical circumstances* subject to which the primary entity exists. There are no other circumstances. If the state on a primary entity changes then there must be changes in its "circumstances" whose consequences are that the primary entity changes state. And such "circumstance"–changes **will imply primary entity state changes**. We shall use the term **'cause'** for a preceding "circumstance"–change that implies a state change of a primary entity. So now we can conclude *that every change of state of a primary entity must have a cause*, and *that "equivalent circumstances" must have "equivalent effects"*. This form of implication is called **causal implication**. And the principle of implication for **causal principle**. So every possible world enjoys the *causal principle*. Kant's transcendental deduction is fundamentally built on the the *possibility of self-awareness*. Sørlander's transcendental deduction is fundamentally built on the *possibility of truth*. In Kant's thinking the *causal principle* is a prerequisite for possibility of self-awareness. In this way Sørlander avoids Kant's *solipsism*, i.e., *"that only one's own mind is sure to exist"* a solipsism that, however, flaws Kant's otherwise great thinking.

6.2.7 Rejection, also, of Hegel's Philosophy

Just as we reject *Descartes, Spinoza's, Locke's, Berkeley's, Hume's,* and *Kant's* Philosophies – *as leading to* **contradictions**, so we must reject *Hegel's* Philosophy: We must reject Hegel's *thesis, antithesis, synthesis*. By relativising philosophy wrt. history Hegel has removed necessity. By thus postulating that *"it is an eternal truth that we cannot achieve eternal truths"*. *Hegel's main contribution ends up in* **contradiction**.

6.2.8 Kinematics

So *primary entities* exist in *space* and *time*. They must have *spatial extent* and *temporal extent*. They must therefore be able to *change* their *spatial properties*. Both as concerns *form* and *location*. But a spatial change in form presupposes a change in location – as the more fundamental. A *primary entity* which changes location is said to be *moving*. If a *primary entity* which does not change location is said to be *resting*. The *velocity* of a primary entity expresses the distance and direction it moves in a given time interval. Change in velocity of a primary entity is called its *acceleration*. Acceleration involves either change in velocity, or change in direction of movement, or both. **So far Sørlander has reasoned us to fundamental concepts of kinematics.**

6.2.9 Dynamics

When we "add" causality" to kinematics we obtain *dynamics*. We can do so, because primary entities are in time. Kinematics imply that that a primary entity changes when it goes from being at rest to moving. Likewise when it goes from movement to rest. Et cetera. So a primary entity has same state of movement if it has same velocity and moves in the same direction. Primary entities change state of movement if they change velocity or direction. So, combining kinematics and the principle of causality, we can deduce that **if** a primary entity changes state of movement **then** there must be a cause, and we call that cause a **force**.

6.2.10 Newton's Laws

Newton's First Law: Combining *kinematics* and the *principle of causality*, and the therefrom deduced concept of *force*, we can deduce that any *change of movement* is proportional[2] to the *force*. This implies that a primary entity which is not under the influence of an external force will continue in the same state of movement – that is, be at rest or conduction a linear movement at constant velocity. **This is Newton's First Law**. **Newton's Second Law:** That a certain, non-zero force implies change of movement, imply that the primary entity must excert a certain *resistance* to that change. It must have what we shall call a certain *mass*.[3] From this it follows that *the change in the state of movement of a primary entity.* not only is proportional to the excerted force, but also inversely proportional[4] to the mass of that entity. **This is Newton's Second Law**. **Newton's Third Law:** In a possible world, the forces that affect primary entities must come from "other" primary entities. Primary entities are located in different volumes of space. Their location may interfere with one another in the sense at least of "obstructing" their mutual movements – leading to clashes. In principle we must assume that even primary entities "far away from one another" obstruct. If they clash it must be with *oppositely directed and equal forces*. **This is Newton's Third Law**.

6.2.11 Gravitation and Quantum Mechanics

Mutual Attraction: How can primary entities possibly be the *source* of *force*s that *influence* one another ? How can primary entities at all have a *mass*[5] such that it requires *force*s to change their *state of movement* ? The answer must be that primary entities *excert* a *mutual influence* on one another – that is there is a *mutual attraction*. **Gravitation:** This must be the case for all primary entities. This must mean that all primary entities can be characterised by a *universal mutual attraction*: a *universal gravitation* **Finite Propagation – A Gravitational Constant:** Thus *mutual attraction* must *propagate* at a certain, finite, velocity. If that velocity was infinite, then it is everywhere and cannot therefore have its source in concretely existing primary entities. But having a finite velocity implies that there must be a *propagational speed limit*. It must be a *constant of nature*. **Gravitational "Pull":** The nature of *gravitational "pull"* can be deduced, basically as follows: *Primary entities must basically consist of elements that attract one another, but which are stable, and that is only possible if it is, in principle, impossible to describe these elementary particles precisely. If there is a fundamental limit to how these basic particles can be described, then it is also precluded that they can undergo continuous change. Hence there is a basis for stability despite mutual attraction. There must be a foundational limit for* **how precise these**

[2] Observe that we have "only" said: *proportional*, meaning also directly proportional, not whether it is logarithmically, or linearly, or polynomially, or exponentially, ..., so.

[3] *Mass* refers loosely to the amount of *matter* in an entity. This is in contrast to *weight* which refers to the *force* exerted on an entity by *gravity*.

[4] Cf. Footnote 2.

[5] cf. Footnote 3 Pg. 17

descriptions can be — *which implies that* **the elementary particle as a whole can be described statistically**. **Quantum Mechanics:** The rest is physics: unification of quantum mechanics and Einstein's special relativity has been done; unification of gravitation with Einstein's general theory of relativity has still to be done. **A Summary:** Philosophy lends to physics its results a necessity that physics cannot give them. Experiments have shown that Einstein's results – with propagation limits – indeed hold for this world. Philosophy shows that every possible world is subject to a fixed propagation limit. Philosophy also shows that for a possible world to exist it must be built from elementary particles which cannot be individually described (with Newton's theory)

6.3 The Logical Conditions for Describing Living Species

6.3.1 Purpose, Life and Evolution

Causality of Purpose: **If** there is to be *the possibility of language and meaning*, **then** there must exist primary entities which are *not entirely encapsulated within the physical conditions*; that they are stable and can influence one another. This is only possible if such primary entities are subject to a *supplementary causality directed at the future*: a **causality of purpose** These primary entities are here called **living species**. **Living Species:** What can be deduced about them ? **They must have some form they can be developed to reach which they must be causally determined to maintain**. This development and maintenance must further in **an exchange of matter with an environment**. . . . It must be possible that living species occur in one of two forms: one form which is characterised by **development, form** and **exchange**, and another form which, **additionally**, can be characterised by the ability to **purposeful movement**s. The first we call **plant**s, the second we call **animal**s.

Animate Entities: For an animal to purposefully move around there must be "additional conditions" for such self-movements to be in accordance with the principle of causality: [**(i)**] they must have **sensory organ**s sensing among others the immediate purpose of its movement; [**(ii)**] they must have **means of motion** so that it can move; and [**(iii)**] they must have **instinct**s, **incentive**s and **feeling**s as causal conditions that what it senses can drive it to movements. And all of this **in accordance with the laws of physics.**

Animal Structure: Animals, to possess these three kinds of "additional conditions", must be built from special units which have an inner relation to their function as a whole; Their **purposefulness** must be built into their physical building units, that is, as we can now say, their **genome**s. That is, animals are built from genomes which give them the **inner determination** to such building blocks for **instinct**s, **incentive**s and **feeling**s. Similar kinds of deduction can be carried out with respect to plants. **Transcendentally one can deduce basic principles of evolution** but not their details.

6.3.2 Consciousness, Learning and Language

The existence of animals is a necessary condition for there being language and meaning in any world. That there can be **language** means that animals are capable of *developing language*. And this must presuppose that animals can **learn** *from their experience*. To learn implies that animals can **feel** pleasure and distaste. One can therefore deduce that animals must possess such building blocks whose inner determination is a basis for learning and **consciousness**. **Language:** Animals with higher social interaction uses **sign**s, eventually developing a **language**. These languages adhere to the same system of defined concepts **which are a prerequisite for any description of any world: namely the system that philosophy lays bare from a basis of transcendental deductions** and **the principle of contradiction** and **its implicit meaning theory**.

6.3.3 Humans, Consciousness and Knowledge

A human is an animal which has a language. Humans must be **conscious** of having **knowledge** of its concrete situation, and as such that person can have knowledge about what he feels and eventually that person can know whether what he feels is true or false. Consequently *a human can describe his situation correctly*.

6.3.4 Responsibility

In this way one can deduce that **humans** can thus have **memory** and hence can have **responsibility** , be **responsible** . Further deductions lead us into **ethics** .

7 The Main Example Continued: Intentional "Pull"

We refer to the example of Sect. 2. The *human-assistedness* of our main example is reflected in the *automobile* artifacts. We do not describe, i.e. model, humans. Instead we let automobiles subsume human character. The *artifacts* of our main example are those of the road net and the autombiles.

46 To automobiles we ascribe an *intent* of *transport*.

47 And to road hubs and links we ascribe an *intent* of *transport*.

48 Seen from the point of view of an automobile there is its own traffic history, A_Hist Item 28c Pg. 5, which is a (time ordered) sequence of timed automobile positions;

49 seen from the point of view of a hub there is its own traffic history, H_Traffic Item 25 Pg. 5, which is a (time ordered) sequence of timed maps from automobile identities into automobile positions; and

50 seen from the point of view of a link there is its own traffic history, L_Traffic Item 26 Pg. 5, which is a (time ordered) sequence of timed maps from automobile identities into automobile positions.

The *intentional "pull"* of these manifestations is this:

51 The union, i.e. proper merge of all automobile traffic histories, AllATH, must now be identical to the same proper merge of all hub, AllHTH, and all link traffic histories, AllLTH.

type
28c, pp.5 A_Hist $= (\mathcal{T} \times \mathsf{APos})^*$
25, pp.5 H_Traffic = A_UI $\overrightarrow{m}$ $(\mathcal{T} \times \mathsf{APos})^*$
26, pp.5 L_Traffic = A_UI $\overrightarrow{m}$ $(\mathcal{T} \times \mathsf{APos})^*$
51 AllATH = $\mathcal{T}$ $\overrightarrow{m}$ (AUI $\overrightarrow{m}$ APos)
51 AllHTH = $\mathcal{T}$ $\overrightarrow{m}$ (AUI $\overrightarrow{m}$ APos)
51 AllLTH = $\mathcal{T}$ $\overrightarrow{m}$ (AUI $\overrightarrow{m}$ APos)
axiom
51 **let** allA = proper_merge_into_AllATH({(a,attr_A_Hist(a))|a:A•a $\in$ as}),
51 allH = proper_merge_into_AllHTH({attr_H_Traffic(h)|h:H•h $\in$ hs}),
51 allL = proper_merge_into_AllLTH({attr_L_Traffic(l)|l:L•h $\in$ ls}) **in**
51 allA = H_and_L_Traffic_merge(allH,allL) **end**

We leave the definition of the four merge functions to the reader! We now discuss the concept of *intentional "pull"*. To each automobile we can, of course, associate its history of timed positions and to each hub and link, similarly their histories of timed automobile positions. These histories are facts! They are not something that is laboriously recorded, where such recordings may be imprecise or cumbersome[6]. The facts are there, so we can, but may not necessarily, talk about these histories as facts. It is in that sense that the purpose ('transport') for which man let automobiles, hubs and link be made with their 'transport' intent are subject to an *intentional "pull"*. It can be no other way: if automobiles "record" their history, then hubs and links must together "record" identically the same history! ■ We have tentatively proposed a concept of *intentional "pull"*. That proposal is in the form, I think, of a transcendental deduction; it has to be further studied.

8 Closing

We have introduced two major and **new**, concepts: (i) **domain analysis & description** *as a precursor to software development*; and (ii) **philosophy** *as a basis for determining major elements on a domain analysis & description method*. We claim these, (i) and (ii), as **new** elements of computer science.

[6] or thought technologically in-feasible – at least some decades ago!

9 Acknowledgements

The author is immensely indebted to Kai Sørlander for the wonderful insight he has gained from his books.

10 Bibliography

Notes

We apologise for the large number of references to own work.

References

[1] Kai Sørlander. *Det Uomgængelige – Filosofiske Deduktioner [The Inevitable – Philosophical Deductions, with a foreword by Georg Henrik von Wright]*. Munksgaard · Rosinante, 1994. 168 pages.

[2] Kai Sørlander. *Under Evighedens Synsvinkel [Under the viewpoint of eternity]*. Munksgaard · Rosinante, 1997. 200 pages.

[3] Kai Sørlander. *Den Endegyldige Sandhed [The Final Truth]*. Rosinante, 2002. 187 pages.

[4] Kai Sørlander. *Indføring i Filosofien [Introduction to The Philosophy]*. Informations Forlag, 2016. 233 pages.

[5] Dines Bjørner. Manifest Domains: Analysis & Description. *Formal Aspects of Computing*, 29(2):175–225, Online: July 2016, Journal: March 2017.

[6] Dines Bjørner. *A Rôle for Mereology in Domain Science and Engineering*. Synthese Library (eds. Claudio Calosi and Pierluigi Graziani). Springer, Amsterdam, The Netherlands, October 2014.

[7] Dines Bjørner. Formal Software Techniques in Railway Systems. In Eckehard Schnieder, editor, *9th IFAC Symposium on Control in Transportation Systems*, pages 1–12, Technical University, Braunschweig, Germany, 13–15 June 2000. VDI/VDE-Gesellschaft Mess– und Automatisiering-stechnik, VDI-Gesellschaft für Fahrzeug– und Verkehrstechnik. Invited talk.

[8] Dines Bjørner, Chris W. George, and Søren Prehn. Computing Systems for Railways — A Rôle for Domain Engineering. Relations to Requirements Engineering and Software for Control Applications. In *Integrated Design and Process Technology. Editors: Bernd Kraemer and John C. Petterson*, P.O.Box 1299, Grand View, Texas 76050-1299, USA, 24–28 June 2002. Society for Design and Process Science. Extended version.

[9] Dines Bjørner. Dynamics of Railway Nets: On an Interface between Automatic Control and Software Engineering. In *CTS2003: 10th IFAC Symposium on Control in Transportation Systems*, Oxford, UK, August 4-6 2003. Elsevier Science Ltd. Symposium held at Tokyo, Japan. Editors: S. Tsugawa and M. Aoki. Final version.

[10] Martin Pěnička, Albena Kirilova Strupchanska, and Dines Bjørner. Train Maintenance Routing. In *FORMS'2003: Symposium on Formal Methods for Railway Operation and Control Systems*. L'Harmattan Hongrie, 15–16 May 2003. Conf. held at Techn.Univ. of Budapest, Hungary. Editors: G. Tarnai and E. Schnieder, Germany. Final version.

[11] Albena Kirilova Strupchanska, Martin Pěnička, and Dines Bjørner. Railway Staff Rostering. In *FORMS2003: Symposium on Formal Methods for Railway Operation and Control Systems*. L'Harmattan Hongrie, 15–16 May 2003. Conf. held at Techn.Univ. of Budapest, Hungary. Editors: G. Tarnai and E. Schnieder, Germany. Final version.

[12] Dines Bjørner. A Container Line Industry Domain. Techn. report, Fredsvej 11, DK-2840 Holte, Denmark, June 2007. Extensive Draft.

[13] Dines Bjørner. The Tokyo Stock Exchange Trading Rules. R&D Experiment, Fredsvej 11, DK-2840 Holte, Denmark, January and February, 2010. Version 1, Version 2.

[14] Dines Bjørner. What are Documents ? Research Note, July 2017. http://www.imm.dtu.dk/-˜dibj/2017/docs/docs.pdf.

[15] Dines Bjørner. Pipelines – a Domain Description[7]. Experimental Research Report 2013-2, DTU Compute and Fredsvej 11, DK-2840 Holte, Denmark, Spring 2013.

[16] Dines Bjørner. Domain Models of "The Market" — in Preparation for E–Transaction Systems. In *Practical Foundations of Business and System Specifications (Eds.: Haim Kilov and Ken Baclawski)*, The Netherlands, December 2002. Kluwer Academic Press. Final draft version.

[17] Dines Bjørner. Weather Information Systems: Towards a Domain Description. Technical Report: Experimental Research, Fredsvej 11, DK–2840 Holte, Denmark, November 2016. http://www.imm.dtu.dk/˜dibj/2016/wis/wis-p.pdf.

[18] Dines Bjørner. A Credit Card System: Uppsala Draft. Technical Report: Experimental Research, Fredsvej 11, DK–2840 Holte, Denmark, November 2016. http://www.imm.dtu.dk/˜dibj/2016/credit/accs.pdf.

[19] Dines Bjørner. Urban Planning Processes. Research Note, July 2017. http://www.imm.dtu.dk/-˜dibj/2017/up/urban-planning.pdf.

[20] Dines Bjørner. A Space of Swarms of Drones. Research Note, November–December 2017. http://www.imm.dtu.dk/˜dibj/2017/docs/docs.pdf.

[21] Dines Bjørner. From Domain Descriptions to Requirements Prescriptions – A Different Approach to Requirements Engineering. 2016. Extensive revision of [22].

[22] Dines Bjørner. From Domains to Requirements. In *Montanari Festschrift*, volume 5065 of *Lecture Notes in Computer Science (eds. Pierpaolo Degano, Rocco De Nicola and José Meseguer)*, pages 1–30, Heidelberg, May 2008. Springer.

[7] http://www.imm.dtu.dk/˜dibj/pipe-p.pdf

[23] H. Bekič, D. Bjørner, W. Henhapl, C. B. Jones, and P. Lucas. A Formal Definition of a PL/I Subset. Technical Report 25.139, IBM Laboratory, Vienna, December 1974.

[24] Dines Bjørner. Programming Languages: Formal Development of Interpreters and Compilers. In *International Computing Symposium 77 (eds. E. Morlet and D. Ribbens)*, pages 1–21. European ACM, North-Holland Publ.Co., Amsterdam, 1977.

[25] Dines Bjørner and Ole N. Oest, editors. *Towards a Formal Description of Ada*, volume 98 of *LNCS*. Springer, 1980.

[26] Dines Bjørner. Rigorous Development of Interpreters and Compilers. In *See [32]*, chapter 9, pages 271–320. Prentice-Hall, 1982.

[27] Hans Henrik Løvengreen and Dines Bjørner. On a formal model of the tasking concepts in Ada. In *ACM SIGPLAN Ada Symp.*, Boston, 1980.

[28] P.L. Haff, editor. *The Formal Definition of CHILL*. ITU (Intl. Telecmm. Union), Geneva, Switzerland, 1981.

[29] G.B. Clemmensen and O. Oest. Formal specification and development of an Ada compiler – a VDM case study. In *Proc. 7th International Conf. on Software Engineering, 26.-29. March 1984, Orlando, Florida*, pages 430–440. IEEE, 1984.

[30] Ole N. Oest. VDM from research to practice (invited paper). In *IFIP Congress*, pages 527–534, 1986.

[31] Dines Bjørner. To Every Manifest Domain a CSP Expression — A Rôle for Mereology in Computer Science. *Journal of Logical and Algebraic Methods in Programming*, (94):91–108, January 2018.

[32] Dines Bjørner and Cliff B. Jones, editors. *Formal Specification and Software Development*. Prentice-Hall, 1982.

