A Philosophy of Domain Science & Engineering*

An Interpretation of Kai Sørlander's Philosophy

An Incomplete Work in Progress Research Report

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

May 20, 2018: 11:20 am

1
2

Abstract

We show how the domain analysis & description calculi of [1] satisfy Kai Sørlander's
Philosophy, but also that Sørlander's Philosophy, notably [2] and [3] mandates extensions
to the calculi in order to form a more consistent "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**. (i) The **physical parts** are not made by man, but are in 3
space and *time*; these are *endurants* 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 – but were not so explicated. They are the parts we treated in [1]. (ii) 4
The **living species parts** are *plants* and *animals*; they are still subject to the laws and prin-
ciples of physics, but additionally *unavoidably* endowed with such properties as *causality
of purpose*. Animals have *sensory organs*, *means of motion*, *instincts*, *incentives* and *feelings*.
Among animals we single out **humans** as parts that are further characterisable: possessing 5
language, *learning skills*, being *consciousness*, and having *knowledge*. These aspects were
somehow, by us, subsumed in our analysis & description by partially endowing *physical
parts* with such properties. (iii) Then there are the parts made by humans, i.e., **artifacts**. 6
Artifacts have a usual set of attributes of the kind *physical parts* can have; but in addi-
tion 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"*. Also these were subsumed in [1]
– by either partially endowing *physical parts* with such properties, or by *ignoring* them!
We thus suggest a **philosophy basis** for **domain science & engineering**. This paper is 7
based on recent research [4, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] into methods for analysing
and describing human-centered universes of discourses such as transport nets, container

* **First reading: The Victor Ivannikov Memorial Event, May 3–4, 2018, Yerevan, Armenia**

† Margin numbers refer to slide numbers of a lecture (series) version of this paper.

‡ This paper and its slide presentation version can be found at: http://www.imm.dtu.dk/˜dibj/2018/-philosophy/filo.pdf, respectively http://www.imm.dtu.dk/˜dibj/2018/philosophy/[4-1-]filo-oh.pdf

lines, pipelines, drones, urban planning, etc. The present paper is motivated by speculations about possible "interfaces" between domain analysis & description methods and the reality they model. A major section of the paper is based on 10 years of research into and experimental use of (the citation-referenced) calculi for domain analysis & description. Another major segment of the paper is based on the philosophy of Kai Sørlander
8 [15, 16, 17, 18][1].

In the first part of the paper we present two calculi, one for **analysing** manifest "worlds" and one for **describing** those "realities". And we "interpret" *manifest endurant entities* as *behaviours* i.e., as *perdurants*. This interpretation is, from the point-of-view of post-Kantian philosophy, a **transcendental deduction**, i.e. cannot be logically explained, but can be understood meta-physically. In a more-or-less summary section we shall then show that the calculi are necessary and sufficient, in that they have a basis in philosophical
9 reasoning. But, what is to us more interesting, we show how the Sørlander Philosophy "kicks back" and either mandates or requires domain properties not covered in my earlier papers on the domain analysis & description method [4, 1].

Initial versions of this document are in the form of a report. As such it collects far more material than should be contained in a proper paper. Most of the "extra" report material is collected from various sources but drastically edited by me. Most of the material of Sect. **9** is extracted from [18] some from [15, 21, 22, 23].

Contents

[1]Other Sørlander books are [2, 19, 20, 3]

1 Introduction 10

Definition 1 Domain: *By a* **domain** *we shall understand a* **rationally describable** *segment of a* **human assisted** *reality, i.e., of the world, its* **physical parts**, *and* **living species**. *These are* **endurants** *("still"), existing in space, as well as* **perdurants** *("alive"), existing also in*
11 *time. 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** ■

12 The *science and engineering of domain analysis & description* is *different* from the science of physics and the core of its derived engineerings: building (civil), chemical, mechanical, electrical, electronics, et cetera. All of these engineerings emerged out of *the natural sciences*. These classical engineering disciplines have increasingly included many facets of *man-machine interface* concerns, but their core is still in the *the natural sciences*. We assume that the readers
13 are familiar with the above notions.

The core of *domain science & engineering* such as we shall pursue it, is in two disciplines: *mathematics*, notably *mathematical logic* and *abstract algebra*, and *philosophy*, notably *meta physics* and *epistemology*. We assume that the readers are familiar with the above-mentioned
14 notions of mathematics.

Definition 2 Metaphysics: *By metaphysics we shall understand a branch of philosophy that explores fundamental questions, including the nature of concepts like being, existence, and reality. Traditional metaphysics seeks to answer, in a "suitably abstract and fully general manner", the questions: What is there? and And what is it like?*[2] ■

15 Topics of metaphysical investigation include existence, objects and their properties, space
16 and time, cause and effect, and possibility.

Definition 3 Epistemology: *By epistemology [from epistēmē, 'knowledge', and logos, 'logical discourse'] is the branch of philosophy concerned with the theory of knowledge* ■[3]

17 The philosophy aspect of our study is primarily epistemological, i.e., not metaphysical.

Epistemology studies the nature of knowledge, justification, and the rationality of belief. Much of the debate in epistemology centers on four areas: (1) the philosophical analysis of the nature of knowledge and how it relates to such concepts as truth, belief, and justification, (2) various problems of skepticism, (3) the sources and scope of knowledge and justified belief, and (4) the criteria for knowledge and justification.[4] A central branch of epistemology is *ontology*,
18 the investigation into the basic categories of being and how they relate to one another.[5]
Observe the distinction in the definitions of metaphysics and epistemology between [metaphysics] *"explores fundamental questions, including the nature of concepts like being, existence, and reality"* and [epistemology] *"the philosophical analysis of the nature of knowledge and how*
19 *it relates to such concepts as truth, belief, and justification, etc."*. Epistemology addresses such questions as *What makes justified beliefs justified?"*; *"What does it mean to say that we know something?"* and, fundamentally, *"How do we know that we know?"*[6]

[2] https://en.wikipedia.org/wiki/Metaphysics
[3] https://en.wikipedia.org/wiki/Epistemology
[4] https://en.wikipedia.org/wiki/Epistemology
[5] https://en.wikipedia.org/wiki/Metaphysics
[6] https://en.wikipedia.org/wiki/Epistemology

1.1 Two Views of Domains 20

There are two aspects to this paper: (i) the analysis & description of fragments of the context in which software, to be developed, is to serve, (ii) and the general, basically philosophical, problem of the absolutely necessary conditions for describing the world.

1.1.1 The Computing Science View 21

In twelve papers we have put forward a method for analysing and describing the domains for which software is developed:

- [4, 1] Manifest Domains: Analysis & Description **FAoC, March 2017**
- [5, 6] Domain Facets: Analysis & Description
- [7, 8] Formal Models of Processes and Prompts
- [9, 10] To Every Manifest Domain Mereology a CSP Expression **LAMP, Jan. 2018**
- [11, 12] From Domain Descriptions to Requirements Prescriptions
- [13, 14] Domains: Their Simulation, Monitoring and Control

These methods involve new principles, techniques and tools – the *calculi*. The calculi has been 22
applied in around 20+ experimental researches to as diverse domains as

- railways,
- IT security,
- container shipping lines,
- "the market",
- pipelines,
- road transport systems,
- stock exchanges,
- credit card systems,
- swarms of drones,
- documents and
- urban planning.

The calculi, we claim, has withstood some severe "tests". The experiments are referenced in Sect. 13.1 [pp. 79].

1.1.2 The Philosophy View 23

In four books the Danish philosopher Kai Sørlander has investigated the philosophical issues alluded to above.

- [15] Kai Sørlander . *Det Uomgængelige – Filosofiske Deduktioner [The Inevitable – Philosophical Deductions] Forord/Foreword: Georg Henrik von Wright*. Munksgaard · Rosinante, 1994. 168 pages.
- [16] Kai Sørlander . *Under Evighedens Synsvinkel [Under the viewpoint of eternity]*. Munksgaard · Rosinante, 1997. 200 pages.
- [17] Kai Sørlander . *Den Endegyldige Sandhed [The Final Truth]*. Rosinante, 2002. 187 pages.
- [18] Kai Sørlander . *Indføring i Filosofien [Introduction to The Philosophy]*. Informations Forlag, 2016. 233 pages.

A main contribution of Sørlander is, on the philosophical basis of the *possibility of truth* 24
(in contrast to Kant's *possibility of self-awareness*) to *rationally* and *transcendentally deduce* **the absolutely necessary conditions for describing any world**. These conditions presume a *principle of contradiction* and lead to the *ability* to *reason* using *logical connectives* and to *handle asymmetry, symmetry* and *transitivity*. *Transcendental deductions* then lead to *space* and
25 *time*, not as priory assumptions, as with Kant, but derived facts of any the world. From this basis Sørlander then, by further transcendental deductions arrive at kinematics, dynamics and the bases for Newton's Laws. And so forth. We build on Sørlander's basis to argue that the domain analysis & description calculi are necessary and sufficient and that a number of relations between domain entities can be understood transcendentally and as "variants" of Newton's Laws!

1.1.3 First Two Independent Treatments, then An Interpretation 26

First we present the two views independent of one-another.

In Segment **I** we present the *domain analysis & description method:* its *principles, techniques* and *tools*, Sects. **2**–**5**, and a *substantial example*, Sect. **6**, to *support understanding* the *domain*
27 *analysis & description method*.

In Segment **III** we present in Sect. **8** a brief motivation of the task of philosophy; in Sect. **9** an extensive review is presented of metaphysical and epistemological issues in philosophy, from the ancient Greeks up til the mid 1900's; in Sect. **10** an extensive review is then given
28 of Sørlander's Philosophy.

Then, in Segment **IV**'s Sect. **11**, we bring the two studies — the *domain analysis & description calculi* and the *Kai Sørlander Philosophy* — together: It is here that, as a consequence of Sørlander's Philosophy, we modify the domain analysis & description method, of Segment **I**, in suggesting extensions.

The Main Contribution

With Segment **IV** the *the main contribution* of this report is achieved: (i) establishing *a basis* for *domain science & engineering* in *philosophy*; and (ii) the *specific modifications* required by and the *founding* of the domain analysis & description calculi in *philosophy*.

29

In Segment **II**, in-between Segments **I** and **III**, we present in Sect. **7**, a short review of *space* and *time*.

1.2 The Computing Science Background 30

1.2.1 Computer & Computing Science

- By **computer science** I understand the study and knowledge of the "things" that can "exist inside" computing devices (i.e., data and computations) – and the study and knowledge of these computing devices.
- By **computing science** I understand the study and knowledge of how to construct "those things", i.e., **programming methodology**.

I consider myself a computing scientist primarily interested in programming methodology.

1.2.2 Formal Methods 31

- By a **method** I understand
  a set of **principles** for **selecting** and **applying** a set of **techniques** and **tools**
  for the **construction** of an artifact, as here, software.

- By a **formal method** I understand I understand
  a method whose principles, techniques and tools
  can be understood in a mathematical framework –
  for example where, among the tools, the **specification languages** can be given
  a **mathemtical syntax**, a **mathematical semantics** and a **mathematical proof system**.

I consider myself to have primarily contributed to the area of formal methods, as exemplified by `VDM` and `RAISE`.

1.2.3 A Triptych of Engineering 32

- Before software can be designed we must be familiar with its requirements.
- Before requirements can be prescribed we must be familiar with the context of the software to be developed, that is, the domain. 33
- Hence the triptych of software development:
  - first (ideally) the domain engineering of an appropriate domain description;
  - then (ideally) the requirements engineering of the requirements prescription – formally related to the domain description;
  - finally the software design "derived" from the requirements prescription and (ideally) formally reasoned to meet customers' expectations, that is, to satisfy the domain description and be correct wrt. the requirements prescription.

34

My contributions in the last many years has been to establish a proper domain science & engineering. My main focus, since 1977, has been on the development of "large" software: compilers (like for `CHILL` and `Ada`), and infrastructure software – for pipelines, railways, health care, banking, road traffic, etc.

1.3 Domains, their Analysis & Description, and a Method 35

In Definition 1 [pp. 8] we gave a rough characterisation of what we man by domain. In this section we shall brief outline what we mean by **domain analysis & description**, and what we mean by a **method for analysing & describing domains**.

1.3.1 Domain Analysis & Description 36

Definition 4: **Domain Analysis and Description:** By **domain analysis and description** we shall understand the analysis & description of domains ■

1.3.2 A Domain Analysis & Description Method 37

Definition 5: **A Domain Analysis and Description Method:** By a **domain analysis and description method** we shall understand a set of principles, techniques and tools for the construction, i.e., analysis & description of a domain model ■

38 The terms *description* and *model* are here considered synonymous.

Segment I: The Domain Analysis & Description Calculi

2 Endurants – cf. s. 6.2 Pg. 36 39

In a series of *definitions*, most of which are rather like *characterisations*[7], we shall *explicate* a number of domain concepts. These definitions will lead to the introduction of first *domain*
40 *analysis prompts*, then also *domain description prompts*. Think of a **prompt** as a *cue*, a *hint*, a *suggestion*, in German, a *stichwort, suchbegriff*, in French, a *signal théâtre*, that the domain analyser is told, by the principles of the domain analysis & description method, to act upon.

2.1 The Universe of Discourse – cf. s. 6.1 Pg. 36 41

Analysis Prompt 1 *is_universe_of_discourse:* *By a* universe of discourse *for domain science & engineering we shall mean a human-centered area of concern, one that involves, as "main players": endurants and perdurants such that at least one of the endurants is man-made and and either represents a human or at least another one is a human* ■

42

Example 1 **Man-made Automobiles and Drivers**: In the large example of Sect. 6 automobiles and road nets are endurants, and automobiles "subsume" their human drivers ■

43

Domain Description Prompt 1 *observe_universe_of_discourse*: *The domain-of-interest needs first be briefly narrated. Just a simple story. One that emphasises the "main players": the endurants and the perdurants such that at least one of the endurants is man-made and and either represents a human or at least another one is a human* ■

2.2 Basic Domain Concepts 44

Definition 6 Entity: *By an* **entity** *we shall understand a* **phenomenon**, *i.e., something that can be* observed, *i.e., be seen or touched by humans,* or *that can be* conceived *as an* abstraction *of an entity; alternatively, a phenomenon is an entity,* if it exists, it is **"being"**, it is that which makes a "thing" what it is: essence, essential nature *[24, Vol. I, pg. 665]* ■

45

[7] Usually, in computer science papers, definitions are terse and based on more-or-less implicit reference to a mathematically precise model. Since domains do not have an a-priori mathematically precise model our definitions cannot be precise. Most of the definitions are taken from such dictionaries as [24, *The Oxford Shorter English Dictionary*] and from the Internet based [25, *The Stanford Encyclopedia of Philosophy*].

Example 2 **Entities and Non-entities**: The following are entities: a stone, say, laying on the ground – which is an entity; a pencil, say, that I, a humen entity, hold in my hand; a rhododendron, in my garden – which is an entity. The following are not entities: the blue sky of my imagination; a fleeting moment of sadness; being drunk ■

46

Analysis Prompt 2 *`is_entity`:* *The domain analyser analyses "things" (θ) into either entities or non-entities. The method can thus be said to provide the* **domain analysis prompt***:*

- *`is_entity` – where `is_entity(θ)` holds if θ is an entity* ■[8]

47

Definition 7 **Endurant***: By an* **endurant** *we shall understand an entity that can be observed or conceived and described as a "complete thing" at no matter which given snapshot of time; alternatively an entity is endurant if it is capable of* enduring*, that is* persist*, "*hold out*" [24, Vol. I, pg. 656]. Were we to "freeze" time we would still be able to observe the full endurant* ■

48

Example 3 **Endurants**: The following are examples of endurants: the lake of a landscape such as a tourist (i.e., an amimal entity) photographs it; the engine train of an automobile such as an automobile mechanic (a human entity) repairs it; and the horse such as a jockey (a human entity) prepares it for a race ■

49

Analysis Prompt 3 *`is_endurant`:* *The domain analyser analyses an entity, e, into an endurant as prompted by the* **domain analysis prompt***:*

- *`is_endurant` – ϕ is an endurant if `is_endurant(e)` holds.*

`is_entity` is a prerequisite prompt *for `is_endurant`* ■

50

Definition 8 **Perdurant***: By a* **perdurant** *we shall understand an entity for which only a fragment exists if we look at or touch them at any given snapshot in time, that is, were we to freeze time we would only see or touch a fragment of the perdurant, alternatively an entity is perdurant if it endures continuously, over time, persists, lasting [24, Vol. II, pg. 1552]* ■

51

Example 4 **Perdurants**: The following are examples of perdurants: the flow of water in a river; the human life, from birth to death; the car driving down a road ■

52

Analysis Prompt 4 *`is_perdurant`:* *The domain analyser analyses an entity e into perdurants as prompted by the* **domain analysis prompt***:*

- *`is_perdurant` – e is a perdurant if `is_perdurant(e)` holds.*

`is_entity` is a prerequisite prompt *for `is_perdurant`* ■

53

Definition 9 Discrete Endurant: *By a* **discrete endurant** *we shall understand an endurant which is separate, individual or distinct in form or concept* ■

54

Example 5 Discrete Endurants: The following are examples of discrete endurants: planets in space; automobiles (in a car sales office); and students at a lecture in a college classroom.

55

Analysis Prompt 5 *is_discrete: The domain analyser analyses endurants e into discrete entities as prompted by the* **domain analysis prompt**:

- *is_discrete – e is discrete if is_discrete(e) holds* ■

56

Definition 10 Continuous Endurant: *By a* **continuous endurant** *we shall understand an endurant which is prolonged, without interruption, in an unbroken series or pattern* ■

57

Example 6 Continuous Endurants: The following are examples of continuous endurants: springs, brooks, rivers and lakes of a landscape; and gas in a pipeline.

58

Analysis Prompt 6 *is_continuous: The domain analyser analyses endurants e into continuous entities as prompted by the* **domain analysis prompt**:

- *is_continuous – e is continuous if is_continuous(e) holds* ■

59

Continuity shall here not be understood in the sense of mathematics. Our definition of 'continuity' focused on *prolonged, without interruption, in an unbroken series or pattern.* In that sense materials (water, oil, sand, gravel, ...) shall be seen as 'continuous',

2.3 An Upper Ontology Diagram of Domains – A Preview 60

Figure 1 [facing page] shows a so-called upper ontology for manifest domains. So far we have covered only a fraction of this ontology, as noted. By ontologies we shall here understand *formal representations of a set of concepts within a domain and the relationships between those*
61 *concepts.*

2.4 Structures – cf. s. 6.2.1 Pg. 36 62

Definition 11 Structure: *By a* **structure** *we shall understand a discrete endurant which the domain engineer chooses to describe as itself consisting of structures, parts, components and materials but to* **not** *endow itself with* **internal qualities**: *unique identifiers, mereology or attributes* ■

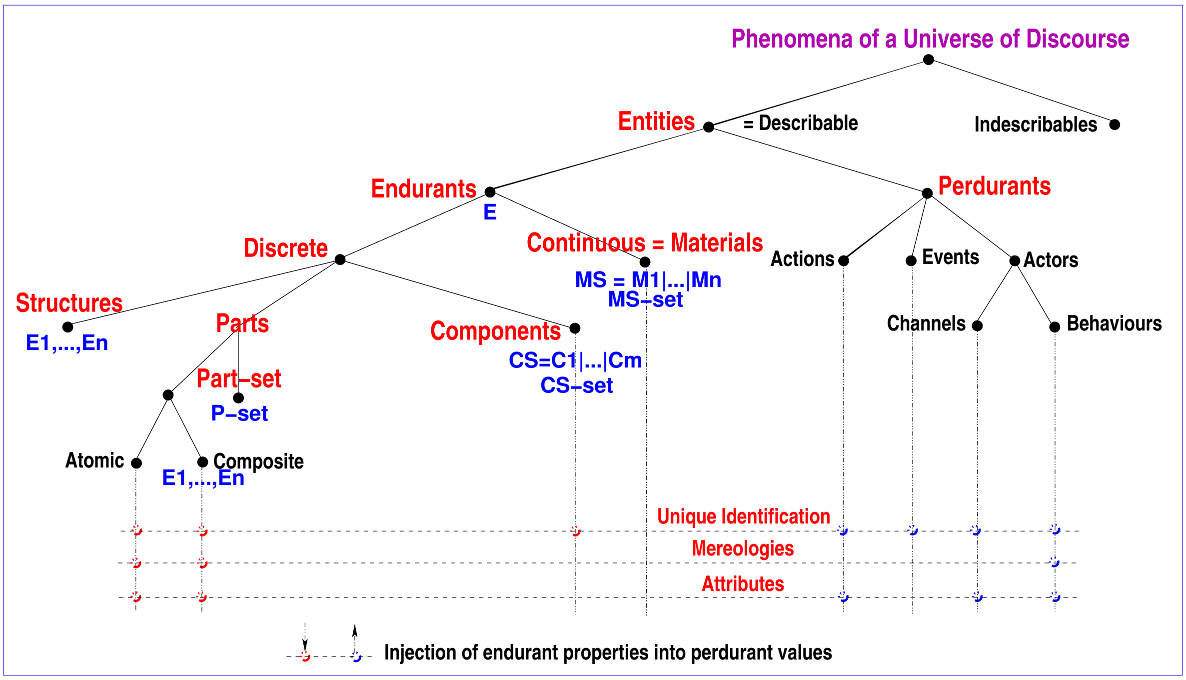


Figure 1: An Upper Ontology for Domains

We shall define the terms parts, components and materials, as well as unique identifica-
63 tion, mereology and attributes later. Structures are introduced in the domain analysis & description method for pragmatic reasons. When modelling an endurant as a structure we are disragarding that the endurant may have a physically "separate" form, treating that endurant as a concept rather than someting manifest. Endurants "first" modelled as structures
64 may, subsequently, or also, be modelled as (usually composite) parts (see below).

Analysis Prompt 7 *is_structure: The domain analyser analyse endurants, e, into structure entities as prompted by the* **domain analysis prompt***:*

- *is_structure* ■ 65

Structures are thus composite endurants which consist of other endurants: discrete as well as continuous, i.e., structures, [physical] parts[, living species] and components, as well as materials. Parts, components and material will soon be defined. The [...] bracketed concepts will not be defined till late in this report.

2.5 Parts, Components and Materials – cf. s. 6.2.2 Pg. 36 66

2.5.1 Parts – cf. s. 6.2.3 Pg. 37

Characterisation 1 **Parts***: Parts are manifest in the sense that we can see them, touch them: we can uniquely identify them (unique identification); relate them to other parts (mereology); and "measure" some of their characteristics (attributes);*

[8] Analysis prompt definitions and description prompt definitions and schemes are delimited by ■.

Parts are going to be the "work horse" of domain descriptions. Our primary focus will be on
67 man-made parts (artifacts). We leave it to physics (i.e., physicists) to model natural parts.

Definition 12 Part: *By a* **part** *we shall understand a discrete endurant which the domain engineer chooses to endow with all three* **internal qualities**: *unique identification, mereology, and one or more attributes* ■

68

Example 7 Examples of Parts: Examples of natural parts are: a raw diamond (as found in the ground); the *Rock of Gibraltar*[9]; *The Equator*[10]. Examples of man-made parts, that is, artifacts are: an armchair; the *Empire State Building*; and a canal lock.

69

Analysis Prompt 8 *is_part*: *The domain analyser analyse endurants, e, into part entities as prompted by the* **domain analysis prompt**:

- *is_part* ■

70

Definition 13 Atomic Part: **Atomic parts** *are those which, in a given context, are deemed to not consist of meaningful, separately observable proper sub-parts. A* **sub-part** *is a part* ■

71

Example 8 Atomic Parts: These are examples of atomic (man-made) parts: a bolt, a screw, a nail; an automobile as bought by the owner; and a pipe, valve, pump, fork, and join of a pipeline.

72

Analysis Prompt 9 *is_atomic*: *The domain analyser analyses a discrete endurant, i.e., a part p into an atomic endurant:*

- *is_atomic: p is an atomic endurant if* `is_atomic(p)` *holds* ■

73

Definition 14 Composite Part: **Composite parts** *are those which, in a given context, are deemed to indeed consist of meaningful, separately observable proper sub-parts* ■

74

Example 9 Composite Parts: These are examples of composite (man-made) parts: a nut (bolt) and screw assembly; an automobile as put together or serviced by a factory, resp. a mechanic; and a pipeline (consisting of pipes, valves, pumps, forks, joins etc.).

75

Analysis Prompt 10 *is_composite*: *The domain analyser analyses a discrete endurant, i.e., a part p into a composite endurant:*

- *is_composite: p is a composite endurant if* `is_composite(p)` *holds* ■

76

Analysis Prompt 11 *observe_endurants*: *The* **domain analysis prompt**:

- *observe_endurants*

directs the domain analyser to observe the sub-endurants of an endurant e and to suggest their sorts. Let, schematically, **observe_endurants**(e) *be* $\{e_1{:}E_1, e_2{:}E_2, \ldots, e_m{:}E_m\}$ ■

77 **Domain Description Prompt 2** *observe_endurant_sorts*: *If* **is_composite**(p) *holds, then the analyser "applies" the* **domain description prompt**

- *observe_endurant_sorts(p)*

resulting in the analyser writing down the endurant sorts and endurant sort observers *domain description text according to the following schema:* 78

2. observe_endurant_sorts schema

Narration:
[s] ... narrative text on sorts ...
[o] ... narrative text on sort observers ...
[η] ... narrative text on sort type observers ...
[i] ... narrative text on sort recognisers ...
[p] ... narrative text on proof obligations ...
Formalisation:
type
[s] P,
[s] E_i i:[1..m] **comment:** E_i i:[1..m] abbreviates E_1, E_2, ..., E_m
value
[o] **obs_endurant_sorts_**E_i: P → E_i i:[1..m]
[η] **if** is_part(e_i): η(e_i) ≡ ⋘ E_i ⋙ i:[1..m]
[i] **is_**E_i: $(E_1|E_2|...|E_m)$ → **Bool** i[1..m]

proof obligation [Disjointness of endurant sorts]
[p] $\mathcal{PO}$: ∀ e:($E_1|E_2|...|E_m$) •
[p] $\bigwedge$ {**is_**E_i(e) ≡ $\bigwedge$ {∼**is_**E_j(p) | j:[1..m] \ {i}} | i:[1..m]}

79

Example 10 **Observe Transport System Endurants**: We refer to example Sect. 6.2.1 [pp. 36] annotation and formalisation Items 8–10; and to example Sect. 6.2.2 [pp. 36] annotation and formalisation Items 11–12a ■

80

Some composite parts can suitably be modelled as sets of parts of the same sort.

Analysis Prompt 12 *has_concrete_type*: *The domain analyser may decide that it is expedient, i.e., pragmatically sound, to render a part sort, P, whether atomic or composite, as a concrete type, T. That decision is prompted by the holding of the* **domain analysis prompt**:

- *has_concrete_type(p).*

is_discrete *is a* **prerequisite prompt***has_concrete_type of* **has_concrete_type** ■

81

Domain Description Prompt 3 *observe_part_type*: *The domain analyser applies the* **domain description prompt***:*

- *observe_part_type(p)*[11]

to parts p:P which then yield the part type and part type observers *domain description text according to the following schema:* 82

3. observe_part_type schema

Narration:

[t_1] ... narrative text on sorts and types S_i ...
[t_2] ... narrative text on types T ...
[t_3] ... narrative text on type of value observer
[o] ... narrative text on type observers ...

Formalisation:

type
[t_1] S_1, S_2, ..., S_m, ..., S_n,
[t_2] $T = \mathcal{E}(S_1,S_2,...,S_n)$
[t_3] $\eta(s_i) \equiv \lll S \ggg$, i:[1..n],s_i:S_i
value
[o] **obs_part_T**: P → T ■

2.5.2 Components – cf. s. 6.2.4 Pg. 37 83

Some discrete composite endurants can suitably be modelled as sets of parts of possibly different sorts but for which there is no need to model their mereology, that is, how the parts in the set relate to one another.

Definition 15 Component: *By a* **component** *we shall understand a discrete endurant which we, the domain analyser cum describer chooses to* **not** *endow with* **mereology** ■

84 Parts may or may not contain, i.e., "have", components.

Example 11 Components of Parts: a part, like a mail-box, may contain letters, newspapers, small packages, advertisement brochures, etc.; a part, like a household shop shelf, may contain bread toasters, blenders, coffee grinders, coffee machines, etc.; and a part, like a book case, may contain books, journals, bric-à-brac, etc. ■

85

[9] Later, when having introduced continuous endurants, i.e., materials, one may claim that the physical aspects of the enclave of *Gibraltar* could also be modelled as a material.

[10] One may claim that *The Equator* is a non-physical concept. To this one may counter-claim that *The Equator* is physically delineable: can be "marked down"!

[11] has_concrete_type is a *prerequisite prompt* of observe_part_type.

Analysis Prompt 13 *has_components:* *The domain analyser inquire endurants e as to whether they have, i.e., contain, components, as prompted by the* **domain analysis prompt***:*

- *has_components* ■

86

Analysis Prompt 14 *is_component:* *The domain analyser analyse endurants e into component entities as prompted by the* **domain analysis prompt***:*

- *is_component* ■

87

Domain Description Prompt 4 *observe_component_sorts:* *The* **domain description prompt***:*

- *observe_component_sorts_P(p)*

yields the component sorts and component sort observer *domain description text according to the following schema – whether or not the actual part p contains any components:* 88

4. observe_component_sorts_P schema

Narration:
[s] ... narrative text on component sorts ...
[o] ... narrative text on component observers ...
[i] ... narrative text on component sort recognisers ...
[u] ... narrative text on unique identifier ...
[p] ... narrative text on component sort proof obligations ...
Formalisation:
type
[s] K1, K2, ..., Kn
[s] K = K1| K2 | ... | Kn
[s] KS = K-**set**
value
[o] **obs_components_**P: P → KS
[i] **is_**K_i: (K_1|K_2|...|K_n) → **Bool** i:[1..n]
[u] **uid_**K_i
Proof Obligation: [Disjointness of Component Sorts]
[p] $\mathcal{PO}$: ∀ k_i:(K_1|K_2|...|K_n) •
[p] $\bigwedge$ {**is_**$K_i(k_i)$ ≡ $\bigwedge${∼**is_**$K_j(k_j)$|j:[1..n] \ {i}}} i:[1..n] ■

89

Example 12 **Observe Transport System Component Sorts**: We refer to example Sect. 6.2.4 [pp. 37] annotation and formalisation Items 16–17 ■

2.5.3 Materials – cf. s. 6.2.5 Pg. 37 90

Definition 16 Material: *By a* **material** *we shall understand a continuous endurant* ■

Parts may or may not contain, i.e., "have", materials. 91

Example 13 Materials of Parts: a part, like a pipe-line pipe, may contain oil; a part, like a timber yard, may contain boards, lumber, etc., of different sizes and qualities; and a part, like a building materials shop, may contain concrete, sand, gravel, bricks, etc., in different bags, containers and sizes ■

92

Example 14 Observe Transport Component Sorts: We refer to example Sect. 6.2.4 [pp. 37] annotation and formalisation Items 16–17 ■

93

Analysis Prompt 15 *has_materials: The domain analyser inquire endurants e as to whether they have, i.e., contains, material, as prompted by the* **domain analysis prompt**:

- *has_materials* ■

Analysis Prompt 16 *is_material: The domain analyser analyse endurants e into material entities as prompted by the* **domain analysis prompt**:

- *is_material* ■

94

Domain Description Prompt 5 *observe_material_sorts_P*: *The* **domain description prompt**:

- *observe_material_sorts_P(e)*

yields the material sorts and material sort observers' domain description text according to the
95 *following schema whether or not part p actually contains materials:*

5. observe_material_sorts_P schema

Narration:
[s] ... narrative text on material sorts ...
[o] ... narrative text on material sort observers ...
[i] ... narrative text on material sort recognisers ...
[p] ... narrative text on material sort proof obligations ...
Formalisation:
type
[s] M1, M2, ..., Mn
[s] M = M1 | M2 | ... | Mn
[s] MS = M-**set**
[a] Ai = A11 | A12 | ... | A1n
value
[o] **obs_mat_sort_**M_i: P → M, [i:1..n]
[o] **obs_materials_**P: P → MS

[i] **is**_M_i: M → **Bool** [i:1..n]
[a] **attr**_A_{i_j}: M_i → A_{i_j} [i:...,j:...]
proof obligation [Disjointness of Material Sorts]
[p] $\mathcal{PO}$: ... ■

96

Example 15 Observe Transport System Materials: We refer to example Sect. 6.2.5 [pp. 37] annotation and formalisation Items 18–19 ■

2.6 Unique Part and Component Identifiers – cf. s. 6.2.7 Pg. 37 97

We introduce a notion of unique identification of parts and components. We assume (i) that all parts and components, p, of any domain P, have *unique identifiers*, (ii) that *unique identifiers* (of parts and components p:P) are *abstract values* (of the *unique identifier* sort PI of parts p:P), (iii) such that distinct part or component sorts, P_i and P_j, have distinctly named *unique identifier* sorts, say PI_i and PI_j, (iv) that all π_i:PI_i and π_j:PI_j are distinct, and (v) that the observer function **uid**_P applied to p yields the unique identifier, say π:PI, of p. 98

Analysis Prompt 17 *type_name: The description language function* **type_name** *applies to unique identifiers, p_{ui}:P_{UI}, and yield the name of the type, P, of the parts having unique identifiers of type P_{UI}:*

- *type_name – where* type_name(p_{ui}) *yields P* ■

Representation of Unique Identifiers: Unique identifiers are abstractions. When we endow two parts (say of the same sort) with distinct unique identifiers then we are simply saying that these two parts are distinct. We are not assuming anything about how these identifiers otherwise come about. 99

Domain Description Prompt 6 *observe_unique_identifier*: *We can therefore apply the* **domain description prompt**:

- *observe_unique_identifier*

to parts p:P resulting in the analyser writing down the unique identifier type and observer domain description text according to the following schema: 100

6. observe_unique_identifier schema

Narration:
[s] ... narrative text on unique identifier sort PI ...
[u] ... narrative text on unique identifier observer **uid**_P ...
[η] ... narrative text on type name, an RSL^+Text observer ...
[a] ... axiom on uniqueness of unique identifiers ...
Formalisation:
type

[s] PI
value
[u] **uid**_P: P → PI
[u] η PI → ≪ P ≫
axiom [Disjointness of Domain Identifier Types]
[a] $\mathcal{A}$: $\mathcal{U}$(PI,PI_i,PI_j,...,PI_k) ■

101

Example 16 Observe Transport System Identifiers: We refer to example Sect. 6.2.7 [pp. 37] annotation and formalisation Items 26–28d ■

2.7 Part Mereologies – cf. s. 6.2.9 Pg. 38 102

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 [26, 27].

2.7.1 Part Relations 103

Which are the relations that can be relevant for part-hood? We give some examples. (i) Two otherwise distinct parts may *"share"* values.[12] By *'sharing'* values we shall, as a generic example, mean that two parts of different sorts has the same attributes but that one *'defines'* the attribute, like, for example *'programming'* its values, cf. df.27 pp.26, whereas the other *'uses'* these values, like, for example considering them *'inert'*, cf. df.22 pp26. (ii) Two otherwise distinct parts may be said to, for example, be topologically "adjacent" or one "embedded"
104 within the other. These examples are in no way indicative of the "space" of part relations that may be relevant for part-hood. The domain analyser is expected to do a bit of experimental research in order to discover necessary, sufficient and pleasing "mereology-hoods"!

2.7.2 Part Mereology: Types and Functions 105

Analysis Prompt 18 *has_mereology: To discover necessary, sufficient and pleasing "mereology-hoods" the analyser can be said to endow a truth value,* **true**, *to the* **domain analysis prompt***:*

- *has_mereology*

When the domain analyser decides that some parts are related in a specifically enunciated mereology, the analyser has to decide on suitable *mereology types* and *mereology observers*
106 (i.e., part relations).

Domain Description Prompt 7 *observe_mereology*: *If has_mereology(p) holds for parts p of type P, then the analyser can apply the* **domain description prompt***:*

- *observe_mereology*

to parts of that type and write down the mereology types and observer domain description text
107 *according to the following schema:*

[12] For the concept of attribute value see Sect. 2.8.2 [pp. 24].

7. observe_mereology schema

Narration:

[t] ... narrative text on mereology type ...
[m] ... narrative text on mereology observer ...
[a] ... narrative text on mereology type constraints ...

Formalisation:

type
[t] MT[13]
value
[m] **obs_mereo**_P: P → MT
axiom [Well−formedness of Domain Mereologies]
[a] $\mathcal{A}$: $\mathcal{A}$(MT)

108

Example 17 Observe Transport System Mereology: We refer to example Sect. 6.2.9 [pp. 38] annotation and formalisation Items 40–43 ■

2.8 Part Attributes – cf. s. 6.2.10 Pg. 39 109

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 more "free-wheeling" sets of **internal qualities**. 110

Example 18 Example Part Attributes: These are examples of part attributes: the carat of a diamond; the number of residents of Gibraltar; the medium diameter and length of the equator; and the length and location[14] of a street segment (i.e., a link).

2.8.1 Inseparability of Attributes from Parts and Materials 111

Parts and materials are typically recognised because of their spatial form and are otherwise characterised by their intangible, but measurable attributes. That is, whereas endurants, whether discrete (as are parts and components) or continuous (as are materials), are physical, tangible, in the sense of being spatial [or being abstractions, i.e., concepts, of spatial endurants], attributes are intangible: cannot normally be touched[15], or seen[16], but can be objectively measured[17]. Thus, in our quest for describing domains where humans play an

[14]Note that we do not presently describe what a location is.
[15]One can see the red colour of a wall, but one touches the wall.
[16]One cannot see electric current, and one may touch an electric wire, but only if it conducts high voltage can one know that it is indeed an electric wire.
[17]That is, we restrict our domain analysis with respect to attributes to such quantities which are observable, say by mechanical, electrical or chemical instruments. Once objective measurements can be made of human feelings, beauty, and other, we may wish to include these "attributes" in our domain descriptions.

active rôle, we rule out subjective "attributes": feelings, sentiments, moods. Thus we shall abstain, in our domain science also from matters of aesthetics. We equate all endurants which, besides possible type of unique identifiers (i.e., excepting materials) and possible type of mereologies (i.e.,, excepting components and materials), have the same types of attributes, with one sort. Thus removing a quality from an endurant makes no sense: the endurant of that type either becomes an endurant of another type or ceases to exist (i.e., becomes a
112 non-entity) !

Example 19 **Inseparability of Attributes**: Let the part be a link (i.e., street segment). It must have a length a link without a length is meaningless. It must have a location a link without a location is meaningless.

2.8.2 Attribute Quality and Attribute Value 113

We distinguish between an attribute (as a logical proposition, of a name, i.e.) type, and an attribute value, as a value in some value space.

Analysis Prompt 19 *attribute_types: One can calculate the set of attribute types of parts and materials with the following* **domain analysis prompt***:*

- *attribute_types*

Thus for a part p we may have **attribute_types**$(p) = \{A_1, A_2, ..., A_m\}$.

114

Example 20 **Example Attribute Sorts**: Let the part be a pipeline unit such as a pipe, a pump, a valve, a fork, or a join. the *material* "flowed" by the pipeline; the *location* of the unit; the *diameter* of a pipe; the [dynamically changeable] *valve position* (`open, closed`, ...); the current and (for guaranteeing laminar flow) maximal in- and out-flows[18] of the pipeline units; et cetera. Notice that there are possibly very many other attributes: we may model some of these; others we may choose to ignore.

2.8.3 Part and Material Attributes: Types and Functions 115

Let us recall that attributes cover qualities other than unique identifiers and mereology. Let us then consider that parts and materials have one or more attributes. These attributes are qualities which help characterise "what it means" to be a part or a material. Note that we expect every part and material to have at least one attribute. The question is now, in general,
116 how many and, particularly, which.

Domain Description Prompt 8 *observe_attributes*: *The domain analyser experiments, thinks and reflects about part attributes. That process is initiated by the* **domain description prompt***:*

- *observe_attributes.*

The result of that **domain description prompt** *is that the domain analyser cum describer writes down the* attribute (sorts or) types and observers *domain description text according to the*
117 *following schema:*

8. `observe_attributes` **schema**

Narration:

[t] ... narrative text on attribute sorts ...
[o] ... narrative text on attribute sort observers ...
[v] ... narrative text on set **of** attribute **value** observers ...
[i] ... narrative text on attribute sort recognisers ...
[p] ... narrative text on attribute sort proof obligations ...

Formalisation:

type
[t] A_i $[1 \leq i \leq n]$
value
[o] **attr**_A_i:P→A_i i:[1..n]
[v] **obs_attrib_values**_P(p) ≡ { **attr**_A_1(p),**attr**_A_2(p),...,**attr**_A_n(p) }
[i] **is**_A_i:(A_1|A_2|...|A_n)→**Bool** i:[1..n]
proof obligation [Disjointness of Attribute Types]
[p] $\mathcal{PO}$: **let** P be any part sort **in** [the domain description]
[p] **let** a:(A_1|A_2|...|A_n) **in** **is**_A_i(a) ≠ **is**_A_j(a) **end end** [i≠i, i,j:[1..n]]

118

Example 21 **Road Transport System Attribute Observers**: We refer to example Sect. 6.2.10 narrative and formulas Items 46 [pp. 39] to 56d [Page 40].

2.8.4 Attribute Categories 119

Michael A. Jackson [28] has suggested a hierarchy of attribute categories: static or dynamic values – and within the dynamic value category: inert values or reactive values or active values – and within the dynamic active value category: autonomous values or biddable values or programmable values. We now review these attribute value types. The review is based on [28, M.A. Jackson]. **Part attributes** are either constant or varying, i.e., **static** or **dynamic** 120
attributes.

Analysis Prompt 20 *`is_static_attribute`: By a* **static attribute**, *a:A, we shall understand an attribute whose values are constants, i.e., cannot change.*

Analysis Prompt 21 *`is_dynamic_attribute`: By a* **dynamic attribute**, *a:A, we shall understand an attribute whose values are variable, i.e., can change. Dynamic attributes are either inert, reactive or active attributes.*

121

[18] Note that we do not presently describe the units in which flow are measured.

Analysis Prompt 22 *is_inert_attribute:* *By an* **inert attribute***, a:A, we shall understand a dynamic attribute whose values only change as the result of external stimuli where these stimuli prescribe new values.*

Analysis Prompt 23 *is_reactive_attribute:* *By a* **reactive attribute***, a:A, we shall understand dynamic attributes whose value, if they vary, change in response to external stimuli, where these stimuli come from outside the domain of interest.*

122

Analysis Prompt 24 *is_active_attribute:* *By an* **active attribute***, a:A, we shall understand a dynamic attribute whose values change (also) of its own volition. Active attributes are either* autonomous, biddable *or* programmable *attributes.*

Analysis Prompt 25 *is_autonomous_attribute:* *By an* **is_autonomous_attribute***(a), we shall understand a dynamic active attribute whose values change value only "on their own volition". The values of an autonomous attributes are a "law onto themselves and their surroundings".*

123

Analysis Prompt 26 *is_biddable_attribute:* *By a* **biddable attribute***, a:A, we shall understand a dynamic active attribute whose values* are prescribed but may fail to be observed as such.

Analysis Prompt 27 *is_programmable_attribute:* *By a* **programmable attribute***, a:A, we shall understand a dynamic active attribute whose values can be prescribed.*

124

Figure 2 captures an attribute value ontology.

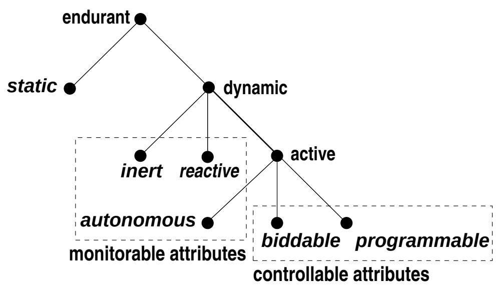


Figure 2: Attribute Value Ontology

125

Example 22 **Road Transport System Attribute Categories**: These are examples of attribute categories of the road transport system of Sect. 6: *static:* link and hub locations, link lengths, automobile brand names; *inert:* ... TO COME ...; *reactive:* ... TO COME ...; *autonomous:* ... TO COME ...; *biddable:* ... TO COME ...; *programmable:* automobile position and automobile, link and hub histories.

126

1 Given a part p we can calculate its **static attributes**.

2 Given a part p we can calculate its **controllable attributes**, i.e., the biddable and programmable attributes.

3 And given a part p we can calculate its **monitorable attributes**, i.e., the inert, reactive and autonomous attributes.

4 These three sets make up all the attributes of part p.

127

```
value
1   stat_attr_typs: P → ≪ SA1×SA2×...×SAs ≫
2   ctrl_attr_typs:  P → ≪ CA1×CA2×...×CAc ≫
3   mon_attr_typs:  P → ≪ MA1×MA2×...×MAm ≫
axiom
4   ∀ p:P •
4       let ≪ SA1×SA2×...×SAs ≫  = stat_attr_typs(p),
4           ≪ CA1×CA2×...×CAc ≫  = ctrl_attr_typs(p),
4           ≪ MA1×MA2×...×MAm ≫  = mon_attr_typs(p) in
4       card{SA1,SA2,...,SAs}+card{CA1,CA2,...,CAc}+card{MA1,MA2,...,MAm}
4       = card{SA1,SA2,...,SAs,CA1,CA2,...,CAc,MA1,MA2,...,MAm} end
```

128

5 Given a part p we can calculate its static attribute values.

6 Given a part p we can calculate its controllable, i.e., the biddable and programmable attribute values.

```
value
5   stat_attr_vals: P → SA1×SA2×...×SAs
5   stat_attr_vals(p) ≡
5       let ≪  SA1×SA2×...×SAs ≫  = stat_attr_typs(p) in
5       (attr_SA1(p),attr_SA2(p),...,attr_SAs(p)) end

6   ctrl_attr_vals: P → CA1×CA2×...×CAc
6   ctrl_attr_vals(p) ≡
6       let ≪  CA1×CA2×...×CAc ≫  = ctrl_attr_typs(p) in
6       (attr_CA1(p),attr_CA2(p),...,attr_CAc(p)) end
```

3 A Transcendental Transformation – cf. s. 6.3 Pg. 40 129

It should be clear to the reader that in *domain analysis & description* we are reflecting on a number of *philosophical issues*. First and foremost on those of *epistemology* and *ontology*. In this section on a sub-field of epistemology, namely that of a number of issues of *transcendental* nature. We refer to [29, pp 878–880] [30, pp 807–810] [31, pp 54–55 (1998)]. 130

Definition 17 Transcendental*: By* **transcendental** *we shall understand the philosophical notion:* **the a priori or intuitive basis of knowledge, independent of experience.**

A priori knowledge or intuition is central: By *a priori* we mean that it not only precedes, but also determines rational thought.

Definition 18 Transcendental Transformation*: By a* **transcendental transformation** *we shall understand the philosophical notion:* **a transcendental "conversion" of one kind of knowledge into a seemingly different kind of knowledge.**

Definition 19 Transcendentality*: By* **transcendentality** *we shall here mean the philosophical notion:* the state or condition of being transcendental.

131

Example 23 Transcendentality: We can speak of a bus in at least three *senses:*

(i) The bus as it is being `"serviced"` (maintained) at an automobile garage;

(ii) the bus as it `"speeds"` down its route; and

(iii) the bus as it `"appears"` (listed) in a bus time table.

The three *senses* are:

(i) as an **endurant** (here a *part*),

(ii) as a **perdurant** (as we shall see a *behaviour*), and

(iii) as an **attribute**[19] ■

132 Example 23, we claim, reflects *transcendentality* as follows:

- We have knowledge of an endurant (i.e., a part) being an endurant.
- We are then to assume that the perdurant referred to in (ii) is an aspect of the endurant mentioned in (i) – where perdurants are to be assumed to represent a different kind of knowledge.
- And, finally, we are to further assume that the attribute mentioned in (iii) is somehow related to both (i) and (ii) – where at least this attribute is to be assumed to represent yet a different kind of knowledge.

4 Perdurants – cf. s. 6.4 Pg. 41 133

So the transcendental deduction to be performed here is that of associating with each part – "existing" in space – a behaviour – "existing" in time.

Perdurants can thus be explained in terms of a notion of *state* and a notion of *time*. We
134 refer to Sect. 7.2 for a discussion of the concept of time.

[19] – in this case rather: as a fragment of a bus time table *attribute*

To speak about behaviours, that is, to describe behaviours, we choose a model for behaviours. We choose that of CSP [32]. With CSP is associated the notions of *processes* (which serve to model behaviours), *channels*, ch, (which serve to model communication between behaviours), and *output/input* clauses: ch!v, respectively ch? which serves to express the offering of a value, v on channel ch, respectively the offering to accept such a value. We shall use these notions freely.

4.1 States – cf. s. 6.2.6 Pg. 37 135

Definition 20 State: *By a* **state** *we shall understand any collection of* **parts** *or* **components** *or* **materials** ■

4.2 On Actions, Events, Behaviours and Actors 136

To us perdurants are further, pragmatically, analysed into *action*s, *event*s, and *behaviour*s. We shall define these terms below. Common to all of them is that they potentially change a state. Actions and events are here considered atomic perdurants. For behaviours we distinguish between discrete and continuous behaviours.

4.2.1 Actors 137

Definition 21 Actor: *By an* **actor** *we shall understand something that is capable of initiating and/or* **carrying out** *actions, events or behaviours* ■

138

Actors will play an important rôle in our domain analysis & description. By what we learn from our study of Sørlander's Philosophy some endurants (of a kind we shall introduce much later[20]) can, by a *transcendental deduction*, "become" perdurants some of which thereby "acting" in rôles of *actors*. 139

Example 24 Actors: Automobile *endurants* *"transmogrify"* into automobile *perdurants* which "subsume" rôles of *humans* in that we "include" humans in the form of automobile drivers in the non-deterministic behaviour automobile perdurants ■

4.2.2 Discrete Actions 140

Definition 22 Discrete Action: *By a* **discrete action** *[33, Wilson and Shpall] we shall understand a* **foreseeable** *thing which deliberately and potentially changes a well-formed state, in one step, usually into another, still well-formed state, and for which an actor can be made responsible* ■

141

Example 25 Discrete Actions: Here are some examples of discrete actions: the removal, i.e., closing of a street segment, i.e., a link, from a road net; the insertion of a street segment-between two street intersections, i.e., hubs, of a road net; and the removal of an automobile from the road net.

[20] *humans* [Sect. 10.5 Pg. 66] and, although not a concept in [15, 18], their *artifacts* [Sect. 10.7 Pg. 66]

4.2.3 Discrete Events 142

Definition 23 Event: *By an* **event** *we shall understand some* **unforeseen** *thing, that is, some 'not-planned-for' "action", one which surreptitiously, non-deterministically changes a well-formed state into another, but usually not a well-formed state, for which no particular domain actor can be made responsible* ■

143

Example 26 Discrete Events: Here are some examples of discrete events: a mud slide which effectively blocks, i.e., closes, a link; and the crashing of two automobiles.

4.2.4 Discrete Behaviours 144

Definition 24 Discrete Behaviour: *By a* **discrete behaviour** *we shall understand a set of sequences of potentially interacting sets of discrete actions, events and behaviours* ■

145

Example 27 Discrete Behaviours: Here are some examples of discrete behaviours: the drive of an automobile along a road net; the sequence of pumping and not-pumping, concurrent with and/or before/after opening and closing valves of a pipeline system; the waiting of an automobile stopped at a traffic light for it turning green; and the road (hub or link) "carrying" automobiles ■

• • •

In this paper we shall omit consideration of concepts of continuous actions, events and behaviours.

4.3 Channels – cf. s. 6.4.2 Pg. 41 146

The fact that a part, p of sort P with unique identifier p_i, has a mereology, for example the set of unique identifiers $\{q_{a_i}, q_{b_i}, ..., q_{d_i}\}$ identifying parts $\{q_a, q_b, ..., q_d\}$ of sort Q, **may mean** that parts p and $q \in \{qa, qb, ..., qd\}$ may wish to exchange – for example, attribute – values,
147 one way (from p to the q's) or the other (vice versa) or in both directions.

Figure 3 Pg. 31 shows (left) two dotted rectangle box (part) and (right) two corresponing,
148 rounded box (behaviour and channel) diagrams. We explain the figure: The left fragment of the figure intends to show a `1:1 Constellation` of a single p:P box and a single q:Q part, respectively, indicating, within these parts, their unique identifiers and mereologies. The right fragment of the figure intends to show a `1:n Constellation` of a single p:P box and a set of q:Q parts, now with arrowed lines connecting the p part with the q parts. These lines are intended to show channels. We show them with two way arrows. We could instead have chosen one way arrows, in one or the other direction. The directions are intended to show a direction of value transfer. We have given the same channel names to all examples,
149 ch_PQ. We have ascribed channel message types MPQ to all channels.[21] Figure 4 shows an
150 arrangement similar to that of Fig. 3 [next page], but for an `m:n Constellation`. The channel declarations corresponding to Figs. 3 and 4 are:

[21]Of course, these names and types would have to be distinct for any one domain description.

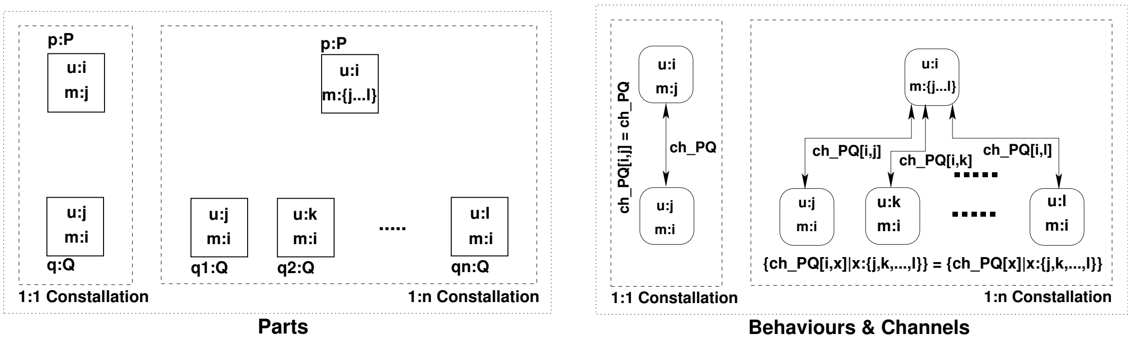


Figure 3: Two Part and Behaviour/Channel Constellations: *u:p* unique id. *p*; *m:p* mereology *p*

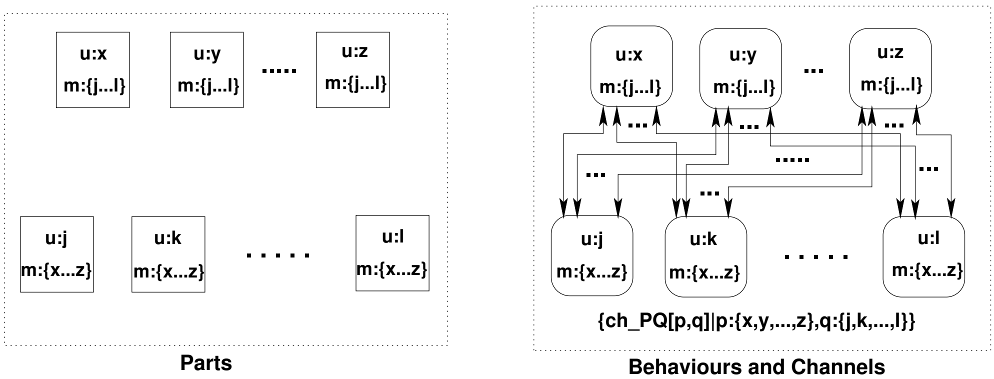


Figure 4: Multiple Part and Behaviour/Channel Constellations: *u:p* unique id. *p*; *m:p* mereology *p*

channel

[1] ch_PQ[i,j]:MPQ
[2] { ch_PQ[i,x]:MPQ | x:{j,k,...,l} }
[3] { ch_PQ[p,q]:MPQ | p:{x,y,...,z}, q:{j,k,...,l} }

Since there is only one index i and j for channel [1], its declaration can be reduced. Similarly there is only one i for declaration [2]:

channel

[1] ch_PQ:MPQ
[2] { ch_PQ[x]:MPQ | x:{j,k,...,l} }

151

7 The following description identities holds:

7 { ch_PQ[x]:MPQ | x:{j,k,...,l} } ≡ ch_PQ[j],ch_PQ[k],...,ch_PQ[l],


```
7  { ch_PQ[p,q]:MPQ | p:{x,y,...,z}, q:{j,k,...,l} } ≡
7      ch_PQ[x,j],ch_PQ[x,k],...,ch_PQ[x,l],
7      ch_PQ[y,j],ch_PQ[y,k],...,ch_PQ[y,l],
7      ...,
7      ch_PQ[z,j],ch_PQ[z,k],...,ch_PQ[z,l]
```

4.4 Behaviours 152

4.4.1 Behaviour Signatures – cf. s. 6.4.3 Pg. 41

We associate with each part, p:P, a behaviour $\mathcal{M}_P$. Behaviours have, as first argument, their unique part identifier: **uid**_P(p). Behaviours evolves around a state in the form of a set of
153 values: its possibly changing mereology, mt:MT and the attributes of the part.[22] A behaviour signature is therefore:

$\mathcal{M}_P$: ui:UI×me:MT×sa:stat_attr_typs(p) → ca:ctrl_attr_typs(p) → calc_i_o_chn_refs(p) **Unit**

where (i) ui:UI is the unique identifier value and type of part p; (ii) me:MT is the value and type mereology of part p, me = **obs_mereo**_P(p); (iii) sa:stat_attr_typs(p): *static attribute* types of part p:P; (iv) ca:ctrl_attr_typs(p): *controllable attribute* types of part p:P; (v) calc_i_o_chn_refs(p) calculates channel references to the **in**put channels reflecting the *monitorable attributes* of p and the **in**put/**out**put and the **out**put channels designated in the mereology, me, of p.

4.4.2 Behaviour Definitions – cf. s. 6.4.4 Pg. 42 154

Let P be a composite sort defined in terms of endurant[23] sub-sorts E_1, E_2, ..., E_n. The behaviour description *translated* from p:P, is composed from a behaviour description, $\mathcal{M}_P$, relying on and handling the unique identifier, mereology and attributes of part p to be *translated* with behaviour descriptions $\beta_1, \beta_2, \ldots, \beta_n$ where: β_1 is *translated* from e_1:E_1, β_2 is *translated* from e_2:E_2, ..., and β_n is *translated* from e_n:E_n. The domain description *transcendental schema* below "formalises" the above.
155

Transcendental Schema 1

Abstract is_composite(p)

```
value
   Translate_P: P → RSL+Text
   Translate_P(p) ≡
      let ui = uid_P(p), me = obs_mereo_P(p),
            sa = stat_attr_vals(p), ca = ctrl_attr_vals(p),
            MT = mereo_type(p), ST = stat_attr_typs(p), CT = ctrl_attr_typs(p),
            IOR = calc_i_o_chn_refs(p), IOD = calc_all_ch_dcls(p) in
      ≪ channel
```

[22] We leave out consideration of possible components and materials of the part.

[23] – structures or composite

IOD
value
$\mathcal{M}_P$: P_UI × MT × ST CT IOR **Unit**
$\mathcal{M}_P$(ui,me,sta)(ca) ≡ $\mathcal{B}_P$(ui,me,sta)(ca)
,⟫ Translate$_{P_1}$(**obs_endurant_sorts**_E$_1$(p))
⟪⟫ Translate$_{P_2}$(**obs_endurant_sorts**_E$_2$(p))
⟪⟫ ...
⟪⟫ Translate$_{P_n}$(**obs_endurant_sorts**_E$_n$(p))
end

Expression $\mathcal{B}_P$(ui,me,sta)(ca,pa) stands for the *behaviour definition body* in which the names ui, me, sta, ca and pa are bound to the *behaviour definition head*, i.e., the left hand side of the ≡. Endurant sorts E$_1$, E$_2$, ..., E$_n$ are obtained from the `observe_endurant_sorts` prompt, Page 17. We informally explain the **Translate**$_{\mathsf{P}_i}$ function. It takes endurants and produces RSL$^+$Text. Resulting texts are bracketed: ⟪rsl_text⟫ For the case that an endurant is a structure there is only its elements to compile; otherwise Schema 2 is as Schema 1 ■ 156 157

Transcendental Schema 2

is_structure(e)

value
Translate$_P$(p) ≡
Translate$_{P_1}$(**obs_endurant_sorts**_P$_1$(p))
⟪⟫ **Translate**$_{P_2}$(**obs_endurant_sorts**_P$_2$(p))
⟪⟫ ...
⟪⟫ **Translate**$_{P_n}$(**obs_endurant_sorts**_P$_n$(p))

Let P be a composite sort defined in terms of the concrete type **Q-set**. The process definition compiled from p:P, is composed from a process, $\mathcal{M}_P$, relying on and handling the unique identifier,the mereology and the attributes of process p as defined by P operating in parallel with processes q:**obs_part**_Qs(p). The domain description "compilation" schematic below "formalises" the above ■ 158 159

Transcendental Schema 3

Concrete is_composite(p)

type
Qs = Q-**set**
value
qs:Q-**set** = **obs_part**_Qs(p)
Translate$_P$(p) ≡
let ui = **uid**_P(p), me = **obs_mereo**_P(p),
sa = stat_attr_vals(p), ca = ctrl_attr_vals(p)

ST = stat_attr_typs(p), CT = ctrl_attr_typs(p),
IOR = calc_i_o_chn_refs(p), IOD = calc_all_ch_dcls(p) **in**
$\lll$ **channel**
IOD
value
$\mathcal{M}_P$: P_UI×MT×ST CT IOR **Unit**
$\mathcal{M}_P$(ui,me,sa)(ca) ≡ $\mathcal{B}_P$(ui,me,sa)(ca) $\ggg$
{ $\lll$,$\ggg$ Translate$_Q$(q)|q:Q•q ∈ qs }
end ■

160

Transcendental Schema 4

is_atomic(p)

value
Translate$_P$(p) ≡
let ui = **uid**_P(p), me = **obs_mereo**_P(p),
sa = stat_attr_vals(p), ca = ctrl_attr_vals(p),
ST = stat_attr_typs(p), CT = ctrl_attr_typs(p),
IOR = calc_i_o_chn_refs(p), IOD = calc_all_chs(p) **in**
$\lll$ **channel**
IOD
value
$\mathcal{M}_P$: P_UI×MT×ST PT IOR **Unit**
$\mathcal{M}_P$(ui,me,sa)(ca) ≡ $\mathcal{B}_P$(ui,me,sa)(ca) $\ggg$
end ■

161

Transcendental Schema 5

Core Behaviour

The core processes can be understood as never ending, “tail recursively defined” processes:

$\mathcal{B}_P$: uid:P_UI×me:MT×sa:SA
→ ct:CT
→ **in** in_chns(p) **in,out** in_out_chns(me) **Unit**
$\mathcal{B}_P$(p)(ui,me,sa)(ca) ≡
let (me′,ca′) = $\mathcal{F}_P$(ui,me,sa)(ca) **in** $\mathcal{M}_P$(ui,me′,sa)(ca′) **end**

$\mathcal{F}_P$: P_UI×MT×ST → CT→ in_out_chns(me) → MT×CT ■

4.5 Initial Running Systems – cf. s. 6.4.5 Pg. 43 162

To round it all off a narrative and a formalisation must be done of *"a running system"*. Up till now the behaviours for all relevant parts have been defined. Now a decision must be made as to which of these are the basis for an initial system. There may be several candidates for initial running systems, that is, collection of concurrently operating behaviours. So the domain analyser cum describer selects all or some candidates. For each the chosen behaviours are properly initialised. And that is that!

5 A Coin Has Two Sides 163

The transcendental deduction that "turns" parts into behaviours can also be interpreted as follows: The part and the "corresponding" behaviour "exist" at one and the same time: the part is characterised by its *internal qualities*, and these are the arguments, in one form or another of the behaviour. The properties of the internal qualities of parts, expressed, for example, in the form of **axioms**, hold for all times (a concept not present in the treatment of endurants), and are to be maintained by the corresponding behaviours, as expressed, for example, in **pre/post** conditions. Let us recall essential "features" of parts and behaviours. For parts, $p{:}P$, we can 164
generally express the following:

Pg. 19:	**uid**_P: P → PI	
Pg. 22:	**obs_mereo**_P: P → $\mathcal{E}$(PI$_1$,PI$_2$,...,PI$_m$)	
Pg. 24:	**attr**_sA$_1$: P → sA$_1$	is_static_attribute
	...	is_static_attribute
	attr_sA$_n$: P → sA$_{n_s}$	is_static_attribute
	attr_cA$_1$: P → cA$_1$	is_controllable_attribute
	...	is_controllable_attribute
	attr_cA$_{n_c}$: P → cA$_{n_c}$	is_controllable_attribute
	attr_mA$_1$: P → mA$_1$	is_monitorable_attribute
	...	is_controllable_attribute
	attr_mA$_{n_m}$: P → mA$_{n_m}$	is_monitorable_attribute

where $n_s \geq 0$, $n_c \geq 0$, and $n_m \geq 0$. For "corresponding" behaviours, $\mathcal{M}_P$, we have (cf. Process 165
Schema 1 [pp. 32]):

let ui = **uid**_P(p), me = **obs_mereo**_P(p),
  sv = stat_attr_vals(p), cv = ctrl_attr_vals(p),
  MT = mereo_type(p), ST = stat_attr_typs(p), CT = ctrl_attr_typs(p),
  IOR = calc_i_o_chn_refs(p), IOD = calc_all_ch_dcls(p) **in**
≪ **channel**
    IOD
  value
    $\mathcal{M}_P$: ui:P_UI × me:MT × sv:ST cv:CT IOR **Unit**
    $\mathcal{M}_P$(ui,me,sv)(cv) ≡ $\mathcal{B}_P$(ui,me,sv)(cv) ≫
end

We leave it to the reader to study these two sets of formulas.

6 An Example: A Road Transport System 166

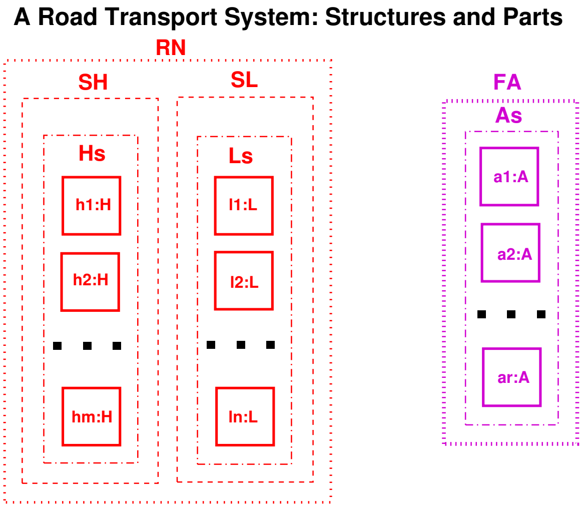


Figure 5: A Road Transport System

6.1 The Universe of Discourse – cf. s. 2.1 pp. 12 167

The universe of discourse is *road transport systems*. We analyse & describe not the class of all road transport systems but a representative subclass, UoD, is *structured* into such notions as a road net, RN, of hubs, H, (intersections) and links, L, (street segments between intersections); a fleet of automobiles, FA, of automobiles, A; et cetera. See Fig. 5 Pg. 36

The delineation of *the universe of discourse* satisfies the characterisation of what a domain must "at least" contain – only if we assume that automobiles include humans — in a sense we do not have to explicate.

6.2 Endurants – cf. s. 2 pp. 12 169

6.2.1 Structures – cf. s. 2.4 pp. 14

See Description Prompt 2, Pg. 17.

8 There is the *universe of discourse*, UoD. It is structured into

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

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

type
8 UoD **axiom** ∀ uod:UoD • is_structure(uod).
9 RN **axiom** ∀ rn:RN • is_strucure(rn).
10 FA **axiom** ∀ fa:FA • is_structure(fa).
value
9 obs_RN: UoD → RN
10 obs_FA: UoD → FA

6.2.2 Parts, Components and Materials – cf. s. 2.5 pp. 15 170

See Description Prompt 2, Pg. 17.

11 The road net consists of

a a structure, SH, of hubs and

b a structure, SL, of links.

12 The fleet of automobiles consists of

a a set, As of automobiles.

type
11a SH **axiom** ∀ sh:SH • is_structure(sh)
11b SL **axiom** ∀ sl:SL • is_structure(sl)
12a As = A-set
value
11a obs_SH: RN → SH
11b obs_SL: RN → SL
12a obs_As: FA → As

6.2.3 Parts – cf. s. 2.5.1 pp. 15 172

See Description Prompt 3, Pg. 18.

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

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

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

```
type
13  H, sH = H-set  axiom ∀ h:H • is_atomic(h)
14  L, sL = L-set  axiom ∀ l:L • is_atomic(l)
15  A, sA = A-set  axiom ∀ a:A • is_atomic(a)
value
13  obs_sH: SH → sH
14  obs_sL: SL → sL
15  obs_sA: SA → sA
```

6.2.4 Components – cf. s. 2.5.2 pp. 18 174

See Description Prompt 4, Pg. 19.

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

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

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

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

6.2.5 Materials – cf. s. 2.5.3 pp. 20 176

See Description Prompt 5, Pg. 20.

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

18 Links may contain material.

19 That material is water, W.

```
type
19  W
value
18  obs_material: L → W
18  pre: obs_material(l) ≡ has_material(h)
```

6.2.6 States – cf. s. 4.1 pp. 29 177

20 Let there be given a universe of discourse, *rts*. It is an example of a state.

From that state we can calculate other states.

21 The set of all hubs, *hs*.

22 The set of all links, *ls*.

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

24 The set of all automobiles, *as*.

25 The set of all parts, *ps*.

```
value
20  rts:UoD
21  hs:H-set ≡ ≡ obs_sH(obs_SH(obs_RN(rts)))
22  ls:L-set ≡ ≡ obs_sL(obs_SL(obs_RN(rts)))
23  hls:(H|L)-set ≡ hs∪ls
24  as:A-set ≡ obs_As(obs_FV(rts))
25  ps:(H|L|BC|B|A)-set ≡ hls∪bcs∪bs∪as
```

6.2.7 Unique Identifiers – cf. s. 2.6 pp. 21 179

See Description Prompt 6, Pg. 21

Part Identifiers

26 We assign unique identifiers to all parts.

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

28 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.
- d All parts have distinct identifiers.

```
type
26  H_UI, L_UI, A_UI
27  R_UI = H_UI | L_UI
value
28a  uid_H: H → H_UI
28b  uid_L: L → L_UI
28c  uid_A: A → A_UI
```

Extract Parts from Unique Identifiers

29 From the unique identifier of a part we can retrieve, $\wp$, the part having that identifier.

type
29 P = H | L | A
value
29 $\wp$: H_UI→H | L_UI→L | A_UI→A
29 $\wp$(ui) ≡ **let** p:(H|L|A)•p∈ps∧uid_P(p)=ui **in** p **end**

Unique Identifier Constants: We can calculate:

30 the *set*, $h_{ui}s$, of *u*nique *h*ub *i*dentifiers;

31 the *set*, $l_{ui}s$, of *u*nique *l*ink *i*dentifiers;

32 the *map*, $hl_{ui}m$, from *u*nique *h*ub *i*dentifiers to the *set* of *u*nique *l*ink *i*identifiers of the links connected to the zero, one or more identified hubs,

33 the *map*, $lh_{ui}m$, from *u*nique *l*ink *i*dentifiers to the *set* of *u*nique *h*ub *i*identifiers of the two hubs connected to the identified link;

34 the *set*, $r_{ui}s$, of all *u*nique hub and link, i.e., *r*oad *i*dentifiers;

35 the *set*, $a_{ui}s$, of *u*nique *a*utomobile *i*dentifiers;

value
30. $h_{ui}s$:H_UI-**set** ≡ {uid_H(h)|h:H•h ∈ hs}
31. $l_{ui}s$:L_UI-**set** ≡ {uid_L(l)|l:L•l ∈ ls}
34. $r_{ui}s$:R_UI-**set** ≡ $h_{ui}s$∪$l_{ui}s$
32. $hl_{ui}m$:(H_UI $\overrightarrow{m}$ L_UI-**set**) ≡
32. [h_ui↦luis|h_ui:H_UI,luis:L_UI-**set**•h_ui∈$h_{ui}s$
32. ∧(_,luis,_)=mereo_H(η(h_ui))] [cf. Item 40]
33. $lh_{ui}m$:(L+UI $\overrightarrow{m}$ H_UI-**set**) ≡
33. [l_ui↦huis [cf. Item 41]
33. | h_ui:L_UI,huis:H_UI-**set** • l_ui∈$l_{ui}s$
33. ∧ (_,huis,_)=mereo_L(η(l_ui))]
35. $a_{ui}s$:A_UI-**set** ≡ {uid_A(a)|a:A•a ∈ as}

6.2.8 Uniqueness of Part Identifiers 184

We must express the following axioms:

36 All hub identifiers are distinct.

37 All link identifiers are distinct.

38 All automobile identifiers are distinct.

39 All part identifiers are distinct.

axiom
36 **card** hs = **card** $h_{ui}s$
37 **card** ls = **card** $l_{ui}s$
38 **card** as = **card** $a_{ui}s$
39 **card** {$h_{ui}s$∪$l_{ui}s$∪$a_{ui}s$}
39 = **card** $h_{ui}s$+**card** $l_{ui}s$+**card** $a_{ui}s$

6.2.9 Part Mereologies – cf. s. 2.7 pp. 22 185

See Description Prompt 7, Pg. 22

40 The mereology of hubs is a triple: (i) the set of all automobile identifiers[24], (ii) the set of unique identifiers of the links that it is connected to and the set of all unique identifiers of all automobiles.[25], and (iii) an empty set.[26]

41 The mereology of links is a triple: (i) the set of all automobile identifiers, (ii) the set of the two distinct hubs they are connected to, and (iii) an empty set.

42 The mereology of an automobiles is a triple: (i) an empty set, (ii) an empty set, and (iii) the set of the unique identifiers of all links and hubs[27].

43 Empty sets are modelled as empty sets of tokens where tokens are further undefined.

type
43 ES = TOKEN-**set**
43 **axiom** ∀ es:ES•es={}
40 H_Mer = V_UI-**set**×L_UI-**set**×ES
40 **axiom** ∀ (vuis,luis,_):H_Mer • luis⊆$l_{ui}s$ ∧ vuis=$v_{ui}s$
41 L_Mer = V_UI-**set**×H_UI-**set**×ES
41 **axiom** ∀ (vuis,huis,_):L_Mer •
41 vuis=$v_{ui}s$ ∧ huis⊆$h_{ui}s$ ∧ **card**huis=2
42 A_Mer = ES×ES×R_UI-**set**
42 **axiom** ∀ (_,ruis,_):A_Mer • ruis=$r_{ui}s$

value
40 mereo_H: H → H_Mer
41 mereo_L: L → L_Mer
42 mereo_A: A → A_Mer

[24]This is just another way of saying that the meaning of hub mereologies involves the unique identifiers of all the automobiles that might pass through the hub `is_of_interest` to it

[25]... its link identifiers designate the links, zero, one or more, that a hub is connected to `is_of_interest` to both the hub and that these links is `interested` in the hub.

[26]... the hubs are not "proactive", i.e., that the universe of discourse have no parts that are `interested` in the hub.

[27]that the automobile might pass through

We can express some additional axioms, in this case for relations between hubs and links:

44 If hub, h, and link, l, are in the same road net,

45 and if hub h connects to link l then link l connects to hub h.

```
axiom
44  ∀ h:H,l:L • h ∈ hs ∧ l ∈ ls ⇒
44    let (_,luis,_)=mereo_H(h),(_,huis,)=mereo_L(l)
45    in uid_L(l) ∈ luis ⇒ uid_H(h) ∈ huis end
```

More mereology axioms need be expressed – but we leave, to the reader, to narrate and formalise those.

6.2.10 Part Attributes – cf. s. 2.8 pp. 23 189

We treat part attributes, sort by sort. See Description Prompt 8, Pg. 24

Hubs: We show just a few attributes:

46 There is a hub state. It is a set of pairs, (l_f,l_t) of link identifiers, where these link identifiers are in the mereology of the hub. The meaning of the hub state, in which, e.g., (l_f,l_t) is an element, is that the hub is open, **"green"**, for traffic *from* link l_f *to* link l_t. If a hub state is empty then the hub is closed, i.e., **"red"** for traffic from any connected links to any other connected links.

47 There is a hub state space. It is a set of hub states. The meaning of the hub state space is that its states are all those the hub can attain. The current hub state must be in its state space.

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

49 The link identifiers of hub states must be in the set, $l_{ui}s$, of the road net's link identifiers.

```
type
46  HΣ = (L_UI×L_UI)-set          [programmable, df.27 pp.26]
axiom
46  ∀ h:H • obs_HΣ(h) ∈ obs_HΩ(h)
type
47  HΩ = HΣ-set                   [static, df.20 pp.25]
48  H_Traffic                     [programmable, df.27 pp.26]
48  H_Traffic = A_UI →m (𝒯 × APos)*
axiom
48  ∀ ht:H_Traffic,ui:A_UI •
48      ui ∈ dom ht ⇒ time_ordered(ht(ui))
value
46  attr_HΣ: H → HΣ
47  attr_HΩ: H → HΩ
48  attr_H_Traffic: : → H_Traffic
```

```
axiom
49  ∀ h:H • h ∈ hs ⇒
49    let hσ = attr_HΣ(h) in
49    ∀ (l_ui i,li_ui i′):(L_UI×L_UI) •
49        (l_ui i,l_ui i′) ∈ hσ ⇒ {l_ui_i,l′_ui_i} ⊆ l_ui s end
value
48  time_ordered: 𝒯* → Bool
48  time_ordered(tvpl) ≡ ...
```

Links: We show just a few attributes:

50 There is a link state. It is a set of pairs, (h_f,h_t), of distinct hub identifiers, where these hub identifiers are in the mereology of the link. The meaning of a link state in which (h_f,h_t) is an element is that the link is open, **"green"**, for traffic *from* hub h_f *to* hub h_t. Link states can have either 0, 1 or 2 elements.

51 There is a link state space. It is a set of link states. The meaning of the link state space is that its states are all those the which the link can attain. The current link state must be in its state space. If a link state space is empty then the link is (permanently) closed. If it has one element then it is a one-way link. If a one-way link, l, is imminent on a hub whose mereology designates that link, then the link is a "trap", i.e., a "blind cul-de-sac".

52 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.

53 The hub identifiers of link states must be in the set, $h_{ui}s$, of the road net's hub identifiers.

```
type
50  LΣ = H_UI-set                 [programmable, df.27 pp.26]
axiom
50  ∀ lσ:LΣ•card lσ=2
50  ∀ l:L • obs_LΣ(l) ∈ obs_LΩ(l)
type
51  LΩ = LΣ-set                   [static, df.20 pp.25]
52  L_Traffic                     [programmable, df.27 pp.26]
52  L_Traffic = A_UI →m (𝒯×APos)*
value
50  attr_LΣ: L → LΣ
51  attr_LΩ: L → LΩ
52  attr_L_Traffic: : → L_Traffic
```

```
axiom
52  ∀ lt:L_Traffic,ui:A_UI•ui ∈ dom ht
52      ⇒ time_ordered(ht(ui))
53  ∀ l:L • l ∈ ls ⇒
53    let lσ = attr_LΣ(l) in
53    ∀ (h_ui i,h_ui i′):(H_UI×K_UI) •
53        (h_ui i,h_ui i′) ∈ lσ ⇒
53          {h_ui_i,h′_ui_i} ⊆ h_ui s end
```


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

54 Automobiles have a time attribute.

55 Automobiles have static number plate registration numbers.

56 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 hub*s, 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
54   𝒯                                        [inert, df.22 pp.26]
55   RegNo                                    [static, df.20 pp.25]
56   APos   == atHub | onLink [programmable, df.27 pp.26]
56a  atHub     :: h_ui:H_UI
56b  onLink    :: fh_ui:H_UI×l_ui:L_UI×frac:Fract×th_ui:H_UI
56b  Fract     = Real, axiom frac:Fract • 0<frac<1
56c  A_Hist  = (T × APos)*   [programmable, df.27 pp.26]
value
54   attr_T: A → 𝒯
55   attr_RegNo: A → RegNo
56   attr_APos: A → APos
56c   attr_A_Hist: A → A_Hist
axiom
56d  ∀ a:A •
56d      let (_,apos) = hd(attr_A_Hist(a)) in
56d      apos = attr_APos(a) end
```

Obvious attributes that are not illustrated are those of velocity and acceleration, forward or backward movement, turning right, left or going straight, etc.

The *acceleration, deceleration, even velocity,* or *turning right, turning left, moving straight*, or *forward* or *backward* are seen as *command actions*. As such they denote actions by the automobile — such as pressing the accelerator, or lifting accelerator pressure or *braking*, or *turning the wheel* in one direction or another, etc. As actions they have a kind of counterpart in the velocity, the acceleration, etc. attributes.

6.2.11 Discussion of Edurants, I 199

In Items 48 Pg. 39 and 52 Pg. 39, we illustrated an aspect of domain analysis & description that may seem, and at least some decades ago would have seemed, strange: namely that if we can think, hence speak, about it, then we can model it "as a fact" in the domain. The case in point is that we include among hub and link attributes their histories of the timed whereabouts of automobiles.[28]

6.2.12 Discussion of Endurants, II 200

We have chosen to model some discrete endurants as structures others as parts (usually composite). Those choices are made mostly to illustrate that the domain analysis & description has a choice. If a choice is made to model a discrete endurant as a structure then it entails that the domain analysis & description does not wish to "implement" that discrete endurant as a behaviour separate from its sub-endurants; If the choice is made to model a discrete endurant as a part then it entails that the domain analysis & description wishes to "implement" that discrete endurant as a behaviour separate from its sub-endurants. The following discrete endurants which are modelled as structures above, could, instead, if modelled as parts, have the entailed behaviours reflect the following possibilities: *road net*, rn:RN: The road net behaviour could be that of a road net authority charged with building, servicing, operating and maintaining the road net. Building and maintaining the road net could mean the insertion of new or removal of old links or hubs. Operating the road net could mean the gathering of automobile traffic statistics, the setting of hub states (traffic signal monitoring and control), etc. *aggregate of automobiles*, ps:PA: The aggregate of automobiles could be that of one or more *automobile clubs*, etc.

6.3 Transcendentality – cf. s. 3 pp. 27 202

We refer to Sect. 6.3 Defn. 23 Page 28.

Example 28 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[29] ■

[28] In this day and age of road cameras and satellite surveillance these traffic recordings may not appear so strange: We now know, at least in principle, of technologies that can record approximations to the hub and link traffic attributes.

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

6.4 Perdurants – cf. s. 4 pp. 28 203

6.4.1 States

Constants: We refer to Sect. 6.2.6 Pg. 37, and to App. 4.1 Pg. 29 We assume, as a constant, an arbitrarily selected universe of discourse, *uod*, and calculate from *uod* all its endurants.

value
20 rts:UoD
21 hs:H-**set** ≡:H-**set** ≡ obs_sH(obs_SH(obs_RN(rts)))
22 ls:L-**set** ≡:L-**set** ≡ obs_sL(obs_SL(obs_RN(rts)))
23 hls:(H|L)-**set** ≡ $hs \cup ls$
24 as:A-**set** ≡ obs_As(obs_FV(rts))

Indexed States: We shall

57 index automobiles

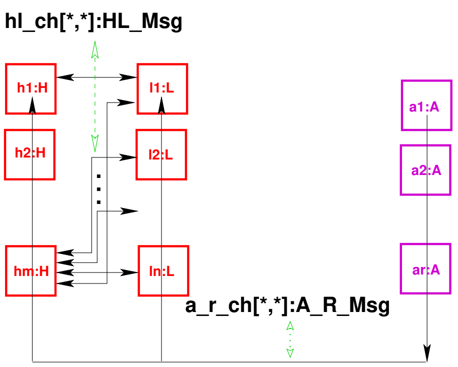
using the unique identifiers of these parts.

type
57 A_{ui}
value
57 ias:A_{ui}-**set** ≡
57 {a_{ui}|a:A,a:A_{ui}:A_{ui}•a∈as∧ui=uid_A(a)}

6.4.2 Channels – cf. s. 4.3 pp. 30 205

We shall argue for hub-to-link channels based on the mereologies of those parts. Hub parts may be topologically connected to any number, 0 or more, link parts. Only instantiated road nets knows which. Hence there must be channels between any hub behaviour and any link behaviour. Vice versa: link parts will be connected to exactly two hub parts. Hence there must be channels from any link behaviour to two hub behaviours. See the figure below:

206



Hub–to–Link Channels and Automobile to Road Channels

Channel Message Types: We ascribe types to the messages offered on channels.

58 Hubs and links communicate, both ways, with one another, over channels, `hl_ch`, whose indexes are determined by their mereologies.

59 Hubs send one kind of messages, links another.

60 Automobiles offer their current, timed positions to the road element, hub or link they are on, one way.

type
59 H_L_Msg, L_H_Msg
58 HL_Msg = H_L_Msg | L_F_Msg
60 A_R_Msg = T × APos

Channel Declarations

61 This justifies the channel declaration which is calculated to be:

channel
61 { hl_ch[h_ui,l_ui]:H_L_Msg
61 | h_ui:H_UI,l_ui:L_UI•i ∈ $h_{ui}s$∧j ∈ $lh_{ui}m$(h_ui) }
61 ∪
61 { hl_ch[h_ui,l_ui]:L_H_Msg
61 | h_ui:H_UI,l_ui:L_UI•l_ui ∈ $l_{ui}s$∧i ∈ $lh_{ui}m$(l_ui) }

We shall argue for automobile to road element channels based on the mereologies of those parts. Automobiles need communicate to all hubs and all links.

62 This justifies the channel declaration which is calculated to be:

channel
62 {a_r_ch[a_ui,r_ui]:A_R_Msg
62 |a_ui:A_UI,r_ui:R_UI•a_ui ∈ $a_{ui}s$∧r_ui ∈ $r_{ui}s$}

6.4.3 Behaviour Signatures – cf. s. 4.4.1 pp. 32 210

We first decide on names of behaviours. In Sect. 4.4.2, Pages 32–34, we gave schematic names to behaviours of the form $\mathcal{M}_P$. We now assign mnemonic names: from part names to names of transcendentally interpreted behaviours and then we assign signatures to these behaviours.

63 $\mathsf{hub}_{h_{ui}}$:

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


```
value
63   hub_{h_ui}:
63a  h_ui:H_UI×(auis,luis,_):H_Mer×HΩ
63b    → (HΣ×H_Traffic)
63c    → in,out { h_l_ch[ h_ui,l_ui ] | l_ui:L_UI:l_ui ∈ luis }
63d          { a_r_ch[ h_ui,a_ui ] | a_ui:A_UI•a_ui∈auis } Unit
63a    pre: auis = a_ui s ∧ luis = l_ui s
```

64 $\mathsf{link}_{l_{ui}}$:

a there is the usual "triplet" of arguments: unique identifier, mereology and static attributes;

b then there are the programmable attributes;

c and finally there are the input/output channel references: first those allowing communication between hub and link behaviours,

d and then those allowing communication between link and automobile behaviours.

```
value
64   link_{l_ui}:
64a  l_ui:L_UI×(auis,huis,_):L_Mer×LΩ
64b    → (LΣ×L_Traffic)
64c    → in,out { h_l_ch[ h_ui,l_ui ] | h_ui:H_UI:h_ui ∈ huis }
64d          { a_r_ch[ l_ui,a_ui ] | a_ui:A_UI•a_ui∈auis } Unit
64a    pre: auis = a_ui s ∧ huis = h_ui s
```

65 $\mathsf{automobile}_{a_{ui}}$:

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
65   automobile_{a_ui}:
65a  a_ui:A_UI×(_,_,ruis):A_Mer×rn:RegNo
65b    → apos:APos
65c    → in attr_T_ch
65d      in,out {a_r_ch[ a_ui,r_ui ]
65d               | r_ui:(H_UI|L_UI)•r_ui∈ruis} Unit
65a    pre: ruis = r_ui s ∧ a_ui ∈ a_ui s
```

6.4.4 Behaviour Definitions – cf. s. 4.4.2 pp. 32 217

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.

Automobiles:

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

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

68 informing the hub behaviour,

69 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,

70 or, again internally non-deterministically,

71 the automobile "disappears — off the radar" !

```
66   automobile_{a_ui}(a_ui,({},(ruis,auis),{}),rn)
66          (apos:atH(fl_ui,h_ui,tl_ui)) ≡
67     (ba_r_ch[ a_ui,h_ui ] ! (attr_T_ch?,atH(fl_ui,h_ui,tl_ui));
68      automobile_{a_ui}(a_ui,({},(ruis,auis),{}),rn)(apos))
69    ⌈⌉
69a   (let ({fh_ui,th_ui},ruis′)=mereo_L(℘(tl_ui)) in
69a         assert: fh_ui=h_ui ∧ ruis=ruis′
66      let onl = (tl_ui,h_ui,0,th_ui) in
69b     (ba_r_ch[ a_ui,h_ui ] ! (attr_T_ch?,onL(onl)) ∥
69b      ba_r_ch[ a_ui,tl_ui ] ! (attr_T_ch?,onL(onl))) ;
69c     automobile_{a_ui}(a_ui,({},(ruis,auis),{}),rn)
69c            (onL(onl)) end end)
70    ⌈⌉
71      stop
```

72 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**,

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" !

```
72        automobile_a_ui(a_ui,({},ruis,{}),rno)
72                      (vp:onL(fh_ui,l_ui,f,th_ui)) ≡
72(a)ii   (ba_r_ch[thui,aui]!atH(lui,thui,nxt_lui) ;
72(a)i     automobile_a_ui(a_ui,({},ruis,{}),rno)(vp))
72b       ⊓
72(b)i    (if not_yet_at_hub(f)
72(b)i      then
72(b)iA       (let incr = increment(f) in
66             let onl = (tl_ui,h_ui,incr,th_ui) in
72(b)iB         a−r_ch[l_ui,a_ui] ! onL(onl) ;
72(b)iC         automobile_a_ui(a_ui,({},ruis,{}),rno)
72(b)iC                          (onL(onl))
72(b)i          end end)
72(b)ii     else
72(b)iiA      (let nxt_lui:L_UI•nxt_lui ∈ mereo_H(℘(th_ui)) in
72(b)iiB       a_r_ch[thui,aui]!atH(l_ui,th_ui,nxt_lui) ;
72(b)iiC       automobile_a_ui(a_ui,({},ruis,{}),rno)
72(b)iiC                       (atH(l_ui,th_ui,nxt_lui)) end)
72(b)i     end)
72c       ⊓
72d        stop
72(b)iA    increment: Fract → Fract
```

Hubs: We model the hub behaviour vis-a-vis automobiles.

73 The hub behaviour

a non-deterministically, externally offers

b to accept timed automobile positions —

c which will be at the hub, from some vehicle, v_ui.

d The timed automobile hub position is appended to the front of that automobile's entry in the hub's traffic table;

e whereupon the hub proceeds as a hub behaviour with the updated hub traffic table.

f The hub behaviour offers to accept from any automobile.

g A **post** condition expresses what is really a **proof obligation**: that the hub traffic, ht′ satisfies the **axiom** of the endurant hub traffic attribute Item 48 Pg. 39.

value

```
73    hub_h_ui(h_ui,(,(luis,vuis)),hω)(hσ,ht) ≡
73a      ⌷
73b      { let m = ba_r_ch[h_ui,v_ui] ? in
73c          assert: m=(_,atHub(_,h_ui,_))
73d        let ht′ = ht † [a_ui ↦ ⟨m⟩⌢ht(a_ui)] in
73e        hub_h_ui(h_ui,(,(luis,auis)),(hω))(hσ,ht′)
73f      | a_ui:A_UI•a_ui∈auis end end }
73g    post: ∀ a_ui:A_UI•a_ui ∈ dom ht′
73g             ⇒ time_ordered(ht′(a_ui))
```

Links: Similarly we model the link behaviour vis-a-vis automobiles.

74 The link behaviour non-deterministically, externally offers

75 to accept timed automobile positions —

76 which will be on the link, from some automobile, a_ui.

77 The timed automobile link position is appended to the front of that automobile's entry in the link's traffic table;

78 whereupon the link proceeds as a link behaviour with the updated link traffic table.

79 The link behaviour offers to accept from any automobile.

80 A **post** condition expresses what is really a **proof obligation**: that the link traffic, lt′ satisfies the **axiom** of the endurant link traffic attribute Item 52 Pg. 39.

```
74   link_l_ui(l_ui,(_,(huis,auis),_),lω)(lσ,lt) ≡
74     ⌷
75     { let m = ba_r_ch[l_ui,a_ui] ? in
76          assert: m=(_,onLink(_,l_ui,_,_))
77       let lt′ = lt † [a_ui ↦ ⟨m⟩⌢lt(a_ui)] in
78       link_l_ui(l_ui,(huis,auis),hω)(hσ,lt′)
79     | a_ui:A_UI•a_ui∈auis end end }
80   post: ∀ a_ui:A_UI•a_ui ∈ dom lt′
80            ⇒ time_ordered(lt′(a_ui))
```

6.4.5 A Running System – cf. s. 4.5 pp. 35 227

Preliminaries: We recall the *hub*, *link* and the *automobile states* first mentioned in Sect. 6.2.6 Page 37.

value

21 hs:H-set ≡ ≡ obs_sH(obs_SH(obs_RN(rts)))

22 ls:L-set ≡ ≡ obs_sL(obs_SL(obs_RN(rts)))

24 as:A-set ≡ obs_As(obs_FA(rts))

Starting Initial Behaviours: We are reaching the end of this domain modelling example. Behind us there are narratives and formalisations 8 Pg. 36 – 80 Pg. 43. Based on these we now express the signature and the body of the definition of a *"system build and execute"* function.

81 The system to be initialised is

a the parallel composition (‖) of
b the distributed parallel composition (‖{...|...}) of
c all the hub behaviours,
d all the link behaviours, and
e all the automobile behaviours.

```
value
81   initial_system: Unit → Unit
81   initial_system() ≡
81c     ‖ { hub_{h_ui}(h_ui,me,hω)(htrf,hσ)
81c       | h:H•h ∈ hs,
81c         h_ui:H_UI•h_ui=uid_H(h),
81c         me:HMetL•me=mereo_H(h),
81c         hω:HΩ•hω=attr_HΩ(h),
81c         htrf:H_Traffic•htrf=attr_H_Traffic_H(h),
81c         hσ:HΣ•hσ=attr_HΣ(h)∧hσ ∈ hω
81c     }
81d     ‖ { link_{l_ui}(l_ui,me,lω)(ltrf,lσ)
81d         l:L•l ∈ ls,
81d         l_ui:L_UI•l_ui=uid_L(l),
81d         me:LMet•me=mereo_L(l),
81d         lω:LΩ•lω=attr_LΩ(l),
81d         ltrf:L_Traffic•ltrf=attr_L_Traffic_H(l),
81d         lσ:LΣ•lσ=attr_LΣ(l)∧lσ ∈ lω
81d     }
81e     ‖ { automobile_{a_ui}(a_ui,me,rn)(apos)
81e         a:A•a ∈ as,
81e         a_ui:A_UI•a_ui=uid_A(a),
81e         me:AMet•me=mereo_A(a),
81e         rn:RegNo•rno=attr_RegNo(a),
81e         apos:APos•apos=attr_APos(a)
81e     }
```

6.5 Space and Time Considerations: A Specific Critique 233

We have not dealt with space and time in a fully satisfactory manner.

6.5.1 Space

We have referred, in Sect. 2, more-or-less explicitly, to **space** in Items 52 [pp. 39], 56 [pp. 40], 56b [pp. 40], 56c [pp. 40], and 56d [pp. 40]. 234
And in Sect. 4 we have also referred to space: 60 Pg. 41, 69b Pg. 42, 72(a)ii and 72(b)i Pg. 42; 72(b)iB and 72(b)iC Pg. 43; 72(b)iiC, 73b and 73d Pg. 43; 75 and 77 Pg. 43. The Sect. 2 references relate to the references of Sect. 4. 235

The problem here is the following: We have not analysed & described the fact that links may be single, double, triple, or more lane links, and hence not whether automobiles may be in identical link positions either moving in different lanes in the same direction; or "piling up" in crashes in the same lane whether "moving" (i.e., being) in the same direction or "moving" in opposite directions; or moving in opposite directions in different lanes. 236

That problem can, of course, be avoided. One can simply augment the analysis & description by introducing appropriate link attributes and appropriate axioms concerning traffic and histories. We leave that the the reader.

6.5.2 Time 237

We have In Sect. 2 referred to **time** in Items 48 Pg. 39, 52 Pg. 39; 54 and 56c Pg. 40. In Sect. 4 we have, correspondingly, also referred to **time** in Items 60 Pg. 41; 65c Pg. 42; 73b Pg. 43 and 73d Pg. 43; 75 Pg. 43 and 77 Pg. 43. 238

It is not the trivial matter of representation of time. One representation of, for example the time this document that you are now reading was compiled, could be May 20, 2018: 11:20 am. Here we have only "refined" the time to within minutes. One could easily represent time "down" to picoseconds ! No, the problem is that of *how often we sample time*. What do the formulas of Items 73b and 73d Pg. 43, and 75 and 77 Pg. 43 express ? *Are they sampled continuously or discretely ?* 239

We shall take the view, here, that the semantics of $\mathtt{RSL}^+$ expresses a discrete sampling, that is, that each iteration of the automobile, the hub and the link behaviours, *take time*, but that the *concurrently behaving automobiles* indeed *may assemble their timed positions simultaneously !* This means that positions recorded for any one particular automobile are all distinct with respect to time, have different time designations.

6.6 The End !

Yes, this is the end of the main example.

6.7 Example Index

6.7.1 Sorts

6.7.2 Types

6.7.3 Functions

6.7.4 Values

6.7.5 Channels

6.7.6 Behaviours

→ **in** {a_r_ch[l_ui,a_ui]|a_ui:A_UI • a_ui∈auis}→ **in,out** {h_l_ch[h_ui,l_ui]|h_ui:H_UI:h_ui∈huis}**Unit**
64, 42

240

Segment II: Space and Time

We have separated out a treatment of the notions of space and time as these are at the very basis of our ability to describe "the world". That is, has deep implications for our attempt to relate the mundane activity of analysing & describing domains to the philosophical issue of *"what can be described"*.

7 Space Time 241

The presentation of the domain analysis & description calculi avoided, in principle, references to space and time; but these concepts are there: "buried" as follows: endurants can be said to "exist" in space and perdurants to "exist" in time. We shall briefly examine these two concepts as they have been the concern of mathematicians. We shall not be interested in the physicists' ***spacetime*** mathematical model that fuses the three dimensions of space and the one dimension of time into a single four-dimensional continuum.

7.1 Space 242

> *Space is the boundless three-dimensional extent in which objects and events have relative position and direction*[30]*. Physical space is often conceived in three linear dimensions, although modern physicists usually consider it, with time, to be part of a boundless four-dimensional continuum known as spacetime. The concept of space is considered to be of fundamental importance to an understanding of the physical universe. However, disagreement continues between philosophers over whether it is itself an entity, a relationship between entities, or part of a conceptual framework*[31]*.*

To us *space* is a conceptual framework. That is, it is not an entity, hence neither an endurant nor a perdurant. Here we shall primarily look at space as a mathematical construction. In Sect. **10** we shall widen that consideration considerably.

7.1.1 Topological Space 243

One notion of space, in mathematics, is that of a Hausdorf (or topological) space:

Definition 25 Topological Space: *A* **topological space** *is an ordered pair (X,τ), where X is a set and τ is a collection of subsets of X, satisfying the following axioms:*[32]

- *The empty set and X itself belong to τ.*

[30] https://www.britannica.com/science/space-physics-and-metaphysics
[31] https://en.wikipedia.org/wiki/Space
[32] Armstrong, M. A. (1983) [1979]. Basic Topology. Undergraduate Texts in Mathematics. Springer. ISBN 0-387-90839-0.

- *Any (finite or infinite) union of members of* τ *still belongs to* τ.
- *The intersection of any finite number of members of* τ *still belongs to* τ ■

The elements of τ *are called* **open sets** *and the collection* τ *is called a* **topology** *on* X.

7.1.2 Metric Space 244

A metric spaces is a set for which distances between all members of the set are defined. Those distances, taken together, are called a metric on the set. A metric on a space induces topological properties like open and closed sets, which lead to the study of more abstract topological spaces.

Definition 26 Metric Space*: A* **metric space** *is an ordered pair* (M, d) *where* M *is a set and* d *is a metric on* M*, i.e., a function*

- $d : M \times M \to \mathbb{R}$

such that for any $x, y, z : M$*, the following holds:*[33]

- *1.* $d(x, y) \geq 0$ *non-negativity or separation axiom*
- *2.* $d(x, y) = 0 \Leftrightarrow x = y$ *identity of indiscernibles*
- *3.* $d(x, y) = d(y, x)$ *symmetry*
- *4.* $d(x, z) \leq d(x, y) + d(y, z)$ *subadditivity or triangle inequality* ■

7.1.3 Euclidian Space 245

The notion of *Euclidian Space* is due to *Euclid of Alexandria* [325–265]. Euclid postulated

Example 29 Euclid's Postulates:

- To draw a straight line from any point to any point.
- To produce [extend] a finite straight line continuously in a straight line.
- To describe a circle with any centre and distance [radius].
- That all right angles are equal to one another.
- [The parallel postulate] That, if a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which are the angles less than the two right angles ■

We refer to Euclidean space. Encyclopedia of Mathematics. URL: http://www.encyclopediaofmath.org/index.php?title=Euclidean_space&oldid=38673 The European Mathematical Society and Springer. 246

[33] B. Choudhary (1992). The Elements of Complex Analysis. New Age International. p.20. ISBN 978-81-224-0399-2.

Example 30 **Euclid's Plane Geometry**: The Euclidean geometry informally described in Example 29 can be formally axiomatised by first introducing the sorts P and L:

```
type
   P, L
value
   [0] obs_Ps: L → P-infset
       parallel: L × L → Bool
```

Observe how the informal axiom in Example 29 has been modelled by the *observer function*
247 obs_Ps. It applies to lines and yields possibly infinite sets of points.

Now we can introduce the axioms proper:

```
axiom
   [1] ∃ p,q:P • p ≠ q,
   [2] ∀ p,q:P • p ≠ q ⇒
             ∃! l:L • p ∈ obs_Ps(l) ∧ q ∈ obs_Ps(l),
   [3] ∀ l:L • ∃ p:P • p ∉ obs_Ps(l),
   [4] ∀ l:L • ∃ p:P • p ∉ obs_Ps(l) ⇒
             ∃ l′:L • l≠l′ ∧ p ∈ obs_Ps(l′) ∧ parallel(l,l′)
```

The concept of being parallel is modelled by the predicate symbol of the same name, by its signature and by axiom [4] ■

We leave it to the reader to reconcile the models of topological space, Defn. 25 [pp. 46], and metric space, Defn. 26 [preceding page], with the axiom systems of examples 29 [previous page] and 30.

7.2 Time 248

`(i) A moving image of eternity;`
`(ii) The number of the movement`
`in respect of the before and the after;`
`(iii) The life of the soul in movement as it passes`
`from one stage of act or experience to another;`
`(iv) A present of things past: memory,`
`a present of things present: sight,`
`and a present of things future: expectations.`

[30, (i) Plato, (ii) Aristotle, (iii) Plotinus, (iv) Augustine].

7.2.1 Time — General Issues 249

In the next sections we shall focus on various models of time, and we shall conclude with a simple view of the operations we shall assume when claiming that an abstract type models time. These sections are far from complete. They are necessary, but, as a general treatment of notions of time, they are not sufficient. We refer the interested reader to special monographs: [34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44].

7.2.2 "A-Series" and "B-Series" Models of Time 250

Colloquially, in ordinary, everyday parlance, we think of time as a dense series of time points. We often illustrate time by a usually horizontal line with an arrow pointing towards the right. Sometimes that line arrowhead is labeled with either a t or the word *time*, or some such name. J.M.E. McTaggart (1908, [36, 35, 44]) discussed theories of time around two notions:

- **"A-series":** has terms like "past", "present" and "future".
- **"B-series":** has terms like "precede", "simultaneous" and "follow".

251

McTaggart argued that the B-series presupposes the A-series: If t precedes t' then there must be a "thing" t'' at which t is past and t' is present. He argued that the A-series is incoherent: What was once 'future', becomes 'present' and then 'past'; and thus events 'will be events', 'are events' and 'were events', that is, will have all three properties.

7.2.3 A Continuum Theory of Time 252

The following is taken from Johan van Benthem [34]: Let P be a point structure (for example, a set). Think of time as a continuum; the following axioms characterise ordering ($<$, $=$, $>$) relations between (i.e., aspects of) time points. The axioms listed below are not thought of as an axiom system, that is, as a set of independent axioms all claimed to hold for the time concept, which we are encircling. Instead van Benthem offers the individual axioms as possible "blocks" from which we can then "build" our own time system — one that suits the application at hand, while also fitting our intuition. 253

Time is transitive: If $p<p'$ and $p'<p''$ then $p<p''$. Time may not loop, that is, is not reflexive: $p \not< p$. Linear time can be defined: Either one time comes before, or is equal to, or comes after another time. Time can be left-linear, i.e., linear "to the left" of a given time.
The following is taken from Johan van Benthem [34]: Let P be a point structure (for example, a set). Think of time as a continuum; the following axioms characterise ordering ($<$, $=$, $>$) relations between (i.e., aspects of) time points. The axioms listed below are not thought of as an axiom system, that is, as a set of independent axioms all claimed to hold for the time concept, which we are encircling. Instead van Benthem offers the individual axioms as possible "blocks" from which we can then "build" our own time system — one that suits the application at hand, while also fitting our intuition. 254

Time is transitive: If $p<p'$ and $p'<p''$ then $p<p''$. Time may not loop, that is, is not reflexive: $p \not< p$. Linear time can be defined: Either one time comes before, or is equal to, or comes after another time. Time can be left-linear, i.e., linear "to the left" of a given time.
One could designate a time axis as beginning at some time, that is, having no predecessor times. And one can designate a time axis as ending at some time, that is, having no successor times. General, past and future successors (predecessors, respectively successors in daily talk) can be defined. Time can be dense: Given any two times one can always find a time between them. Discrete time can be defined. 255

axiom
[TRANS: Transitivity] $\forall$ p,p′,p″:P • $p < p' < p'' \Rightarrow p < p''$
[IRREF: Irreflexivity] $\forall$ p:P • $p \not< p$
[LIN: Linearity] $\forall$ p,p′:P • $(p=p' \vee p<p' \vee p>p')$

[L−LIN: Left Linearity] ∀ p,p′,p″:P • (p′<p ∧ p″<p) ⇒ (p′<p″ ∨ p′=p″ ∨ p″<p′)
[BEG: Beginning] ∃ p:P • ∼∃ p′:P • p′<p
[END: Ending] ∃ p:P • ∼∃ p′:P • p<p′
[SUCC: Successor]
  [PAST: Predecessors] ∀ p:P,∃ p′:P • p′<p
  [FUTURE: Successor] ∀ p:P,∃ p′:P • p<p′
[DENS: Dense] ∀ p,p′:P (p<p′ ⇒ ∃ p″:P • p<p″<p′)
[DENS: Converse Dense] ≡ [TRANS: Transitivity]
[DISC: Discrete]
  ∀ p,p′:P • (p<p′ ⇒ ∃ p″:P • (p<p″ ∧ ∼∃ p‴:P • (p<p‴<p″))) ∧
  ∀ p,p′:P • (p<p′ ⇒ ∃ p″:P • (p″<p′ ∧ ∼∃ p‴:P • (p″<p‴<p′)))

A strict partial order, SPO, is a point structure satisfying TRANS and IRREF. TRANS, IRREF and SUCC imply infinite models. TRANS and SUCC may have finite, "looping time" models.

7.3 Wayne D. Blizard's Theory of Space–Time 256

We now bring space and time together in an axiom system (Wayne D. Blizard, 1980 [45]) which relate abstracted entities to spatial points and time. Let $A, B, \ldots$ stand for entitites, $p, q, \ldots$ for spatial points, and t, τ for times. 0 designates a first, a begin time. Let t' stand for the discrete time successor of time t. Let $N(p,q)$ express that p and q are spatial neighbours. Let $=$ be an overloaded equality operator applicable, pairwise to entities, spatial locations and times, respectively. A_p^t expresses that entity A is at location p at time t. The axioms — where we omit (obvious) typings (of A, B, P, Q, and T): $'$ designates the time successsor
257 function: t'.

(I)	$\forall A \forall t \exists p$	$:$	A_p^t	
(II)	$(A_p^t \wedge A_q^t)$	$\supset$	$p = q$	
(III)	$(A_p^t \wedge B_p^t)$	$\supset$	$A = B$	
$(IV)(?)$	$(A_p^t \wedge A_p^{t'})$	$\supset$	$t = t'$	
$(V\ \ i)$	$\forall p, q$	$:$	$N(p,q) \supset p \neq q$	Irreflexivity
$(V\ \ ii)$	$\forall p, q$	$:$	$N(p,q) = N(q,p)$	Symmetry
258 $(V\ \ iii)$	$\forall p \exists q, r$	$:$	$N(p,q) \wedge N(p,r) \wedge q \neq r$	No isolated locations
$(VI\ \ i)$	$\forall t$	$:$	$t \neq t'$	
$(VI\ \ ii)$	$\forall t$	$:$	$t' \neq 0$	
$(VI\ \ iii)$	$\forall t$	$:$	$t \neq 0 \supset \exists \tau : t = \tau'$	
$(VI\ \ iv)$	$\forall t, \tau$	$:$	$\tau' = t' \supset \tau = t$	
(VII)	$A_p^t \wedge A_q^{t'}$	$\supset$	$N(p,q)$	
$(VIII)$	$A_p^t \wedge B_q^t \wedge N(p,q)$	$\supset$	$\sim (A_q^{t'} \wedge B_p^{t'})$	

We comment on these axioms:

- II–IV,VII–VIII: The axioms are universally 'closed'; that is: We have omitted the usual $\forall A, B, p, q, t$s.
- (I): For every entity, A, and every time, t, there is a location, p, at which A is located at time t.
- (II): An entity cannot be in two locations at the same time.
- (III): Two distinct entities cannot be at the same location at the same time.

- (IV): Entities always move: An entity cannot be at the same location at different times. *This is more like a conjecture: Could be questioned.*
- (V): These three axioms define N.
- (V i): Same as $\forall p :\sim N(p,p)$. "Being a neighbour of", is the same as "being distinct from".
- (V ii): If p is a neighbour of q, then q is a neighbour of p.
- (V iii): Every location has at least two distinct neighbours. 259
- (VI): The next four axioms determine the time successor function $'$.
- (VI i): A time is always distinct from its successor: time cannot rest. There are no time fix points.
- (VI ii): Any time successor is distinct from the begin time. Time 0 has no predecessor.
- (VI iii): Every non–begin time has an immediate predecessor.
- (VI iv): The time successor function $'$ is a one–to–one (i.e., a bijection) function.
- (VII): The *continuous path axiom:* If entity A is at location p at time t, and it is at location q in the immediate next time (t'), then p and q are neighbours.
- (VIII): *No "switching":* If entities A and B occupy neighbouring locations at time t them it is not possible for A and B to have switched locations at the next time (t').

260

Except for Axiom (IV) the system applies both to systems of entities that "sometimes" rests, i.e., do not move. These entities are spatial and occupy at least a point in space. If some entities "occupy more" space volume than others, then we may suitably "repair" the notion of the point space P (etc.). We do not show so here.

Segment III: A Philosophy Basis

261

8 A Task of Philosophy

Philosophy is the study of general and fundamental problems concerning matters such as *existence*, *knowledge*[34], *values*, *reason*, *mind*, and *language*.

8.1 Epistemology 262

We shall focus on *existence*, specifically on *epistemology* – meaning 'knowledge' and 'logical discourse' – it is the branch of philosophy concerned with the theory of knowledge. Episte- 263
mology studies the nature of knowledge, justification, and the rationality of belief. Much of the debate in epistemology centers on four areas: (1) the philosophical analysis of the nature

[34] including Scientific Knowledge: Mathematics, Physics, Computer Science, etc.

of knowledge and how it relates to such concepts as truth, belief, and justification, (2) various problems of skepticism, (3) the sources and scope of knowledge and justified belief, and (4) the criteria for knowledge and justification. Epistemology addresses such questions as "What makes justified beliefs justified?", "What does it mean to say that we know something ?", and fundamentally "How do we know that we know ?"

8.2 Ontology 264

A *"corollary"* of epistemology is *ontology*: the philosophical study of the nature of *being, becoming, existence,* or *reality,* as well as the *basic categories of being and their relations.*

8.3 The Quest 265

The *quest* is now threefold.

(i) First to prepare the ground for a discussion of possible philosophical issues of the domain analysis & description calculi. We do so by a review of philosophy (Pages 53–59) focusing on epistemology and ontology problems – from the ancient Greek philosophers till
266 Bertrand Russell.

(ii) Then to follow that up with a review of the Philosophy of Kai Sørlander as it is, most recently, expressed in [18], and as refined from earlier works: [15, 16, 17]. This is done in Sect. **10**, Pages 59–68.

(iii) Finally to show, issue-by-issue how concepts of the domain analysis & description calculi more have a basis in philosophy than in mathematics and computer science. This is done in Sect. **11**, Pages 69–77.

8.4 Schools of Philosophy 267

We shall only cover Western Philosophy to some depth. A seven line summary will be give, in Sect. **8.4.2**, of a possibly relevant aspect of Indian Philosophy. We'll leave it at that. The fact is that Indian Philosophy has not, it appears, influenced Western Philosophy. That short summary are in line the choice of issues that we seek to uncover.

8.4.1 Western Philosophy 268

Section **9** presents a "capsule" summary of Western Philosophy. It is, at present, a "tour de force", seven pages. One purpose of presenting it is that we are then able to enumerate and date the issues relevant to our quest while discarding some of the proposed theories. Another purpose is to remind the reader of the depth, breadth and plurality of issues of Western Philosophy.

8.4.2 Indian Philosophy 269

Pramana, literally means "proof" and "means of knowledge", refers to epistemology in Indian philosophies, The focus of Pramana is how correct knowledge can be acquired, how one knows, how one doesn't, and to what extent knowledge pertinent about someone or something can be acquired. Ancient and medieval Indian texts identify six pramanas as correct means of accurate knowledge and to truths: (1) perception, (2) inference, (3) comparison and analogy, *(4) postulation, (5) derivation from circumstances, non-perception, negative/cognitive proof, and (6) word, testimony of past or present reliable experts*[35].

9 From Ancient to Kantian Philosophy and Beyond ! 270

The review of this section, i.e., Sect. 9, is based primarily on [15]. It is exclusively "slanted" towards those aspects of the thinking of these philosophers with respect to the *task of philosophy* as we defined it in Sect. 8. In this review we reject the contributions of these great philosophers that is contradictory. This presentational "bias" should in no way stand in way of our general admiration for their otherwise profound thinking.

9.1 Pre-Socrates 271

A number of pre-Socratian thinkers speculated on how the world was "constructed". The earlier thinkers were pre-occupied with *matter*, that is, *substance*; what did the world consist of, how was it constructed ? In doing that these thinkers were trying to be scientists, they were not, in this philosophers. We briefly review some of the pre-Socratian thinkers and 272
philosophers.

Thales of Miletus, 624–546 BC [18, pp 35] *"claimed*[36] *that all existing, i.e., base matter, derived from water"*; **Anaximander of Miletus, 610–546 BC** [18, pp 35-36] *"that base matter all came from apeiron, some further unspecified substance"*; **Anaximenes of Miletus, 585–528 BC** [18, pp 36] *"that base matter was air"*; **Heraklit of Efesos, a. 500 BC** [18, pp 37] *"claimed that* 273
fire was the base matter; and extended the concern from substance to permanence and based the thinking not only on (empirical) observations but also on logical reasoning claiming that everything in the world was in a constant struggle, all the time changing – so since all is changing, i.e., that nothing is stable, he concludes that nothing exists." In that Heraklit was
a philosopher. 274

And, from now, philosophy reigned.

Parmenides of Elea, 501–470 BC [18, pp 37-38, 48-49] *"counterclaimed that that which actually exists is eternal and unchanging – is logically impossible"*; **Zeno of Elea, 490–430 BC** 275
[18, pp 38-39] *"supported Parmenindes' claim by claiming some paradox, i.e., the well-known Achilles and the tortoise – thereby introducing dialectic reasoning and proof by contradiction (reductio ad absurdum)"*; **Demokrit, 460–370 BC** [18, pp 40-42] *"tried to unify Heraklit's concept of changeability and Parmenides' concept of permanence in a new way; everything in the world is built from, consists of atoms and change is due to movement of atoms"*. **The Sophists,** 276
5th Century BC [18, pp 43-44] *"doubted, or even refuted, that we can arrive at universal truths about the world purely through reasoning. They refute that there is an objectively true reality which we can obtain knowledge about. So, instead, skepticism reigned"*. 277

What is interesting, to us, is that, the thinking of even the early Greek thinkers delineates the realms of religion and mythology on one side, and those of science and philosophy, on the other side.

[35] https://en.wikipedia.org/wiki/Pramana
[36] [18, pp 35] refers to Sørlander's book [18] Page 35.

9.2 Plato, Socrates and Aristotle 278

Socrates, 470–399 BC [18, pp 44-45] *"protested against the sophists' refusal of reason, common sense, sanity and prudence"*. We know of Socrates' thinking almost exclusively through

279 **Plato, 427–347 BC**: [18, pp 46-49] *"We shall focus on Plato's* theory of ideas. *His argument is that non-physical (but substantial)* ideas *represent the most accurate reality. Abstract and common concepts obtain meaning through standing for ideas that are eternal and unchangeable.*

280 *In contrast to* ideas *Plato considers the concept of a* phenomenon. Phenomena are instances of ideas. We recognize a phenomenon because it embodies an idea. *So, according to Plato, the changeable world that surrounds us, one which we experience through our senses, is only a reflection of a, or the, real world. That real world is unchangeable and "consists" of ideas"*.[37]

281 **Aristotle, 384–322 BC**. [18, pp 50-53] *"For Aristotle it was not Plato's* abstract ideas *that "existed" but the* concrete world *of which we are a part of with our body. The abstract ideas,*

282 *however, in Aristotle's thinking, constitute a system for describing the world.*[38] *We shall very briefly list two of the concept clusters that Aristotle made to our thinking of the world: (i)* modalities *and (ii)* explanations *– the latter also referred to as* causes*. The* modalities *are: (i.1)* necessity*, that which is unavoidably so; (i.2)* reality*, that which we observe; and (i.3)* possibility*, that which might be. The* causes *(or* explanations*) are: (ii.1)* matter *or* material cause*,*

283 *(ii.2)* form cause *or* formal cause *(ii.3)* agent cause *and (ii.4)* end cause *or* purpose cause *(ii.1) By* material cause *Aristotle means the aspect of the change or movement which is determined by the material that composes the moving or changing things. (ii.2) By* form *or* formal cause *Aristotle means a change or movement's* formal cause*, is a change or movement caused by*

284 *the arrangement, shape or appearance of the thing changing or moving. (ii.3) By* agent cause *Aristotle means a change or movement's efficient or moving cause, consists of things apart from the thing being changed or moved, which interact so as to be an agency of the change or movement. (ii.4) By* end cause *or* purpose cause *Aristotle means a change or movement's*

285 *final cause, is that for the sake of which a thing is what it is. Aristotle's contributions are, for us, decisive. Aristotle reveals how* being *is by revealing the* irreducible types of predicates *which we can actually use when* describing the world*. Aristotle thus examines the* categories: substance *(human, horse),* quantity *(6 feet tall),* quality *(white, red),* relation *(larger, shorter),* location *(in Athens),* time *(yesterday, last year),* position *(lying, sitting),* posture *(wearing shoes),* action *(running, singing), and* suffering *(being cut). This enumeration*[39] *is certainly*

286 *not definitive. Kant, two thousand years later, revives this idea: a* system of unavoidable basic concepts *for the description of the world and our situation in it."*[40]

[37] One may, rather crudely, interpret Plato's concept of ideas with that of types. A value of some type is then a 'phenomenon'.

[38] It should be quite clear, to the reader, that, in this, we follow Aristotle: A main descriptional, in fact, specificational, tool is that of *type definitions*.

[39] *"Of things said without any combination, each signifies either substance or quantity or qualification or a relative or where or when or being-in-a-position or having or doing or being-affected. To give a rough idea, examples of substance are man, horse; of quantity: four-foot, five-foot; of qualification: white, grammatical; of a relative: double, half, larger; of where: in the Lyceum, in the market-place; of when: yesterday, last-year; of being-in-a-position: is-lying, is-sitting; of having: has-shoes-on, has-armour-on; of doing: cutting, burning; of being-affected: being-cut, being-burned."* Ackrill, John (1963). Aristotle, Categories and De Interpretatione. Oxford: At the Clarendon Press. ISBN 0198720866.

[40] It should likewise be obvious to the reader that the notion of *categories* is central to our ontological structuring of domain entities.

9.3 The Stoics: 300 BC–200 AD 287

We shall just focus on one aspect of their contribution to logic and philosophy, that of logic. [22, pp 22-23] *"They distinguish between simple propositions and composite propositions. They also distinguish between three kinds of propositions. implication, conjunction and disjunction. They* 288 *had a special understanding of implication: A proposition is, to the Stoics, of the composite form: $A \Rightarrow B$; A; B. For example:* If it is day then it is light; it is day; therefore it is light. *In this and many other ways they contributed to the philosophy of logic (from which, it seems Gottlob Frege was inspired)"*. **Chrysippus of Soli: 279–206 BC** was a prominent early Stoic. 289

• • •

Almost two thousand years passed before philosophy again flourished. *Christianity*, in Europe, in a sense, "monopolised" critical thinking. With the *Renaissance* and *Martin Luther's Protestantism* thinkers again turned to philosophy.

9.4 The Rational Tradition: Descartes, 290

René Descartes: 1596–1650 [18, pp 72–74] *"rejected the splitting of corporeal substance into matter and form. His main focus was on the relations between mind and form: as thinking substance we recognize material substance"*. **Baruch Spinoza: 1632–1677** [18, pp 74-78] *"rejected Descartes's two substances: there is, he claims, is only one substance; for Spinoza God and nature was one and the same"*. **Gottfried Wilhelm Leibniz: 1646–1716** [18, pp 78-79] *"introduced* 291 *the Law of the Indiscernability of Identicals, It is still in wide use today. It states that if some object x is identical to some object y, then any property that x has, y will have as well"*.[41]

9.5 The Empirical Tradition: Locke, Berkeley and Hume 292

John Locke: 1632–1704. We focus on Locke's ideas of *sensing*. He defines himself[42]:

> as that conscious thinking thing,
> (whatever substance, made up of whether spiritual,
> or material, simple, or compounded, it matters not)
> which is sensible, or conscious of pleasure and pain,
> capable of happiness or misery,
> and so is concerned for itself,
> as far as that consciousness extends.

293

[18, pp 80-82] *"According to Locke, humans obtain their knowledge about the world through sensory perception. At one level, he claims, the world is "mechanical", so our sensory apparatus is influenced mechanically, for example through tactile or visual means. This sense information is then communicated to our brains. First the mechanical sense data become sense ideas, The sense ideas then become reflection ideas."* In the "jargon" of our domain analysis & description method the *sense ideas* are *values* and the *reflection ideas* become *types*. So a central 294

[41] We refer, forward, to Sect. 10.2.1 [pp. 62], and, 'backward', to Sect. 2.6 [pp. 21] [*unique identifiers*], for our "response" to Leibniz's *Law of the Indiscernability of Identicals*.

[42] Locke, John (1997), Woolhouse, Roger, ed., *An Essay Concerning Human Understanding*, New York: Penguin Books

idea in Locke's theory is that all *cognition* builds on our *reflection* over *sense ideas*. In other words: *"Can we conclude anything from our sense ideas to knowledge about those "outer" things which cause the sense ideas ?"* [18, pg. 85] To answer that question Locke goes on to
295 distinguish[43] between *"primary qualities*[44] *and secondary qualities*[45]. In the jargon of domain analysis & description the primary qualities correspond to "our" *external qualities*, the secondary qualities to "our" *internal qualities*, but not quite ! *"Locke views primary qualities as measurable aspects of physical reality and secondary qualities as subjective aspects of physical reality, where "our" domain analysis & description takes both to be somehow measurable. We*
296 *must therefore claim that our distinction is purely pragmatic"*. Locke now claims: *"(i) that we can, with respect to the primary qualities, deduce from our sense ideas to the reality, the world behind these; (ii) that the primary qualities exist in reality independent of whether we "experience" them or not; and (iii) that this is not the case for the secondary qualities which exist*
297 *only in our consciousness"*. **George Berkeley: 1685–1753** [18, pp 82-84] *"points out a problem in Locke's theory: namely that Locke's distinction between primary qualities as being objective and secondary qualities as being subjective does not hold. He argues that primary qualities*
298 *can be subjective. To solve that problem Berkeley denied the existence of a reality "behind" the sense ideas: there is no material reality; reality is our sense ideas: esse est precipi*[46] *! The material reality is there because it is continuously experienced by 'God'. The problem now is can we, at all, determine fundamental characteristics about the world and our situation as hu-*
299 *mans in that world without assuming the concept of independently existing substance"*. **David Hume, 1711–1776**. Hume's major work was *An Enquiry Concerning Human Understanding* [46]. [18, pp 85-87] *"Where Berkeley eliminated material substance Hume also eliminated Berkeley's concepts of 'God' and 'Consciousness'. He claimed that the basic sense-impressions, which to Hume were the basis for all valid human recognition, made it impossible to arrive at a valid recognition of 'God' and a substantial 'I'. They must therefore be eliminated when*
300 *trying to describe the world and our situation in it. According to Hume all that we know are sense impressions and the conceptions derived from these. Hume further distinguishes between composite and simple (not-composite) sense impressions. Correspondingly Hume distinguishes*
301 *between composite and simple (non-composite) ideas. As a consequence there is no necessity in the world, nor in possible relations between cause and effect This renders Hume's thinking in this area very problematic"*.

9.6 Immanuel Kant: 1720–1804 302

[22, pp 280-282] *"Kant was "shaken" by Hume's critique of causality. As a response – along one line of thought – Kant introduced two notions: "Das Ding an sich" is the world that we know, that we sense, and "Das Ding für uns" is a world prior to, outside our cognition. Along another*

[43] https://en.wikipedia.org/wiki/Primary/secondary_quality_distinction

[44] Primary qualities are thought to be properties of objects that are independent of any observer, such as solidity, extension, motion, number and figure. These characteristics convey facts. They exist in the thing itself, can be determined with certainty, and do not rely on subjective judgments. For example, if an object is spherical, no one can reasonably argue that it is triangular.

[45] Secondary qualities are thought to be properties that produce sensations in observers, such as color, taste, smell, and sound. They can be described as the effect things have on certain people. Knowledge that comes from secondary qualities does not provide objective facts about things.

[46] *"to-be-is-to-be-perceived"*

*line of thought Kant claimed that there is our cognition.By means of the cognitive tools with
which our reason is equipped we reach out for "Das Ding an sich" and forms it according to
our cognition. The result is the world as we know it. This means that reality never means the* 303
*"Das Ding an sich", the world "outside" us, "independent" of us. We are excluded from that
world".*

[18, pp 88-92] *"Kant turns the reasoning around. What we empirically observe is determined
by our "reasoning apparatus". We do not observe "things" as they are in themselves ("Das
Ding an sich"), but we "recognize" them as they are formed by our own reasoning apparatus.
This "reasoning apparatus" includes some intuition forms: space and time. These, space and* 304
*time, are therefore, to Kant, not characteristics of the world as it is, but are some intuition
forms that determine our view of the world. How can it now be possible that we can have self-* 305
*awareness on the basis of what we are confronted with – what we see? Here Kant introduces
what he termsthe* **transcendental deduction**. *We can only have self awareness under the
assumption that we experience our views (outlook) as expression of objects, "things", that
exist independent of our experiencing them!"* 306

[18, pp 90-91] *"But Kant's concept of "Das Ding an sich" is inconsistent. It is in contradiction,
because it itself is knowable as being unknowable; and it is in contradiction, because it, in a
mystical sense, is the cause of the thing which we know as a phenomenon, but (we) cannot
apply the cause effect category outside the world of phenomena".* 307

A main contribution of Kant however, is his concept of *Transcendental Schemata*[47]. *"If
pure concepts of the understanding (categories) and sensations are radically different, what
common quality allows them to relate?"* Kant wrote the chapter on Schemata in his *Critique
of Pure Reason* to solve the problem of *"...how we can ensure that categories have 'sense and
significance'"*. Transcendental schema are not related to empirical concepts or to mathemat- 308
ical concepts. These schemata connect pure concepts of the understanding, or categories, to
the phenomenal appearance of objects in general, that is, objects as such, or all objects[48].
Example *categorical schemas* are: The categories of quantity all share the schema of number. 309
The categories of quality all have degrees of reality as their schema. "The schema of the
category of relation is the order of time"[49]. "The schema of the category of modality is time
itself as related to the existence of the object"[50].

9.7 Post-Kant 310

Johann Gottlieb Fichte, 1752–1824 [18, pp 93-94] *"tried to avoid Kant's Das Ding an sich/Das
Ding für uns dualism by letting the subject, the I, determine the object, the not-I, but ends
up in contradiction".* **Georg Wilhelm Friedrich Hegel, 1770–1831** [18, pp 94-97] *"also dissolves* 311
the Kantian dualism. He builds an impressive theory. The basis for this theory is the as-

[47] In Kantian philosophy, a transcendental schema (plural: schemata; from Greek: $\sigma\chi\eta\mu\alpha$, "form, shape, figure") is the procedural rule by which a category or pure, non-empirical concept is associated with a sense impression. A private, subjective intuition is thereby discursively thought to be a representation of an external object. Transcendental schemata are supposedly produced by the imagination in relation to time `https://en.wikipedia.org/wiki/Schema_(Kant)#Transcendental_schemata`.

[48] Körner, S., Kant, Penguin Books, 1990. p. 72

[49] William Henty Stanley Monck, Introduction to the Critical Philosophy. Publ. Dublin, W. McGee, 1874, p.44.

[50] See footnote 49 above.

sumption of a deep-seated identity between reason (sense) and reality: "the reasonable is real" and "the real is reasonable". Hegel saw his understanding of this duality in the light of his-
312 *tory. Hegel thus saw truth, reason and reality historically. "Modern" dialectism was born. Now two contradictory philosophies could now be both true. From this Hegel developed an impressive "apparatus": From "nothingness" via "creation", "quality", quantity" to "essence", "cause", "reality", "causality", and on to "concept", "life" and "cognition" ending with the*
313 *"absolute""*! And there we end! 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
314 up in contradiction. **Friedrich Schelling, 1775–1854**, [18, pp 94] *"goes further by removing the subject/object distinction claiming an underlying identity between these, that is, between mind and matter: nature is the visible mind, and mind is the invisible nature. Again this attempt*
315 *brings Schelling's work into contradictions"*. **Friedrich Ludwig Gottlob Frege, 1848–1925**. Although primarily a mathematician and logician, Frege contributed to Philosophy. Amongst his contributions were the distinction between *"sinn"* (*sense*), and *"bedeutung"* (*reference*). The distinction[51] is: the reference (or "referent"; bedeutung) of a proper name is the object it means or indicates (bedeuten), its sense (Sinn) is what the name expresses. The reference
316 of a sentence is its truth value, its sense is the thought that it expresses. **Edmund Husserl, 1859–1938**, [18, pp 115-116] *"founded a school of phenomenology. To Husserl our conscience is characterised by intentionality. Cognition is an act which is directed at something. When I see, I see something. When I think, I think something. Philosophy, to Husserl, should build on this insight. It should investigate that which conscience is directed at from "within", and without prejudice of what it might be. Husserl expressed clearly the difference between meaning and object"*. But as [15, pp 115-116] shows, Husserl thereby ends up in an inconsistent theory.
317 **Bertrand Russell, 1872–1970**, [18, pp 117-118] *"amongst very many contributions put forward a Philosophy of Logical Atomism [47]. It is based on the formal logic developed Russell and Whitehead in [48, Principia Mathematica]. That formal logic distinguishes between simple and complex propositions; the latter being truth functions over simple propositions. Logical Atomism now claims that the world must be describable by independent simple propositions. This requires that simple empirical propositions must be logically independent of one another.*
318 *This again requires that the meaning of a simple empirical proposition alone must depend on a relation between the simple proposition and that which it stands for in reality. The meaning of a word is that "object" which the word "denotes". This is similar to Wittgenstein's theory.*
319 *The problem is that the requirement that the simple, elementary propositions must be logically independent of one another makes it impossible to find such elementary propositions. It is therefore impossible to find those "objects" that the elementary propositions are supposed to denote. The whole of Logical Atomism thus builds on an erroneous extrapolation from formal*
320 *logic"*. **Logical Positivism: 1920s–1936** was a "circle" if philosophers, mostly based in Vienna, cf. **Wiener Kreis**. [18, pp 119-121] *"They did not adopt Russell's Logical Atomism. Instead they claimed that the meaning of a sentence is its conditions for being true: i.e., a description of all facts that must be the case in order for the sentence to be judged true; that is, the verification conditions. But the problem here is that if the verification conditions are a valid meaning criterion, then its own formulation cannot be meaningful! So logical positivism ends up in*

[51] *On Sense and Reference* [*"Über Sinn und Bedeutung"*], Zeitschrift für Philosophie und philosophische Kritik, vol. 100 (1892), pp. 25–50

contradiction". Some philosophers of the Vienna Circle were **Moritz Schlick, 1882–1936; Alfred Jules Ayer, 1910–1989; Rudolf Carnap, 1891–1970** and **Otto Neurath, 1882–1945**. **Ludwig Wittgenstein, 1889–1951** was not a member of the Vienna Circle, but his early work 321
was much discussed in the Circle. [18, pp 121-124] *"This work of Wittgenstein was Tractatus Logico-Philosophicus [49, 1921]. Tractatus, as did Logical Positivism, basically takes language as a departure point for a philosophical analysis of the world and our situation in it. But both these theories build on self-refusing bases. Wittgenstein understood that his Tractatus was built on a too simple meaning theory, i.e., a theory of how meaning is ascribed to sentences. In Philosophische Untersuchungen* [50] *Wittgenstein explores new directions – which have no bearing on our quest."*

9.8 Bertrand Russell – Again ! 322

We bring an excerpt from Russell's [51, *History of Western Philosophy*, Chap. XXXI: The Philosophy of Logical Analysis, pp 786–788]. The excerpt that we bring reflects Russell's thinking, around 1945, as influenced, no doubt, by developments in quantum physics. *From all this it seems to follow that events, not particles, must be the 'stuff' of physics. What has been thought of as a particle will have to be thought of as a series of events. The series of events that replaces a particle has certain important physical properties, and therefore demands our attention; but it has no more substantiality than any other series of events that we might arbitrarily single out. Thus 'matter' is not part of the ultimate material of the world, but merely a convenient way of collecting events into bundles."* 323

We cannot, but point out, the "similarity" of these observations to our transcendental deduction of behaviours from parts.

• • •

We have surveyed ideas of 32 philosophers – ideas relevant to our quest: that of understanding borderlines between philosophical arguments and formal, mathematical arguments as they relate to domain analysis & description. We shall now turn to elucidate these.

10 The Kai Sørlander Philosophy 324

We shall review an essence of [15, 18]. Kai Sørlander 's objective [18, pp 131] *"is to investigate the philosophical question: 'what are the necessary characteristics of each and every possible world and our situation in it'. We can reformulate this question into the task of determining the necessary logical conditions for every possible description of the world and our situation in it"*.

10.1 The Basis 325

In this section we shall mostly quote from [15]. *"The world is all that is the case. All that can be described in true propositions." "In science we investigate how the world is factually." "Philosophy puts forward another question. We ask of what could not consistently be otherwise."* [52 :1,2,3] **The Inescapable Meaning Assignment**: *"It is thus the task of philos-* 326

[52] [15], : [1] pg. 13, ℓ 2–3, [2] pg. 13, ℓ 7–8, [3] pg. 13, ℓ 11–12

ophy to determine the inescapable characteristics of the world and our situation in it." In determining these inescapable characteristic *"we cannot refer to our experience ... since the experience cannot tell us anything that could not consistently be otherwise." "Two demands must be satisfied by the philosophical basis. The first is that it must not be based on empirical premises. The other is that it cannot consistently be refuted by anybody under any conceivable*
327 *circumstances. These demands can only be satisfied by one assumption."* We shall refer to this assumption as:

The Inescapable Meaning Assignment

- The *The Inescapable Meaning Assignment* is[53] the recognition of the mutual dependency between
  - *the meaning of designations and*
  - *the consistency relations between propositions.*

As an example of what "goes into" *the inescapable meaning assignment* we bring, albeit from the world of computer science, that of the description of the *stack* data type (its entities and
328 operations).

The Meaning of Designations

Stacks - A Narrative

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

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

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

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

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

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

The consistency relations: 329

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

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

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

The meaning of designations: 330

[53] [15], pg. 13-14, ℓ13-ℓ1

type
82. E, S
value
83. empty_S: **Unit** → S
84. is_empty_S: S → **Bool**
85. stack: E × S → S
86. unstack: S $\tilde{\rightarrow}$ S
87. top: S $\tilde{\rightarrow}$ E

The consistency relations:

88. is_empty(empty_S()) = **true**
88. is_empty(stack(e,s)) = **false**
89. unstack(stack(e,s)) = s
90. top(stack(e,s)) = e

331
Necessary and Empirical Propositions: *"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." "**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."* The definition *"the world is all that is the case. All that can be described in true* 332
propositions." [54 :1,2,3,4,5] 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."* [55] **Primary Objects**: *"an empirical* 333
*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."* [56] **Two Re-** 334
quirements 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."* [57] **The inescapable meaning
assignment:** *'the meaning of designations and the consistency relations between propositions'* [58]
... satisfies this basis.[59] **The Possibility of Truth**: Where Kant builds on the *contradictory* 335
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*: [18, pp 136] *"since the possibility of truth cannot
in a consistent manner be denied we can hence assume the* **contradiction principle:** *'a propo-
sition and its negation cannot both be true'. We assume that the contradiction principle is a
necessary truth*[60]*"* **The Logical Connectives**: Sørlander now deduces the logical connectives: 336

[54] [15], : [1] pg. 13, ℓ 16–17; [2] pg. 13, ℓ 17–18; [3] pg. 13, ℓ 20–21; [4] pg. 14, ℓ 26–30; [5] pg. 13, ℓ 2–3
[55] [15], pg.15, ℓ15-18
[56] [15], pg.15, ℓ23-30
[57] [15], pg. 30, ℓ 6–12
[58] [15], pg. 13-14, ℓ13-ℓ1
[59] [15], pg. 30, ℓ 16–28
[60] [18, pp 136] *"A necessary truth, on one side, follows from the meaning of the designations by means of which it is expressed, and, on the other side and at the same instance, define these designations and their mutual meaning."*

337 *conjunction* ('and' ∧), *disjunction* ('or', ∨), and *implication* (⇒ or ⊃). **Necessity and Possibility**: [18, pp 142] *"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.*
338 *A proposition is possibly true, if its negation is not necessarily true"*. **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. The entities that are referenced in empirical propositions have not been completely characterised by these propositions; they are simply those that can be referenced in empirical propositions.

10.2 Logical Conditions for Describing Physical Worlds 339

So which are the logical conditions of descriptions of any world ? In [15] and [18] Kai Sørlander , through a series of transcendental deductions "unravels" the following logical conditions: (i) symmetry and asymmetry (ii) transitivity and intransitivity, (iii) space: direction, distance, etc., (iv) time: before, after, in-between etc., (v) states and causality, (vi) kinematics, dy-
340 namics, etc., and (vii) 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.

10.2.1 Symmetry and Asymmetry 341

[18, pp 152] *"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"*.

10.2.2 Transitivity and Intransitivity 342

[18, pp 148] *"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"*.

10.2.3 Space 343

[18, pp 154] *"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. Direction and distance can be understood as spatial relations. From these relations are derived the relation in-between. 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*.

10.2.4 States 344

[18, pp 158-159] *"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. For it to be logically possible, that one-and-the-same primary entity can* *be ascribed incompatible predicates, is only logically possible if any primary entity can exist in different states. 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".*

10.2.5 Time 345

[18, pp 159] *"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".*

10.2.6 Causality 346

[18, pp 162-163] *"States are related by the time relations "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,* 347
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 348
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 349
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.* 350
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.

10.2.7 Kinematics 351

[18, pp 164–165] *"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 in movement. If a primary entity which does not change location is said to be at rest. The velocity*[61] *of a* 352 *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 we have reasoned us to fundamental concepts of *kinematics*.

10.2.8 Dynamics 353

[18, pp 165-165] *"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* be moving*. Likewise when it goes from movement to rest. And similarly, when it accelerates (decelerates). So a primary entity has same* state of movement *if it has same velocity and moves in the same direction. Primary entities change*
354 *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*".*

10.2.9 Newton's Laws 355

Newton's First Law: [18, pp 165-166] *"Combining* kinematics *and the* principle of causality*, and the therefrom deduced concept of* force*, we can deduce that any* change of movement *is proportional*[62] *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 conduc-*
356 *tion a linear movement at constant velocity. This is* Newton's First Law*".* **Newton's Second Law:** [18, pp 166] *"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*.*[63] *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*[64] *to the*
357 *mass of that entity. This is* Newton's Second Law*".* **Newton's Third Law:** [18, pp 166-167] *"In a possible world, the forces that affects 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*".*

[61] Velocity has a *speed* and a *vectorial direction*. *Speed* is a scalar, for example of type `kilometers per hour`. *Vectorial direction* is a scalar structure, for example for a spatial direction consisting of geographical elements: x `degrees North,` y `degrees East` $(x+y=90)$, `and` z `degrees Up or Down` $(0 \le z \le 90$, `where, if` $z=90$ `we have that both` x `and` y `are` $0)$.

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

[63] *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*.

[64] Cf. Footnote 62.

10.3 Gravitation and Quantum Mechanics 358

Mutual Attraction: [18, pp 167-168] *"How can primary entities possibly be the* source *of* forces
that influence *one another ? How can primary entities at all have a* mass[65] *such that it*
requires forces *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:** 359
[18, pp 168] *"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** 360
Propagation – A Gravitational Constant: [18, pp 168] *"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.*"*[66]
Gravitational "Pull": [18, pp 169-170] *"The nature of* gravitational "pull" *can be deduced, basically* 361
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
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 362
Einstein's special relativity has been done; unification of gravitation with Einstein's general
theory of relativity is still to be done. **A Summary:** [18, pp 170-173] *"Philosophy lends to physics* 363
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) "

10.4 The Logical Conditions for Describing Living Species 364

10.4.1 Purpose, Life and Evolution

Causality of Purpose: [18, pp 174] *"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*"* **Living Species:** [18, pp 174-175] *"These primary entities are here called* living species. 365
What can be deduced about them ? They must have some form they can be developed to reach;
and 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

[65] cf. Footnote 63 Pg. 64

[66] Let two entities have respective masses m_1 and m_2. Let the forces with which they attract each other be f_1, respectively f_2. Then the *law of gravitation* – *as it can be deduced by philosophical arguments* – can be expressed as $f_1 = f_2$. The specific force, expressed using Newton's constant G is $f = G \times m_1 \times m_2 \times r^{-2}$ where r is the distance between the two entities and $G = 6.674 \times 10^{-11} \times m^3 \times kg^{-1} \times s^{-2}$ [m:meter, kg:kilogam s:second] – *as derived by physicists.*

and exchange, and another form which, additionally, can be characterised by the ability to
366 *purposeful movement. The first we call plants, the second we call animals"* **Animate Entities:** [18, pp 176] *"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: they must have sensory organs sensing among others the immediate purpose of its movement; they must have means of motion so that it can move; and they must have instincts, incentives and feelings as causal conditions that what it senses can drive it to movements"* And all of this in accordance
367 with the laws of physics. **Animal Structure:** [18, pp 177-178] *"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 genomes; that is, animals are built from genomes which give them the inner determination to such building blocks for instincts, incentives and feelings. Similar kinds of deduction can be carried out with respect to plants. Transcendentally one can deduce basic principles of evolution but not its details"*

10.4.2 Consciousness, Learning and Language 368

Consciousness and Learning: [18, pp 180-181] *"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 and can learn. ... One can therefore deduce that animals must possess such building blocks whose inner*
369 *determination is a basis for learning and consciousness "* **Language:** [18, pp 181-182] *"Animals with higher social interaction uses signs, 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"*

10.5 Humans, Knowledge, Responsibility 370

Humans: [18, pp 184] *"A human is an animal which has a language"* **Knowledge:** [18, pp 184] *"Humans must be conscious of having knowledge of its concrete situation, and as such that humans can have knowledge about what they feel, and eventually that humans can know whether what they*
371 *feel is true or false. Consequently humans can describe their situation correctly"* **Responsibility:** [18, pp 184] *"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"*

10.6 An Augmented Upper Ontology 372

We now augment our upper-ontology, to include *living species*, from that of Fig. 1 Pg. 15 to
373 that of Fig. 6 Pg. 67. We leave it to the reader to "fill in the details !"

10.7 Artifacts: Man-made Entities 374

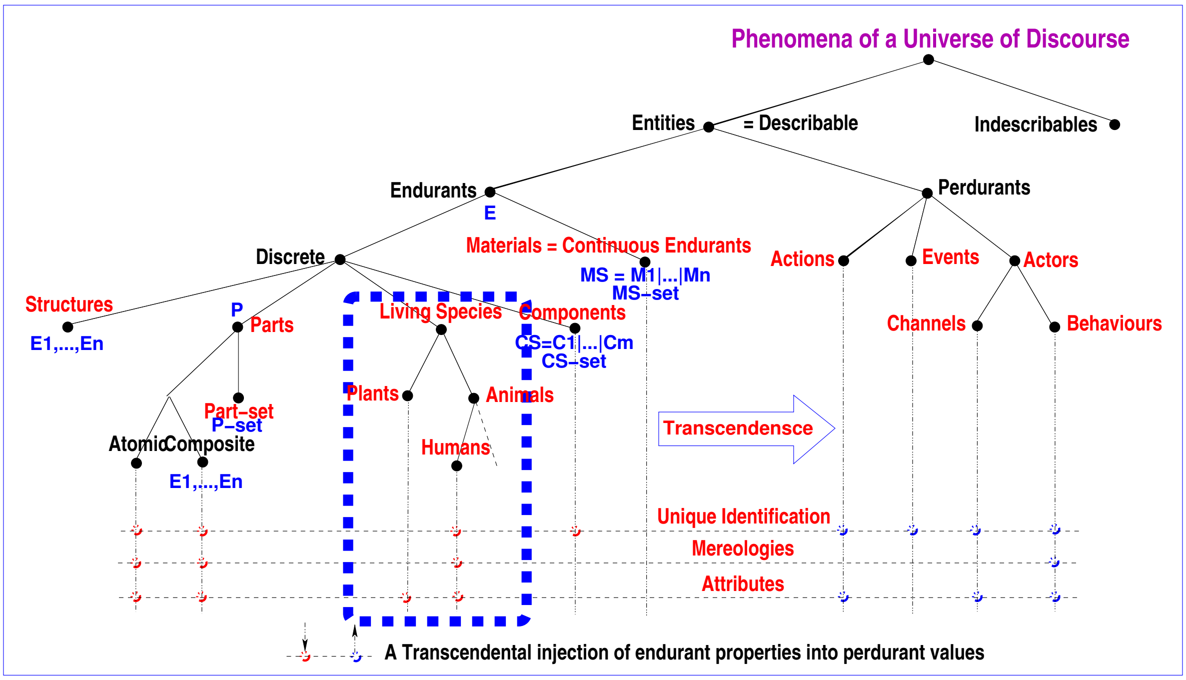


Figure 6: An Upper Ontology for Domains – with **Living Species**

Definition 27 Artifact: *By an* artifact *we shall understand a* man-made entity: *usually an* endurant *in* space, *one that satisfies the laws of physics, and sometimes one that, by a* transcendental deduction, *can take on the rôle of a* perdurant; *but the artifact can also, for example, by* intended *as a* piece of art, *something for our enjoyment and reflection.*

375

We then augment our upper-ontology, to include *artifacts*, from that of Fig. 6 Pg. 67 to that of Fig. 7 Pg. 68. We leave it to the reader to "fill in the details!" 376

10.8 Intentionality 377

We have ended our presentation of Sørlander's Philosophy. Before going into justifications of our *domain analysis & description calculi* with respect to this philosophy we shall briefly comment on the concept of *intentionality*. 378

Intentionality is a philosophical concept and is defined by the `Stanford Encyclopedia of Philosophy`[67] *as "the power of minds to be about, to represent, or to stand for, things, properties and states of affairs." The puzzles of intentionality lie at the interface between the philosophy of mind and the philosophy of language. The word itself, which is of medieval Scholastic origin, was rehabilitated by the philosopher Franz Brentano towards the end of the nineteenth century. and adopted by Edmund Husserl. 'Intentionality' is a philosopher's word. It derives* 379

[67] Jacob, P. (Aug 31, 2010). *Intentionality*. `Stanford Encyclopedia of Philosophy` (`https://seop.illc.uva.nl/entries/intentionality/`) October 15, 2014, retrieved April 3, 2018.

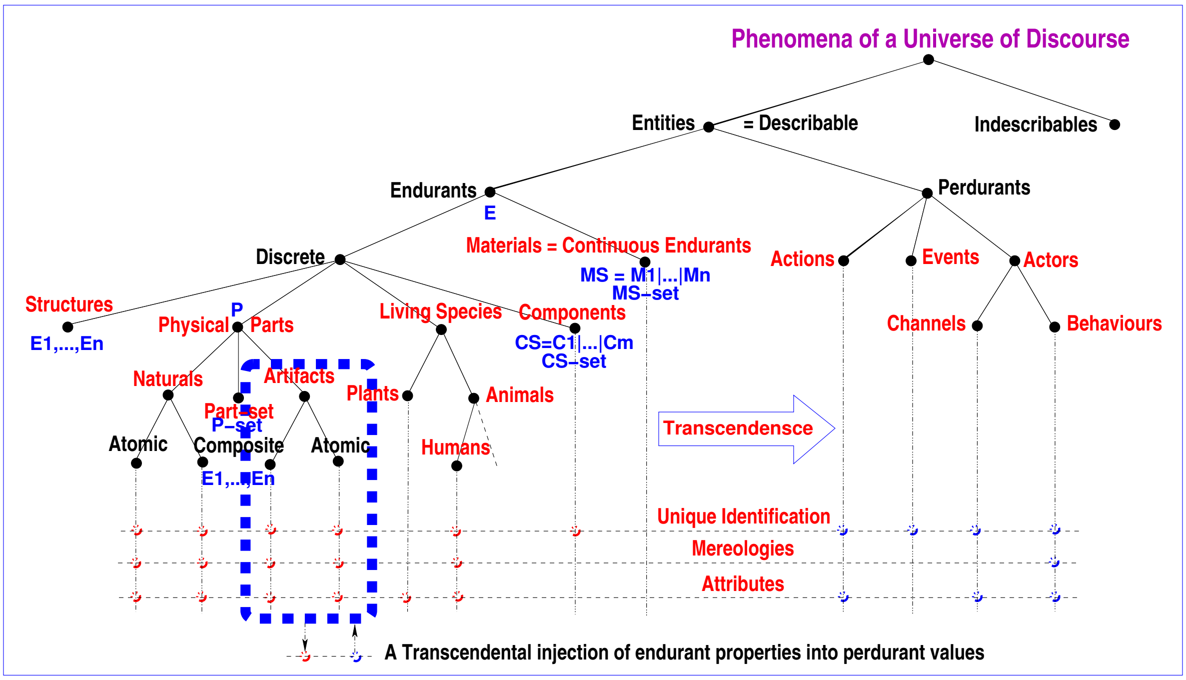


Figure 7: An Upper Ontology Extended with **Artifacts**

from the Latin word intentio*, which in turn derives from the verb intendere, which means being directed towards some goal or thing. The earliest theory of intentionality is associated with St. Anselm's ontological argument for the existence of God, and with his tenets distinguishing*
380 *between objects that exist in the understanding and objects that exist in reality.*

We shall here endow the concept of 'intentionality' with the following interpretation. Man-made artifacts are made for specific purposes. Often two or more artifacts are intended to serve a purpose, that is, to represent an intent. We speculate as follows:

Definition 28 On Intentional Pull*:* *Two or more artifactual parts of different sorts, but with overlapping sets of intents may excert an intentional "pull" on one another* ■

381

This *intentional "pull"* may take many forms. Let $p_x{:}X$ and $p_y{:}Y$ be two parts of *different sorts* (X, Y), and with *common intent*, ι. *Manifestations* of these, their common intent must somehow be *subject to constraints*, and these must be *expressed predicatively*.

We return, in Sect. **11.1.4** [pp. 73], with an *example of* what we claim to be an *intentional "pull"*, that is, Example 34 [pp. 73].

Segment IV: Fusing Philosophy into Computer Science

382

11 Philosophical Issues of The Domain Calculi 383

We now interpret the *domain analysis & description analysis calculus* of Segment **I** in the light of Sørlander's *Philosophy* of Sect. **10**.

We re-examine all analysis calculus prompts with references to their prompt number or the section – and the page on which their definition is given.

11.1 The Analysis Calculus Prompts 384

11.1.1 External Qualities

- Item 1, pp. 12: `is_universe_of_discourse:` After a rough sketch narrative of the contemplated domain, the informal justification to be given for this query should be along these lines: the chosen universe-of-discourse is one that can be described in true propositions; that is, one that is based in *space* and *time*; subject to *Laws of Newton*; etc., and, indispensably so, involves *persons* with *language*, *responsibility* and *intents*. 385

- Item 2, pp. 13: `is_entity:` So entities are just that: describable, based in either space (as are endurants) or in both space and time (as are perdurants), and involving persons. That is, entities are the "stuff" that philosophy cares about in its quest to understand the world. What lies outside may be in the realm of superstition, "mumbo-jumbo", et cetera; "things" that are neither in space nor time; figments of the mind. 386

- Item 3, pp. 13: `is_endurant:` An endurant is an entity which we characterise in propositions without reference to (actual, i.e. "real") time. There is no notion of state changes in describing entities. Endurants are either based in physics or based in living species: plants and animals including persons, or are artifacts which build on endurants. Endurants are, in the words of Whitehead, [52], *continuants*. 387

- Item 4, pp. 13: `is_perdurant:` And, consequently, a perdurant is an entity which we characterise in propositions with more-or-less explicit reference to (actual, i.e. "real") time, focusing on state-changes and/or interaction between perdurants. Perdurants are either *action*s or *event*s or *behaviour*s. **Definition**: *Behaviours* are defined as sets of sequences of actions, events and behaviours ■ Philosophical treatments are given of 388
the notions of *time* in [53, 35, 45, 34], [discrete] *action*s in [33], *event*s in [54, 55, 56, 57, 58, 59, 60, 61, 62, 63], and *behaviour*s in, for example, the Internet based articles on `plato.stanford.edu/entries/behaviorism/` and `www.behavior.org/search.php?-q=behavior+and+philosophy`. Most of the literature on behaviours focus on psychological aspects which we consider outside the realm of our form of domain analysis & description,

  The interplay between endurants and perdurants is studied in [64]. 389

- Item 5, pp. 14: `is_discrete:` [We re-emphasize that the notion of *discreteness* of *endurants* such as we "need" it here, is not related to the notion of *discreteness* in physics or mathematics.] The terms *separate, individual* and *distinct* characterise *discreteness*.

It is up to the *domain analysis & description scientist cum engineer* to decide whether en
390 entity should be characterised as primarily distinguished by these ‘qualities’ – or not.

- Item 6, pp. 14: `is_continuous:` [We re-emphasize that the notion of *continuity* of *endurants* such as we “need” it here, is not related to the notion of *continuity* in physics or mathematics.] The terms: *prolonged*, *without interruption*, and *unbroken series or pattern* characterise *continuity* of endurants. It is up to the *domain analysis & description scientist cum engineer* to decide whether en entity should be characterised as primarily
391 distinguished by these ‘qualities’, or not.
- Item 7, pp. 15: `is_structure:` Whether a discrete endurant is considered a *structure*, or a *part*, or *a set of components* is a *pragmatic* decision. So has no bearings in the Sørlander Philosophy outside its possible bearings in language where the notion of language can
392 be motivated philosophically.
- Item 8, pp. 16: `is_part`, Item 14, pp. 19: `is_component` and Item 16, pp. 20: `is_material:` All entities, whether non-living species, including artifactual, or living species (plants and animals, incl. humans) are subject to *the inescapable meaning assignment*, the *principle of contradiction* and its *implicit meaning theory*. They are also subject
393 to the notions of *space* and *time* and to the *Laws of Newton*, etc. The living species entities are *additionally* subject to *causality of purpose* with humans having *language*, *memory* and *responsibility*. These notions can be assumed, but we do not, at present, i.e., in this report, suggest any means of modelling language, memory and responsibil-
394 ity. Following Sørlander’s Philosophy there are the (atomic, see below) part p living species: **is**_LIVE_SPECIES(p), of which there are plants, **is**_PLANT(p), and there are animals, **is**_ANIMAL(p), of which (latter) some are humans, **is**_HUMAN(p), and some are not; and there are the non-living-species parts, p, of which some are made by man (or by other artifacts), **is**_ARTIFACT(p), and some are not, we refer to them as *physical part*s. We therefore now, as a consequence of Sørlander’s Philosophy, suggest the domain analysis
395 prompts: **is**_LIVE_SPECIES, **is**_PLANT, **is**_ANIMAL, **is**_HUMAN and **is**_ARTIFACT.

All this means that the Sørlander Philosophy, in a sense, mandates us to introduce the following *new analysis prompts:*

Analysis Prompt 28 *`is_physical:` The domain analyser analyses discrete endurants (d) into physical parts:*

- *`is_physical` – where `is_physical(d)` holds if d is a physical part* ▪

Analysis Prompt 29 *`is_living:` The domain analyser analyses discrete endurants (d) into living species:*

- *`is_living` – where `is_living(d)` holds if θ is a living species.* ▪

396

Analysis Prompt 30 *`is_natural:` The domain analyser analyses physical parts (p) into natural:*

- ⊗ *is_natural* – *where* **is_natural**(p) *holds if p is a natural part* ■

Analysis Prompt 31 *is_artifactual: The domain analyser analyses physical parts (p) into artifactual physical parts:*

- ⊗ *is_artifactual* – *where* **is_artifactual**(p) *holds if p is a man-made part* ■ 397

Analysis Prompt 32 *is_plant: The domain analyser analyses living species (ℓ) into plants:*

- ⊗ *is_plant* – *where* **is_plant**(ℓ) *holds if ℓ is a plant* ■

Analysis Prompt 33 *is_animal: The domain analyser analyses living species (ℓ) into animals:*

- ⊗ *is_animal* – *where* **is_animal**(ℓ) *holds if ℓ is an animal* ■ 398

Analysis Prompt 34 *is_human: The domain analyser analyses animals (α) into humans:*

- ⊗ *is_human* – *where* **is_human**(α) *holds if α is a human* ■

Analysis prompts, is_XXX, similar to is_human, can be devised for other animal species. 399

- Item 9, pp. 16: is_atomic: and Item 10, pp. 16: is_composite: The notion of atomicity here has nothing to do with that of the the Greeks [Demokrit, pp. 53]. Here it is a rather pragmatic issue, void, it seems, of philosophical challenge. It is a purely pragmatic issue with respect to any chose domain whether the domain scientist cum engineer decides to analyse & describe a part into being atomic or composite.

  Example 31 Automobile: Atomic or Composite: Thus, *for example*, you the reader may consider your automobile as atomic, whereas your mechanic undoubtedly considers it composite ■

11.1.2 Unique Identifiers 400

Sect. 2.6, pp. 21–22: unique identifiers:

Uniqueness of entities follows from the basic logic of symmetry etc. Uniqueness or rather *identity*, is an thus important philosophical notion [cf. Sect. 10.2.1 [pp. 62]]. Notice that we are not concerned with any representation of unique part and component identifiers. So please, dear reader, do not speculate on that! The uniqueness of part or component identifiers "follows" the part and component, irrespective of the spatial location and time of the possibly "movable" part or component, i.e., irrespective of its state!

11.1.3 Mereology 401

Sect. 2.7, pp. 22–23: `mereology:`

There are some new aspects of the concept of mereology – which, in light of the Sørlander Philosophy, were not considered in Sect. 2.7, and which it is now high time to consider, and,
402 for some of these aspects, *to include in the domain analysis & description method.*

- **Philosophy:** Mereology, such as we use it, derives from *Stanisław Leśniewski*, Polish mathematician, logician, philosopher (1886–1939) [65, 66, 67, 68, 69, 70]. Wikipedia presents an overview of aspects of mereology.[68]. Related to our "use" of the concept of mereology are the studies of Henry S. Leonard and Nelson Goodman [71, 72, 73, 1940–2008], Bowman L. Clarke [74, 75, 1981–1985], Douglass T. Ross [76, 1976], Mario Bunge [77, 78, 1977–1979], Peter Simons [79, 1987], Barry Smith [80, 81, 82, 83, 84, 85, 1993–
403 2004] and Roberto Casati and Achille C. Varzi [86, 87, 26, 1993–1999].

- **Topologies and Intents:** To us mereology, in light of Sørlander's Philosophy, now becomes either of two relations (or possibly both): (i) spatial relations, as for *Stanisław Leśniewski* and the cited references, and (ii) *intensional* relations. We characterise the latter as follows:

  Definition 29 Intentional Relations: *By an intensional relation we shall understand a relation between distinct endurants which manifests two (or more) designations and at least one meaning* ■

404

  Example 32 Transport: Automobiles and roads, i.e. hubs and links, have distinct sorts and designations, but share the *intent* (*meaning*) of technologically *supporting traffic* ■

405 We refer to [5, *Domain Facets: Analysis & Description*].

- **Part Mereologies:** Thus the mereology of parts shall be sought in either their topological, i.e., spatial, arrangements, or their intents – with parts of same intent being mereologically related, or possibly some combination of both.

  Example 33 Traffic: Hence, in reference to the example of Sect. 6, we have that the mereologies of each automobile include the set of unique identifiers of all hubs and links, and the mereologies of each hub and link include the set of unique identifiers of all automobiles ■

406

- **Further Studies:** It appears that the concept of mereology, in light of Sørlander's Philosophy, warrants further scrutiny, philosophically well as from the point of view of domain analysis & description method. Should discrete endurants be further analysed into structures, parts and components, as now, and *natural discrete endurants* or *artifact discrete endurants* or should discrete endurants have attribute values of *natural discrete endurant values* or *artifact discrete endurant values*.

[68] https://en.wikipedia.org/wiki/Mereology#Metaphysics

11.1.4 Attributes 407

Sect. 2.8, pp. 23–27: `attributes:`

Attributes, their type and value, are the main means for *expressing propositions about primary entities.*[69] Let us first recall: *parts* and *components* have **unique identifiers**, *parts* have **mereologies** and *parts* and *materials* have **attributes.** Let us also "remember" that these differences are purely pragmatic. All endurants are subject to being in *space* and *time*, and 408
being subject to the *principle of causality*. Three sets of attributes follow from the Sørlander's Philosophy: (i) attributes of non-life-specifies entities; (ii) attributes of life-specifies entities, but additionally subject to *purpose*, *language*, *responsibility*, and *causality of principle*; and those (iii) attributes that are additional and more individually determined by the kind of the part. We shall now summarise these.

Non-Species Parts: These are the parts that were actually treated in Sect. 2. To them, 409
as a consequence of Sørlander's Philosophy, one can ascribe the following attribute observers: **attr**_$\mathbb{SPACE}$ and **attr**_$\mathbb{TIME}$. No explanation seems necessary here. Attribute observers related to the above could be: **attr**_$\mathbb{LOCATION}$ where the *location* to be yielded is some spatial point within the space yielded by the $\mathbb{SPACE}$ observer. **attr**_$\mathbb{VOLUME}$ where the *volume* is the volume (in some units) of the space yielded by the $\mathbb{SPACE}$ observer. **attr**_$\mathbb{MASS}(p)$ where the *mass* is the mass (in some units) of the part p. Et cetera. We leave it to the reader to "think up" Boolean and other algebraic operators over time, space, location, mass, etc.

Artifacts: To remind, *artifacts* are parts made by man and/or other artifacts. They 410
have all the same attributes (i.e. attribute observers) as has non-species parts. In addition they may have such attribute observes as **attr_Intent**, **attr**_Maker, **attr**_Brand_Name, **attr**_Production_Year, **attr**_Owner, **attr**_Purchase_Price, **attr**_Current_Value and **attr**_Condition. The idea of the **attr_Intent** attribute observer is to yield a token that somehow identifies the *purpose* of the artifact: `transport, "measurement-of-this", "measurement-of-that", "food-stuff",` etc. We leave it to the reader to figure out the idea of the other attributes.

Artifactual Intents: In the world of physics, since Isaac Newton, the mutual attraction of bod- 411
ies (with mass) and in the context of gravitation leads to the **gravitational pull**, cf. Sect. 10.3 pp. 65. Now, in the context of artifactual parts with intents we may speak of **intentional "pull"**.

Definition 30 Intentional Pull*: Two or more artifactual parts of different sorts, but with overlapping sets of intents may excert an* intentional "pull" *on one another* ■

412

This *intentional "pull"* may take many forms. Let $p_x : X$ and $p_y : Y$ be two parts of *different sorts* (X, Y), and with *common intent*, ι. *Manifestations* of these, their common intent must somehow be *subject to constraints*, and these must be *expressed predicatively*. 413

Example 34 Automobile and Road Transport: For the main example, Sect. 6,

91 *automobiles* shall now include the intent of `'transport'`,

92 and so shall *hubs* and *links*.

91 **attr**_Intent: A → (`'transport'`|...)-**set**

[69] *The world is all that is the case. All that can be described in true propositions.* [15, pp.13, ℓ 2–3]

92 **attr**_Intent: H → ('transport'|...)-**set**
92 **attr**_Intent: L → ('transport'|...)-**set**

Manifestations of 'transport' is reflected in *automobiles* having the automobile position attribute, APos, Item 56 Pg. 40, *hubs* having the *hub traffic* attribute, H_Traffic, Item 48 Pg. 39,
414 and in *links* having the *link traffic* attribute, L_Traffic, Item 52 Pg. 39.

93 Seen from the point of view of an automobile there is its own traffic history, A_Hist Item 56c Pg. 40, which is a (time ordered) sequence of timed automobile's positions;

94 seen from the point of view of a hub there is its own traffic history, H_Traffic Item 48 Pg. 39, which is a (time ordered) sequence of timed maps from automobile identities into automobile positions; and

95 seen from the point of view of a link there is its own traffic history, L_Traffic Item 52 Pg. 39, which is a (time ordered) sequence of timed maps from automobile identities into automobile positions.

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

96 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.

415

type
56c, pp.40 A_Hist = $(\mathcal{T} \times \text{APos})^*$
48, pp.39 H_Traffic = A_UI $\overrightarrow{m}$ $(\mathcal{T} \times \text{APos})^*$
52, pp.39 L_Traffic = A_UI $\overrightarrow{m}$ $(\mathcal{T} \times \text{APos})^*$
96 AllATH = $\mathcal{T}$ $\overrightarrow{m}$ (AUI $\overrightarrow{m}$ APos)
96 AllHTH = $\mathcal{T}$ $\overrightarrow{m}$ (AUI $\overrightarrow{m}$ APos)
96 AllLTH = $\mathcal{T}$ $\overrightarrow{m}$ (AUI $\overrightarrow{m}$ APos)
axiom
96 **let** allA = proper_merge_into_AllATH({(a,attr_A_Hist(a))|a:A•a ∈ as}),
96 allH = proper_merge_into_AllHTH({attr_H_Traffic(h)|h:H•h ∈ hs}),
96 allL = proper_merge_into_AllLTH({attr_L_Traffic(l)|l:L•h ∈ ls}) **in**
96 allA = H_and_L_Traffic_merge(allH,allL) **end**

416 We leave the definition of the four merge functions to the reader!

We now discuss the concept of *intentional "pull"*. We endow each automobile with its history of timed positions and each hub and link with 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[70]. 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! ■

417 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.

Humans[71]**:** *Humans* have *sensory organs* and *means of motion*; *inner determination* for 418
instincts, *incentives* and *feelings*; *purpose*; and *language*; and can *learn*[72]. We leave it, to the reader, as a *research topic*: to suggest means for expressing analysis prompts that cover these kinds of attributes. 419

For this report we have little to say on the issue of *humans*. Rather much more work has to be done for any meaningful writing. So, here is a challenge to the readers!

11.1.5 A Summary of Domain Analysis Prompts 420

11.2 The Description Calculus Prompts 421

MORE TO COME

- Item 1, pp. 12: `observe_universe_of_discourse:`
- Item 2, pp. 17: `observe_endurant_sorts:`
- Item 3, pp. 18: `observe_part_type:`
- Item 4, pp. 19: `observe_component_sorts:`
- Item 5, pp. 20: `observe_material_sorts:`
- Item 6, pp. 21: `observe_unique_identifier:`

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

[71] We focus on humans, but the discussion can be "repeated", in modified form, for plants and animals in general.

[72] cf. Sect. 10.4.2 [pp. 66]

- Item 7, pp. 22: `observe_mereology:`
- Item 8, pp. 24: `observe_attributes:`

MORE TO COME

11.2.1 A Summary of Domain Description Prompts 422

MORE TO COME

[1] `observe_ universe_ of_ discourse`, 12
[2] `observe_ endurant_ sorts`, 17
[3] `observe_ part_ type`, 18
[4] `observe_ component_ sorts_ P`, 19
[5] `observe_ material_ sorts_ P`, 20
[6] `observe_ unique_ identifier`, 21
[7] `observe_ mereology`, 22
[8] `observe_ attributes`, 24

MORE TO COME

11.3 The Behaviour Schemata 423

TO BE WRITTEN

11.4 Wrapping Up 424

We summarise the above in a revision of the *ontology diagram* first given in Fig. 1 Pg. 15 and used, in more-or-less that form, in several publications: [1, 4, 7, 88]. The revision is shown in
425 Fig. 8:
426 Figure 8 emphasis the analytic, "upper" structure of domains and emphasises endurants: **Black** names attached to diagram nodes designate "upper" categories of entities. **Red** names similarly attached designate manifest categories of entities. **Blue** names also so attached are the sort names of values of manifest endurants. Both naturals and artifacts have atomic and composite values. We only hint (⋱) at other (than human) animal species. The lower dashed horizontal lines with pairs of **-o- - -o-** hint at the internal endurant qualities that are "transferred"

11.5 Discussion 427

11.5.1 Review of Revisions

We have related a number of the domain analysis & description method's analysis prompts to Sørlander's Philosophy – and have found that a number of corrections has to be made to the understanding of these: the basis for *unique identifiers* and the categories of endurants and
428 attributes. With [1] endurants came in three forms: *structures*, *parts* (atomic and composite), and *materials*. Now we must *refine* the notion of parts into: *physical parts* (as assumed in [1]),
429 *artifactual parts* and *living species parts*. We must further articulate the notion of attributes: as before, for *physical parts*, to necessarily include the in-avoidable classical physics attributes[73] and be subject to the *principle of causality* and *gravitational pull*; but now additionally also to *artifactual parts*, still subject to the attributes of physical parts but now additionally subject to

[73] *space, time, mass, velocity,* etc.

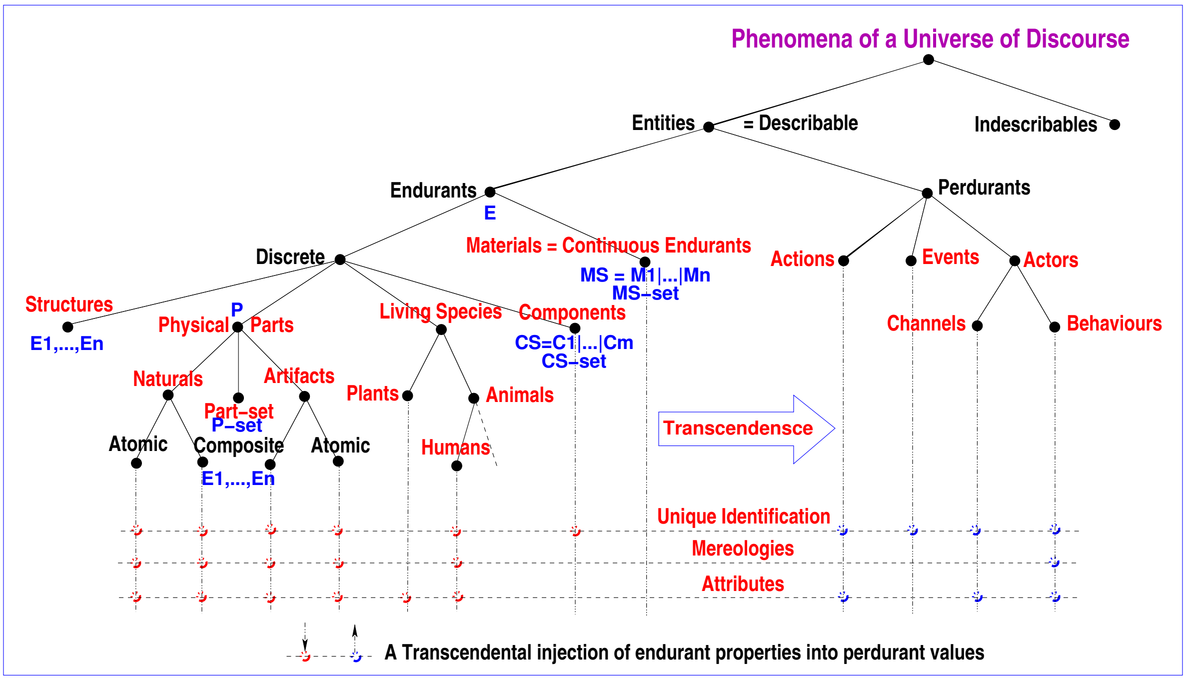


Figure 8: A Revised Upper Ontology for Domains

additional in-avoidable attributes such as *intent* and to both *gravitational pull* and *intentional "pull"*; and to *living species parts*, notably, in this report, *humans* with their attributes. 430

11.5.2 General

It is only of interest to study the domain analysis & description method *analysis calculus* with respect to Sørlander's Philosophy. The corresponding *description calculus* and schemata are not analytic. They represent our "response" to the domain analysis. So our "quest" has ended. It is time to "sum up". 431

Segment V: Summing Up

Although there is obviously a lot more to study we stop here, for a while, to wrap up this report. With what we have presented we can, however, make several conclusions – and that will now be done!

12 Conclusion 432

12.1 General Remarks

When I have informed my colleagues of this work their reactions have been mixed. *Oh yes, philosophy, yes, I referred to Plato in one of my papers, ages ago!*, or – *does it relate to*

the recent Facebook scandal ?, and other such deeply committing and understanding uttering.
433 Philosophy is actually hard. Anyone can claim to reflect philosophically, and many do, and some even refer, in their newspaper columns, to being philosophers, but it does take some
434 practice to actually do philosophy. Good schooling, up to senior high, is required. Having learned to reason, in classical disciplines like mathematics and physics; being able to read in two or more foreign languages; having learned history, real history, for us, in the Western world, from before the ancient Greeks, and on-wards; these seems to be prerequisites for a
435 serious study of philosophy.

In grammar school I passed the little test in Greek and the "large" test in Latin at the age of 14–15. I had wonderful teachers. I learned about the *history of ideas* from Johs. Sløk [23]. My university did not offer courses in philosophy. Over the years I acquired many [and browsed some additional] philosophy books: Karl Jaspers [89], Bertrand Russell [90, 91, 51], [Alfred North Whitehead [92, 52, 93],] Willard van Orme Quine [94, 95, 96], [Martin Heidegger [53],] Ludwig Johan Josef Wittgenstein [97, 50], Karl Popper [98, 99, 100, 101, 102, 103], Imre Lakatos [104], David Favrholdt [105, 106], John Sowa [107], as well as some dictionaries: [30,
436 29, 108, 31, Cambridge, Oxford, Blackwell] and [109]. In this century I started looking at a number of epistemological essays: [110, Logic and Ontology], [77, 78, 82, 111, 112, Objects], [79, 80, 81, 113, 85, Ontology], [114, 33, 57, Actions], [54, 55, 59, 115, 61, 63, 62, 58, 57, Events], [66, 67, 74, 75, 71, 86, 87, 83, 62, 26, Mereology], [116, 117, 118, 119, Qualities,
437 Properties] and [56, SpaceTime]. But although wonderful "reads", it was not until Sørlander's [15, 16, 2, 17, 19, 20, 3, 18] that philosophy really started meaning something. *'Philosophy is*
438 *useless'* it is said. *' "Results" of philosophy are not meant to solve problems '*, it is said. But Sørlander's Philosophy, [15, 18], have definitely helped shape the *domain analysis & description analysis calculus* into a form that makes it rather definitive!

Before my study of Kai Sørlander's Philosophy the upper ontology – like shown in Fig. 1 Pg. 15 – was based on empirical observations.

After my study the upper ontology – now shown in Fig. 7 Pg. 68 – is based on philosophical reasoning and is definite, is unavoidable!

12.2 Revisions to the Calculi and Further Studies 439

Yes, our study of Sørlander's Philosophy, [15, 18], has led to the following modifications of the *domain analysis & description analysis calculus*: (i) a more refined view of *discrete endurants*; (ii) "refinements" of *attributes* need be studied further; (iii) the *intentional "pull"* between *artifactual parts* need be studied further; and (iv) the *transcendental deduction* that "translates"
440 *endurants* into *behaviours* need be studied further see, however, below.

(i) Refined View of Discrete Endurants: Where *discrete endurants* before were (i.1) *parts* and (i.2) *components*, they are now (i.1a) *physical*, (i.2) *components*, (i.3) *live species parts* and (i.1b) *artifacts*. of which the *live species parts* are (i.3a) *plants* and (i.3b) *animals*, (i.3c)
441 for which latter we focus on *humans*,

(iv) Which Endurants are Candidates for Perdurancy ? (iv.1) **Naturals:** It seems that if we only focus on transcendentally deducing *natural endurants* into behaviours then we are
442 really studying or doing **physics:** *mechanics*, *chemistry*, *electricity*, et cetera. (iv.2) **Living Species:** It seems that if we only focus on transcendentally deducing (iv.2.1) *living species* into behaviours then we are really studying or doing **life sciences:** *botanics*, *zoology*, *biology*, et

cetera. (iv.2.2) or if we just focus on *humans*, then we are really studying or doing **behavioral sciences**. (iv.3) **Artifacts:** (iv.3.1) We have seen that it makes sense to "transmogrify" many 443
artifacts into behaviours. But how characterise those for which that deduction makes, or does not make sense? (iv.3.2) It seems that if we only focus on transcendentally deducing *artifacts* into behaviours then we are really studying or doing **engineering:** *mechanical*, *chemical*, *electrical*, *electronics*, et cetera, engineering.

12.3 Remarks on Classes of Artifactual Perdurants 444

We can rather immediately identify the following "classes" of *artifactual perdurants*:

- **Computerised Command & Control Systems:** Here we have several, i.e. more than just a few distinct artifacts, interacting with human operators for the purpose of command, monitoring and controlling some of these artifacts and humans. Examples are *pipelines* [120] and *swarms of drones* [121]. 445
- **Logistics: Planning & Monitoring:** Here again we have several, i.e. more than just a few distinct artifacts, but the emphasis is on operational planning and the monitoring of plan fulfillment. Examples are *container lines* [122] and *railways* [123, 124, 125, 126, 127]. 446
- **Monitoring:** Usually the systems here are just monitoring a single endurant. Examples are *weather forecast* [128] and *health care*. 447
- **Mechanics:** Here we are dealing with the operation of just one artifact: a *lathe* a *machine saw*, etc., an *automobile*, et cetera. 448
- **The "End" Result:** Here we are dealing with computers being the artifacts – "final" instruments in achieving some purpose! Examples are *urban planning* [129] *stock exchange* [130] *credit card system* [131] *documents* [132] *Web systems* [133] *E-market* [134]

449
We refer to [14] for a discussion of domain models as a basis for software demos, software simulators, software monitoring and software monitoring and control.

12.4 Acknowledgements 450

First and foremost I acknowledge the deep inspiration drawn from the study of Sørlander's Philosophy, notably [2] and [3]. Several people have commented, in various more-or-less spurious ways, not knowing really, what I was up to, when I informed them of my current study and writing on "applying" Sørlander's Philosophy, notably [2] and [3] to my work on domain analysis & description. Several of these comments, however uncommitted, have, however – strangely enough, upon reflection, helped me to even better grasp what it was I was trying to unravel. Let my acknowledgments to them remain anonymous.

13 Bibliography 451

13.1 Bibliographical Notes

We list a number of reports all of which document descriptions of domains. These descriptions were carried out in order to research and develop the domain analysis and description concepts now summarised in the present paper. These reports ought now be revised, some slightly, others less so, so as to follow all of the prescriptions of the current paper. Except where a URL is given in full, please prefix the web reference with: `http://www2.compute.dtu.dk/~dibj/`.

452

1 *A Railway Systems Domain:* `racosy/domains.ps` (2003)

2 *Models of IT Security:* `it-security.pdf` (2006)

3 *A Container Line Industry Domain:* `container-paper.pdf` (2007)

4 *The "Market": Buyers, Sellers, Traders:* `themarket.pdf` (2007)

5 *What is Logistics ?:* `logistics.pdf` (2009)

6 *A Domain Model of Oil Pipelines:* `pipeline.pdf` (2009)

7 *Transport Systems:* `comet/comet1.pdf` (2010)

8 *The Tokyo Stock Exchange:* `todai/tse-1.pdf` and `todai/tse-2.pdf` (2010)

9 *On Development of Web-based Software:* `wfdftp.pdf` (2010)

10 *A Credit Card System:* `/2016/uppsala/accs.pdf` (2016)

11 *Documents:* `/2017/docs.pdf` (2017)

12 *A Context for Swarms of Drones:* `/2016/uppsala/accs.pdf` (2017)

13 *A Framework for Urban Planning:* `/2018/accs.pdf` (2018)
http://www.imm.dtu.dk/ dibj/2017/urban-planning.pdf

13.1 References 453

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

[2] Kai Sørlander. *Om Menneskerettigheder*. Rosinante, 2000. 171 pages.

[3] Kai Sørlander. *Fornuftens Skæbne – Tanker om Menneskets Vilkår*. Informations Forlag, 2014. 238 pages.

[4] Dines Bjørner. A Domain Analysis & Description Method – Principles, Techniques and Modeling Languages. Research Note based on [1], February 20 2018. http://www.imm.dtu.dk/~dibj/2018/adam/2018daad.pdf.

[5] Dines Bjørner. Domain Facets: Analysis & Description. November 2016. Extensive revision of [6]. http://www.imm.dtu.dk/~dibj/2016/facets/faoc-facets.pdf.

[6] Dines Bjørner. Domain Engineering. In Paul Boca and Jonathan Bowen, editors, *Formal Methods: State of the Art and New Directions*, Eds. Paul Boca and Jonathan Bowen, pages 1–42, London, UK, 2010. Springer.

[7] Dines Bjørner. Domain Analysis and Description – Formal Models of Processes and Prompts. 2016. Extensive revision of [8]. http://www.imm.dtu.dk/~dibj/2016/process/process-p.pdf.

[8] Dines Bjørner. Domain Analysis: Endurants – An Analysis & Description Process Model. In Shusaku Iida, José Meseguer, and Kazuhiro Ogata, editors, *Specification, Algebra, and Software: A Festschrift Symposium in Honor of Kokichi Futatsugi*. Springer, May 2014.

[9] 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.

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

[11] Dines Bjørner. From Domain Descriptions to Requirements Prescriptions – A Different Approach to Requirements Engineering. 2016. Extensive revision of [12].

[12] 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.

[13] Dines Bjørner. Domains: Their Simulation, Monitoring and Control – A Divertimento of Ideas and Suggestions. Technical report, Fredsvej 11, DK–2840 Holte, Denmark, 2016. Extensive revision of [14]. http://www.imm.dtu.dk/~dibj/2016/demos/faoc-demo.pdf.

[14] Dines Bjørner. Domains: Their Simulation, Monitoring and Control – A Divertimento of Ideas and Suggestions. In *Rainbow of Computer Science, Festschrift for Hermann Maurer on the Occasion of His 70th Anniversary.*, Festschrift (eds. C. Calude, G. Rozenberg and A. Saloma), pages 167–183. Springer, Heidelberg, Germany, January 2011.

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

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

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

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

[19] Kai Sørlander. *Forsvaret for Rationaliteten*. Informations Forlag (Publ.), 2008. 232 pages.

[20] Kai Sørlander. *Den Politiske Forpligtelse – Det Filosofsike Fundament for Demokratisk Stillingtagen*. Informations Forlag, 2011. 280 pages.

[21] Johannes Sløk. *Platon*. Berlingske Filosofi Bibliotek, 1964.

[22] Johannes Sløk. *Stoikerne*. Berlingske Filosofi Bibliotek, 1966.

[23] Erik Lund, Mogens Pihl, and Johannes Sløk. *De europæiske ideeres historie*. Gyldendal, 1962.

[24] W. Little, H.W. Fowler, J. Coulson, and C.T. Onions. *The Shorter Oxford English Dictionary on Historical Principles*. Clarendon Press, Oxford, England, 1973, 1987. Two vols.

[25] Edward N. Zalta. The Stanford Encyclopedia of Philosophy. 2016. Principal Editor: https://plato.stanford.edu/.

[26] Roberto Casati and Achille C. Varzi. *Parts and Places: the structures of spatial representation*. MIT Press, 1999.

[27] 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.

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

[29] Ted Honderich. *The Oxford Companion to Philosophy*. Oxford University Press, Walton St., Oxford OX2 6DP, England, 1995.

[30] Rober Audi. *The Cambridge Dictionary of Philosophy*. Cambridge University Press, The Pitt Building, Trumpington Street, Cambridge CB2 1RP, England, 1995.

[31] Nicholas Bunnin and E.P. Tsui-James, editors. *The Blackwell Companion to Philosophy*. Blackwell Companions to Philosophy. Blackwell Publishers, 108 Cowley Road, Oxford OX4 1JF, UK, 1996.

[32] C.A.R. Hoare. *Communicating Sequential Processes*. C.A.R. Hoare Series in Computer Science. Prentice-Hall International, 1985. Published electronically: http://www.usingcsp.com/cspbook.pdf (2004).

[33] George Wilson and Samuel Shpall. Action. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2012 edition, 2012.

[34] Johan van Benthem. *The Logic of Time*, volume 156 of *Synthese Library: Studies in Epistemology, Logic, Methhodology, and Philosophy of Science (Editor: Jaakko Hintika)*. Kluwer Academic Publishers, P.O.Box 17, NL 3300 AA Dordrecht, The Netherlands, second edition, 1983, 1991.

[35] David John Farmer. *Being in time: The nature of time in light of McTaggart's paradox*. University Press of America, Lanham, Maryland, 1990. 223 pages.

[36] J. M. E. McTaggart. The Unreality of Time. *Mind*, 18(68):457–84, October 1908. New Series. See also: [37].

[37] Robin Le Poidevin and Murray MacBeath, editors. *The Philosophy of Time*. Oxford University Press, 1993.

[38] Arthur Prior. *Changes in Events and Changes in Things*, chapter in [37]. Oxford University Press, 1993.

[39] Arthur N. Prior. *Logic and the Basis of Ethics*. Clarendon Press, Oxford, UK, 1949.

[40] Arthur N. Prior. *Formal Logic*. Clarendon Press, Oxford, UK, 1955.

[41] Arthur N. Prior. *Time and Modality*. Oxford University Press, Oxford, UK, 1957.

[42] Arthur N. Prior. *Past, Present and Future*. Clarendon Press, Oxford, UK, 1967.

[43] Arthur N. Prior. *Papers on Time and Tense*. Clarendon Press, Oxford, UK, 1968.

[44] Gerald Rochelle. *Behind time: The incoherence of time and McTaggart's atemporal replacement*. Avebury series in philosophy. Ashgate, Brookfield, Vt., USA, 1998. vii + 221 pages.

[45] Wayne D. Blizard. A Formal Theory of Objects, Space and Time. *The Journal of Symbolic Logic*, 55(1):74–89, March 1990.

[46] David Hume. *An Enquiry Concerning Human Understanding*. Washington Square Press, 1963 (1748, 1750). Ed.Ernest C. Mossner.

[47] Bertrand Russell. The Philosophy of Logical Atomism. *The Monist: An International Quarterly Journal of General Philosophical Inquiry,*, xxxviii–xxix:495–527, 32–63, 190–222, 345–380, 1918–1919.

[48] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica, 3 vols.* Cambridge University Press, 1910, 1912, and 1913. Second edition, 1925 (Vol. 1), 1927 (Vols 2, 3), also Cambridge University Press, 1962.

[49] Ludwig Wittgenstein. *Tractatus Logico–Philosophicus*. Taylor & Francis Books Ltd, 1921 (1975).

[50] Ludwig Johan Josef Wittgenstein. *Philosophical Investigations*. Oxford Univ. Press, 1958.

[51] Bertrand Russell. *History of Western Philosophy*. George Allen and Unwin Ltd, Unwin University Books, London, 1974 (1945, 1961).

[52] A.N. Whitehead. *The Concept of Nature*. Cambridge University Press, Cambridge, 1920.

[53] Martin Heidegger. *Sein und Zeit (Being and Time)*. Oxford University Press, 1927, 1962.

[54] F. Dretske. Can Events Move? *Mind*, 76(479-492), 1967. Reprinted in [61, 1996], pp. 415-428.

[55] A. Quinton. Objects and Events. *Mind*, 88:197–214, 1979.

[56] D.H. Mellor. Things and Causes in Spacetime. *British Journal for the Philosophy of Science*, 31:282–288, 1980.

[57] Donald Davidson. *Essays on Actions and Events*. Oxford University Press, 1980.

[58] P.M.S. Hacker. Events and Objects in Space and Time. *Mind*, 91:1–19, 1982. reprinted in [61], pp. 429-447.

[59] Alain Badiou. *Being and Event*. Continuum, 2005. (Lêtre et l'événements, Edition du Seuil, 1988).

[60] Jaegwon Kim. *Supervenience and Mind*. Cambridge University Press, 1993.

[61] Roberto Casati and Achille C. Varzi, editors. *Events*. Ashgate Publishing Group – Dartmouth Publishing Co. Ltd., Wey Court East, Union Road, Farnham, Surrey, GU9 7PT, United Kingdom, 23 March 1996.

[62] Chia-Yi Tony Pi. *Mereology in Event Semantics*. Phd, McGill University, Montreal, Canada, August 1999.

[63] Roberto Casati and Achille Varzi. Events. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2010 edition, 2010.

[64] Thomas Bittner, Maureen Donnelly, and Barry Smith. Endurants and Perdurants in Directly Depicting Ontologies. *AI Communications*, 17(4):247–258, December 2004. IOS Press, in [135].

[65] . Technical report. .

[66] E.C. Luschei. *The Logical Systems of Leśniewksi*. North Holland, Amsterdam, The Netherlands, 1962.

[67] C. Lejewski. A note on Leśniewksi's Axiom System for the Mereological Notion of Ingredient or Element. *Topoi*, 2(1):63–71, June, 1983.

[68] J.T.J. Srzednicki and Z. Stachniak, editors. *Leśniewksi's Lecture Notes in Logic*. Dordrecht, 1988.

[69] J.T.J. Srzednicki and Z. Stachniak. *Leśniewksi's Systems Protothetic*. . Dordrecht, 1998.

[70] S. J. Surma, J. T. Srzednicki, D. I. Barnett, and V. F. Rickey, editors. *Stanisław Leśniewksi: Collected works (2 Vols.)*. Dordrecht, Boston – New York, 1988.

[71] Henry S. Leonard and Nelson Goodman. The Calculus of Individuals and its Uses. *Journal of Symbolic Logic*, 5:45–44, 1940.

[72] Nelson Goodman. *The Structure of Appearance*. 1st.: Cambridge, Mass., Harvard University Press; 2nd.: Indianapolis, Bobbs-Merrill, 1966; 3rd.: Dordrecht, Reidel, 1977, 1951.

[73] Marcus Rossberg. *Leonard, Goodman, and the Development of the Calculus of Individuals*. Ontos, Frankfurt, Germany, 2009.

[74] Bowman L. Clarke. A Calculus of Individuals Based on 'Connection'. *Notre Dame J. Formal Logic*, 22(3):204–218, 1981.

[75] Bowman L. Clarke. Individuals and Points. *Notre Dame J. Formal Logic*, 26(1):61–75, 1985.

[76] Douglas T. Ross. Toward foundations for the understanding of type. In *Proceedings of the 1976 conference on Data: Abstraction, definition and structure*, pages 63–65, New York, NY, USA, 1976. ACM. http://doi.acm.org/10.1145/800237.807120.

[77] M. Bunge. *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*, volume 3. Reidel, Boston, Mass., USA, 1977.

[78] M. Bunge. *Treatise on Basic Philosophy: Ontology II: A World of Systems*, volume 4. Reidel, Boston, Mass., USA, 1979.

[79] Peter M. Simons. *Parts: A Study in Ontology*. Clarendon Press, 1987.

[80] Barry Smith. Ontology and the Logistic Analysis of Reality. In N. Guarino and R. Poli, editors, *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*. Institute for Systems Theory and Biomedical Engineering of the Italian National Research Council, Padua, Italy, 1993. Revised version in G. Haefliger and P. M. Simons (eds.), Analytic Phenomenology, Dordrecht/Boston/London: Kluwer.

[81] Barry Smith. Ontology and the Logistic Analysis of Reality. In G. Haefliger and P. M. Simons, editors, *Analytic Phenomenology*. Dordrecht/Boston/London: Kluwer, Padua, Italy, 1993.

[82] Barry Smith. Fiat Objects. In L. Vieu N. Guarino and S. Pribbenow, editors, *Parts and Wholes: Conceptual Part-Whole Relations and Formal Mereology*, 11th European Conference on Artificial Intelligence, pages 15–23. European Coordinating Committee for Artificial Intelligence, Amsterdam, 8 August 1994. (Revised version: Topoi, 2001, vol.20, no.2, pp.131-148).

[83] Barry Smith. Mereotopology: A Theory of Parts and Boundaries. *Data and Knowledge Engineering*, 20:287–303, 1996.

[84] Pierre Grenon and Barry Smith. SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition & Computation*, 4(1):69 – 104, 2004.

[85] Pierre Grenon and Barry Smith. SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition and Computation*, 4(1):69–104, 2004.

[86] Achille C. Varzi. *Spatial Reasoning in a Holey[74] World*, volume 728 of *Lecture Notes in Artificial Intelligence*, pages 326–336. Springer, 1994.

[87] Achille C. Varzi. *On the Boundary between Mereology and Topology*, pages 419–438. Hölder-Pichler-Tempsky, Vienna, 1994.

[74]holey: something full of holes

[88] Dines Bjørner. The Manifest Domain Analysis & Description Approach to Implicit and Explicit Semantics. *EPTCS: Electronic Proceedings in Theoretical Computer Science, Yasmine Ait-Majeur, Paul J. Gibson and Dominique Méry*, 2018. First International Workshop on Handling IMPlicit and EXplicit Knowledge in Formal Fystem Development, 17 November 2017, Xi'an, China.

[89] Karl Jaspers. *Philosophy, Vols.1–3*. The University of Chicago Press, 1969. Translated by E.B. Ashton.

[90] Bertrand Russell. *The Problems of Philosophy*. Home University Library, London, 1912. Oxford University Press paperback, 1959 Reprinted, 1971-2.

[91] Bertrand Russell. *Introduction to Mathematical Philosophy*. George Allen and Unwin, London, 1919.

[92] A.N. Whitehead. *An Enquiry Concerning the Principles of Natural Knowledge*. Cambridge University Press, Cambridge, 1929.

[93] Alfred North Whitehead. *Science and the Modern World*. The Free Press, New York, 1925.

[94] Willard van Orman Quine. *From a Logical Point of View*. Harvard Univ. Press, Cambridge, Mass., USA, 1953, 1980. Collection of papers: Language and Ontology.

[95] Willard van Orman Quine. *Word and Object*. The MIT Press, Cambridge, Mass., USA, 1960. Naturalism: Philosophy as part of Natural Science.

[96] Willard van Orman Quine. *Pursuit of Truth*. Harvard Univ. Press, Cambridge, Mass., USA, paperback edition, 1992. Clear, concise formulation of Quine's philosophical position.

[97] Ludwig Johan Josef Wittgenstein. *Tractatu Logico–Philosophicus*. Oxford Univ. Press, London, (1921) 1961.

[98] Karl R. Popper. *The Logic of Scientific Dicovery*. Hutchinson of London, 3 Fitzroy Square, London W1, England, 1959,. . . ,1979. Translated from [136].

[99] Karl R. Popper. *Conjectures and Refutations. The Growth of Scientific Knowledge*. Routledge and Kegan Paul Ltd. (Basic Books, Inc.), 39 Store Street, WC1E 7DD, London, England (New York, NY, USA), 1963,. . . ,1981.

[100] Karl R. Popper. *Autobiography of Karl Popper*. in The Philosohy of Karl Popper in *The Library of Living Philosophers*, Ed. Paul Arthur Schilpp. Open Court Publishing Co., Illionos, USA, 1976. See also [101].

[101] Karl R. Popper. *Unended Quest: An Intellectual Autobiography*. Philosophy and Autobiography. Fontana/Collins, England, 1976–1982. Originally in [100].

[102] Karl R. Popper. *A Pocket Popper*. Fontana Pocket Readers. Fontana Press, England, 1983. An edited collection, Ed. David Miller.

[103] Karl R. Popper. *The Myth of the Framework. In defence of science and rationality*. Routledge, 11 New Fetter Lane, London EC4P 4EE, England, 1994, 1996. An edited collection of essays, Ed. M.A. Notturno.

[104] Imre Lakatos. *Proofs and Refutations: The Logic of Mathematical Discovery (Eds.: J. Worrall and E. G. Zahar)*. Cambridge University Press, The Edinburgh Building, Shaftesbury Road, Cambridge CB2 2RU, England, 2 September 1976. ISBN: 0521290384. Published in 1963-64 in four parts in the British Journal for Philosophy of Science. (Originally Lakatos' name was Imre Lipschitz.).

[105] David Favrholdt. *Filosofisk Codex — Om begrundelsen af den menneskelige erkendelse*. Gyldendal, Nordisk Forlag, Klareboderne, Copenhagen K, Denmark, 1999. In Danish. English/German 'translation': Philosophical Codex — On motivating human 'erkenntniss'. 361 pages.

[106] David Favrholdt. *Æstetik og filosofi*. Høst Humaniora. Høst & Søn, Købmagergade 62, DK–1150 Copenhagen K, Denmark, Fall 2000.

[107] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole Thompson Learning, August 17, 1999.

[108] Jonathan Dancy and Ernest Sosa, editors. *The Blackwell Companion to Epistemology*. Blackwell Companions to Philosophy. Blackwell Publishers, 108 Cowley Road, Oxford OX4 1JF, UK, 1994.

[109] Arne Naess. *Philosophy*. Oslo, Norway, 1980.

[110] Thomas Hofweber. Logic and ontology. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2012 edition, 2012.

[111] Henry Laycock. Object. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2011 edition, 2011.

[112] Henry Laycock. Object. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2011 edition, 2011.

[113] Thomas Bittner. Ontology, Vagueness, and Indeterminacy — Extended Abstract. Technical report, Centre de recherche en geomatique, Laval University, Quebec, Canada. Thomas.Bittner@scg.ulaval.ca.

[114] George Wilson and Samuel Shpall. Action. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2012 edition, 2012.

[115] Roberto Casati and Achille Varzi. Events. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2010 edition, 2010.

[116] Nelson Goodman. *A Study of Qualities*. Garland, New York, 1990. Ph.D. dissertation thesis, Harvard, 1940.

[117] D. H. Mellor and Alex Oliver. *Properties*. Oxford Readings in Philosophy. Oxford Univ Press, May 1997. ISBN: 0198751761, 320 pages.

[118] D. H. Mellor and Alex Oliver. *Properties*. Oxford Readings in Philosophy. Oxford Univ Press, May 1997. ISBN: 0198751761, 320 pages.

[119] Chris Fox. *The Ontology of Language: Properties, Individuals and Discourse*. CSLI Publications, Center for the Study of Language and Information, Stanford University, California, ISA, 2000.

[120] Dines Bjørner. Pipelines – a Domain Description[75]. Experimental Research Report 2013-2, DTU Compute and Fredsvej 11, DK-2840 Holte, Denmark, Spring 2013.

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

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

[123] 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 Automatisieringstechnik, VDI-Gesellschaft für Fahrzeug– und Verkehrstechnik. Invited talk.

[124] 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.

[125] 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.

[126] 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.

[127] 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.

[128] 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.

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

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

[130] 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.

[131] 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.

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

[133] Dines Bjørner. On Development of Web-based Software: A Divertimento of Ideas and Suggestions. Technical, Technical University of Vienna, August–October 2010. http://www.imm.dtu.dk/˜dibj/wfdftp.pdf.

[134] 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.

[135] Jochen Renz and Hans W. Guesgen, editors. *Spatial and Temporal Reasoning*, volume 14, vol. 4, Journal: AI Communications, Amsterdam, The Netherlands, Special Issue. IOS Press, December 2004.

[136] Karl R. Popper. *Logik der Forschung*. Julius Springer Verlag, Vienna, Austria, 1934 (1935). English version [98].

[137] Dines Bjørner. *Software Engineering, Vol. 2: Specification of Systems and Languages*. Texts in Theoretical Computer Science, the EATCS Series. Springer, 2006. Chapters 12–14 are primarily authored by Christian Krog Madsen.

454

Segment VI: Appendix

455

A RSL: The RAISE Specification Language – A Primer 456

A.1 Type Expressions

Type expressions are expressions whose value are types, that is, possibly infinite sets of values (of "that" type).

A.1.1 Atomic Types

Atomic types have (atomic) values. That is, values which we consider to have no proper constituent (sub-)values, i.e., cannot, to us, be meaningfully "taken apart".

RSL has a number of *built-in* atomic types. There are the Booleans, integers, natural numbers,
457 reals, characters, and texts.

type
- [1] **Bool** **true**, **false**
- [2] **Int** ... , −2, −2, 0, 1, 2, ...
- [3] **Nat** 0, 1, 2, ...
- [4] **Real** ..., −5.43, −1.0, 0.0, 1.23···, 2,7182···, 3,1415···, 4.56, ...
- [5] **Char** "a", "b", ..., "0", ...
- [6] **Text** "abracadabra"

A.1.2 Composite Types

Composite types have composite values. That is, values which we consider to have proper constituent (sub-)values, i.e., can be meaningfully "taken apart". There are two ways of expressing composite types: either explicitly, using concrete type expressions, or implicitly, using sorts (i.e., abstract types) and observer functions.

Concrete Composite Types From these one can form type expressions: finite sets, infinite sets, Cartesian products, lists, maps, etc.

Let A, B and C be any type names or type expressions, then the following are type expressions:

- [7] A-**set**
- [8] A-**infset**
- [9] A × B × ... × C
- [10] A^*
- [11] A^ω
- [12] $A \xrightarrow[m]{} B$
- [13] A → B
- [14] $A \stackrel{\sim}{\to} B$
- [15] (A)
- [16] A | B | ... | C
- [17] mk_id(sel_a:A,...,sel_b:B)
- [18] sel_a:A ... sel_b:B

The following the meaning of the atomic and the composite type expressions:

1 The Boolean type of truth values **false** and **true**.

2 The integer type on integers ..., –2, –1, 0, 1, 2, ...

3 The natural number type of positive integer values 0, 1, 2, ...

4 The real number type of real values, i.e., values whose numerals can be written as an integer, followed by a period ("."), followed by a natural number (the fraction).

5 The character type of character values ″a″, ″bb″, ...

6 The text type of character string values ″aa″, ″aaa″, ..., ″abc″, ...

7 The set type of finite cardinality set values.

8 The set type of infinite and finite cardinality set values.

9 The Cartesian type of Cartesian values.

10 The list type of finite length list values.

11 The list type of infinite and finite length list values.

12 The map type of finite definition set map values.

13 The function type of total function values.

14 The function type of partial function values.

15 In (A) A is constrained to be:

- either a Cartesian B × C × ... × D, in which case it is identical to type expression kind 9,
- or not to be the name of a built-in type (cf., 1–6) or of a type, in which case the parentheses serve as simple delimiters, e.g., (A $\overrightarrow{m}$ B), or (A*)-**set**, or (A-**set**)list, or (A|B) $\overrightarrow{m}$ (C|D|(E $\overrightarrow{m}$ F)), etc.

16 The postulated disjoint union of types A, B, . . . , and C.

17 The record type of mk_id-named record values mk_id(av,...,bv), where av, . . . , bv, are values of respective types. The distinct identifiers sel_a, etc., designate selector functions.

18 The record type of unnamed record values (av,...,bv), where av, . . . , bv, are values of respective types. The distinct identifiers sel_a, etc., designate selector functions.

Sorts and Observer Functions

type
  A, B, C, ..., D
value
  obs_B: A → B, obs_C: A → C, ..., obs_D: A → D

The above expresses that values of type A are composed from at least three values — and these are of type B, C, ..., and D. A concrete type definition corresponding to the above presupposing material of the next section

```
type
  B, C, ..., D
  A = B × C × ... × D
```

A.2 Type Definitions

A.2.1 Concrete Types

Types can be concrete in which case the structure of the type is specified by type expressions:

```
type
  A = Type_expr
```

Some schematic type definitions are:

```
[19]  Type_name = Type_expr /∗ without |s or subtypes ∗/
[20]  Type_name = Type_expr_1 | Type_expr_2 | ... | Type_expr_n
[21]  Type_name ==
          mk_id_1(s_a1:Type_name_a1,...,s_ai:Type_name_ai) |
          ... |
          mk_id_n(s_z1:Type_name_z1,...,s_zk:Type_name_zk)
[22]  Type_name :: sel_a:Type_name_a ... sel_z:Type_name_z
[23]  Type_name = {| v:Type_name′ • 𝒫(v) |}
```

458 where a form of [20]–[21] is provided by combining the types:

```
  Type_name = A | B | ... | Z
  A == mk_id_1(s_a1:A_1,...,s_ai:A_i)
  B == mk_id_2(s_b1:B_1,...,s_bj:B_j)
  ...
  Z == mk_id_n(s_z1:Z_1,...,s_zk:Z_k)
```

Types A, B, ..., Z are disjoint, i.e., shares no values, provided all mk_id_k are distinct and due to the use of the disjoint record type constructor ==.

```
axiom
  ∀ a1:A_1, a2:A_2, ..., ai:Ai •
    s_a1(mk_id_1(a1,a2,...,ai))=a1 ∧ s_a2(mk_id_1(a1,a2,...,ai))=a2 ∧
    ... ∧ s_ai(mk_id_1(a1,a2,...,ai))=ai ∧
  ∀ a:A • let mk_id_1(a1′,a2′,...,ai′) = a in
    a1′ = s_a1(a) ∧ a2′ = s_a2(a) ∧ ... ∧ ai′ = s_ai(a) end
```

A.2.2 Subtypes

In RSL, each type represents a set of values. Such a set can be delimited by means of predicates. The set of values b which have type B and which satisfy the predicate $\mathcal{P}$, constitute the subtype A:

type
  A = {| b:B • $\mathcal{P}$(b) |}

A.2.3 Sorts — Abstract Types

Types can be (abstract) sorts in which case their structure is not specified:

type
  A, B, ..., C

A.3 The RSL Predicate Calculus

A.4 Propositional Expressions

Let identifiers (or propositional expressions) a, b, ..., c designate Boolean values (**true** or **false** [or **chaos**]). Then:

  false, **true**
  a, b, ..., c ∼a, a∧b, a∨b, a⇒b, a=b, a≠b

are propositional expressions having Boolean values. ∼, ∧, ∨, ⇒, = and ≠ are Boolean connectives (i.e., operators). They can be read as: *not*, *and*, *or*, *if then* (or *implies*), *equal* and *not equal*.

A.4.1 Simple Predicate Expressions

Let identifiers (or propositional expressions) a, b, ..., c designate Boolean values, let x, y, ..., z (or term expressions) designate non-Boolean values and let i, j, ..., k designate number values, then:

  false, **true**
  a, b, ..., c
  ∼a, a∧b, a∨b, a⇒b, a=b, a≠b
  x=y, x≠y,
  i<j, i≤j, i≥j, i≠j, i≥j, i>j

are simple predicate expressions.

A.4.2 Quantified Expressions

Let X, Y, ..., C be type names or type expressions, and let $\mathcal{P}(x)$, $\mathcal{Q}(y)$ and $\mathcal{R}(z)$ designate predicate expressions in which x, y and z are free. Then:

∀ x:X • $\mathcal{P}(x)$
∃ y:Y • $\mathcal{Q}(y)$
∃ ! z:Z • $\mathcal{R}(z)$

are quantified expressions — also being predicate expressions.

They are "read" as: For all x (values in type X) the predicate $\mathcal{P}(x)$ holds; there exists (at least) one y (value in type Y) such that the predicate $\mathcal{Q}(y)$ holds; and there exists a unique z (value in type Z) such that the predicate $\mathcal{R}(z)$ holds.

A.5 Concrete RSL Types: Values and Operations

A.5.1 Arithmetic

type
  Nat, **Int**, **Real**
value
  $+,-,*$: **Nat**×**Nat**→**Nat** | **Int**×**Int**→**Int** | **Real**×**Real**→**Real**
  /: **Nat**×**Nat**$\xrightarrow{\sim}$**Nat** | **Int**×**Int**$\xrightarrow{\sim}$**Int** | **Real**×**Real**$\xrightarrow{\sim}$**Real**
  $<,\leq,=,\neq,\geq,>$ (**Nat**|**Int**|**Real**) → (**Nat**|**Int**|**Real**)

A.5.2 Set Expressions

Set Enumerations Let the below a's denote values of type A, then the below designate simple set enumerations:

{{}, {a}, {e_1,e_2,...,e_n}, ...} ∈ A-**set**
{{}, {a}, {e_1,e_2,...,e_n}, ..., {e_1,e_2,...}} ∈ A-**infset**

Set Comprehension The expression, last line below, to the right of the ≡, expresses set comprehension. The expression "builds" the set of values satisfying the given predicate. It is abstract in the sense that it does not do so by following a concrete algorithm.

type
  A, B
  P = A → **Bool**
  Q = A $\xrightarrow{\sim}$ B
value
  comprehend: A-**infset** × P × Q → B-**infset**
  comprehend(s,P,Q) ≡ { Q(a) | a:A • a ∈ s ∧ P(a)}

A.5.3 Cartesian Expressions

Cartesian Enumerations Let e range over values of Cartesian types involving A, B, ..., C, then the below expressions are simple Cartesian enumerations:

```
type
  A, B, ..., C
  A × B × ... × C
value
  (e1,e2,...,en)
```

A.5.4 List Expressions

List Enumerations Let a range over values of type A, then the below expressions are simple list enumerations:

```
{⟨⟩, ⟨e⟩, ..., ⟨e1,e2,...,en⟩, ...} ∈ A*
{⟨⟩, ⟨e⟩, ..., ⟨e1,e2,...,en⟩, ..., ⟨e1,e2,...,en,... ⟩, ...} ∈ A^ω

⟨ a_i .. a_j ⟩
```

The last line above assumes a_i and a_j to be integer-valued expressions. It then expresses the set of integers from the value of e_i to and including the value of e_j. If the latter is smaller than the former, then the list is empty.

List Comprehension The last line below expresses list comprehension.

```
type
  A, B, P = A → Bool, Q = A ~→ B
value
  comprehend: A^ω × P × Q ~→ B^ω
  comprehend(l,P,Q) ≡ ⟨ Q(l(i)) | i in ⟨1..len l⟩ • P(l(i))⟩
```

A.5.5 Map Expressions

Map Enumerations Let (possibly indexed) u and v range over values of type $T1$ and $T2$, respectively, then the below expressions are simple map enumerations:

```
type
  T1, T2
  M = T1 →m T2
value
  u,u1,u2,...,un:T1, v,v1,v2,...,vn:T2
  [ ], [u↦v], ..., [u1↦v1,u2↦v2,...,un↦vn] all ∈ M
```

Map Comprehension The last line below expresses map comprehension:

type
U, V, X, Y
$M = U \xrightarrow{m} V$
$F = U \stackrel{\sim}{\to} X$
$G = V \stackrel{\sim}{\to} Y$
$P = U \to$ **Bool**
value
comprehend: $M \times F \times G \times P \to (X \xrightarrow{m} Y)$
comprehend(m,F,G,P) $\equiv$ [F(u) $\mapsto$ G(m(u)) | u:U • u $\in$ **dom** m $\wedge$ P(u)]

A.5.6 Set Operations

Set Operator Signatures

value
19 $\in$: A $\times$ **A-infset** $\to$ **Bool**
20 $\notin$: A $\times$ **A-infset** $\to$ **Bool**
21 $\cup$: **A-infset** $\times$ **A-infset** $\to$ **A-infset**
22 $\cup$: (**A-infset**)-**infset** $\to$ **A-infset**
23 $\cap$: **A-infset** $\times$ **A-infset** $\to$ **A-infset**
24 $\cap$: (**A-infset**)-**infset** $\to$ **A-infset**
25 $\setminus$: **A-infset** $\times$ **A-infset** $\to$ **A-infset**
26 $\subset$: **A-infset** $\times$ **A-infset** $\to$ **Bool**
27 $\subseteq$: **A-infset** $\times$ **A-infset** $\to$ **Bool**
28 $=$: **A-infset** $\times$ **A-infset** $\to$ **Bool**
29 $\neq$: **A-infset** $\times$ **A-infset** $\to$ **Bool**
30 **card**: **A-infset** $\stackrel{\sim}{\to}$ **Nat**

Set Examples

examples
a $\in$ {a,b,c}
a $\notin$ {}, a $\notin$ {b,c}
{a,b,c} $\cup$ {a,b,d,e} = {a,b,c,d,e}
$\cup${{a},{a,bb},{a,d}} = {a,b,d}
{a,b,c} $\cap$ {c,d,e} = {c}
$\cap${{a},{a,bb},{a,d}} = {a}
{a,b,c} $\setminus$ {c,d} = {a,bb}
{a,bb} $\subset$ {a,b,c}
{a,b,c} $\subseteq$ {a,b,c}
{a,b,c} = {a,b,c}
{a,b,c} $\neq$ {a,bb}
card {} = 0, **card** {a,b,c} = 3

Informal Explication

19 $\in$: The membership operator expresses that an element is a member of a set.

20 $\notin$: The nonmembership operator expresses that an element is not a member of a set.

21 $\cup$: The infix union operator. When applied to two sets, the operator gives the set whose members are in either or both of the two operand sets.

22 $\cup$: The distributed prefix union operator. When applied to a set of sets, the operator gives the set whose members are in some of the operand sets.

23 $\cap$: The infix intersection operator. When applied to two sets, the operator gives the set whose members are in both of the two operand sets.

24 $\cap$: The prefix distributed intersection operator. When applied to a set of sets, the operator gives the set whose members are in some of the operand sets. 459

25 $\setminus$: The set complement (or set subtraction) operator. When applied to two sets, the operator gives the set whose members are those of the left operand set which are not in the right operand set.

26 $\subseteq$: The proper subset operator expresses that all members of the left operand set are also in the right operand set.

27 $\subset$: The proper subset operator expresses that all members of the left operand set are also in the right operand set, and that the two sets are not identical.

28 $=$: The equal operator expresses that the two operand sets are identical.

29 $\neq$: The nonequal operator expresses that the two operand sets are *not* identical.

30 **card**: The cardinality operator gives the number of elements in a finite set.

Set Operator Definitions The operations can be defined as follows ($\equiv$ is the definition symbol):

value

$s' \cup s'' \equiv \{ a \mid a{:}A \bullet a \in s' \vee a \in s'' \}$
$s' \cap s'' \equiv \{ a \mid a{:}A \bullet a \in s' \wedge a \in s'' \}$
$s' \setminus s'' \equiv \{ a \mid a{:}A \bullet a \in s' \wedge a \notin s'' \}$
$s' \subseteq s'' \equiv \forall\, a{:}A \bullet a \in s' \Rightarrow a \in s''$
$s' \subset s'' \equiv s' \subseteq s'' \wedge \exists\, a{:}A \bullet a \in s'' \wedge a \notin s'$
$s' = s'' \equiv \forall\, a{:}A \bullet a \in s' \equiv a \in s'' \equiv s{\subseteq}s' \wedge s'{\subseteq}s$
$s' \neq s'' \equiv s' \cap s'' \neq \{\}$
card $s \equiv$
  if $s = \{\}$ **then** 0 **else**
  let $a{:}A \bullet a \in s$ **in** $1 +$ **card** $(s \setminus \{a\})$ **end end**
  pre s /∗ is a finite set ∗/
card $s \equiv$ **chaos** /∗ tests for infinity of s ∗/

A.5.7 Cartesian Operations

type
  A, B, C
  g0: G0 = A × B × C
  g1: G1 = (A × B × C)
  g2: G2 = (A × B) × C
  g3: G3 = A × (B × C)

value
  va:A, vb:B, vc:C, vd:D
  (va,vb,vc):G0,
  (va,vb,vc):G1
  ((va,vb),vc):G2
  (va3,(vb3,vc3)):G3

decomposition expressions
  let (a1,b1,c1) = g0,
    (a1′,b1′,c1′) = g1 **in** .. **end**
  let ((a2,b2),c2) = g2 **in** .. **end**
  let (a3,(b3,c3)) = g3 **in** .. **end**

A.5.8 List Operations

List Operator Signatures

value
  hd: $A^\omega \xrightarrow{\sim} A$
  tl: $A^\omega \xrightarrow{\sim} A^\omega$
  len: $A^\omega \xrightarrow{\sim}$ **Nat**
  inds: $A^\omega \rightarrow$ **Nat-infset**
  elems: $A^\omega \rightarrow$ A-**infset**
  .(.): $A^\omega \times$ **Nat** $\xrightarrow{\sim} A$
  $^\frown$: A*: ≠A:$^\omega$AA$^{\omega\omega}$ A$^\omega$* A$^\omega$→ **BoBbol**

List Operation Examples

examples
  hd⟨a1,a2,...,am⟩=a1
  tl⟨a1,a2,...,am⟩=⟨a2,...,am⟩
  len⟨a1,a2,...,am⟩=m
  inds⟨a1,a2,...,am⟩={1,2,...,m}
  elems⟨a1,a2,...,am⟩={a1,a2,...,am}
  ⟨a1,a2,...,am⟩(i)=ai
  ⟨a,b,c⟩$^\frown$⟨a,b,d⟩ = ⟨a,b,c,a,b,d⟩
  ⟨a,b,c⟩=⟨a,b,c⟩
  ⟨a,b,c⟩ ≠ ⟨a,b,d⟩

Informal Explication

- **hd**: Head gives the first element in a nonempty list.
- **tl**: Tail gives the remaining list of a nonempty list when Head is removed.
- **len**: Length gives the number of elements in a finite list.

- **inds**: Indices give the set of indices from 1 to the length of a nonempty list. For empty lists, this set is the empty set as well.
- **elems**: Elements gives the possibly infinite set of all distinct elements in a list.
- $\ell(i)$: Indexing with a natural number, i larger than 0, into a list ℓ having a number of elements larger than or equal to i, gives the ith element of the list. 460
- $\frown$: Concatenates two operand lists into one. The elements of the left operand list are followed by the elements of the right. The order with respect to each list is maintained.
- $=$: The equal operator expresses that the two operand lists are identical.
- $\neq$: The nonequal operator expresses that the two operand lists are *not* identical.

The operations can also be defined as follows:

List Operator Definitions

```
value
   is_finite_list: A^ω → Bool

   len q ≡
      case is_finite_list(q) of
         true → if q = ⟨⟩ then 0 else 1 + len tl q end,
         false → chaos end

   inds q ≡
      case is_finite_list(q) of
         true → { i | i:Nat • 1 ≤ i ≤ len q },
         false → { i | i:Nat • i≠0 } end

   elems q ≡ { q(i) | i:Nat • i ∈ inds q }
```

461

```
   q(i) ≡
      if i=1
         then
            if q≠⟨⟩
               then let a:A,q′:Q • q=⟨a⟩⌢q′ in a end
               else chaos end
         else q(i−1) end

   fq ⌢ iq ≡
         ⟨ if 1 ≤ i ≤ len fq then fq(i) else iq(i − len fq) end
          | i:Nat • if len iq≠chaos then i ≤ len fq+len end ⟩
      pre is_finite_list(fq)
```


$\mathsf{iq}' = \mathsf{iq}'' \equiv$
$\quad \mathbf{inds}\ \mathsf{iq}' = \mathbf{inds}\ \mathsf{iq}'' \wedge \forall\ \mathsf{i}{:}\mathbf{Nat} \bullet \mathsf{i} \in \mathbf{inds}\ \mathsf{iq}' \Rightarrow \mathsf{iq}'(\mathsf{i}) = \mathsf{iq}''(\mathsf{i})$

$\mathsf{iq}' \neq \mathsf{iq}'' \equiv \sim(\mathsf{iq}' = \mathsf{iq}'')$

A.5.9 Map Operations

Map Operator Signatures and Map Operation Examples

value

$\mathsf{m(a)}{:}\ \mathsf{M} \to \mathsf{A} \stackrel{\sim}{\to} \mathsf{B},\ \mathsf{m(a)} = \mathsf{b}$

$\mathbf{dom}{:}\ \mathsf{M} \to \mathsf{A}\text{-}\mathbf{infset}\ [\text{domain of map}]$
$\quad \mathbf{dom}\ [\mathsf{a1}{\mapsto}\mathsf{b1},\mathsf{a2}{\mapsto}\mathsf{b2},...,\mathsf{an}{\mapsto}\mathsf{bn}] = \{\mathsf{a1},\mathsf{a2},...,\mathsf{an}\}$

$\mathbf{rng}{:}\ \mathsf{M} \to \mathsf{B}\text{-}\mathbf{infset}\ [\text{range of map}]$
$\quad \mathbf{rng}\ [\mathsf{a1}{\mapsto}\mathsf{b1},\mathsf{a2}{\mapsto}\mathsf{b2},...,\mathsf{an}{\mapsto}\mathsf{bn}] = \{\mathsf{b1},\mathsf{b2},...,\mathsf{bn}\}$

$\dagger{:}\ \mathsf{M} \times \mathsf{M} \to \mathsf{M}\ [\text{override extension}]$
$\quad [\mathsf{a}{\mapsto}\mathsf{b},\mathsf{a}'{\mapsto}\mathsf{bb}',\mathsf{a}''{\mapsto}\mathsf{bb}''] \dagger [\mathsf{a}'{\mapsto}\mathsf{bb}'',\mathsf{a}''{\mapsto}\mathsf{bb}'] = [\mathsf{a}{\mapsto}\mathsf{b},\mathsf{a}'{\mapsto}\mathsf{bb}'',\mathsf{a}''{\mapsto}\mathsf{bb}']$

462

$\cup{:}\ \mathsf{M} \times \mathsf{M} \to \mathsf{M}\ [\text{merge } \cup]$
$\quad [\mathsf{a}{\mapsto}\mathsf{b},\mathsf{a}'{\mapsto}\mathsf{bb}',\mathsf{a}''{\mapsto}\mathsf{bb}''] \cup [\mathsf{a}'''{\mapsto}\mathsf{bb}'''] = [\mathsf{a}{\mapsto}\mathsf{b},\mathsf{a}'{\mapsto}\mathsf{bb}',\mathsf{a}''{\mapsto}\mathsf{bb}'',\mathsf{a}'''{\mapsto}\mathsf{bb}''']$

$\backslash{:}\ \mathsf{M} \times \mathsf{A}\text{-}\mathbf{infset} \to \mathsf{M}\ [\text{restriction by}]$
$\quad [\mathsf{a}{\mapsto}\mathsf{b},\mathsf{a}'{\mapsto}\mathsf{bb}',\mathsf{a}''{\mapsto}\mathsf{bb}'']\backslash\{\mathsf{a}\} = [\mathsf{a}'{\mapsto}\mathsf{bb}',\mathsf{a}''{\mapsto}\mathsf{bb}'']$

$/{:}\ \mathsf{M} \times \mathsf{A}\text{-}\mathbf{infset} \to \mathsf{M}\ [\text{restriction to}]$
$\quad [\mathsf{a}{\mapsto}\mathsf{b},\mathsf{a}'{\mapsto}\mathsf{bb}',\mathsf{a}''{\mapsto}\mathsf{bb}'']/\{\mathsf{a}',\mathsf{a}''\} = [\mathsf{a}'{\mapsto}\mathsf{bb}',\mathsf{a}''{\mapsto}\mathsf{bb}'']$

$=,\neq{:}\ \mathsf{M} \times \mathsf{M} \to \mathbf{Bool}$

$^\circ{:}\ (\mathsf{A} \underset{m}{\overrightarrow{\ }} \mathsf{B}) \times (\mathsf{B} \underset{m}{\overrightarrow{\ }} \mathsf{C}) \to (\mathsf{A} \underset{m}{\overrightarrow{\ }} \mathsf{C})\ [\text{composition}]$
$\quad [\mathsf{a}{\mapsto}\mathsf{b},\mathsf{a}'{\mapsto}\mathsf{bb}'] \circ [\mathsf{bb}{\mapsto}\mathsf{c},\mathsf{bb}'{\mapsto}\mathsf{c}',\mathsf{bb}''{\mapsto}\mathsf{c}''] = [\mathsf{a}{\mapsto}\mathsf{c},\mathsf{a}'{\mapsto}\mathsf{c}']$

Map Operation Explication

- $m(a)$: Application gives the element that a maps to in the map m.
- **dom**: Domain/Definition Set gives the set of values which *maps to* in a map.
- **rng**: Range/Image Set gives the set of values which *are mapped to* in a map.
- †: Override/Extend. When applied to two operand maps, it gives the map which is like an override of the left operand map by all or some "pairings" of the right operand map.

- ∪: Merge. When applied to two operand maps, it gives a merge of these maps.
- \: Restriction. When applied to two operand maps, it gives the map which is a restriction of the left operand map to the elements that are not in the right operand set. 463
- /: Restriction. When applied to two operand maps, it gives the map which is a restriction of the left operand map to the elements of the right operand set.
- =: The equal operator expresses that the two operand maps are identical.
- ≠: The nonequal operator expresses that the two operand maps are *not* identical.
- °: Composition. When applied to two operand maps, it gives the map from definition set elements of the left operand map, m_1, to the range elements of the right operand map, m_2, such that if a is in the definition set of m_1 and maps into b, and if b is in the definition set of m_2 and maps into c, then a, in the composition, maps into c.

Map Operation Redefinitions The map operations can also be defined as follows:

```
value
  rng m ≡ { m(a) | a:A • a ∈ dom m }

  m1 † m2 ≡
    [ a↦b | a:A,b:B •
      a ∈ dom m1 \ dom m2 ∧ bb=m1(a) ∨ a ∈ dom m2 ∧ bb=m2(a) ]

  m1 ∪ m2 ≡ [ a↦b | a:A,b:B •
               a ∈ dom m1 ∧ bb=m1(a) ∨ a ∈ dom m2 ∧ bb=m2(a) ]

  m \ s ≡ [ a↦m(a) | a:A • a ∈ dom m \ s ]
  m / s ≡ [ a↦m(a) | a:A • a ∈ dom m ∩ s ]

  m1 = m2 ≡
    dom m1 = dom m2 ∧ ∀ a:A • a ∈ dom m1 ⇒ m1(a) = m2(a)
  m1 ≠ m2 ≡ ∼(m1 = m2)

  m°n ≡
    [ a↦c | a:A,c:C • a ∈ dom m ∧ c = n(m(a)) ]
    pre rng m ⊆ dom n
```

A.6 λ-Calculus + Functions

A.6.1 The λ-Calculus Syntax

```
type /* A BNF Syntax: */
  ⟨L⟩ ::= ⟨V⟩ | ⟨F⟩ | ⟨A⟩ | ( ⟨A⟩ )
```

```
    ⟨V⟩ ::= /∗ variables, i.e. identifiers ∗/
    ⟨F⟩ ::= λ⟨V⟩ • ⟨L⟩
    ⟨A⟩ ::= ( ⟨L⟩⟨L⟩ )
value /∗ Examples ∗/
    ⟨L⟩: e, f, a, ...
    ⟨V⟩: x, ...
    ⟨F⟩: λ x • e, ...
    ⟨A⟩: f a, (f a), f(a), (f)(a), ...
```

A.6.2 Free and Bound Variables 464

Let x, y be variable names and e, f be λ-expressions.

- ⟨V⟩: Variable x is free in x.
- ⟨F⟩: x is free in $\lambda y \bullet e$ if $x \neq y$ and x is free in e.
- ⟨A⟩: x is free in $f(e)$ if it is free in either f or e (i.e., also in both).

A.6.3 Substitution 465

In RSL, the following rules for substitution apply:

- **subst**([N/x]x) ≡ N;
- **subst**([N/x]a) ≡ a,

  for all variables a≠ x;
- **subst**([N/x](P Q)) ≡ (**subst**([N/x]P) **subst**([N/x]Q));
- **subst**([N/x]($\lambda x \bullet P$)) ≡ λ y•P;
- **subst**([N/x](λ y•P)) ≡ $\lambda y \bullet$ **subst**([N/x]P),

  if x≠y and y is not free in N or x is not free in P;
- **subst**([N/x](λy•P)) ≡ λz•**subst**([N/z]**subst**([z/y]P)),

  if y≠x and y is free in N and x is free in P
  (where z is not free in (N P)).

A.6.4 α-Renaming and β-Reduction 466

- α-renaming: λx•M

  If x, y are distinct variables then replacing x by y in λx•M results in λy•**subst**([y/x]M). We can rename the formal parameter of a λ-function expression provided that no free variables of its body M thereby become bound.

- β-reduction: (λx•M)(N)

  All free occurrences of x in M are replaced by the expression N provided that no free variables of N thereby become bound in the result. (λx•M)(N) ≡ **subst**([N/x]M)

A.6.5 Function Signatures 467

For sorts we may want to postulate some functions:

```
type
   A, B, C
value
   obs_B: A → B,
   obs_C: A → C,
   gen_A: BB×C → A
```

A.6.6 Function Definitions 468

Functions can be defined explicitly:

```
value
   f: Arguments → Result
   f(args) ≡ DValueExpr

   g: Arguments →~ Result
   g(args) ≡ ValueAndStateChangeClause
   pre P(args)
```

Or functions can be defined implicitly: 469

```
value
   f: Arguments → Result
   f(args) as result
   post P1(args,result)

   g: Arguments →~ Result
   g(args) as result
   pre P2(args)
   post P3(args,result)
```

The symbol $\stackrel{\sim}{\rightarrow}$ indicates that the function is partial and thus not defined for all arguments. Partial functions should be assisted by preconditions stating the criteria for arguments to be meaningful to the function.

A.7 Other Applicative Expressions 470

A.7.1 Simple let Expressions

Simple (i.e., nonrecursive) **let** expressions:

$$\textbf{let } \mathsf{a} = \mathcal{E}_d \textbf{ in } \mathcal{E}_b(\mathsf{a}) \textbf{ end}$$

is an "expanded" form of:

$$(\lambda \mathsf{a}.\mathcal{E}_b(\mathsf{a}))(\mathcal{E}_d)$$

A.7.2 Recursive let Expressions

Recursive **let** expressions are written as:

$$\textbf{let } \mathsf{f} = \lambda \mathsf{a}:\mathsf{A} \bullet \mathsf{E}(\mathsf{f}) \textbf{ in } \mathsf{B}(\mathsf{f},\mathsf{a}) \textbf{ end}$$

is "the same" as:

$$\textbf{let } \mathsf{f} = \textbf{Y}\mathsf{F} \textbf{ in } \mathsf{B}(\mathsf{f},\mathsf{a}) \textbf{ end}$$

where:

$$\mathsf{F} \equiv \lambda \mathsf{g}\bullet\lambda \mathsf{a}\bullet(\mathsf{E}(\mathsf{g})) \text{ and } \mathsf{YF} = \mathsf{F}(\mathsf{YF})$$

A.7.3 Predicative let Expressions

Predicative **let** expressions:

$$\textbf{let } \mathsf{a}:\mathsf{A} \bullet \mathcal{P}(\mathsf{a}) \textbf{ in } \mathcal{B}(\mathsf{a}) \textbf{ end}$$

express the selection of a value a of type A which satisfies a predicate $\mathcal{P}(\mathsf{a})$ for evaluation in the body $\mathcal{B}(\mathsf{a})$.

A.7.4 Pattern and "Wild Card" let Expressions

Patterns and *wild cards* can be used:

let {a} ∪ s = set **in** ... **end**
let {a,_} ∪ s = set **in** ... **end**

let (a,b,...,c) = cart **in** ... **end**
let (a,_,...,c) = cart **in** ... **end**

```
let ⟨a⟩^ℓ = list in ... end
let ⟨a,_,bb⟩^ℓ = list in ... end

let [a↦bb] ∪ m = map in ... end
let [a↦b,_] ∪ m = map in ... end
```

A.7.5 Conditionals

Various kinds of conditional expressions are offered by RSL:

```
if b_expr then c_expr else a_expr
end

if b_expr then c_expr end ≡ /* same as: */
   if b_expr then c_expr else skip end

if b_expr_1 then c_expr_1
elsif b_expr_2 then c_expr_2
elsif b_expr_3 then c_expr_3
...
elsif b_expr_n then c_expr_n end

case expr of
   choice_pattern_1 → expr_1,
   choice_pattern_2 → expr_2,
   ...
   choice_pattern_n_or_wild_card → expr_n
end
```

A.7.6 Operator/Operand Expressions

```
⟨Expr⟩ ::=
          ⟨Prefix_Op⟩ ⟨Expr⟩
        | ⟨Expr⟩ ⟨Infix_Op⟩ ⟨Expr⟩
        | ⟨Expr⟩ ⟨Suffix_Op⟩
        | ...
⟨Prefix_Op⟩ ::=
          − | ∼ | ∪ | ∩ | card | len | inds | elems | hd | tl | dom | rng
⟨Infix_Op⟩ ::=
          = | ≠ | ≡ | + | − | ∗ | ↑ | / | < | ≤ | ≥ | > | ∧ | ∨ | ⇒
        | ∈ | ∉ | ∪ | ∩ | \ | ⊂ | ⊆ | ⊇ | ⊃ | ^ | † | °
⟨Suffix_Op⟩ ::= !
```

A.8 Imperative Constructs

A.8.1 Statements and State Changes

Often, following the RAISE method, software development starts with highly abstract-applicative constructs which, through stages of refinements, are turned into concrete and imperative constructs. Imperative constructs are thus inevitable in RSL.

```
   Unit
value
   stmt: Unit → Unit
   stmt()
```

- Statements accept no arguments.
- Statement execution changes the state (of declared variables).
- **Unit** → **Unit** designates a function from states to states.
- Statements, stmt, denote state-to-state changing functions.
- Writing () as "only" arguments to a function "means" that () is an argument of type **Unit**.

A.8.2 Variables and Assignment

0. **variable** v:Type := expression
1. v := expr

A.8.3 Statement Sequences and skip

Sequencing is expressed using the ';' operator. **skip** is the empty statement having no value or side-effect.

2. **skip**
3. stm_1;stm_2;...;stm_n

A.8.4 Imperative Conditionals

4. **if** expr **then** stm_c **else** stm_a **end**
5. **case** e **of**: p_1→S_1(p_1),...,p_n→S_n(p_n) **end**

A.8.5 Iterative Conditionals

6. **while** expr **do** stm **end**
7. **do** stmt **until** expr **end**

A.8.6 Iterative Sequencing

8. **for** e **in** list_expr • P(b) **do** S(b) **end**

A.9 Process Constructs

A.9.1 Process Channels

Let A and B stand for two types of (channel) messages and i:KIdx for channel array indexes, then:

channel c:A
channel { k[i]:B • i:KIdx }

declare a channel, c, and a set (an array) of channels, k[i], capable of communicating values of the designated types (A and B).

A.9.2 Process Composition

Let P and Q stand for names of process functions, i.e., of functions which express willingness to engage in input and/or output events, thereby communicating over declared channels. Let P() and Q stand for process expressions, then:

P ∥ Q	Parallel composition
P ⌷ Q	Nondeterministic external choice (either/or)
P ⊓ Q	Nondeterministic internal choice (either/or)
P ⫵ Q	Interlock parallel composition

express the parallel (∥) of two processes, or the nondeterministic choice between two processes: either external (⌷) or internal (⊓). The interlock (⫵) composition expresses that the two processes are forced to communicate only with one another, until one of them terminates.

A.9.3 Input/Output Events

Let c, k[i] and e designate channels of type A and B, then:

c ?, k[i] ?	Input
c ! e, k[i] ! e	Output

expresses the willingness of a process to engage in an event that "reads" an input, respectively "writes" an output.

A.9.4 Process Definitions

The below signatures are just examples. They emphasise that process functions must somehow express, in their signature, via which channels they wish to engage in input and output events.

```
value
   P: Unit → in c out k[i]
   Unit
   Q: i:KIdx →  out c in k[i] Unit

   P() ≡ ... c ? ... k[i] ! e ...
   Q(i) ≡ ... k[i] ? ... c ! e ...
```

The process function definitions (i.e., their bodies) express possible events.

A.10 Simple RSL Specifications

Often, we do not want to encapsulate small specifications in schemes, classes, and objects, as is often done in RSL. An RSL specification is simply a sequence of one or more types, values (including functions), variables, channels and axioms:

```
type
   ...
variable
   ...
channel
   ...
value
   ...
axiom
   ...
```

In practice a full specification repeats the above listings many times, once for each "module" (i.e., aspect, facet, view) of specification. Each of these modules may be "wrapped" into scheme, class or object definitions.[76]

A.11 RSL Index

[76] For schemes, classes and objects we refer to [137, Chap. 10]

B RSL$^+$ 471

TO BE WRITTEN

C A Language of Domain Analysis & Description Prompts 472

TO BE WRITTEN

D A Description Narration Language 473

TO BE WRITTEN

E Indexes

474

E.1 Philosophy Index

E.2 Domain Analysis Index

E.2.1 Concepts

E.2.2 Definitions

E.2.3 Analysis Predicates

E.2.4 Description Observers

E.2.5 *Proof Obligations and Axioms*

E.2.6 Observer Function Literals