Domains: Ontologies and Taxonomies: $2+3\mathcal{N}$

Dines Bjørner

Inst. of Mathematics & Computer Science
Technical University of Denmark
DK-2800 Kgs. Lyngby, Denmark
E–Mail: bjorner@gmail.com, URL: www.imm.dtu.dk/~db

June 16, 2026, 07:22

Abstract We study the concepts of domain, ontology and taxonomy as they are used in *domain science & engineering* – a [new] branch of formal software science & engineering.

For the concept of $\mathbb{DOMAIN}$s, such as we characterise that term, there is suggested one analysis & description ontology and one taxonomy; that is: one way of *analysing & describing* and summarily *structuring domains*, and, for each such (domain analysis & description), a taxonomy of the actual domain and a domain specific ontology, et cetera!

The contribution of this paper is that the specific $\mathbb{DOMAIN}$ analysis & description *ontology* [of this paper] (i) explains the conceptual structure of domains and guides the domain analyser cum describers in their analysis – thus "forming" the description; and (ii) that narrative & formalised descriptions of domains structure their *taxonomies* and yields further *ontologies*.

From a common $\mathbb{DOMAIN}$ analysis & description *ontology* to *domain understanding*, and from domain *descriptions* and *taxonomies* to further *ontologies*!

In other words: One pair of common *domain analysis & description ontology* and *taxonomy* and $\mathcal{N}$ pairs of *domain ontologies* and *taxonomies* – hence $2+2\mathcal{N}$!

The Triptych Dogma

In order to *specify* $\mathbb{S}$*oftware*, we must understand its $\mathbb{R}$*equirements*.
In order to *prescribe* $\mathbb{R}$*equirements* we must understand the $\mathbb{D}$*omain*.
So we must study, analyze and describe $\mathbb{D}$*omains*.

$\mathbb{D},\mathbb{S} \models \mathbb{R}$

In proofs of $\mathbb{S}$*oftware* correctness,
with respect to $\mathbb{R}$*equirements*,
assumptions are made with respect to the $\mathbb{D}$*omain*.

1 Introduction

This paper is a *'proof of concept'* paper. We wish to understand the terms *'ontology'* and *'taxonomy'* in the context of the $\mathbb{DOMAIN}$ *method*. We focus on the domain aspects of the Triptych Dogma – and start by characterising the two concepts of *ontology* and *taxonomy* in their most general understanding:

Characterisation: 1 *Ontology: Ontology is the branch of metaphysics dealing with the nature of being; a set of concepts and categories in a subject area or domain that shows their properties and the relations between them [12,13]* `[Wikipedia]`*. An ontology identifies and distinguishes concepts and their relationships; it describes content and relationships; it describes content and relationships. [A taxonomy formalizes the hierarchical relationships among concepts and specifies the term to be used to refer to each; it prescribes structure and terminology.]* [Bob Bater]. Ontologies *specify* ■

This general concept of ontology appears [15, *Tom Gruber 1993*] to mainly being used in connection with **knowledge sharing**. In this paper we shall, in contrast, apply the term to software development, in particular to domain engineering.

Characterisation: 2 *Taxonomy: Taxonomy is a practice and science concerned with classification or categorization. Typically, there are two parts to it: the development of an underlying scheme of classes (a taxonomy) and the allocation of things to the classes (classification)* ■

- **Taxonomy and ontology** are both methods used to **organize** and **structure** information, but they differ significantly in complexity and purpose.
- **Taxonomy** is a hierarchical classification system used to organize items into parent-child categories (a "tree" structure), primarily for navigation and tagging.
- **Ontology** is a more complex, formal representation of a knowledge domain, mapping relationships, properties, and constraints between concepts (a "web" structure) to enable reasoning and inference.

The two concepts appear to be somewhat "muddled up", i.e., confused, by some. So it appears from `Internet` documents[1]

This paper is based on [1,3,4,5,7]. When we use the special term DOMAIN we mean the specific method [7] for analysing & describing domains as we shall soon define that term.

Characterisation: 3 *Domain: By a* ***domain*** *we shall understand a rationally describable segment of a discrete dynamics fragment of a human assisted reality: the world that we daily observe – in which we work and act, a reality made significant by human-created entities. The domain embody endurants and perdurants [DB]* ■

Endurants and perdurants are characterised in Sect. 3.2 on page 4

Characterisation: 4 *The* 𝔻𝕆𝕄𝔸𝕀ℕ *Analysis & Description Method: By the* 𝔻𝕆𝕄𝔸𝕀ℕ ***analysis & description method*** *we understand a set of principles, techniques and tools for analysing and describing domains.* ■

The 𝔻𝕆𝕄𝔸𝕀ℕ method is covered in [1,3,4,5,7] and summarised in Sect. 3.2.

Characterisation: 5 *Domain Descriptions: By* ***domain description*** *we shall understand a set of* ***description units*** *that narrate and formalise individual domain entities: endurants and perdurants, their values and types, their unique identification, mereologies, attributes, and behaviours – where those descriptions refer to, stand for, denote the actual entities of the concrete, manifest domain being described* ■

[1] – for example `https://www.earley.com/insights/what-difference-between-taxonomy-and-ontology-it-matter-complexity` and `https://www.linkedin.com/pulse/classification-process-taxonomy--stan-garfield`

Characterisation: 6 *Domain Analysis & Description Ontology: By a* ***domain analysis & description ontology*** *we mean a summary of the domain analysis & description method where that ontology is expressed in terms of concepts* ■

Characterisation: 7 *Domain Analysis & Description Taxonomy: By a* ***domain analysis & description taxonomy*** *we mean a taxonomy, cf. characterisation 2 on the preceding page* ■

2 A Review

2.1 Earlier and Our Work on Ontology

This paper treats the concepts of ontology and taxonomy from a different viewpoint than early (1970s-1990s) ontology (and taxonomy) work, for example: [11,15,18], and from numerous `Internet` postings. [2] Much of the work on ontology has so far resulted in ontology description languages, viz. `Cyc`[3] and `OWL`[4]. The background for the early work is McCarthy's [25,26,27]. We also build on that work. But whereas our work also builds on the Landin/Strachey/Scott work on the formal semantics of programming Languages [19,20,23,22,21,14,31,33,28,32,16,30], the 'semantics' influence seems absent from "standard" ontology work. At the risk of simplifying our argument one may claim that past and current ontology work represents a syntactic view of the world whereas the $\mathbb{DOMAIN}$ ontology work is deeply influenced by a, or the, semantic view.

2.2 Earlier and Our Work on Taxonomies

There seems to be little, if any serious work on taxonomy in information and in compuer science. The field originated in botanics. [29] shows scientific work in zoology. Mentioning of taxonomy if computer science and software engineering contexts is of the kind *"Oh ! – and here is the* [or *a*] *taxonomy ..."*. The literature [also on the `Internet`] even refers to the "taxonomy confusion": re. *"What is the difference between ontology and taxonomy ?"*[5] We shall show how domain taxonomies can be directly 'derived', i.e., calculated from domain descriptions that are developed based on the $\mathbb{DOMAIN}$ analysis & description ontology – and we shall suggest that such taxonomies can be the basis for specific domain ontologies !

3 Ontology

We bring some conventional examples.

3.1 Some Examples

"Classical" ontologies are presented in either of two forms: graphical or formal textual.

We shall only refer to the formal textual in the form of literature references: [11,15,18], for general treatments, and `Internet` references to specific ontology languages: `CyC: en.wikipedia.org/wiki/Cyc` and `OWL: en.wikipedia.org/wiki/Web_Ontology_Language`.

[2] – for example `en.wikipedia.org/wiki/Ontology_(information_science)`, `www.sciencedirect.com/-science/article/pii/S1071581985710816?via%3Dihub`, and `en.wikipedia.org/wiki/Suggested_Upper_Merged_Ontology`. Prefix these [and the next] `URL`s with `https://`.

[3] en.wikipedia.org/wiki/Cyc

[4] en.wikipedia.org/wiki/Web__Ontology__Language

[5] – `www.earley.com/insights/what-difference-between-taxonomy-and-ontology-it-matter-complexity`.

We start by showing graphical renditions of ontologies. They vary very (!) much, some are colourful !

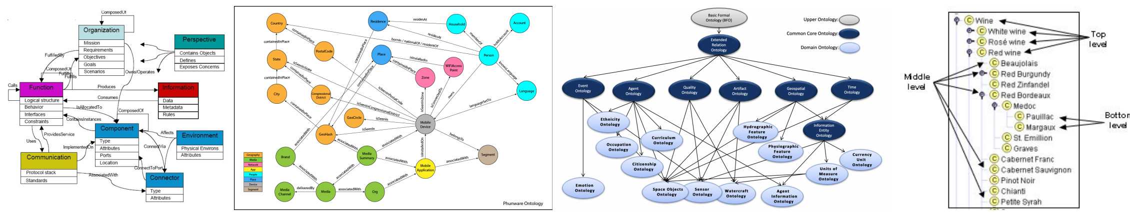


Figure 1. Example Ontologies

The ontologies shown in Fig. 1 are taken from:
`en.wikipedia.org/wiki/Ontology_(information_science)`,
`data.phunware.com/docs/phunware-ontology`,
`ncor-network.org/docs/about/what-is-ontology` and
`protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html`.[6]

3.1.1 Comment on Examples Fig. 1.

"Behind" examples of Fig. 1 lie texts that explain *"what is an ontology ?"*. To us it is quite 'telling' that there are rather many such *"what is an ontology ?"* explanations. It is also telling that ontologies are rendered in graphic form – except, perhaps, the rightmost ontology of Fig. 1. Each of the examples use a different kind of graphics. And finally it is telling that the explanations vary so much.

In other, at least our, words: it seems time to sort out some common aspects of ontologies and, perhaps it may be useful to formalise these.

In the next section we sketch our formal understanding of the DOMAIN analysis ontology.

3.2 The DOMAIN Analysis & Description Ontology

In this paper we restrict the treatment of an ontology concept to that of DOMAINs[7]. Focusing, we emphasize, on the man-made aspects. That is: excluding ontologies that "cover" the God-made worlds of physics (including chemistry), biology, etc.

Characterisation: 8 *Endurants: In the context of domains,* ***endurants*** *are those entities that we can observe (see and touch), in* space*, as complete entities at no matter which point in* time *– material entities that persists, endures – capable of enduring adversity, severity, or hardship [Merriam Webster]* ■

Characterisation: 9 *Perdurants:* ***Perdurants****, in the context of domains, are those entities of domains for which only a fragment exists, in* space*, if we look at or touch them at any given snapshot in* time *[Merriam Webster]* ■

[6] – prefix these URLs with `https:`

[7] By emphasizing the term DOMAIN we mean to "limit" the scope of our treatment to that of the domains as outlined in Characterisation 3 on page 2 and, in particular, to treat these in the light of the DOMAIN method [7]

In the context of endurants one can speak of endurants being either

(i) **solid**[8] or (ii) **fluid**[9]; solids being either
(iii) **parts**[10] or (iv) **living species**[11]; of parts being either
(v) **atomic**[12] or (vi) **compounds**[13]; compounds being either
(vii) **Cartesian**[14] or (viii) **part set parts**[15].

Et cetera.

In the context of perdurants we can speak of **channels, actors, actions, events**, and **behaviours**.

We refer to [1,3,4,5,7] for details!

Figure 2 on the following page shows a graphic rendition, we suggest, of an ontology for analysing, describing and structuring an understanding of DOMAINs.

We now explain Fig. 2 on the next page.

- Focus on the root of the tree-like structure topmost,
- It "lists" (•, $\mathbb{TIME}$, $\mathbb{SPACE}$).
- Interpret hat as follows: Domains, i.e., the •, exists in $\mathbb{TIME}$ and $\mathbb{SPACE}$ – where $\mathbb{TIME}$ and $\mathbb{SPACE}$ and refers to what you, i.e., we "normally" understand by time and space. The • refers to the domain being analysed.
- $\mathbb{TIME}$ and $\mathbb{SPACE}$ exits in any DOMAIN, in any world. They are not entities. They are phenomena that can be transcendentally deduced based on pure reasoning!
- Domain phenomena are either rationally describable, i.e., are *entities* or they are not – in which latter case our domain analysis [and description] ends!
- If they are entities, then we can go on in our analysis.
- The above `either-or, (i-viii)` analysis follows the tree structure to the left of Fig. 2 on the following page – shown as the External Qualities labeled dashed blue line box. It directs the use in the analysis process, or, better, *shows the set of concepts and categories of DOMAINs!*

A similar explanation for *perdurants* can be given.

Now, what is special – about the DOMAIN approach to the understanding, analysis and description of DOMAINs – is the following:

- Certain **parts**, those that the domain analyser cum describers decide as being *"candidates for behaviour-ship"*, that is: are *"behaviourable"*, can *transcendentally deduced*, i.e., *"morphed"* into **behaviours**!

[8] – an endurant which is separate, individual or distinct in form or concept, or, rephrasing: have body [or magnitude] of three-dimensions: length, breadth and depth `[24, Vol. II, pg. 2046]`

[9] – an endurant which is prolonged, without interruption, in an unbroken series or pattern; or, rephrasing: a substance (liquid, gas or plasma) having the property of flowing, consisting of particles that move among themselves [24, Vol. I, pg. 774]

[10] a solid endurant existing in time and subject to laws of physics, including the *causality principle* and *gravitational pull*

[11] a solid endurant, subject to laws of physics – including biology, and additionally subject to *causality of purpose*. plus more!

[12] – a part which the domain analyser considers to be indivisible in the sense of not meaningfully divisible, for the purposes of the domain under consideration, that is, to not meaningfully consist of sub-parts

[13] those parts which can be considered composed from either a definite collection (a Cartesian) of two or more parts (!) or an indefinite collection (a part set) of zero, one or more parts

[14] parts consist of two or more distinctly type-named endurants (solids or fluids)

[15] part set parts consist of an indefinite number of zero, one or more parts

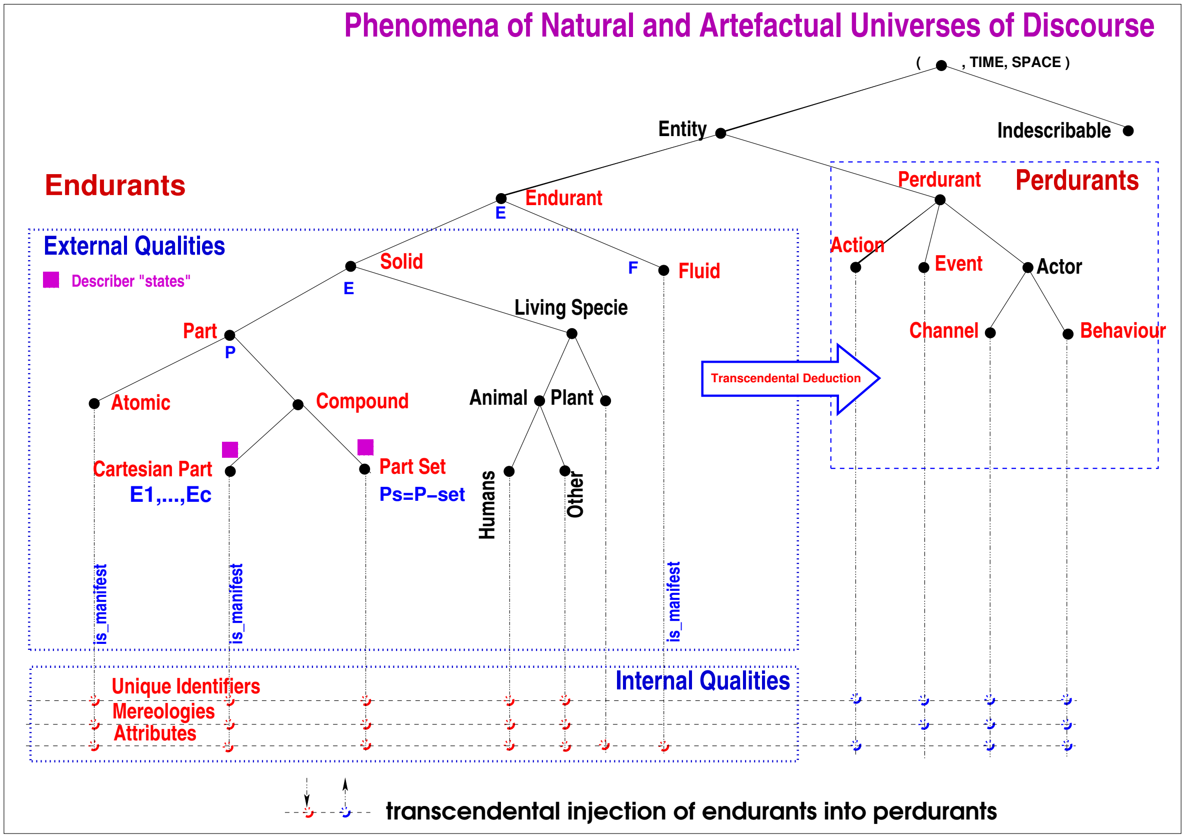
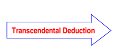


Figure 2. A Domain Analysis Ontology

- We do not have to be concerned with the question as to which these parts are: how the domain analyser cum describers decide upon this – but refer to [1,3,4,5,7].
- The *Domain Analysis Ontology* diagram of Fig. 2 attempts to show, in two ways, how this "morphing" works:
  - ∗ The left-right arrow  "means" the following: To every behaviourable endurant part there corresponds a perdurant in the form of a behaviour function.
  - ∗ The dashed vertical lines "extending from" parts "down" to the horisontal [16] lines, and then "across" to the right vertically "up" to perdurants mean: the arguments of the behaviour function are those of the internal quality values of parts.

We now elaborate on this transcendental deduction. Earlier we focused on the *external qualities* of endurants. Now we turn to their *internal qualities* .

- With parts, `p`, we associate **unique identifiers, mereology** and **attributes**.
- Uniqueness of parts can be postulated: **uid_**`P(p)` "observes" the unique identifier of `p`.
- The **mereology** of `p`, i.e., how `p` relates to other parts – identified by their unique identifiers – is expressed by **mereo_**`P(p)`.

[16] left-hand side, also dashed lines, bottom Internal Qualities labeled dashed blue line box

- The **attributes**, `A1, A2, ..., An`, of `p` – completing the final set of properties that [fully, or at least, for our domain understanding and description purposes] characterises parts – can be known by means of these observers: **attr_**`A1(p)`, **attr_**`A2(p),...,` **attr_**`An(p)`.
- The **signature** of part `p:P` behaviours, $\mathcal{P}$. can then be expressed:
  - *** signature:** $\mathcal{P}$: $\mathtt{PI} \rightarrow \mathtt{MP} \rightarrow (\mathtt{A1} \times \mathtt{A2} \times \cdots \times \mathtt{An}) \rightarrow$ **Unit**
- Invocation of the `p:P` behaviour is then:
  - * $\mathcal{P}$(**uid_**p)(**mereo_**P(p))(**attr_**A1(p),**attr_**A2(p),...,**attr_**An(p)) .

This, then, explains, in a formal way, the meaning of our DOMAIN *analysis ontology* diagram.

3.3 Discussion of the DOMAIN Analysis Ontology

We have shown a graph: Fig. 2 on the facing page. It shows a number of labeled boxes, nodes, lines and a special arrow. The labels name concepts of domains. That is: it is suggested that these concepts, which has been more-or-less rigorously characterised, are indeed concepts central to our understanding of any domain. We can not define these concepts, say in mathematics, as they are postulated to be concepts of an otherwise informal, man-made world! We have then described the meaning of each box, node, line and the special arrow.

This, then, is how we would like to see any proposed ontology be rigorously characterised.

4 Taxonomy

Originally, taxonomy referred only to the classification of organisms on the basis of shared characteristics. Today it also has a more general sense. It may refer to the classification of things or concepts, as well as to the principles underlying such work. Thus a taxonomy can be used to organize species, documents, videos or anything else. ■

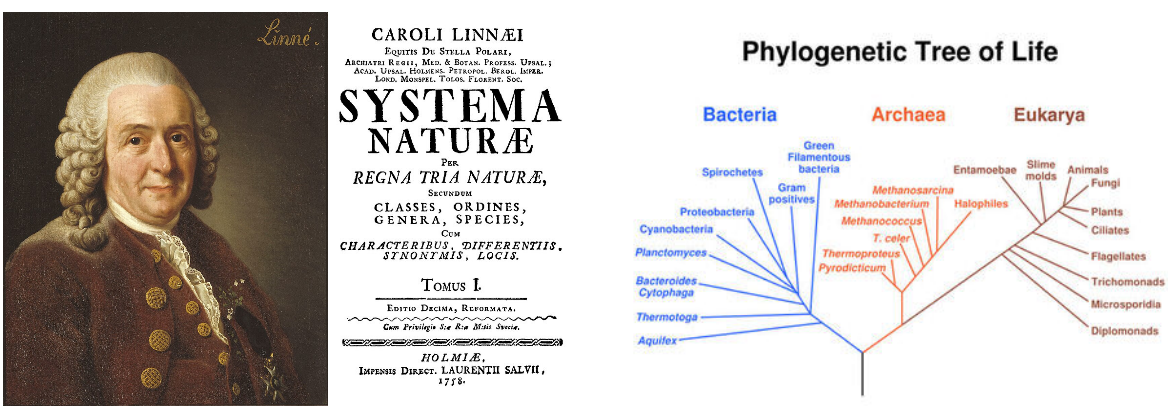


Figure 3. Carl von Linneaeus: The originator of Taxonomy

DOMAIN taxonomies come in two forms: endurant, respective perdurant taxonomies.

- ϵ **Endurant** taxonomies are tree-like and show how parts are composed, ultimately from atoms.

π **Perdurant** "taxonomies" show how part behaviours interact: synchronize and communicate.

We put – "tongue-in-cheek" – double quotes in perdurant "taxonomies" as there are not really of the same "kind" as endurant taxonomies. Endurant taxonomies adheres to Carl von Linneaeus' *Kingdom, Division* (or `Phylum`), `Class, Order, Family, Genus,` and `Species` hierarchy – but now with different classification. See `https://en.wikipedia.org/wiki/Taxonomy`.

DOMAIN perdurant taxonomies are graphs with nodes standing for part behaviours, edges for their interaction – as [indirectly] "forewarned" through the mereology of parts.

4.1 Endurant Taxonomy

We start by giving three examples.

4.1.1 Endurant Taxonomy Examples.

Example: 1 *A Road Transport Endurant Taxonomy:*
We develop the endurant taxonomy by first developing a description of a simple road transport domain:

Narrative:

1. *By a road transport domain,* `RT` *we shall understand*
2. *a Cartesian of a* `road net, RN` *and an* `automobile aggregate, AA`*.*
3. *A road net is observed as [yet] a Cartesian of an aggregate of hubs (i.e., street intersections),* `AH`*, and an aggregate of links (i.e., street segments between immediately neighbouring hubs),* `AL`*.*
4. *An aggregate of hubs is a set of atomic hubs.*
   An aggregate of links is a set of atomic links.
   An aggregate of automobiles is a set of atomic automobiles.

Formalisation:

type
1. *RT*
2. *RN, AA*

value
2. **obs_**RN: $RT \to RN$, **obs_**AA: $RT \to AA$

type
3. *AH, AL*

value
3. **obs_**AH: $RN \to AH$, **obs_**AL: $RN \to AL$

type
4. $SH = H$-**set**, H
4. $SL = L$-**set**, L
4. $SA = A$-**set**, A

value
4 **obs_**SH: $AH \to SH$, **obs_**LS: $AL - LS$
4 **obs_**SA: $AA - SA$

From the formalisation one can calculate the taxonomy, cf. Fig. 4 on the next page.

Example: 2 *A Container Terminal Port Endurant Taxonomy: We narrate, informally, the structure of components that we may observe of a container terminal port, cf. Fig. 5 on page 10: There are containers C, and there are container vessels, V. Vessels transport containers from harbour to harbour: load and unload them at container terminal ports in these harbours – where they are temporarily stored in [container] stacks* ***S*** *– with the vessels also storing them in stacks. Quay cranes* ***QC*** *lift containers between vessels and quay trucks* ***QT****. Quay trucks move containers between the quays, at which the vessels are moored, and [container terminal port] stacks. Vessel and port stacks are arranged in a number of rows* ***R****, and rows then consist of a proper "stack", i.e., a "tower" of containers [2].*

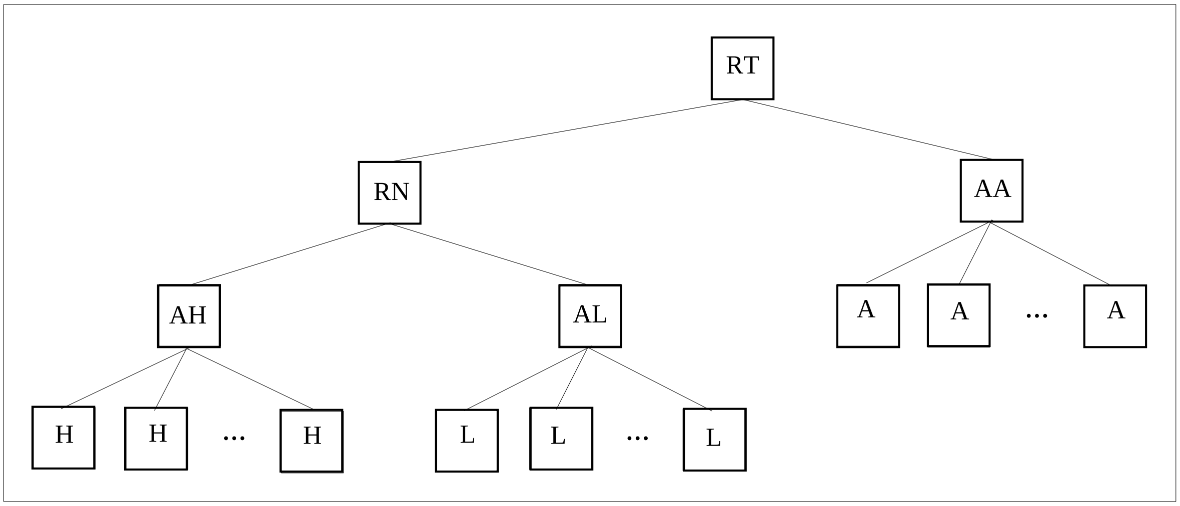


Figure 4. A Road Transport Endurant Taxonomy ■

Example: 3 *A Transport System Endurant Taxonomy:* *Figure 6 on page 11 shows an endurant taxonomy for the following kind of domain:*

- *There is the domain* `T` *[6].*
- *It embodies a* `Transport Net: G`*;* `Merchandise: MA, MS, M`*;* `Kustomers: KA, KS, K`*;* `Conveyor Companies: CKA, CKS, CK`*, etc.; and* `Logistics Companies: LA, LS ,L`*.*
- *The transport net embodies a* `Net Aggregate: NA` *with* `Nodes: N` *and an* `Edge Aggregate: EA` *with* `Edges: E`*.*
- *Nodes stand for land-based trucking centers, railway stations, harbours and airports.*
- *Edges, correspondingly stand for roads, rail lines, air routes and sea lanes.*
- *Merchandise and kustomers are atomic.*
- *Conveyor companies embody sets of conveyors,* `CS,C` *(trucks, freight and passenger trains, aircraft and vessels), and conveyor offices,* `CO`*.*
- *Logistics companies are much like conveyor companies* ■

4.1.2 Discussion of Endurant Taxonomy Examples. All three endurant taxonomy examples (Figures 4, 5 on the next page, 6 on page 11) are shown like ("up-side down") trees. Nodes are shown as boxes and stand for endurant parts. Branches are shown as lines and stand for "contains": the "upper node" is a compound parts and the "lower nodes", seen as a whole, is either a Cartesian of a part set. If there are $\cdots$ between lower nodes then the compound part is a part set – all of whose lower nodes are identically labeled. If not, then the lower (2 or more) nodes are elements of a Cartesian compound. The leaves of the endurant taxonomy trees are atoms.

The endurant taxonomy of Fig. 6 on page 11 is a variant of the former endurant taxonomy figures. It indicates that some (non-manifest) parts [dotted boxes] are not to be "matched" by, i.e., "morphed into", transcendentally deduced in behaviours; other (possibly manifest endurant) parts [dash-lined boxes] may be so morphed.

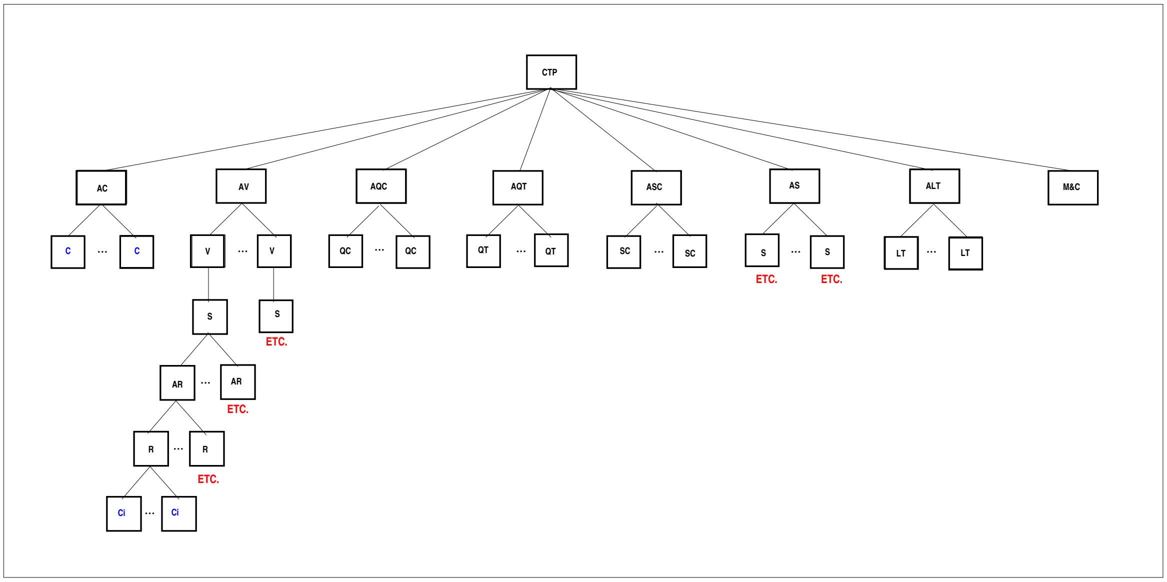


Figure 5. A Container Terminal Port Endurant Taxonomy ■

This, then, is ow we would like to see any proposed endurant taxonomy presented: A tree and its explication as just above.

4.2 Perdurant "Taxonomy"

Perdurant "taxonomies" are shown as graphs: nodes denote part behaviours and directed edges denote communication between part behaviours.

4.2.1 Perdurant "Taxonomy" Examples.

Example: 4 *A Road Transport Perdurant Taxonomy: The domain analyser cum describers have decided to only consider atomic part behaviours. Automobiles notify links and hubs of their entering, remaining in and leaving these – as summarily narrated and formalised below. See Fig. 7 on the facing page.*

Road Transport Perdurants – An Excerpt

5. Automobiles have (i) unique identifiers, (ii) mereology (the unique identifiers of the links and hubs it may enter) and (iii) attributes (whether at a hub or on a link, and a history: a time-stamped ordered sequence of messages as to the whereabouts of the auto).
6. Autos are either at a hub or on a link.
7. Autos at a hub, internal non-deterministically ($\sqcap$) either remains at the hub or enters a link.
8. Autos remaining at hubs
9. inform the hubs of this
10. and proceeds with a correspondingly updated history.
11. Autos entering links
12. [non-deterministically] chooses a[n exit] link (of its mereology) – and records the time;
13. informs the hub of its leaving
14. and the link of its entering,
15. and proceeds to be an auto now on a link.

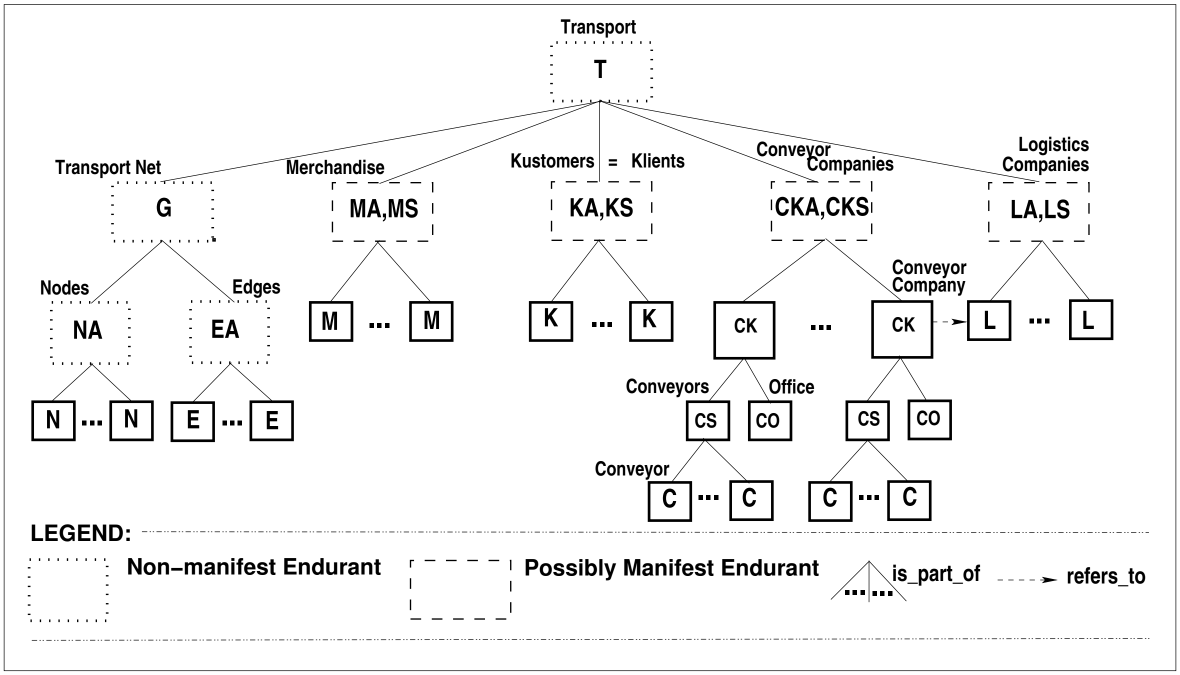


Figure 6. A Transport Domain Endurant Taxonomy

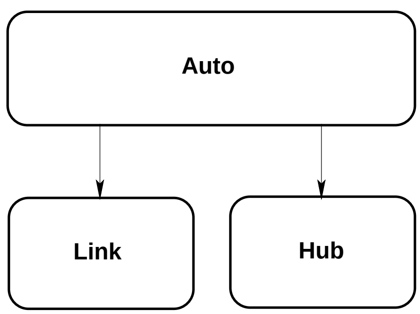


Figure 7. Road Transport Perdurant Taxonomy ■

16. etc.

```
value
5.  auto: AI → MI → (APos × Hist) → Unit
5.  auto(ai,(luis,huis),(pos,...)) ≡
6.     case pos of
6.        atHub(hi)        → auto_at_hub(ai)(luis,huis)(pos,hist)
6.        onLin(_,(_,_,)) → auto_on_link(ai)(luis,huis)(pos,hist) end
7.  auto_at_hub(ai)(luis,huis)(atHub(hi),hist) ≡
7.       auto_remain_at_hub(ai)(luis,huis)(atHub(hi),hist)
7.    ⌈⌉ auto_enters_link(ai)(luis,huis)(atHub(hi),hist,)
8.  auto_remain_at_hub(ai)(luis,huis)(atHub(hi),hist) ≡
9.     let τ = record TIME() in
9.     comm[{ai,hi}] ! Remain(τ,ai);
10.    auto(ai)(luis,huis)(atHub(hi),⟨(τ,atHub(hi))⟩^hist) end
```

```
11.  auto_enters_link(ai)(luis,huis)(atHub(fhi),hist) ≡ ...
12.    let tli:LI • tli ∈ luis, thi = ..., τ = record𝕋𝕀𝕄𝔼() in
13.    ( comm[{ai,fhi}] ! LeaveHub(τ,ai) ∥
14.      comm[{ai,tli}] ! EnterLink(τ,ai) ) ;
15.    auto(ai)(luis,huis)(onLink(tli,(fhi,0,thi)),⟨(τ,EnterLink(tli))⟩^hist) end
8.  etc. ■


17. Links have (i) unique identifiers, (ii) mereology (the autos that may enter the
    link), and (iii) attributes (length, ..., and history: a time-stamped ordered
    sequence of messages as to presence of autos).
    Links (as are hubs) are passive behaviours.
18. They record external non-deterministically (⌷) communications from autos and
19. resume being links, with updated history.

value
17.  link: LI → LMI → (LEN × ... × HIST) → Unit
17.  link(li)(ais)(l,...,hist) ≡
18.     let msg = ⌷ { comm[{ai,li}] ? | ai ∈ ais } in
19.     link(li)(ais)(l,...,⟨(msg)⟩^hist) end
```

The formal **comm**unication items 9, 13, 14 and 18, show the use of the CSP [17] output (!) / input (?) clause and correspond to the downward arrows in Fig. 7 on the previous page.

Example: 5 *A Container Terminal Port Perdurant Taxonomy: Container vessels notify port management (mgt.) of their arrival and departure, and, through port mgt. communicate with quay cranes; Quay cranes "likewise" communicate with quay trucks, these with port stack cranes; these again with stacks; and stacks – indirectly through stack cranes with land trucks [2]. See Fig. 8.*

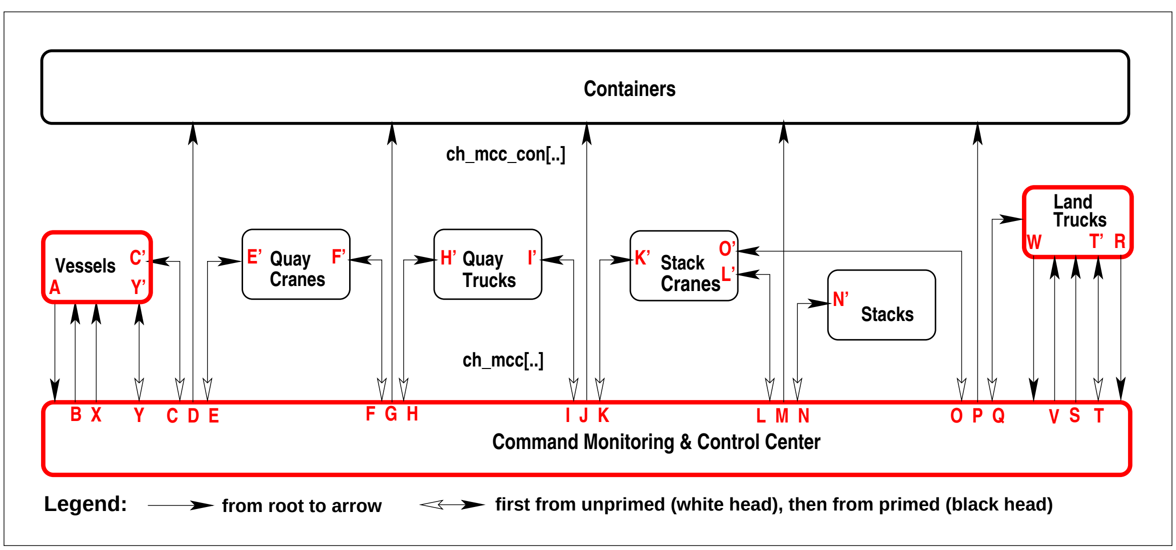


Figure 8. A Container Terminal Port Perdurant Taxonomy

Example: 6 *A Merchandise Transport Domain Perdurant Taxonomy: K*[17]*ustomers negotiate orders with logistics and/or conveyor companies, and these with conveyors. Conveyors communicate with kustomers and the nodes and edges of the route net [6]. See Fig. 9.*

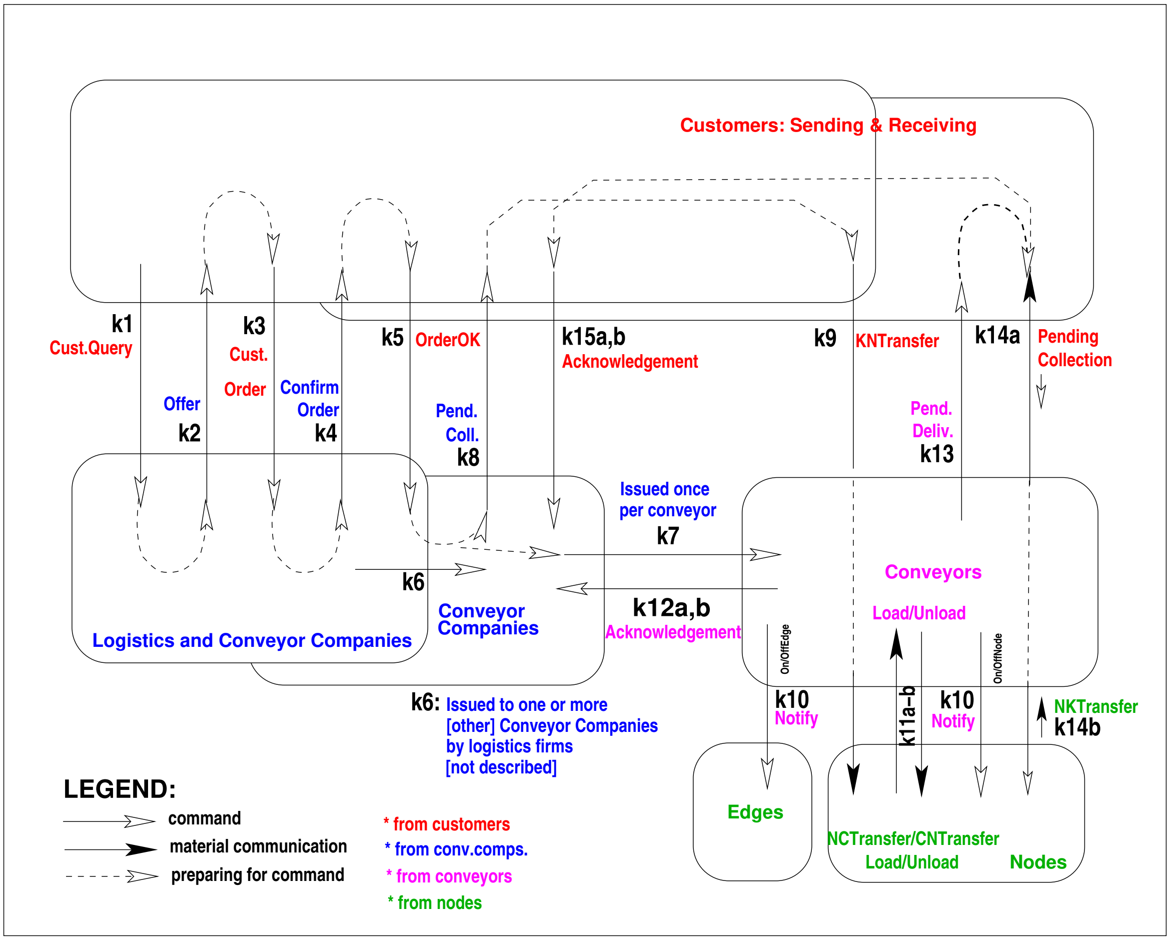


Figure 9. A Merchandise Transport Domain Perdurant Taxonomy ∎

4.2.2 Discussion of Perdurant "Taxonomy" Examples.

The exemplified perdurant "taxonomies" are really not taxonomies! Hence the tongue-in-cheek double quotes. They are not because they are not depicting a tree, but a graph structure. That is, we relax our characterisation of taxonomies, expanding from trees to graphs!

Endurant taxonomies describe, in narrative & formalisation texts and graphically, the "tree-like" decomposition of endurant compound parts into parts – and these are graphed by sharp corner boxes. Perdurant taxonomies describe – based on the mereologies of parts – the communication between part behaviours – and these are shown as "rounded boxes". Endurant taxonomy graphs symbolically indicate part set parts by $\cdots$'ed "continuations". Perdurant

[17] Sorry for spelling customers with k – done to avoid confusion with **C**onveyors

taxonomy graphs let each "rounded box" potentially stand for multiple behaviours of the same signature.

In this paper we only show perdurant taxonomies graphically – but these graphs summarise rigorous narrative and formalised textual descriptions of perdurants: channels and part behaviours. We refer to [4,5,7] for examples of such textual descriptions.

4.2.3 Rendition of Taxonomies.

So how do we show taxonomies of domains analysed & described according to the 𝔻𝕆𝕄𝔸𝕀ℕ method ?

Characterisation: 10 *Domain Taxonomy Rendition: By [a rendition of] a taxonomy of a domain that has been analysed & described according to the* 𝔻𝕆𝕄𝔸𝕀ℕ *method we shall understand a possibly textually annotated "tree" – one whose root node stands for the universe of discourse, whose "leaves" stand for atomic parts, and whose "inner" nodes stand for compounds of either Cartesian or part set parts with their descendant nodes standing for the components of that compound as analysed. Nodes are labeled with entity sort names. The possible annotations may be any kind of comments that the domain analyser cum describers deem relevant* ■

4.3 Ontologies & Taxonomies

4.3.1 From Taxonomy to Ontology !

We distinguish between the *abstract ontology* of domain analysis and description and the *concrete ontology* (as we shall now unfold that term). There is, in this paper one such abstract ontology.

We now claim that one can derive a domain ontology from any domain pair of endurant and perdurant taxonomies.; that is: to any pair of concrete domain taxonomies (the endurant and the perdurant), with their introduction of many parts, `p:P`, of part sets, there is one concrete domain ontology. It goes as follows – referring only to the graphically rendition of these derived and specific domain ontologies:

- There is an initial node of the concrete ontology graph[18] It is labeled with the same name as the "root" nod of the endurant taxonomy graph.
- Successor nodes "follow" those of the endurant taxonomy graph – except, and this is, of course, a difference between the taxonomy tree and "its corresponding" ontology graph, that part set parts are represented by only one node in the concrete ontology.
- Nodes are then otherwise connected as per the mereology of parts.
- Where nodes in taxonomies stand for, refer to, denote real, *manifest* domain entities, nodes in "their derived" ontologies stand for *immaterial* concepts.
- Et cetera !

It is left to the reader to fill in the details !

4.3.2 Rendition of DOMAIN Ontologies.

So how do we show Ontologies of domains analysed & described according to the 𝔻𝕆𝕄𝔸𝕀ℕ method ?

[18] In a textual explanation of the concrete ontology this "initial node" is usually an introductory section or paragraph.

Characterisation: 11 $\mathbb{DOMAIN}$ *Ontology: By an ontology of a domain that has been analysed & described according to the* $\mathbb{DOMAIN}$ *method we shall understand a possibly textually annotated "graph" – one whose nodes stand for atomic parts or compounds of either Cartesian or part set parts with their descendant nodes standing for the components of that compound as analysed – but where the parts of part set are represented just once. Nodes are labeled with entity sort names. The possible annotations may be any kind of comments that the domain analyser cum describers deem relevant* ■

4.3.3 The DOMAIN Analysis & Description Taxonomy. The Domain Analysis & Description Ontology aims a creating documents, i.e., one Endurant! The items of its special ontology are concepts. So how do we interpret the concept of Domain Analysis & Description Taxonomy? The answer is simple:

- The Domain Analysis & Description Taxonomy is the *Table of Contents* of the domain description.
- The leaves of this Table of Contents refer to domain specification units.

The ring has been closed!

4.3.4 The Taxonomy of This Paper. The present paper is also a concept document:

This Paper's Taxonomy

5 Conclusion

General. We have shown how an ontology can be the basis for understanding the structure and entities of a domain as well as the development of formal domain descriptions. It is suggested how endurant [domain] taxonomies can be calculated from domain descriptions as well as indicated how perdurant "taxonomies" can be related to descriptions of domain behaviours. Given the standard page limitations we have omitted the technical details of the "shown", "suggested" and "indicated".

Specific Claims. We have introduced, explained and related the concepts of:

- *Domain Description* (Char. 5 pg. 2)
- *The* $\mathbb{DOMAIN}$ *Method Analysis & Description Ontology* (Char. 6 pg. 3)
  There is one such in this paper. There could be others; but the one used here works.
- *The* $\mathbb{DOMAIN}$ *Method Analysis & Description Taxonomy* (Char. 7 pg. 3)
- *Domain Endurant Taxonomies* & *Perdurant Taxonomies* ($\iota\,\epsilon$ pg. 7 & $\iota\,\pi$ pg. 8)
- *Domain Ontologies* (Char. 11 pg. 15)
  For each of, say $\mathcal{N}$, domains, there is one ontology.
- $\mathbb{DOMAIN}$ *Taxonomy* (Sect. 4.3.3 pg. 15)
  And one endurant taxonomy.
  There is no corresponding perdurant taxonomy. A perdurant taxonomy could reflect how a specific domain analysis & description project went along.

In all $2+3\mathcal{N}$!

Overall Summary. Distinctions — between $\mathbb{DOMAIN}$ method and domain description ontologies and taxonomies — are necessary as some domains are conceptual, others are manifestations of a man-made, material world. Where ontology started out in the direction of *Knowledge Engineering* the ontology and taxonomy notions of this paper are bound to their *Domain Engineering* aim of describing and descriptions of material, human-artifact wolds.

Future Work. Based on the clarifications of this paper we can now develop deeper insight into the concepts of domain ontologies and taxonomies.

References

1. Dines Bjørner. Manifest Domains: Analysis & Description. *Formal Aspects of Computing*, 29(2):175–225, March 2017. Online: 26 July 2016.
2. Dines Bjørner. Container Terminals. www.imm.dtu.dk/~dibj/2018/yangshan/maersk-pa.pdf. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, September 2018. An incomplete draft report; currently 60+ pages.
3. Dines Bjørner. Domain Analysis & Description – Principles, Techniques and Modeling Languages. *ACM Trans. on Software Engineering and Methodology*, 28(2):66 pages, March 2019. www.imm.dtu.dk/~dibj/2018/tosem/Bjorner-TOSEM.pdf.
4. Dines Bjørner. *Domain Science & Engineering – A Foundation for Software Development*. EATCS Monographs in Theoretical Computer Science. Springer, Heidelberg, Germany, January 2020. xii+346 pages. A revised version of this book is [8]. https:///doi.org/10.1007/978-3-030-73484-8.
5. Dines Bjørner. Domain Analysis & Description. In *Theoretical Aspects of Computing, ICTAC 2025*, number 16237 in Lecture Notes in Computer Science, pages 39–66. Springer, November 24–28 2025. A Tutorial. DOI 10.1145/3796228.

6. Dines Bjørner. Transport – A Domain Description. Technical report, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, 12 June 2025. https://www.imm.dtu.dk/~dibj/2025/transport/main.pdf.
7. Dines Bjørner. The $\mathbb{DOMAIN}$ method. Technical report, DTU Compute, Technical University of Denmark, Fredsvej 11, DK-2840 Holte, Denmark, March 2026.
8. Dines Bjørner. Domain Science & Engineering, a Primer[19] Technical University of Denmark. Significantly revised edition of [4]. xii+211 pages., February 2026 To be submitted [2026/2027].
9. Dines Bjørner and Mikhail Chupilko. : . Translation of [8]. [Book], To be submitted [2026/2027].
10. Dines Bjørner and Yang ShaoFa. . Translation of [8]. [Book], To be submitted [2026/2207].
11. Douglas B.Lenar and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Boston, MA, USA, 1989. ISBN 978-0201517521.
12. M. Bunge. *Treatise on Basic Philosophy: Ontology I: The Furniture of the World*, volume 3. Reidel, Boston, Mass., USA, 1977.
13. M. Bunge. *Treatise on Basic Philosophy: Ontology II: A World of Systems*, volume 4. Reidel, Boston, Mass., USA, 1979.
14. R.M. Burstall and Peter J. Landin. Programs and their proofs: An algebraic appraoch. *Machine Intelligence*, 4:17–43, 1969.
15. Tom Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 43(5-6):907–928, 1993.
16. Carl A. Gunter and Dana S. Scott. Semantic domains. In J.v. Leeuwen, editor, *[35] — vol.B.*, pages 633–674. North-Holland Publ.Co., Amsterdam, 1990.
17. Charles Anthony Richard Hoare. *Communicating Sequential Processes*. C.A.R. Hoare Series in Computer Science. Prentice-Hall International, London, England, 1985. Published electronically: `usingcsp.com/cspbook.pdf` (2004).
18. A. Katifori, C.Halatsis, G. Lepouras, and C. Vassilakis amd E. Giannopoulou. Ontology Visualization Methods – A Survey. *ACM Computing Surveys*, 39(4), 2007.
19. Peter J. Landin. The Mechanical Evaluation of Expressions. *Computer Journal*, 6(4):308–320, 1964.
20. Peter J. Landin. A Correspondence Between ALGOL 60 and Church's Lambda-Notation (in 2 parts). *Communications of the ACM*, 8(2-3):89–101 and 158–165, Feb.-March 1965.
21. Peter J. Landin. A Formal Description of ALGOL 60. In *[34]*, pages 266–294, 1966.
22. Peter J. Landin. A Lambda Calculus Approach. In L. Fox, editor, *Advances in Programming and Non-Numeric Computations*, pages 97–141. Pergamon Press, 1966.
23. Peter J. Landin. The Next 700 Programming Languages. *Communications of the ACM*, 9(3):157–166, 1966.
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. John McCarthy. Recursive Functions of Symbolic Expressions and Their Computation by Machines, Part I. *Communications of the ACM*, 3(4):184–195, 1960.
26. John McCarthy. Towards a Mathematical Science of Computation. In C.M. Popplewell, editor, *IFIP World Congress Proceedings*, pages 21–28, 1962.
27. John McCarthy. A Basis for a Mathematical Theory of Computation. In *Computer Programming and Formal Systems*. North-Holland Publ.Co., Amsterdam, 1963.
28. Robert E. Milne and Christopher Strachey. *A Theory of Programming Language Semantics*. Chapman and Hall, London, Halsted Press/John Wiley, New York, 1976.
29. J.M Padial, A. Miralles, and et.al. I De la Riva. The integrative future of taxonomy. *Front Zoology*, 7(16), 2010. https://doi.org/10.1186/1742-9994-7-16.
30. John C. Reynolds. *Theories of Programming Languages*. Cambridge University Press, Edinburgh Building, Shaftesbury Road, Cambridge CB2 2RU, England, 10 December 1998. ISBN 0521594146, 512 Pages
31. Dana S. Scott. Outline of a Mathematical Theory of Computation. In *Proc. 4th Ann. Princeton Conf. on Inf. Sci. and Sys.*, page 169, 1970.
32. Dana S. Scott. Logic and programming languages. *Communications of the ACM*, 20(9):634–641, 1977.
33. Dana S. Scott and Christopher Strachey. Towards a mathematical semantics for computer languages. In *Computers and Automata*, volume 21 of *Microwave Research Inst. Symposia*, pages 19–46, 1971.
34. T. B. Steel, editor. *Formal Language Description Languages for Computer Programming, IFIP TC-2 Working Conference, 2964, Baden*. North-Holland Publ.Co., Amsterdam, 1966.
35. J. van Leeuwen, editor. *Handbook of Theoretical Computer Science, Volumes A and B*. Elsevier, 1990.

[19] This book is currently being translated into Russian, [9], by Dr. Mikhail Chupilko and his colleagues, ISP/RAS (Institute of Systems Programming, Russian Academy of Sciences), Moscow and into Chinese, [10], by Dr. Yang ShaoFa, IoS/CAS (Institute of Software, Chinese Academy of Sciences), Beijing.