Domain Analysis & Description
A Philosophy Basis

Dines Bjørner
Technical University of Denmark,Denmark

August 28, 2019: 15:43

1

Abstract

This paper consists of two parts: A philosophy part and a terse summary of my April 2019 `ACM Trans. on Software Engineering and Methodology` paper on *Domain Analysis & Description*.

In the philosophy part, Sect. 3, we outline Kai Sørlander's philosophy on what must necessarily be in any description of any world.

In the domain analysis & description part, Sects. 4–9, we present a new preamble for software engineering, one that precedes requirements engineering. We outline two calculi: one for the analysis of the endurants of human artefact "dominated" domains, and one for their description. By a transcendental deduction endurant domain descriptions are translated into perdurant domain descriptions: manifest parts becoming behaviours.

We show how the ontology of the second part is basically founded on the necessities of description outlined in the first part — thereby contributing to a philosophy basis for computing.

1 Introduction

2

Before **software** can be **designed** the programmer must **grasp** its requirements. Before **requirements** are **prescribed** the engineer must **grasp** an **adequate extent** of the **domain** in which that software is to serve. **But do software engineers today have a sufficient grasp of their target domains?** 3

By *an adequate* (domain) we do not mean "an entire, full, complete" domain, but a sizable part of it — usually "somewhat more" than entailed by [subsequent] requirements. By *a grasp* (of a domain) we mean an understanding that enables us to **reason** about the domain. Our 'reasonable grasp' is, we suggest, manifested in some text; that is, in some language. 4

In order to *reason* we we expect that language to be **formal**; that is, to have a formal **syntax**, a formal, mathematical **semantics** and a **proof system**. We no not expect the **domain**, the **requirements** and the **software design** [incl. coding] languages to be the same, one **specification** language. 5

Ideally software development, therefore, to us, entails three major phases: **domain engineering** in which we analyze and describe the domain, $\mathscr{D}$, in which the software is to serve; **requirements engineering** in which we "derive" the requirements, $\mathscr{R}$, that the software is to fulfill; and **software design** in which we finally arrive, via one or more steps of software design, $\mathscr{S}$. 6

Domain descriptions express **properties** of the chosen domain. **Requirements prescriptions** express **desired properties** of the **desired software**, **not** how it is implemented. **Software designs** express how requirements are to be **satisfied** by **executable code**. 7

Software, $\mathscr{S}$, is expected to fulfill customer *expectations*, hence must be based on *domain understanding*, $\mathscr{D}$, and to be *correct* with respect to requirements, $\mathscr{R}$. We can express this formally: $\mathscr{R} \models \mathscr{D}$ ($\mathscr{R}$ models $\mathscr{D}$), respectively $\mathscr{D}, \mathscr{S} \models \mathscr{R}$ ($\mathscr{S}$, in the context of $\mathscr{D}$, models $\mathscr{R}$). This obviously means that we

expect developers to consider domain, requirements and software specifications (incl. code) as mathematical objects and that tests, model checks and theorem proofs ideally be carried out accordingly.

The above portrays a **hypothetical situation.** Today's software engineering essentially has no domain engineering phase. This current situation is unlike any other engineering. The classical disciplines of chemical, civil, electrical and mechanical engineering all build on the sciences of physics. Major software systems are today developed without a proper understanding of their underlying domain. It seems that software engineering today starts, "at best", with requirements engineering.

The present paper, as well as its precedent papers [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14], lead us to claim that domain science & engineering offers a compelling *new foundation for software development.* Physics is the basis for all branches classical of engineering. So, we claim, [that] domain science, is a basis for all software. We shall, however, cover only a fragment of this basis.

2 Domain Science & Engineering

DEF.1: By a **domain** we shall understand a **rationally describable** segment of a **human assisted** reality. That is, of the world, its **physical parts: natural** ["God-given"] and **artefactual** ["man-made"], and **living species**: **plants** and **animals** including, notably, **humans** ■ DEF.2: By *rationally describable* we mean describable using, for example, the principles, techniques and analysis & description prompts of this paper and detailed in [12] ■ DEF.3: By *human assisted reality* we mean a universe of discourse with at least one artefactual phenomenon and as monitored & controlled by at least one human ■ We present an essence of two calculi: a *calculus* of **domain analysis prompts** and a *calculus* of **domain description prompts**. We shall only present the *prompts*, not their algebra, that is, not the laws of combined uses of prompts. DEF.4: By a **prompt** we shall here mean an act of encouraging the domain analyser cum describer, that is, a human, to do something, here: to analyse and/or describe ■ DEF.5: By **domain analysis** we mean an *inquiry*, by domain analysers, i.e., humans, into the *make-up* of a domain, with the analysis resulting in affirmative answers, to questions like *"is, what I am observing, such-and-such"* – **true** or **false** ■ DEF.6: By a **domain description** we mean a textual document, both informal, the narrative, and formal, the specification ■ The **narrative** is a natural language text which in terse statements introduces the names of the domain, and, possibly, also the definitions, of sorts (types) of syntactic and semantic entities, actions, events and behaviours, and axioms; not anthropomorphically, but by emphasizing their properties. The **formal specification** is a collection of sort, or type definitions, function and behaviour definitions, together with axioms and proof obligations constraining the definitions. So the problem is to analyse and describe a domain, that is, to describe physical parts, whether natural or man-made, and, in [rare] cases also living species: plants and animals, notably humans. The next many sections shows how we tackle, and hence expect others to tackle that problem. The approach takes its departure in **philosophy**. Most decisively in the philosophy of **Kai Sørlander**.

3 Some Issues of Philosophy

The question is: *"What must inavoidably be in any description ?"* We take it as the necessary characteristics of any domain is equivalent with the conceptual, logical conditions for any possible description of that domain. Sørlander, after careful studies [24, 25, 27, 31] of all possibly relevant philosophers' work, puts forward the thesis of *the possibility of truth* and then basing *transcendental deductions* on indisputable logical relations to arrive at the conceptual, logical conditions for any possible description of that domain. The starting point, now, in a series of deductions, is that of logic and that we can *assert* a property, $\mathscr{P}$, and its negation $\neg\mathscr{P}$; and that *these two assertions cannot both be true*, that is, that $\mathscr{P} \wedge \neg\mathscr{P}$ cannot be true. So the *possibility of truth* is a universally valid condition. When we claim that, we also claim *the contradiction principle*. The *implicit meaning theory* is this: *"in assertions there are mutual dependencies between the meaning of designations and consistency relation between assertions"*. When we claim that a philosophy basis is that of the possibility of truth, then we assume that this basis include the contradiction principle

and the implicit meaning theory. We shall also refer to the implicit meaning theory as the **inescapable meaning assignment**. 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 endurants and operations). 22 23

An Inescapable Meaning Assignment Example, Narrative

The meaning of designations:

1 Stacks, s:S, have elements, e:E;
2 the empty_S operation takes no arguments and yields a result stack;
3 the is_empty_S operation takes an argument stack and yields a Boolean value result.
4 the stack operation takes two arguments: an element and a stack and yields a result stack.
5 the unstack operation takes an non-empty argument stack and yields a stack result.
6 the top operation takes an non-empty argument stack and yields an element result.

The consistency relations:

7 an empty_S stack is_empty, and a stack with at least one element is not;
8 unstacking an argument stack, stack(e,s), results in the stack s; and
9 inquiring as to the top of a non-empty argument stack, stack(e,s), yields e.

The Inescapable Meaning Assignment Example, Formalisation 24

The meaning of designations:

type
1. E, S
value
2. empty_S: **Unit** → S
3. is_empty_S: S → **Bool**
4. stack: E × S → S
5. unstack: S $\xrightarrow{\sim}$ S
6. top: S $\xrightarrow{\sim}$ E

The consistency relations:

7. is_empty(empty_S()) = **true**
7. is_empty(stack(e,s)) = **false**
8. unstack(stack(e,s)) = s
9. top(stack(e,s)) = e

3.1 Logical Connectives 25

3.1.1 Negation: ¬

The logical connective, **negation** (¬), is defined as follows: if assertion $\mathscr{P}$ holds then assertion $\neg\mathscr{P}$ does not hold – the contradiction principle understood as a definition of the concept of negation.

3.1.2 Conjunction and Disjunction: ∧ and ∨

26

Assertion $\mathscr{P} \wedge \mathscr{Q}$ holds, i.e., is true, if both $\mathscr{P}$ and $\mathscr{Q}$ holds.

Assertion $\mathscr{P} \vee \mathscr{Q}$ holds, i.e., is true, if either $\mathscr{P}$ or $\mathscr{Q}$ or both $\mathscr{P}$ and $\mathscr{Q}$ holds.

3.1.3 Implication: ⇒

27

Assertion $\mathscr{P} \Rightarrow \mathscr{Q}$ holds, i.e., is true, if the first assertion, $\mathscr{P}$, holds, tt, and the second assertion, $\mathscr{Q}$, is not false, $\neg ff.[(\mathscr{P},\mathscr{Q}),\mathscr{P} \Rightarrow \mathscr{Q}]$: $[(tt,tt),tt]$, $[(tt,ff),ff]$, $[(ff,tt),ff]$, and $[(ff,ff),tt]$.

3.2 Transcendence

28

DEF.7: By **transcendental** *"we shall understand the philosophical notion: the a priori or intuitive basis of knowledge, independent of experience"* ■ DEF.8: By a **transcendental deduction** *"we shall understand the philosophical notion: a transcendental 'conversion' of one kind of knowledge into a seemingly different kind of knowledge"* ■ We shall take what can be expressed in logical propositions and predicates as "one kind of knowledge". A keyword, in the above definition, is "seemingly". We refer to Sects. 3.3.6-3.3.8.

3.3 Towards a Philosophy Basis for Physics and Biology

30

In a somewhat long series of deductions we shall, based on Sørlander's Philosophy motivate the laws of Newton and more, not on the basis of empirical observations, but on the basis of transcendental deductions based on rational reasoning.

3.3.1 Possibility and Necessity

31

Based on logical implication we can transcendentally define the two *modal operators:* **possibility** and

32 **necessity**. DEF.9: An assertion is **necessarily** true if its truth follows from follows from the definition of the designations by means of which it is expressed ■ DEF.10: An assertion is **possibly** true if its negation is not necessary ■

3.3.2 Empirical Assertions

33

There can be assertions whose truth value does not only depend on the definition of the designations by means of which they are expressed. Those are assertions whose truth value does not follow only from the definition of designations but depend also on the assertions *referring* to something that exists independently of the designations by means of which they are expressed. We shall call such assertions *empirical.*

3.3.3 Existence

34

With Sørlander we shall now argue that there exist many entities in any world: [31, pp 145] *"Entities, in a first step of reasoning, that can be referred to in empirical assertions do not necessarily exist. It is, however,*

35 *an empirical fact that they do exist; hence there is a logical necessity that they do not exist. In a second step of reasoning, these entities must exist as a necessary condition for their actually being ascribed the predicates which they must necessarily befit in their capacity of of being entities referred to in empirical assertions."*

3.3.4 Identity, Difference and Relations

36

[31, pp 146] *"An entity, referred to by A, is* **identical** *to an entity, referred to by B, if A cannot be ascribed a predicate, in-commensurable with a predicate ascribed to B."* That is, if A and B cannot be

37 ascribed in-commensurable predicates. [31, pp 146] *"Entities A and B are* **different** *if they can be ascribed in-commensurable predicates."* [31, pp 147] *"***Identity** *and* **difference** *are thus transcendentally derived through these formal definitions and must therefore be presupposed in any description of any domain and*

38 *must be expressible in any language."* Identity and difference are **relations**. [31, pp 147] *"As a consequence identity and difference imply relations. ...* **Symmetry** *and* **asymmetry** *are also relations: A identical to B is the same as B identical to A. And A different from B is the same as B different from A. Finally* **transitivity** *follows from A identical to B and B identical to C implies A identical to C."*

3.3.5 Sets, Quantifiers and Numbers

39

We can, as a consequence of two or more different entities satisfying a same predicate, say P, *define*

40 the notion of the **set** of all those entities satisfying P. And, we can, as a consequence of two or more

entities, $e_i,...,e_j$, all being *distinct*, therefore implying in-commensurable predicates, $Q_i,...,Q_j$, but still satisfying a common predicate, P, claim that they all belong to a same set. The predicate P can be said to **type** that set. And so forth: following this line of reasoning we can introduce notions of cardinality of sets, finite and infinite sets, existential ($\exists$) and universal ($\forall$) quantifiers, etc.; and we can in this way transcendentally deduce the concept of (positive) numbers, their addition and multiplication; and that such are an indispensable aspect of any domain. We leave it then to mathematics to further study number theory.

3.3.6 Space and Geometry

DEF.11: **Space:** [31, 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"* ■ [31, pp 155] *"Entities, to which reference can be made in simple, empirical assertions, must exist in space; they must be spatial, i.e., have a certain extension in all directions; they must therefore "fill up some space", have surface and form."* From this, by further reasoning one can develop notions of points, line, surface, etc., i.e., Euclidean as well as non-Euclidean **geometry**.

3.3.7 States

We introduce a notion of **state**. [31, pp 158–159] *"Entities may be ascribed predicates which it is not logically necessary that they are ascribed. How can that be possible ? Only if we accept that entities may be ascribed predicates which are in-commensurable with predicates that they are actually ascribed."* That is possible, we must conclude, if entities can **exist** in distinct **states**. We shall let this notion of state further undefined – till Sect. 6.3.4.

3.3.8 Time and Causality

DEF.12: **Time:** [31, pp 159] *"Two 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"* ■

•••

So space and time are not phenomena, i.e., are not entities. They are, by transcendental reasoning, aspects of any possible world, hence, of any description of any domain.

In a concentrated series [31, 160-163], of logical reasoning and transcendental deductions, Sørlander, introduce the concepts of the empirical circumstances under which entities exist, implying *non-logical implication* between one-and-the-same entity at distinct times, leading to the notions of **causal effect** and **causal implication** – all deduced transcendentally. Whereas Kant's *causal implication* is transcendentally deduced as necessary for the *possibility of self-awareness*. Sørlander's *causal implication* does not assume *possibility of self-awareness*. The **principle of causality** is a necessary condition for assertions being about the same entity at different times.

3.3.9 Kinematics

[31, pp 164] *"Entities are in both space and time; therefore it must be assumed that they can change their spatial properties; that is, are subject to* **movement**. *An entity which changes location is said to* **move**. *An entity which does not change location is said to be* **at rest**." In this way [31] transcendentally introduces the notions of *velocity* and *acceleration*, hence *kinematics*.

3.3.10 Dynamics: Newton's Laws

[31, pp 166] *"When combining the causality principle with dynamics we deduce that when an entity changes its state of movement then there must be a cause, and we call that cause a* **force***."* [31, pp 166] *"The* ***change*** *of state of entity movement must be* ***proportional*** *to the applied force; an entity not subject to an external force will remain in its state of movement: This is* **Newton's 1st Law.***"* [31, pp 166] *"But to change an entity's state of movement by some force must imply that the entity exerts a certain* ***resistance*** *to that change; the entity must have a* **mass***. Changes in an entity's state of movement besides being proportional to the external force, must be* ***inverse proportional*** *to its mass. This is* **Newton's 2nd Law.***"* [31, pp 166-167] *"The forces that act upon entities must have as source other entities: entities may* ***collide;*** *and when they collide* ***the forces*** *they exert on each other must be* ***the same but with opposite directions.*** *This is* **Newton's 3rd Law.***"* [31, pp 167-168] *"How can entities be the source of forces ? How can they have a mass ? Transcendentally it must follow from* ***gravitational pull****. Across all entities of mass, there is a mutual attraction,* **universal gravitation***."* [31, pp 168-169] *"Gravitation must, since it has its origin in the individual entities, propagate with a definite velocity; and that velocity must have a limit, a constant of nature, the* ***universal speed limit****."*

3.3.11 From Philosophy to Physics

Based on logical reasoning and transcendental deductions one can thus derive major aspects of that which be (assumed to be) in any description of any world, i.e., domain. In our domain description ontology we shall let the notions of **natural parts** and **continuous endurants** (non-solids) cover what we have covered so far: they are those entities which satisfy the laws of physics, hence are in space and time. In the next sections we shall make further use of Sørlander's Philosophy to logically and transcendentally justify the inevitability of **living species: plants** and **animals** including, notably, **humans**, in any description of any domain.

3.3.12 Purpose, Life and Evolution

[31, pp 174] *"For language and meaning to be possible there must exist entities that are not constrained to just the laws of physics. This is possible if such entities are further subject to a* **"purpose-causality"** *directed at the future. These entities must strive to maintain their own existence."* We shall call such entities **living species**. Living species must maintain and also further develop their form and do so by an exchange of materials with the surroundings, i.e., **metabolism**, with one kind of living species subject only to development, form and **metabolism**, while another kind additionally **move purposefully**, The first we call **plants**, the second **animals**. Animals, consistent with the principle of causality, must possess **sensory organs**, a **motion apparatus**, and **instincts, feelings, promptings** so that what has been sensed, may be responded to [through motion]. The **purpose-directness** of animals must be built into the animals. Biology shows that that is the case. The animal genomes appear to serve the **purpose-directness** of animals. [31, pp 178] *"Biology shows that it is so; transcendental deduction that it must be so."*

3.3.13 Awareness, Learning and Language

[31, pp 180] "
Animals, to **learn** *from* **experience***, must be able to feel* **inclination** *and* **disinclination***, and must be able to remember that it has acted in some way leading to either the feeling of inclination or disinclination. As a consequence, an animal, if when acting in response to sense impression, ι, experiences the positive feeling of inclination (desire), then it will respond likewise when again receiving sense impression ι, until it is no longer so inclined. If, in contrast, the animal feels the negative feeling of disinclination (dislike), upon sense impression ι, then it will avoid responding in this manner when receiving sense impression ι."* [31, pp 181] *"Awareness is built up from the sense impressions and feelings on the basis of, i.e., from what the individual animal has learned. Different animals can be expected to have different levels of*

consciousness; and different levels of consciousness assume different biological bases for learning. This is possible, biology tells us, because of there being a central nervous system with building blocks, the neurons, having an inner determination for learning and consciousness." [31, pp 181–182] *"In the mutual interaction between animals of a higher order of consciousness these animals learn to use* **signs** *developing increasingly complex sign systems, eventually "arriving" at* **languages***."* It is thus we single out **humans**. [31, pp 183] *"Any human language which can describe reality, must assume the full set of concepts that are prerequisites for any world description."*

INTERLUDE

We have concluded the presentation of a major issue of this paper. that of a philosophy that may be a possible basis for domain science & engineering. We now *"apply"* this, Kai Sørlander's, Philosophy to the problem of domain analysis & description.

4 Phenomena and Entities

𝔻𝔼𝔽.13: By an entity, is_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* [20, Vol. I, pg. 665] ■ An entity is what we can analyse and describe using the analysis & description prompts outlined in this paper. Many of the entities that we are concerned with are those with which Kai Sørlanders Philosophy is likewise concerned. They are the ones that are unavoidable in any description of any possible world.

•••

Before main Sects. 5–9, we introduce two categories of entities: endurants and perdurants.

4.1 Endurants

𝔻𝔼𝔽.14: 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 [20, Vol. I, pg. 656]. Were we to "freeze" time we would still be able to observe the entire endurant* ■ Endurants, thus, are capable of enduring. 𝔼𝕩.1. **Endurants:** *A train wagon, a rail track* and *a railway station* ■ We suggest that the concept of endurants can be seen as a transcendental deduction based on the inescapable fact that there is a multitude of entities, cf. Sect. 3.3.5, and that considering these as existing in just space, are the endurants. But note that endurants are [to be] *observed.*

4.2 Perdurants

𝔻𝔼𝔽.15: 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. 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 [20, Vol. II, pg. 1552]* ■ Perdurants are entities that only exists partially at any given point in time. 𝔼𝕩.2. **Perdurant:** *a train ride* ■ We suggest that the concept of perdurants can be seen as a transcendental deduction based on the inescapable fact that there are a multitude of entities, cf. Sect. 3.3.5, and that considering these as existing in both space and time, are the perdurants.

•••

Sections 5–7 and Sect. 9 reviews the complex, conceptual "universes" of endurants, respectively perdurants. Section 8 unveils, by a transcendental deduction, the link between endurants and perdurants: that endurant parts transcend into behaviours. Figure 1 suggests a structuring of endurants, perdurants and their relations.

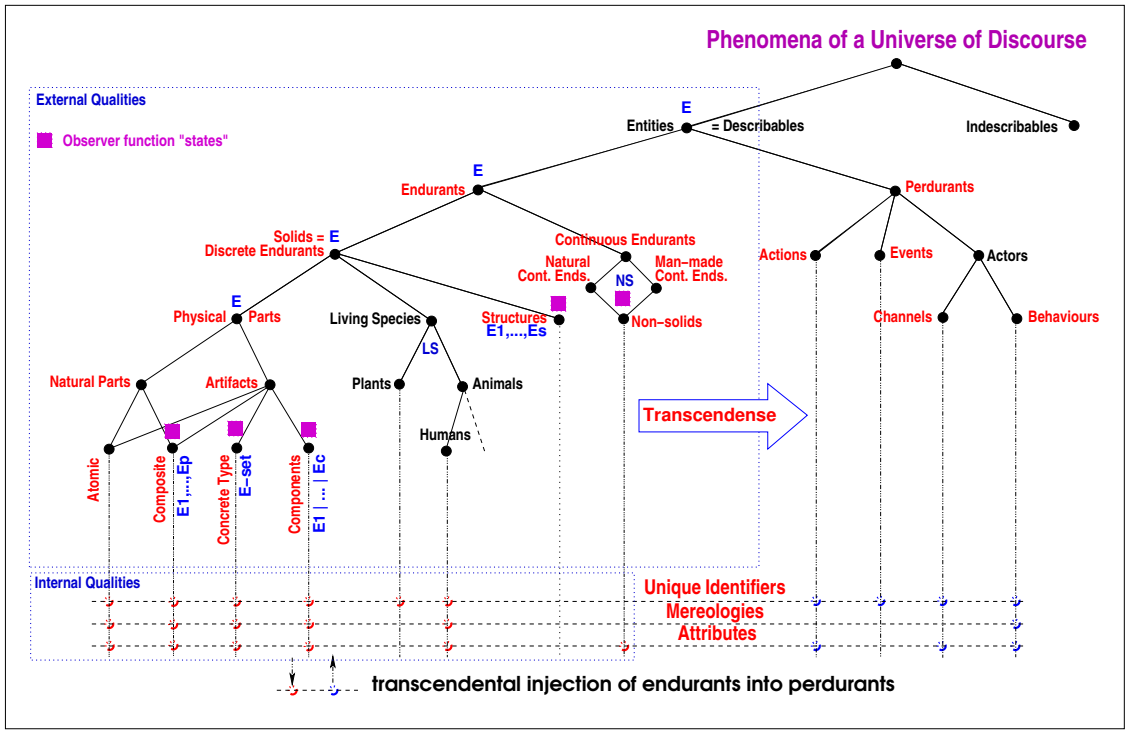

Figure 1: An Upper Ontology for Domain Entities. We shall not discuss black labeled entity classes. The magenta coloured square boxes, ■, designate "analysis states: where respective description prompts apply.

5 Endurants: Analysis of External Qualities

Observable qualities of endurants are those that can be touched. Endurants are either discrete or continuous, that is, solid, respectively non-solid. We choose, it may seem arbitrarily, to analyse endurants into either **discrete** (solid) or **continuous** (non-solid) endurants. That is, we claim that endurants can be so analysed either of one kind, or of the other, but not both ! We justify the distinctions based on physics.

5.1 Discrete and Continuous Endurants

DEF.16: By a **discrete endurant** (a solid) we shall understand an endurant which is **separate**, **individual** or **distinct** in *form or concept* [20, OED] ■ EX.3. **Discrete Endurants:** *a canal lock, a canal link between two adjacent locks, a barge* ■ DEF.17: By a **continuous endurant** (a non-solid) we shall understand an endurant which is **prolonged**, without **interruption**, in an **unbroken series** or **pattern** [20, OED] ■ We think of a non-solid to be either a gas or a plasma or a liquid. EX.4. **Continuous Endurant:** *water* ■ Note, and please accept, the OED definitions. They are not precise in the sense of mathematics. We are not dealing with an exact 'world'. We are dealing with real worlds.

5.2 Solids 87

Solids, i.e., discrete endurants, are either physical parts, or living species, or structures. We shall motivate the first two categories of solids on the background of Sølander's philosophy. Structures are motivated pragmatically.

5.2.1 Physical Parts and Living Species

88

DEF.18: By a **physical part** we shall understand a discrete endurant existing in time and subject to laws of physics, including the causality principle and gravitational pull – which are not living species or animals ■ EX.5. **Physical parts:** *A pipeline system,a pipeline,pipeline units: pipes, valves, pumps*, etc. ■ DEF.19: 89
By a **living species** we shall understand a discrete endurant, subject to laws of physics, and additionally 90
subject to causality of purpose ■ EX.6. **Living Species:** *a garden*, *a flower bed*, *a rhodendenron* ■ In this 91
paper we shall not elaborate on the possibility of natural versus man-made living species.

5.2.2 Natural Parts and Artefacts

92

DEF.20: By a **natural part** we shall understand physical parts, i.e., that are in space and time, are subject to the laws of physics, and also subject to the principle of causality and gravitational pull, but are not man made and not living species ■ EX.7. **Natural Parts:** *a landscape*, *a lake*, *a forest*, *a mountain* ■ 93
DEF.21: By an **artefact** we shall understand physical parts that are man made with one or more intents ■ 94
We shall explain the notion of *intent* later. EX.8. **Artefacts:** *a road network*, with *automobiles*, *hubs (a* 95
road intersections), *links (between two adjacent hubs)*, *routes (contiguous sequences of zero, one or more adjacent, alternating hubs and links)*.The intents are that *automobiles* drive along *routes* and that *hubs* and *links* serve as conduits for *automobiles* ■

5.2.3 Atomic or Composite Parts

96

DEF.22: By an **atomic part** we shall understand a part which, in a given context, deemed to **not** contain of 97
meaningful, separately observable proper sub-part. A sub-part is a part ■ EX.9. **Atomic Artefacts:** *hubs*, 98
links, *automobiles* ■ We shall not consider natural parts as other than that, neither atomic, nor composite 99
in this paper. DEF.23: By a **composite part** we shall mean physical parts which, in a given context, are 100
deemed to indeed consist of meaningful, separately observable proper sub-parts ■ EX.10. **Composite** 101
Artefacts: *a road net*, *a pipeline system*, *a railway system* ■ EX.11. **Elements of a Composite Artefact:** The domain of road transport is assumed to contain a road net which then contains a set of links, a set of hubs, a set of automobiles, a set of zero, one or more road maintenance departments, and a set of zero, one or more automobile clubs. I may contain other parts ■

5.2.4 Varieties of Artefacts

102

In addition to **atomic** and **composite** artefacts there are **concrete type** artefacts and **component** artefacts. The analysis into this variety of four kinds is based on pragmatic grounds. DEF.24: By a **concrete type artefact** we shall, simplifying, mean a set of endurants, all of the same sort ■ EX.12. **Concrete** 103
Type Artefacts: *a set of hubs*, *a set of links*, *a set of automobiles* ■ DEF.25: By a **component type** 104
artefact we shall understand a set of zero, one or more discrete endurants of the same kind which we, the domain analyser cum describer choose to not endow with mereology ■ EX.13. **Components:** *letters* (of 105
a mail box), *sand grains* (of a heap), *bricks* (of a stack) ■

5.2.5 Plants, Animals and Humans

106

Living species are either plants or animals. DEF.26: By a **plant, animal** and **human** we shall understand what Kari Sørlander's Philosophy transcendentally arrives at as such ■ We omit examples !

5.2.6 Structures

107

DEF.27: By a **structure** we shall understand a discrete endurant which the domain engineer chooses to describe as consisting of one or more endurants, whether discrete or continuous, but to not endow with
108 internal qualities: unique identifiers, mereology or attributes ■ Structures are "conceptual endurants". A structure "gathers" one or more endurants under "one umbrella", often simplifying a presentation of some elements of a domain description. Sometimes, in our domain modelling, we choose to model an endurant as a structure, sometimes as a physical part; it all depends on what we wish to focus on in our domain model. As such structures are "compounds" where we are interested only in the (external and internal) qualities of the elements of the compound, but not in the qualities of the structure itself.

5.3 Non-solids

109

We remind you of Sect. 5. En entity may thus be a non-solid. A composite part, p, natural or man-made, may have one or more non-solid entities, though with at least one solid entity — in which case has_non_solids(p).

5.4 The Analysis Prompts

110

We summarise the analysis prompts informally introduced in this section.

- is_entity,
- is_discrete,
- is_continuous,
- is_physical,
- is_living,
- is_structure,
- is_natural,
- is_artefact,
- is_atomic,
- is_composite,
- is_concrete,
- is_component,
- has_non_solids,
- has_components.

6 Endurants: Analysis of Internal Qualities

111

Internal qualities of endurants are those qualities that cannot be touched but can be either conceptualised or measured. We consider the following internal qualities: **unique identifiers**, **mereology**, and **attributes**.
112 Physical parts have the full set of internal qualities. Structures are endurants for which the domain analyser cum describer had decided to not endow with internal qualities. We shall not, in this paper, be concerned with the internal qualities of living species.

6.1 Unique Identifiers

113

It is based on the philosophy idea of *identity*, cf. Sect. 3.3.4, that we associate with each solid a unique identifier. Ex.14. **Road Net Links and Hubs:** The road net of a transport system consists of links, i.e., street segments, and hubs, i.e., street intersections. Links of a road not have unique identifiers. (Links of all road nets are distinctly identified.) Hubs of any road net are distinctly identified.

6.2 Mereology

114

DEF.28: Mereology is the study of parts and the wholes they form ■ Mereology, as here put forward, is due
115 to the Polish philosopher/logician/mathematician Stanisław Leśniewski [21, 15]. There are basically two relations that can be relevant for part-hood (i) a topological one, and (ii) a conceptual one. (i) Physically two or more parts may be adjacent to one another or one within within another. (ii) Conceptually some
116 parts, "relate" to a "therefrom physically distinct" part. Ex.15. **Topological Mereology:** The mereology of links of a road net is a set of two distinct hub identifiers of that net, and of hubs of a road net is a set of zero, one or more link identifiers of that net. The mereology thus defines a concept of routes of a road
117 net, and must be such that there is at least one route from any hub to any other hub of a road net ■ Ex.16.

Conceptual Mereology: The mereology of an automobile (of a road transport system) identifies the hubs and links that it may traverse and the zero, one or more automobile clubs it may be a member of. The mereology of a hub and a link, (of a road transport system) in addition to what has already been ascribed to hubs and links, identifies one road maintenance department ■ 118

We may model the mereology of a part, p, as a triplet: an input set of unique identifiers of parts from which p "receives input" in a sense not further described here; a pair of input/output sets of unique identifiers of parts from which p "receives input" and "delivers output" in a sense not further described here; and an output set of unique identifiers of parts to which p "delivers output" in a sense not further described here.

6.3 Attributes 119

Unique identifications and mereologies form abstract concepts. Although topological mereologies may be observed they, and unique identification, are not manifest – although they can be the quantities that are referred to in empirical assertions. Attributes are measurable properties of endurants, properties that can be referred to in empirical assertions — they, so-to-speak, gives *"flesh and blood"*, that is, substance to endurants. Endurants are typically recognised because of their spatial form and are otherwise characterised 120 by their intangible, but measurable attributes. We equate all endurants which, besides possible type of unique identifiers and possible type of mereologies, have the same types of attributes, with one sort. 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 non-entity) !

6.3.1 Attribute Categories

121

Attributes [19] are either of **static** value, i.e., does change value, or of **monitorable** value, i.e., **inert** or **reactive**: monitorable values can change, or of **controllable** value, i.e., **biddable** or **programmed**: biddable values can be prescribed, but prescription may fail, programmable values can be set. Ex.17. **Link Attributes:** Typical link attributes could be: location (e.g., as a Bézier curve) [static], length [static], road condition (icy, dry, ...) [monitored], state – as a set of pairs of adjacent hub identifiers [controllable], state spaces – as set of all such states [static], and automobile history: recordings of which automobiles have been on the link, at which position and time ■ Ex.18. **Hub Attributes:** Typical hub attributes could be: 122 123 location [static], state – as set of pairs of adjacent link identifiers [controllable], and automobile history: recordings of which automobiles have been on the hub, and at which time. States are abstractions of traffic signals ■ Ex.19. **Automobile Attributes:** Typical automobile attributes could be: position (on hub or 124 link), velocity, etc., road net history: recordings of the hubs and links on which the automobile has been, at which position and time ■

6.3.2 Artefact Intents

125

With artefacts we can associate **intents**. Def.29:By an **intent** of an artefact we shall understand a simple label which informally indicates the purpose for which the artefact is intended ■ An artefact may be ascribed more than one intent. Artefacts are usually ascribed at least one intent. Ex.20. **Intents of Automobiles and Road Nets:** To automobile we may ascribe the intent that they are located on the road net, i.e., on hubs and links; and to hubs and links we then ascribe the intent that they accommodate automobiles ■ 126

6.3.3 Intentional Pull

127

Gravitational pull, cf. Sect. 3.3.10, follows from Newton's Third Law. Intentional pull "follows" from the fact that pairs or triples, etc., of artefacts of different sorts may be ascribed commensurate intents. Ex.21. **Intentional Pull between Automobiles and Road Net:** If an automobile's road net history records 128

that is has visited a road net unit at time t and position π, then that road net unit's automobile history records that very same fact ! And vice versa. It cannot be otherwise ! ■

6.3.4 States

By a state we shall understand any collection of endurants for which any one endurant has at least one dynamic, i.e., non-static, attribute. By the state of a behaviour we shall understand its current *program point*, that is, its point of execution, and the collection of its monitorable and controllable variables, that is, of their current values.

7 Endurants: Description Prompts

So far we have outlined a number of domain analysis prompts, cf. Sect. 5.4. We now summarise some description prompts. We refer to Fig. 1. The "analysis states" marked with magenta colored square boxes, ■ , correspond, left-to-right in the ontology graph to the following description prompts: **observe_endurant_sorts**, **observe_concrete_part**, **observe_component_sort**, **observe_structure_components** and **observe_non_solids**.

type: observe_endurant_sorts: E→**Text**

Narrative:
s. narrative text on sorts $E_1,...,E_n$
o. narrative text on observers $obs\_E_1,...,obs\_E_n$
p. narrative text on proof obligation: $\mathscr{P}$
Formalisation:
s. **type** $E_1,...,E_n$
o. **value** obs_E_1: $E \to E_1,...,obs\_E_n$: $E \to E_n$
p. **proof obligation** $\mathscr{P}$: $\forall\ i{:}\{1..n\} \bullet is\_E_i(e) \equiv \bigwedge\{\sim E_j(e) | j{:}[1..n] \setminus \{i\} | j{:}[1..n]\}$

A similar observer is defined for concrete type parts, cf. Sect. 5.2.4:

type: observe_concrete_part: E→**Text**

Narratives:
s. narrative text on sort P
o. narrative text on observer obs_Ps
Formalisation:
s. **type** P, Ps = P-**set**
o. **value** obs_Ps: E→P-**set**

We refer to [12] for observers of components, structures and non-solids. The above covered the prompts for describing external qualities. Prompts for describing internal qualities are: **observe_unique_identifier**, **observe_mereology** and **observe_attributes**.

type observe_unique_identifier: P→**Text**

Narratives:
i. text on unique identifier: UI
o. text on unique identifier observer: uid_E
Formalisation:
i. **type** UI
o. **value** uid_E: E → UI

type observe_mereology: P→**Text**

Narratives:
m. text on mereology: M
o. text on mereology observer: mereo_E
Formalisation:
m. **type** M = $\mathscr{E}$(UI$_a$, ..., UI$_c$)
o. **value** mereo_E: E → M

139

type observe_attributes: P→**Text**

Narratives:
a. texts on attributes: A$_i$, ..., A$_k$
o. texts on attribute observers: attr_A$_i$, ..., attr_A$_k$
Formalisation:
a. **type** A$_i$, ..., A$_k$
o. **value** obs_A$_i$: E → A$_i$, ..., obs_A$_k$: E → A$_k$

140

Next we informally describe an ("interactive") analysis and description process. It postulates endurant values, V, as either (physical) part values, or component values, or non-solid values. The *process* evolves around: uod:UoD, an endurant, a universe of discourse, value. new, a variable holding a set of endurant values; gen a variable also holding a set of endurant values; and txt, a variable holding (a set) of narrative and formal texts, those generated generated, as illustrated, by observer functions. The analysis function **observe_endurant_values(v)** yields sets of endurant values. 141

A Domain Analysis & Description Process, Part I/II

```
type
    V = Part_VAL | Komp_VAL | Non_Sol_VAL
variable
    new:V-set := {uod:UoD},
    gen:V-set := {},
    txt:Text := {}
value
    discover_sorts: Unit → Unit
    discover_sorts() ≡
        while new ≠ {} do
            let v:V • v ∈ new in
            (new := new \ {v} ‖ gen := gen ∪ {v}) ;
            is_part(v) →
                (is_atomic(v) → skip ,
                 is_composite(v) →
                     (new := new ∪ observe_endurant_values(v) ‖
                      txt := txt ∪ observe_endurant_sorts(v)) ,
                 has_concrete_type(v) →
                     (new := new ∪ {ι observe_endurant_values(v)} ‖
                      txt := txt ∪ observe_concrete_part(v))) ,
            has_components(v) →
                 (new := new ∪  ‖
                  txt := txt ∪ observe_component_sorts(v)) ,
            has_non_solids(v) →
                 txt := txt ∪ observe_non_solid_sorts(v) ,
            is_structure(v) →
                 ... [EXERCISE FOR THE READER !] ...
    end end
```

142

$\iota\ s$ selects an element of the set s. If s is empty $\{\iota\ s\}$ is empty.

A Domain Analysis & Description Process, Part II/II

```
discover_uids: Unit → Unit
discover_uids() ≡
    for ∀ v:(PVAL|KVAL) • v ∈ gen
        do txt := txt ∪ observe_unique_identifier(v) end
discover_mereologies: Unit → Unit
discover_mereologies() ≡
    for ∀ v:PVAL • v ∈ gen
        do txt := txt ∪ observe_mereology(v) end
discover_attributes: Unit → Unit
discover_attributes() ≡
    for ∀ v:(PVAL|MVAL) • v ∈ gen
        do txt := txt ∪ observe_attributes(v) end
analysis+description: Unit → Unit
analysis+description() ≡
    discover_sorts();          discover_uids();
    discover_mereologies();    discover_attributes()
```

143

8 From Parts to Behaviours

144

It is often said *"every noun can be verbed"* and *"every verb can be nouned"*. That may be so. In any case we shall perform the following one-way *transcendental deduction: "to every [endurant] physical part" "we* 145 *shall associated a perdurant behaviour"*. That deduction is "inspired" by the following observations: (i) *there is the train, as an endurant entity, as as it stands, there, on the platform, statically observable, over time;* (ii) *there is the train, as a perdurant behaviour, as it "speeds" down the railway track, only a part of it visible at any given point and time;* and (iii) *there is the train, as a railway system attribute, as it "appears"* 146 *in a time table; programmable.*

By an **action** we shall understand a function which, among its arguments, take a state and delivers an updated state, and where that action has been knowingly, willfully applied. By an **event** we shall understand a state change for which we do not seek its origin, i.e., who or what caused that state change. 147 By a **behaviour** we shall understand a sequence of one or more actions, events and behaviours.

In Sect. 9 we shall formally summarise, cf. [12], that deduction. Since physical parts coexist — their translated behaviours operate concurrently. Since physical parts relate — these behaviours communicate. For that reason we shall express the part behaviours in terms of Hoare's CSP [17, 18, 22, 23]. Hence we shall express communication via channels.

9 Perdurants

148

To simplify matters we shall just deal with artefacts. These are described in terms of their sorts, whether atomic, composite or concrete (like 'sets of'), and unique identifiers, mereology and attributes. Transcendentally deduced artefact behaviours are described in terms of their signatures and definition bodies. 149 We shall now show how to relate all of the endurant descriptions with their perdurant counterpart; that is, how the transcendental deduction works ! Each have a crucial rôle !

9.1 Behaviour Signatures

150

A behaviour signature, for part p, is of the form:

value

```
Name: ui:UI →
      (st1,...sts):Statics →
      me:Mereology →          [me = (ichs,iochs,ochs)]
      (ca1,...,cac):Programmables →
      in      Monitorables, in_Mereology,
      in,out in_out_Mereology,
      out    out_Mereology → Unit
```

151

We explain this signature: Name is an analyser cum describer chosen name, usually a meaningful mnemonic; UI is the type of the unique identifier for the translated sort, i.e., of part p; Statics designates the zero (s=0), one or more static attributes of part p; Mereology designates the triplet mereology of the part, cf. last paragraph of Sect. 6.2; Programmables designates the zero (c=0), one or more programmable attributes of part p; Monitorable designates zero, one or more **in**put channel references designating monitoriable attributes of part p; in_Mereology designates zero, one or more **in**put channel references designating the parts from whom part p "receives" input; in_out_Mereology designates zero, one or more **in**put/**out**put channel references designating the parts from whom part p "receives" input and to whom it "delivers" output; out_Mereology designates zero, one or more **out**put channel references designating the parts to whom part p "delivers" output; and **Unit** designates that the behaviour goes on forever ! Technicalities are given in [12], Sects. 7.4.3–7.4.4.

9.2 Behaviour Definition Bodies: $\mathscr{B}_P$

152

In general the signature expresses that behaviour Name(uid) evolves around (i) constant values whose type is given in Statics; (ii) input from monitorable attributes (of values "residing" in part p, but not otherwise expressible) are expressed in the body of the behaviour definition by the CSP input expression attr_A ? where A is a monitorable attribute of p; (iii) input from topologically related parts, q, are expressed by ch[ui$_p$,ui$_q$] ?; and (iv) output of values v to topologically related parts, q, are expressed by ch[ui$_p$,ui$_q$] ! v. (v) In other words, the channel designations of the signature are of the form: attr_A$_i$, ..., attr_A$_j$ and ch[ui$_p$,ui$_q$]. Further technicalities are given in [12], Sect. 7.4.5.

153

9.3 From Part Descriptions to Behaviour Definitions

Composite parts: **type** Translate$_P$: P → RSL$^+$**Text**

```
value
  Translate_P: P → RSL+Text
  Translate_P(p) ≡
    let ui = uid_P(p),          me = mereo_P(p),     Sects. 6.1,6.2
        sa = stat_attr_vals(p), ca = ctrl_attr_vals(p),    Sect. 6.3.1
        MT = mereo_typs(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
        M_P: P_UI → ST → MT → CT → IOR  Unit
        M_P(ui)(sa)(me)(ca) ≡ B_P(ui)(sa)(me)(ca)
          ,⫸ Translate_P1(obs_endurant_sorts_E1(p))
          ⫷⫸ Translate_P2(obs_endurant_sorts_E2(p))
          ⫷⫸ ...
          ⫷⫸ Translate_Pn(obs_endurant_sorts_En(p))
    end
```

154

The above schema specifies the translation of composite parts into RSL$^+$**Text**, where RSL is the RAISE Specification Language, [16]. The ⫷ ... ⫸ designate the texts, ..., as written.

155

Concrete parts: **type** **Translate**$_P$: P → RSL$^+$**Text**

```
type
    Qs = Q-set
value
    qs:Q-set = obs_Qs(p)
    Translate_P(p) ≡
        let ui = uid_P(p), me = 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
        ⫷ channel IOD
            value
                𝒜_P: UI → ST → CT → IOR Unit
                𝒜_P(ui)(sa)(me)(ca) ≡ ℬ_P(ui)(sa)(me)(ca)  ⫸
                { ⫷,⫸ Translate_Q(q)|q:Q•q ∈ qs }
        end
```

156

Atomic parts: **type** **Translate**$_P$: P → RSL$^+$**Text**

```
value
    Translate_P(p) ≡
        let ui = uid_P(p), me = 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
        ⫷ channel IOD
            value
                𝒜_P: P_UI×MT×ST  PT  IOR  Unit
                𝒜_P(ui)(sa)(me)(ca) ≡ ℬ_P(ui)(sa)(me)(ca)  ⫸
        end
```

$\mathscr{B}_P$(ui)(sa)(me)(ca) designate the "body" of the definition of behaviour $\mathscr{B}_P$(ui). For details we refer to the **Core Behaviour Schema** of [12, Sect. 7.5].

9.4 Channel Declarations

157

Here we shall just mention that the above **Translate** schemas refer to **channel**s. The channel declaration, in RSL, are of the form

- **channel** ch[{ui$_p$,ui$_q$}]: CH_MSG

where CH_MSG is a type expression for the values communicated over CSP channels. That is, the channel array indexes are two element sets of unique identifiers of relevant distinct parts as implied by their respective mereologies.

9.5 Concrete System

158

An instantiation of any given *universe of discourse, uod,* thus amounts to the parallel, ||, composition of behaviours, potentially one for each composite and each atomic part. [12, Sect. 7.6] illustrates an example.

10 Conclusion 159

10.1 What Have We Achieved ?

We have summarised an essence of Kai Sørlander's Philosophy [31]: recounted how, from a basis of the **inescapable meaning theory**, the concepts of *space*, *time* and *Newton's Laws* can be transcendentally deduced, and from these the concepts of *living species: plants* and *animals*. And we have summarised the 160
essence of [12]: an ontology of endurants and perdurants, discrete and continuous (non-solid) endurants, physical parts, living species and structures, natural parts and artefacts; and their internal qualities: unique identifiers, mereology and attributes. Finally we have shown, by a transcendental deduction, how discrete 161
endurants can be "morphed" into perdurants, i.e., [in this paper] CSP behaviours whose signature can be derived from internal qualities of appropriate discrete endurants: unique identifiers, mereology and attributes. Throughout we have related the two areas: philosophy and computing. 162

The Philosophy aspect of this paper is new. That is, it is, to our knowledge, the first time a serious attempt has been made to strongly relate an area of the science of computing to philosophy. The domain science & engineering of [12] is also new. For the first time we see a straight line from the domain of artefact problems to their solution by computing [1, 2, 5, 6, 7, 11]. We find that remarkable.

If you have a need for examples, please consult [12] and [13, twelve domain case studies].

10.2 Open Problems 163

We shall only focus on issues that relate to Sects. 4–9. Further studies seem necessary in order to secure the inevitability of the distinction between discreteness and continuity, justify the presence of structures, and the distinction between atomic and composite parts. The transcendental deduction of endurants into perdurants may not exactly satisfy Kai Sørlander's strict criteria for such deductions.

10.3 Acknowledgment 164

I am grateful to Kai Sørlander for his patience and help in properly understanding his philosophy and for creating that philosophy [24, 25, 26, 27, 28, 29, 30, 31]: truly a remarkable feat — as also observed by Georg Henrik von Wright, Wittgenstein's successor at Cambridge, England, at age 32 [en.wikipedia.-org/wiki/Georg_Henrik_von_Wright] [24]. 165

11 References 166

[1] 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. imm.dtu.dk/~dibj/montanari.pdf.

[2] 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.

[3] Dines Bjørner. Domain Science & Engineering – *From Computer Science to The Sciences of Informatics, Part I of II: The Engineering Part*. *Kibernetika i sistemny analiz*, 2(4):100–116, May 2010.

[4] Dines Bjørner. Domain Science & Engineering – *From Computer Science to The Sciences of Informatics Part II of II: The Science Part*. *Kibernetika i sistemny analiz*, 2(3):100–120, June 2011.

[5] 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. imm.dtu.dk/~dibj/maurer-bjorner.pdf.

[6] Dines Bjørner. *Domain Science and Engineering as a Foundation for Computation for Humanity*, chapter 7, pages 159–177. Computational Analysis, Synthesis, and Design of Dynamic Systems. CRC [Francis & Taylor], 2013. (eds.: Justyna Zander and Pieter J. Mosterman).

[7] Dines Bjørner. Domain Analysis: Endurants – An Analysis & Description Process Model. In Shusaku Iida and José Meseguer and Kazuhiro Ogata, editor, *Specification, Algebra, and Software: A Festschrift Symposium in Honor of Kokichi Futatsugi*. Springer, May 2014. imm.dtu.dk/~dibj/2014/kanazawa/kanazawa-p.pdf.

[8] Dines Bjørner. Domain Engineering – A Basis for Safety Critical Software. Invited Keynote, ASSC2014: Australian System Safety Conference, Melbourne, 26–28 May. , Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, December 2014. imm.dtu.dk/~dibj/2014/assc-april-bw.pdf.

[9] Dines Bjørner. Manifest Domains: Analysis & Description. *Formal Aspects of Computing*, 29(2):1–51, 2017. DOI 10.1007/s00165-016-0385-z.

[10] 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*, 12 May 2018. First International Workshop on Handling IMPlicit and EXplicit Knowledge in Formal Fystem Development, 17 November 2017, Xi'an, China. imm.dtu.dk/~dibj/2017/bjorner-impex.pdf.

[11] Dines Bjørner. To Every Manifest Domain a CSP Expression — A Rôle for Mereology in Computer Science. *Journal of Logical and Algebraic Methods in Programming*, 1(94):91–108, January 2018. compute.dtu.dk/~dibj/2016/mereo/mereo.pdf.

[12] Dines Bjørner. Domain Analysis & Description – Principles, Techniques and Modelling Languages. *ACM Trans. on Software Engineering and Methodology*, 28(2), April 2019. imm.dtu.dk/~dibj/2018/tosem/Bjorner-TOSEM.pdf.

[13] Dines Bjørner. Domain Case Studies:

- *Container Line*, ECNU, 2019, imm.dtu.dk/~db/container-paper.pdf
- *Documents*, TongJi Univ., 2018, imm.dtu.dk/~dibj/2017/docs/docs.pdf
- *Urban Planning*, TongJi Univ., 2018, imm.dtu.dk/~dibj/2017/up/urban-planning.pdf
- *Swarms of Drones*, ISCAS, 2017, imm.dtu.dk/~dibj/2017/swarms/swarm-paper.pdf
- *Road Transport*, DTU, 2013, imm.dtu.dk/~dibj/road-p.pdf
- *Credit Cards*, Uppsala, 2012, imm.dtu.dk/~dibj/2016/credit/accs.pdf
- *Weather Information*, Bergen, 2012, imm.dtu.dk/~dibj/2016/wis/wis-p.pdf
- *Web-based Transaction Processing*, TUV, 2010, imm.dtu.dk/~dibj/wfdftp.pdf
- *The Tokyo Stock Exchange*, Todai, 2010, imm.dtu.dk/~db/todai/tse-1.pdf, imm.dtu.dk/~db/todai/tse-2.pdf
- *Pipelines*, Graz, 2009, imm.dtu.dk/~dibj/pipe-p.pdf
- *The Market*, DTU, 2008, imm.dtu.dk/~dibj/themarket.pdf
- *Container Terminal Port*, DTU, 2007, imm.dtu.dk/~dibj/2018/yangshan/maersk-pa.pdf
- *Railways*, DTU - a compendium, 2004, imm.dtu.dk/ dibj/train-book.pdf

Experimental research reports: 2004–2019, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark.

[14] Dines Bjørner. The Rôle of Domain Engineering in Software Development. Why Current Requirements Engineering Seems Flawed! In *Perspectives of Systems Informatics*, volume 5947 of *Lecture Notes in Computer Science*, pages 2–34, Heidelberg, Wednesday, January 27, 2010. Springer.

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

[16] Chris W. George, Peter Haff, Klaus Havelund, Anne Elisabeth Haxthausen, Robert Milne, Claus Bendix Nielsen, Søren Prehn, and Kim Ritter Wagner. *The RAISE Specification Language*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England, 1992.

[17] Charles Anthony Richard Hoare. Communicating Sequential Processes. *Communications of the ACM*, 21(8), Aug. 1978.

[18] Charles Anthony Richard 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).

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

[20] 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.

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

[22] A. W. Roscoe. *Theory and Practice of Concurrency*. C.A.R. Hoare Series in Computer Science. Prentice-Hall, 1997. http://www.comlab.ox.ac.uk/people/bill.roscoe/publications/68b.pdf.

[23] Steve Schneider. *Concurrent and Real-time Systems — The CSP Approach*. Worldwide Series in Computer Science. John Wiley & Sons, Ltd., Baffins Lane, Chichester, West Sussex PO19 1UD, England, January 2000.

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

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

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

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

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

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

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

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