Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

Topic 26
Jackson's Description Principles

- The **prerequisite** for following this (part of the) lecture is that you have followed the lectures on Description Principles and On Defining and On Definitions.
- The **aims** are
  - ⋆ to introduce Jackson's concepts of designations, definitions and refutable assertions, and
  - ⋆ to provide formal tools for the expression of manifestations of these concepts.

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- The **objective** is
  - ⋆ to ensure that the developer becomes a professional specifier.
- The **treatment** is systematic to formal.

We build on ideas eloquently expressed in:

> Software Requirements & Specifications
> a lexicon of practice, principles and prejudices
> Michael Jackson
> ACM Press and Addison-Wesley Publishing Company
> ISBN 0-201-87712-0, 1995; xvi+228 Pages

- Since all we do is
  - ⋆ construct, analyse and compare descriptions
  - ⋆ we shall analyse
  - ⋆ the concept and constituents of descriptions.[8]

[8] We shall here use the term description as also covering the terms prescription and specification.

- A description is about
  - ⋆ manifest individuals, i.e., phenomena and concepts.
  - ⋆ Some of these represent, or are intended to represent, facts;
  - ⋆ others represent mental constructions, i.e., concepts.
  - ⋆ A description includes *designations*, definitions and refutable descriptions.
  - ⋆ A description can either be formal or informal.
  - ⋆ A description sets a scope and a span, and a description
  - ⋆ expresses moods.
- By an individual we mean a physically manifest phenomenon.

Phenomena, Facts and Individuals

- Phenomena are what appears to exist.
  Domain phenomena are built up from facts about individuals.
- A fact is a simple truth about the world:
  It is the smallest unit of observation.
- Large and complex observations and truths can be broken down into assertions about several facts.

- A fact involves one or more *individuals*. Anything can be an individual. Each individual is identical to itself but distinct from all other individuals.
- If x is identical to y, then x and y are the same individual: $x = y$.
- If we say that x is similar or equivalent to y then x and y are distinct individuals that share some property, characteristic, attribute or quality.

Designations

Characterisation 7.134 • By a *designation description* (for short: *designation*) we syntactically mean a textual triple:

- ⋆ a *designation term*,
- ⋆ a *designation recognition rule*,
- ⋆ and a *designation identification*

. ■

Characterisation 7.135 A *designation term* is a simple, i.e., atomic name. ■

Characterisation 7.136 A *designation recognition rule* is a text which purports to *designate* something. ■

Characterisation 7.137 • The *designation identification* links the *designation recognition rule* to that something (mentioned in the previous characterisation)

. ■

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- To create a *designation* we thus write down two items:
  - ⋆ a *designation description*
  - ⋆ and a *designation identification*.
- The *designation description* is usually of the form:
  - ⋆ *Designated term:* `dt`.
  - ⋆ *Recognition rule:* `dt`'s satisfy the following informally stated properties: ... — where all other words, i.e., terms, are assumed to be well-known!

- That is: *recognition rules* are expressed only in terms that are otherwise well understood, i.e., part of the folklore.
- If a recognition rule thus contains a term that is elsewhere *defined*,
  - ⋆ then it is not a recognition rule
  - ⋆ but becomes a *definition*.

Example 7.79 *Rail Units, I:* We give the first in a series of examples relating to rail nets.

- *Designated term:* `rail_unit`, or just `U`.
- *Recognition rule:* A `rail_unit` is a composition of an even number of parallel positioned rails (long, narrow, profiled iron bars) separated such that one can always identify pairs of rails of the composition that are at a specified distance (the rail gauge), and otherwise held together by a set of ties.

When we write `rail_unit` we mean the extensional meaning (the **type**) of that term. ■

Example 7.80 *Rail Units, II:* We give the second in a series of examples relating to rail nets.

- *Designated term:* `linear_rail_unit`, or just `U`, for which a predicate, to be assumed, holds: `is_linear_rail_unit(u)` for all `u` in the extension `U`, i.e., `u:U`.
- *Recognition rule:* A `linear_rail_unit` is a pair of parallel positioned rails (long, narrow, profiled iron bars) separated at a specified distance (the rail gauge) and held together by a set of ties. (The predicate `is_linear_rail_unit(u)` "arises" from the recognition rule.)

Observe that the designation term and the recognition rule dealt with a concrete concept for which the immediate concretisation points to a set of a phenomena. What is being designated and to be recognised is any one member of this set. ■

- The latter example is a specialisation of the former.
- The former example expressed: ... *composition of an even number of parallel positioned 'rails'* ...
- thus allowing for, say, two pairs as in a switch or in a simple crossover rail unit.
- A *designation identification* is usually of the form:
  - ⋆ "That 'thing' (u) there is a U.
  - ⋆ So is that 'thing' (u') over there!" —
- thus physically pointing out a *designation set*, that is, instances of values that together become part of a type.

Some Observations

- In general a *designation description* can be formalised;
- but one cannot formalise the *identification* — as it relates a formal world to an inherently informal world.
- As illustrated in the two previous examples,
- the latter of which was a specialisation of the former,
- a *designation* may extensionally denote a class (of things) which can be subdivided into subclasses.

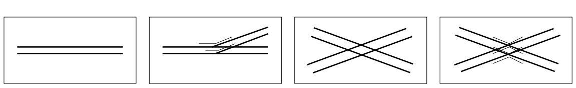

Figure 7.12: Roughly drawn rail units

Example 7.81 *Rail Units, III:* We give the third in a series of examples relating to rail nets.

- *Designated term:* `rail_unit`.
- *Recognition rule:* — as for the above, previous `rail_unit` example.
- *Designated term:* `linear_rail_unit`.
- *Recognition rule:* — as for the above, previous `linear_rail_unit` example, and additionally: Thus a `linear_rail_unit` has two "ends" and a single (two-way) link between these ends. Any other `rail_unit` may be `connected` to either of these ends (and if so, then the end is called a `connector`).

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- To support textually expressed recognition rules it is often useful to deploy graphic means as in Fig. 7.12.
- Even photographic means could be used.
- And then one usually could show several photos of variations of the same kind of designated individuals.

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- *Designated term:* `switchable_rail_unit`.
- *Recognition rule:* — as for `rail_unit` just above, and additionally: A `switchable_rail_unit` has three `connectors` which "define" (allow, permit) two two-way links through the `switchable_rail_unit.`

- *Designated term:* `simple_crossover_rail_unit`.
- *Recognition rule:* — as for `rail_unit` just above, and additionally: A `simple_crossover_rail_unit` has four `connectors` which "define" (allow, permit) two two-way links through the `simple_crossover_rail_unit.`

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- *Designated term:* `switchable_crossover_rail_unit`.
- *Recognition rule:* — as for `rail_unit` just above, and additionally: A `switchable_crossover_rail_unit` has four `connectors` which, together with the switching ability of the unit, "define" (allow, permit) four two-way links through the `switchable_crossover_rail_unit.`

- *Designated term:* `connector`.
- *Recognition rule:* A `connector` is that which allows two `rail_unit`s to be connected, "end to end".

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- Further recognition rules deal with `connector`s and `rail_unit`s.
- *A `connector` is any "end" of a `rail_unit` to which other `rail_units` may be 'connected'.*
- *Any one `connector` is shared by at most two `rail_units`.*
- *A `rail_unit` is mutually exclusive either a `linear_rail_unit` or a `switchable_rail_unit` or a `simpl_crossover_rail_unit` or a `switchable_crossover_rail_unit.`*

See axioms [2,3] in a formalisation given later. ■

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- As we shall see in a later lecture,
- doubt may be raised as to whether some of the text parts of the above recognition rules express assertions.
- (Designations state facts, not assertions.)
- For example: "has two ends" (or three, or four).
- These parts are part of recognition rules,
- but what about:
  - ⋆ "any one connector is shared by at most two rail units"?
- Or the part right after it:
  - ⋆ "... mutually exclusive ..."?

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- For the last (the mutual exclusion) we can claim it to be a fact,
- hence it is part of a recognition rule.
- For the "at most two", the case is a bit more complicated.
- And then,
  - ⋆ when we formalise the whole thing, as we shall see below,
  - ⋆ and when such a formalisation is compared to that of a refutable assertion,
  - ⋆ we shall see that, formally speaking, the difference is almost invisible.

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- ⋆ Thus we must be prepared for the eventuality
  - ◇ that the pragmatics of making a distinction between designations, definitions and refutable assertions
  - ◇ can not be carried visibly over into formal models of the things being designated or defined, or for which assertions are expressed.

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

Formalisation

- Which of the above alternative ways of designations are we going to formalise?
- Typically we formalise a *designation description* as shown in the next example.
- It allows for the general case of several alternatively expressible designations.

Example 7.82 *Units:*

- We choose to introduce just one type (i.e., sort), U, for rail units,
- and we choose to let
  - ⋆ (recognition rule-oriented) observer functions
  - ⋆ and suitable axioms

  help separate rail units into a partition of its more specialised rail units.

Let W denote a concept of undirected "ways" through a unit (Fig. 7.13).

type
 U, C, W
value
 obs_Cs: U→C-**set**
 obs_Ws: U→W-**set**
 is_linear,is_switch,is_simpl_cross,is_switch_cross: U→**Bool**

axiom
[1] $\forall\ u:U,\ \exists\ c,c',c'',c''':C \bullet$
 $\textbf{card}\{c,c',c'',c'''\}=4 \wedge obs\_Cs(u)\subseteq\{c,c',c'',c'''\} \Rightarrow$
  $is\_linear(u)\Rightarrow obs\_Cs(u)\subseteq\{c,c'\}\wedge\textbf{card}\ obs\_Ws(u)=1 \vee$
  $is\_switch(u)\Rightarrow obs\_Cs(u)=\{c,c',c''\}\wedge\textbf{card}\ obs\_Ws(u)=2 \vee$
  $is\_simpl\_cross(u)\Rightarrow obs\_Cs(u)=\{c,c',c'',c'''\}\wedge\textbf{card}\ obs\_Ws(u)=2 \vee$
  $is\_switch\_cross(u)\Rightarrow obs\_Cs(u)=\{c,c',c'',c'''\}\wedge\textbf{card}\ obs\_Ws(u)=4,$

[2] $\forall\ c:C \bullet \textbf{card}\{\ u \mid u:U \bullet c \in obs\_Cs(u)\ \} \leq 2,$

[3] $\textbf{let}\ lus=\{|u:U\cdot is\_linear(u)|\},\ sus=\{|u:U\cdot is\_switch(u)|\},$
  $cus=\{|u:U\cdot is\_simpl\_cross(u)|\},\ scus=\{|u:U\cdot is\_switch\_cross(u)|\}$
 $\textbf{in}\ lus \cap sus = lus \cap scus = lus \cap scus = \{\} \wedge$
 $sus \cap cus = sus \cap scus = cus \cap scus = \{\}\ \textbf{end}$

- U denotes the possibly infinite set of all *designations* satisfying the rail_unit *recognition rule*.
- C stands for the possibly infinite set of all *designations* satisfying the connector *recognition rule*.
- W denotes the concept of way (or link).
- The predicates
  - ⋆ is_linear, is_switch, is_simpl_cross and is_switch_cross

  further constrain the rail_unit *recognition rule*,
- as indicated by the **axiom**.

Observe how the formalisation fits with the narration. ■

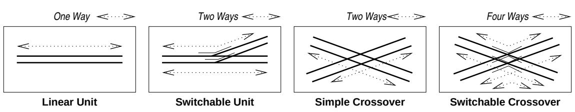

Figure 7.13: Undirected ways through units

Observer Functions and Identification

- The *observer functions*, eg. obs_Cs and obs_Ls, are the closest counterpart to the undefined terms of *recognition rules* that we have.
- These *observer functions* cannot be defined.
- They are postulated. They "arise" as the result of *identification*.

- Given an actual *universe of discourse* — to which the above *designations* are said to apply —
  - ⋆ one can now "define" these *observer functions*
  - ⋆ by "walking out and into" the *universe of discourse* and
  - ⋆ by providing the *identification*.
- To get a better grasp of possible relationships between recognition rules and observer functions let us give a further example.

Example 7.83 *Rail Net: Lines and Stations:*

- *A railway net consists of [one or more] lines and [two or more] stations.*
- *A line is a linear sequence of one or more linear rail units.*
- *A station is any composition (connection) of rail units. [A line connects exactly two distinct stations.]*
- The above (excluding the bracketed parts) may be claimed to be a definition, and we shall soon turn to a treatment of definitions.
- But we claim that each of the (unbracketed parts of the) above notions can be designated.
- The *italic text* above may not look like *recognition rules*, but they are!

- One can indeed "walk out, into" the railway system domain and, with a sweeping hand, express:
  - ⋆ *That "thing" there is a linear rail unit. That one, next to it, is likewise.*
  - ⋆ *That particular sequence of those two linear units I just pointed to forms (part of) a line.*
  - ⋆ *This "thing" here is a rail unit of a station.*
  - ⋆ *It is a crossover.*
  - ⋆ *Here is the connector that separates a line from a station.*
  - ⋆ *No rail unit is both a rail unit of a line and of a station, or of two otherwise distinct lines or stations.*


```
type
  N, L, S, U, C, L
value
  obs_Ls: N → L-set, obs_Ss: N → S-set,
  obs_Us: (N|L|S) → U-set
axiom
  ∀ n:N, l,l′:L, s,s′:S •
    card obs_Ls(n) ≥ 1 ∧ card obs_Ss(n) ≥ 2 ∧
    {l,l′} ⊆ obs_Ls(n) ∧ {s,s′} ⊆ obs_Ss(n) ⇒
      l≠l′ ⇒ obs_Us(l) ∩ obs_Us(l′) = {} ∧
      s≠s′ ⇒ obs_Us(s) ∩ obs_Us(s′) = {} ∧
      obs_Us(l) ∩ obs_Us(s) = {} ∧
      exactly_two_distinct_stations(l,obs_Ss(n))
```

■

Mathematical and Computing Entities

- From a formal point of view, i.e., from the points of view of mathematics and computer and computing science, which are the kinds of designations?
- Which kinds of mathematical entities are they?
- Or, in the jargon of computing: Which types of computer and computing science entities are they?

Mathematical Entities

- When viewed mathematically, are the designations scalars, like numbers (integers, reals (or complex), rationals, transcendental), or truth values, etc.?
- Or are they composite, like sets, Cartesians, lists, maps or functions (or algebras, etc.)?
- And if the latter, then which are their component elements?

Computing Entities

- When viewed computer and computing sciencewise, are the designations values (of, for example, the above-mentioned mathematical kind (where models are like algebras))?
- Or are they types of these, i.e., types as we know them in computer science: simple lattices, Scott domains or otherwise?
- Or are they events — whatever they are?
- If lists, then what do these lists model: behaviours (of processes, in terms of traces of actions and/or events, etc.), or other?
- In any case, when we model computing entities (values, types, events, (process) behaviours, semantic algebras), then we model them in terms of the above-mentioned mathematical entities.

Awareness

- We do not intend to give a full answer to the question:
- which kind (type) of formal entities can designations be modelled by?
- We only advise that the practicing, professional software engineer be reasonably well-versed in these matters:
  - ⋆ modelling designations formally in terms of mathematical entities — perhaps couched in the computing jargon of, for example,
    - ⋄ types and values,
    - ⋄ events and (process) behaviours,
    - ⋄ semantic algebras or other.

Some Guidelines

- In the following we shall try to stay clear of the deeper problems of conceptual modelling as currently studied in computer and computing science
- But we must — since it cannot be avoided (if we are to cover any ground at all) — postulate some possibilities of concept modelling (since that is what it, in essence, is all about).
- We do so in order to identify some designation (etc.) principles and techniques.

- Some *designations* are specific, "one of a kind" things ("that rail unit there"), or they are specific events, or specific behaviours (etc.).
- Or *designations* are types or models (i.e., algebras) over these.
- Some designations are those of components of contexts and states.
- Contexts are entities whose properties — whose attributes, whose values — remain static over time.
- States are such whose values change with time — they are dynamic.

- Examples of contexts are: *road* and *rail nets* (when viewed topologically and over time periods that are not too large), similarly for airline *timetables*.
- Examples of states are: *road*, rail and air traffic, hence *trucks*, trains and aircraft, and the *hubs* (where they meet and passengers embark and disembark, or where *freight receipt*, *transfer* and *delivery* can take place).
- A *bill of lading* is somewhere "in between": The *route* along which a *freight item* should be conveyed is, we assume, static, but the *bill of lading* is (probably) marked ("updated") to show the (believed) current (or last recorded) position of the *freight item* for which it is the *bill of lading*.

- But, in any case, all these examples can be modelled as typed, usually composite values, fixed or variable.
- These values we consider inert:
- They do not change by themselves.
- Some external action has to change them.
- Designations thus could, alternatively, be the, or a, specific action (active phenomenon, *function*, operation, task, procedure) of *inscribing a freight item for conveyance with a logistics firm, loading it onto the truck, unloading it*, etc.
- In this case we refer to the designations as function values.
- Or designations could be the events of *truck departure* from, resp. *arrival at a hub*.

- Or designations could be part or the entire (process) behaviour *from inscription to unloading* described in terms of the specific sequence of inert and dynamic phenomena.
- In this case we refer to the designations as behaviours (a certain kind of function values).

• • •

- The above explication presents just one kind of conceptual modelling approach.
- It is "slanted" towards `CSP` and `RAISE`.

Principles 7.31 *Conceptual Frameworks:* When setting out on a first description, identify which conceptual modelling framework you intend to work within. ■

Techniques 24 *Framework Model:* Among the conceptual modelling frameworks that can be believably offered are:

- `B`, `VDM`, `Z`
- `RAISE`
- `CafeOBJ`, `CASL`

We list three groups. The first are solely model-oriented, the last is solely property-oriented (in the algebraic semantics style). `RAISE` basically offers both styles. . ■

A "Large" Example

Example 7.84 *Transport:*

type
  Fre, BoL, Trk, Hub
value
  m,n:**Nat**
  obs_BoL: Fre → BoL
axiom
  m>0 ∧ n>0
type
  HIdx = {| 1..m |}, TIdx = {| 1..n |}
channel
  { ht[h,t]:(Fre×BoL) · h:HIdx, t:TIdx }

value
  truck: t:TIdx → Trk → **Unit** → **in**,**out** {ht[h,t]|h:HIdx} **Unit**
  truck(t)(tr) ≡ (...; **let** h = ... **in** truck(t)(unload(tr)(h,t)) **end**)⌈⌉(...)
  hub: h:HIdx → Hub → **in**,**out** {ht[h,t]|t:TIdx} **Unit**
  hub(h)(hu) ≡ ⌈⌉ {hub(j)(add(ht[h,t]?)(hu))|t:TIdx} ⌈⌉ (...)
  load: Fre×BoL → t:TIdx × h:HIdx → **in**,**out** ht[h,t] Trk
  unload: Trk → (h:HIdx × t:TIdx) → **out** {ht[i,t]|i:HIdx} Trk
  unload(tr)(h,t) ≡
   let (f,b):(Fre×BoL)·ii(tr,f,b,h) **in** ht[h,t]!(f,b);rem(f,b)(tr) **end**
  ii: tr:Trk×f:Fre×b:BoL×h:HIdx → **Bool** /∗ b of f of tr mentions h ∗/
  rem: Fre×BoL → Trk → Trk
  add: Fre×BoL → Hub → Hub

. ■

Formal Modelling

- It is not the intention of the present section to suggest neither principles nor techniques for exactly how to formally model designations.
- Such principles and techniques were and are the subject of earlier lectures and later lectures.
- Instead it is the aim of the current section to point out the following principle:

Principle 7.32 *Choice of Description Style:* In considering which things to designate and to describe, let the informal description style be governed by the chosen, most suitable formal description style. ■

Discussion: Designations

Principle 7.33 *Type Versus Value (Instantiation) Modelling:*

- In building up designations —
- within the conceptual paradigm (i.e., framework), say, of `RAISE` —
- one must decide
  - ⋆ whether one is trying to designate the type of all those things that are similar to a few designations, i.e., satisfy their common recognition rule — as for Fre, BoL, Trk and Hub above —
  - ⋆ or one is trying to define only a specific value (one particular thing) — as for hub, truck, load, unload, ii, rem or add above

. ■

Explicit Definitions

- First, designations represent one form of definition.
- In any description all terms that are special to the *universe of discourse* being described must be *defined*
  - ⋆ either through *designations*
  - ⋆ or by *explicit definitions.*

Definitions: "The Narrow Bridge"

- The example above contained only designatable quantities, or so we claimed.
- But that may not always be possible, or convenient.
- Sometimes it is easier to *define* a `term` by giving it a *definition*, but (notably) using already *defined* (including *designated*) `terms`.

Example 7.85 *Rail Lines:*

- *A railway line is a linear, acyclic, sequence of linear rail units, that is, a sequence where adjacent elements of the sequence are rail units that share exactly one connector.*

type

U, C
L1′ = U-**set**
L1 = {| us:L1′ · wf_L1(us) |}
L2′ = U*
L2 = {| us:L2′ · wf_L2(us) |}

value

obs_Cs: U → C-**set**
wf_L1: U-**set** → **Bool**
wf_L1(us) ≡ ∃ ul:U* · **len** ul = **card** us ∧ **elems** ul = us ∧ wf_L2(ul)
wf_L2: U* → **Bool**
wf_L2(ul) ≡
  ∀ u:U · u ∈ **elems** ul ⇒ is_linear(u) ∧
  ∀ i:**Nat** · {i,i+1}⊆**inds** ul ⇒
    card(obs_Cs(ul[i]) ∩ obs_Cs(ul[i+1]))=1 ∧
    len ul > 1 ⇒ obs_Cs(**hd** ul) ∩ obs_Cs(ul[**len** ul]) = {}

- Here the terms linear rail unit and connector are already designated terms;
- and the concept of a line is defined in terms of those designations.
- Recall an earlier axiom which stated that for any connector there are at most two units sharing that connector.
- That axiom is assumed above.

■

Techniques 25 *Definitions* must be expressed in terms of designated and/or defined terms, and ultimately definitions must be based on designations. ■

Principles 7.34 Although it may seem utterly obvious we urge the developer to *develop alternative definitions*. ■

- Here we developed alternatives L1 and L2.
- Although one may claim that lines could be designations, since, in a sense, they are tangible, we have here defined the concept of line.
- We follow the advice of Michael Jackson when we raise it to a principle:

Principle 7.35 *The Narrow Bridge:* Seek as few designations as possible: Enough to define all the other possibly designatable as well as all the desirable abstract, nontangible concepts. ■

Definition of Abstract, Intangible Concepts

- We shall next define some abstract, intangible concepts.

Example 7.86 *Path:*

- *A rail unit defines a concept of path.*
- *A path (through a rail unit) represents the ability for a train to move along that path, through the rail unit.*[9]
- (Notice that we neither, at present, designate nor define what we mean by train or move.)
- *We define (i.e., we model) a path as a pair of connectors.*
- (Based on the definition of path we then proceed to define further intangible, i.e., not physically manifest, concepts.)

[9] When we earlier spoke of a concept of link, that could, in a sense, be seen as an abstract concept, much like a pair of opposite direction paths.

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- *With any rail unit we can associate a state: a possibly empty set of paths.*
- *And, finally, we can associate with any rail unit its state space: the set of all the states that a rail unit may "be in" over its lifetime as a rail unit.*

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

type
$\quad$ U, C, P = C×C
$\quad$ Σ = P-**set**, Ω = Σ-**set**
axiom
$\quad$ ∀ (c,c′):P • c≠c′
value
$\quad$ obs_Σ: U → Σ, obs_Ω: U → Ω

We shall soon look at possible relations between designations and definitions of the railway system. ∎

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

How Much, How Little to Define?

- The question is now: how much to *define?*
- For the *universes of discourse* of *requirements* and *software design* there is a *utility* concept: We know what we aim at, so we *define* at least that; but why more than that?
- For the *universes of discourse* of *application domains* we do not, as a matter of principle, know what we are aiming at, so here we go for more: as long as some interesting ideas are being *defined*, then why not?

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- It is only, we believe, in "roaming around" and experimenting with *definitions*, and, later, with *refutable assertions*, that we may discover the *most interesting, most relevant, most fitting* domain concepts.
- It is in this exploration, and based on *definitions* that we can expect to build theories of specific domains.

Principle 7.36 *Exploring Theory Bases:* In constructing *domain models*, i.e., *descriptions* of *domains*, *designate* according to the *"narrow bridge"* principle, and then *define* as many abstract concepts as long as their *definition* (and those of attendant *refutable assertions* help) "reveals" further concepts. ∎

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

A last example may illustrate the previous point.

Example 7.87 *Routes:*

- *(i) A route is a sequence of connected rail units.*
- *(ii) A route is an open route if the states of all rail units of the route are such that there are paths in those states that also connect (in one direction or another).*
- *(iii) Given a unit within a station define as reachable from the station unit all those lines that are incident upon and (thus) emanate from that station, and for which there is a route between the unit and the line.*

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- *(iv) Given any two rail units, an 'origin' and a possible 'destination', of a railway network define a more general notion of reachability, one that is satisfied if there are open routes from the 'origin' to the 'destination'.*
- And so forth.
- The present example illustrates the definition of a number of (viz.: four) concepts without a priori making sure that these concepts will serve a useful purpose.

∎

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

Discussion: Definitions

- Choosing an appropriate balance between a reasonably small set of designations and an appropriate (large, "rich") set of definitions is an art!
- No definite advice can be given, other than perhaps that given above, which, when followed, helps ensure an "appropriate balance".
- We have now covered two of the informal and formal description principles and their formalisation techniques.
- From the examples so far we can already now conclude that one cannot see from the formulas whether a description formalises a designation or a definition!

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- That is, the *pragmatics* of distinguishing between designation and definitions is lost in the formulas.
- Thus it is important with some *syntax* to start the informal and formal texts in order to alert the reader to which is what!

Refutable Assertions

Characterisation 7.138 By a *refutable assertion* we mean a claim that may be shown to be wrong. ■

Designation and Definition Assertions

- In Example 7.83, where we described railway nets of lines and stations, we put in some bracketed sentence parts.
- Those parts did not designate manifest things.
- They expressed suitable, desirable or other relations of designations (and, later, of definitions).
- We shall refer to such constraints as *refutable assertions.*

Example 7.88 *Rail Net Assertion:*

- That *a line connects exactly two distinct stations* is one such example of a refutable assertion.
- It implies that there must be *at least two stations in a railway net* — another refutable assertion — and *at least one line* — yet another refutable assertion.
- We asserted elsewhere that *every connector is shared by at most two (otherwise distinct) rail units;* that *linear rail units have two connectors and define one link;* that *switchable rail units have three connectors (and define two links);* etc.
- These sentence parts can all be considered refutable assertions.

■

Techniques 26 *Refutable assertions* must be expressed in terms of either designated or defined terms, or both kinds of terms. ■

- Designations, in a sense, are facts.
- And facts cannot be refuted.
- Definitions are definitions, and as such must be accepted.
- But definitions may seem general and may thus need to be characterised ("tied down") through some form of constraining assertions.

Example 7.89 *Unit States:*

- We continue the rail unit path, state and state space example given earlier.
- *The paths of a rail unit, i.e., the paths of any of its states, must mention only connectors of that unit.*

axiom

$$\forall\ u{:}U,c{:}C,\sigma{:}\Sigma \cdot$$
$$\quad obs\_\Sigma(u) \in obs\_\Omega(u) \wedge$$
$$\quad \sigma \in obs\_\Omega(u) \Rightarrow \forall\ (c,c'){:}C{\times}C \cdot (c,c') \in \sigma \Rightarrow \{c,c'\} \subseteq obs\_Cs(u)$$

- The above assumes an earlier axiom expressing the fact that the connectors of a path are always distinct.

■

- This is a refutable assertion:
- One might think of cases where there could be connectors of a path in some unit's state that were not connectors of that unit, or could one?
- We basically will not know till someone one day shows us an understanding of a railway net that "favours" such an interpretation!

Analysis

- What was claimed, above, as examples of refutable assertions, certainly were assertions.
- Whether they will be refuted remains to be seen, but they are potentially refutable.
- Take the example of *any connector* being "constrained" *to be shared by at most two rail units.*
- If one could show, in the future, that there were indeed connectors that were shared by three (or more) rail units, then we have indeed refuted the assertion.
- But is it likely?

Example 7.90 *Stations and Lines: A station is connected to at least one line.* ■

- Refuting this assertion would amount to allowing railway nets with completely isolated stations.
- Is that likely?

Example 7.91 *Stations and Open Routes: For any station, there is at least one potentially open route from, and at least one such route to, some other station of the net.* ■

- If this assertion is refuted then there will be at least one station from, or into which one cannot "travel"!
- Is that likely?

"Dangling" Assertions

- It is much too easy to write down texts that "weave a web" of pseudodesignations, pseudodefinitions and pseudoassertions.

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

Example 7.92 *Freight Transport Bill of Lading:*

- Suppose the following text was all we had: *A bill of lading lists the transport hubs where the referenced freight items first loaded onto a conveyor are being transferred from one conveyor to a next, and are finally unloaded.*
- *A transport according to a bill of lading does not visit a hub more than at most once.*
- Then what were we to mean?

Suppose further that we went straight ahead and formalised the above *italic text:*

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

type
Hub, HIdx, Fre, Con
BoL′ = (F × (HIdx × HIdx* × HIdx))
BoL = {| b:BoL′ · wf_BoL(b) |}
Transport = Fre × (Con × Hub)*
value
wf_BoL: BoL′ → **Bool**
wf_BoL(f,(o,hl,d)) ≡ **card**({o,d} ∪ **elems** hl)=2+**len** bol

obs_Fn: Fre → FIdx
load,unload: Fre × BoL → Con → Con
transfer: Fre × BoL → (Con × Con) → (Con × Con)
M: BoL → Transport-**infset**

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

- What is the problem?
- The problem is that we have taken the liberty of mentioning such possibly designatable phenomena as Hub, Hub index, Freight, Freight index, Conveyor, load, unload and transfer, and given signature to the function values without having attempted the slightest bit of designations!
- We have likewise defined a Meaning function which defines the meaning of a bill of lading as a possibly infinite set of defined *Transports*.
- The assertion *does not visit a hub more than at most once* is thus pretty meaningless.

■

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

Techniques 27 *Dangling Assertions:* Make sure all terms of an assertion are either designated, defined or otherwise bound in the assertion. ■

Example 7.93 *Transport Subtypes:* With respect to the previous example we should thus also have properly subtyped the *Transport* type, etc. ■

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

Discussion: Refutable Assertions

- Assertions are like axioms. They are self-evident truths — until refuted by observations made in the referenced universe of discourse.
- Assertions are thus not facts, nor are they theorems.
- Theorems are predicate expressions that can be proven to follow from designations, definitions and (the axioms of) refutable assertions.
- Theorems (lemmas, propositions) together with designations, definitions, the axioms of assertions and other theorems (lemmas, propositions) form a theory of the described universe of discourse.

Department of Computer Science and Engineering
Institute of Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, DK-2800 Kgs.Lyngby, Denmark

Discussion: Description Principles

- We have related the description principle concepts of designations, definitions and refutable assertions to (albeit sketchy examples of) formal specifications.
- We have, most recently above, seen how formal specifications,
  - ⋆ cf. the latest formalisation example above,
  - ⋆ without proper, necessarily informal descriptions that adhere to Jackson's principles,
- can too easily "drug" us into believing that the model identification principle can be skipped.
- We ourselves readily admit to having, far too often for anybody's good, fallen into that trap!