(2. 2.4.)

Start of Lecture 3: DOMAIN ENGINEERING

3. Domain Engineering

- We focus on the *facet* components of a domain description
- and shall not here cover such aspects of domain engineering as
  - stakeholder identification and liaison,
  - domain acquisition and analysis,
  - terminologisation,
  - verification, testing, model-checking, validation and
  - domain theory formation.

(3. Domain Engineering)

- By understanding, first, the *facet* components
  - the domain engineer is in a better position to effectively
  - establish the regime of stakeholders,
  - pursue acquisition and analysis,
  - and construct a necessary and sufficient terminology.
- The domain description components each cover their domain facet.

(3. Domain Engineering)

- We outline six such facets:
  - intrinsics,
  - support technology,
  - rules and regulations,
  - scripts (licenses and contracts),
  - management and organisation, and
  - human behaviour.
- But first we cover a notion of business processes.

3.1. Business Processes

- By a business process we understand
  - a set of one or more, possibly interacting behaviours
  - which fulfill a business objective.
- We advocate that domain engineers,
  - typically together with domain stakeholder groups,
  - rough-sketch their individual business processes.

Example 6 – Some Transport Net Business Processes

- With respect to one and the same underlying road net
- we suggest some business-processes
- and invite the reader to rough-sketch these.

27 **Private citizen automobile transports:** Private citizens use the road net for pleasure and for business, for sightseeing and to get to and from work.

A private citizen automobile transport "business process rough-sketch" might be:

> A car owner drives to work: *Drives out, onto the street, turns left, goes down the street, straight through the next three intersections, then turns left, two blocks straight, etcetera, finally arrives at destination, and finally turns into a garage.*

28 **Public bus (&c.) transport:** Province and city councils contract bus (&c.) companies to provide regular passenger transports according to timetables and at cost or free of cost.

A public bus transport "business process rough-sketch" might be:

> A bus drive from station of origin to station of final destination: *Bus driver starts from station of origin at the designated time for this drive; drives to first passenger stop; open doors to let passenger in; leaves stop at time table designated time; drives to next stop adjusting speed to traffic conditions and to "keep time" as per the time table; repeats this process: "from stop to stop", letting passengers off and on the bus; after having (thus, i.e., in this manner) completed last stop "turns" bus around to commence a return drive.*

29 **Road maintenance and repair:** Province and city councils hire contractors to monitor road (link and hub) surface quality, to maintain set standards of surface quality, and to "emergency" re-establish sudden occurrences of low quality.

30 **Toll road traffic:** State and province governments hire contractors to run toll road nets with toll booth plazas.

31 **Net revision: road (&c.) building:** State government and province and city councils contract road building contractors to extend (or shrink) road nets.

- The detailed description of the above rough-sketched business process synopses now becomes part of the domain description as partially exemplified in the previous and the next many examples.

■ End of Example 6

- Rough-sketching such business processes helps bootstrap the process of domain acquisition.

3.2. Intrinsics

- By intrinsics we shall understand
  - the very basics,
  - that without which none of the other facets can be described,
  - i.e., that which is common to two or more, usually all of these other facets.

Example 7 – Intrinsics

- Most of the descriptions of earlier examples model intrinsics.
- We add a little more:

32 A link traversal is a triple of a (from) hub identifier, an along link identifier, and a (towards) hub identifier

33 such that these identifiers make sense in any given net.

34 A link state is a set of link traversals.

35 And a link state space is a set of link states.

value
  n:N
type
  [32] $\mathsf{LT}' = \mathsf{HI} \times \mathsf{LI} \times \mathsf{HI}$
  [33] $\mathsf{LT} = \{|\mathsf{lt}:\mathsf{LT}' \cdot \mathsf{wfLT}(\mathsf{lt})(\mathsf{n})|\}$
  [34] $\mathsf{L}\Sigma' = \mathsf{LT}\text{-}\mathbf{set}$
  [34] $\mathsf{L}\Sigma = \{|\mathsf{l}\sigma:\mathsf{L}\Sigma' \cdot \mathsf{wf\_L}\Sigma(\mathsf{l}\sigma)(\mathsf{n})|\}$
  [35] $\mathsf{L}\Omega' = \mathsf{L}\Sigma\text{-}\mathbf{set}$
  [35] $\mathsf{L}\Omega = \{|\mathsf{l}\omega:\mathsf{L}\Omega' \cdot \mathsf{wf\_L}\Omega(\mathsf{l}\omega)(\mathsf{n})|\}$
value
  [33] $\mathsf{wfLT}: \mathsf{LT} \rightarrow \mathsf{N} \rightarrow \mathbf{Bool}$
  [33] $\mathsf{wfLT}(\mathsf{hi},\mathsf{li},\mathsf{hi}')(\mathsf{n}) \equiv$
  [33] $\quad \exists\ \mathsf{h},\mathsf{h}':\mathsf{H} \cdot \{\mathsf{h},\mathsf{h}'\} \subseteq \omega\mathsf{Hs}(\mathsf{n}) \wedge$
  [33] $\quad\quad \omega\mathsf{HI}(\mathsf{h}) = \mathsf{hi} \wedge \omega\mathsf{HI}(\mathsf{h}') = \mathsf{hi}' \wedge$
  [33] $\quad\quad \mathsf{li} \in \omega\mathsf{LIs}(\mathsf{h}) \wedge \mathsf{li} \in \omega\mathsf{LIs}(\mathsf{h}')$

■ End of Example 7

3.3. Support Technologies

- By support technologies we shall understand
  - the ways and means by which
    - ∗ humans and/or
    - ∗ technologies
    - ∗ support
      - · the representation of entities and
      - · the carrying out of actions.

Example 8 – Support Technologies

- Some road intersections (i.e., hubs) are controlled by semaphores
  - alternately shining **red**–**yellow**–**green**
  - in carefully interleaved sequences
  - in each of the in-directions from links incident upon the hubs.
- Usually these signalings are initiated as a result of road traffic sensors placed below the surface of these links.
- We shall model just the signaling:

36 There are three colours: **red**, **yellow** and **green**.

37 Each hub traversal is extended with a colour and so is the hub state.

38 There is a notion of time interval.

39 Signaling is now a sequence, $\langle(h\sigma', t\delta'), (h\sigma'', t\delta''), \ldots, (h\sigma'^{\cdots\prime}, t\delta'^{\cdots\prime})\rangle$ such that the first hub state $h\sigma'$ is to be set first and followed by a time delay $t\delta'$ whereupon the next state is set, etc.

40 A semaphore is now abstracted by the signalings that are prescribed for any change from a hub state $h\sigma$ to a hub state $h\sigma'$.


```
type
 [36] Colour == red | yellow | green
 [37] X = LI×HI×LI×Colour [crossings of a hub]
 [37] HΣ = X-set [hub states]
 [38] TI [time interval]
 [39] Signalling = (HΣ × TI)*
 [40] Semaphore = (HΣ × HΣ) →m Signalling
value
 [37] ωHΣ: H → HΣ
 [40] ωSemaphore: H → Sema,
 [41] chg_HΣ: H × HΣ → H
 [41] chg_HΣ(h,hσ) as h′
 [41]   pre hσ ∈ ωHΩ(h) post ωHΣ(h′)=hσ
```



```
 [39] chg_HΣ_Seq: H × HΣ → H
 [39] chg_HΣ_Seq(h,hσ) ≡
 [39]   let sigseq = (ωSemaphore(h))(ωΣ(h),hσ) in
 [39]   sig_seq(h)(sigseq) end
 [39] sig_seq: H → Signalling → H
 [39] sig_seq(h)(sigseq) ≡
 [39]   if sigseq=⟨⟩ then h else
 [39]   let (hσ,tδ) = hd sigseq in let h′ = chg_HΣ(h,hσ);
 [39]   wait tδ;
 [39]   sig_seq(h′)(tl sigseq) end end end
```

■ End of Example 8

3.4. Rules and Regulations

- By a **rule** we shall understand
  - a text which describe how the domain is
    — i.e., how people and technology are —
  - expected to behave.
- The meaning of a rule is
  - a predicate over "before/after" states of actions
    (simple, one step behaviours):
  - if the predicate holds then the rule has been obeyed.

- By a **regulation** we shall understand
  - a text which describes actions to be performed
  - should its corresponding rule fail to hold.
- The meaning of a regulation is therefore
  - a state-to-state transition,
  - one that brings the domain into a rule-holding "after" state.

Example 9 – Rules We give two examples related to railway systems where train stations are the hubs and the rail tracks between train stations are the links:

41 Trains arriving at or leaving train stations:

(a) (In China:) No two trains
(b) must arrive at or leave a train station
(c) in any two minute time interval.

42 Trains travelling "down" a railway track. We must introduce a notion of links being a sequence of adjacent sectors.

(a) Trains must travel in the same direction;
(b) and there must be at least one "free-from-trains" sector
(c) between any two such trains.

We omit showing somewhat "lengthy" formalisations.
. ■ End of Example 9

We omit exemplification of regulations.

(3. Domain Engineering 3.4. Rules and Regulations)

3.5. Scripts, Licenses and Contracts

3.5.1. Scripts

- By a script we understand
  - a usually structured set of pairs of rules and regulations —
  - structured, for example, as a simple "algorithm description".

(3. Domain Engineering 3.5. Scripts, Licenses and Contracts 3.5.1. Scripts)

Example 10 – Timetable Scripts

43 Time is considered discrete. Bus lines and bus rides have unique names (across any set of time tables).

44 A *T*ime*T*able associates *B*us *L*ine *Id*entifiers (*blid*) to sets of *Journies*.

45 *Journies* are designated by a pair of a *BusRoute* and a set of *BusRides*.

46 A *BusRoute* is a triple of the *Bus Stop* of origin, a list of zero, one or more intermediate *Bus Stop*s and a destination *Bus Stop*.

47 A set of *BusRides* associates, to each of a number of *B*us *Id*entifiers (*bid*) a *Bus Sched*ule.

48 A *Bus Sched*ule is a triple of the initial departure *T*ime, a list of zero, one or more intermediate bus stop *T*imes and a destination arrival *T*ime.

(3. Domain Engineering 3.5. Scripts, Licenses and Contracts 3.5.1. Scripts)

49 A *Bus Stop* (i.e., its position) is a *Frac*tion of the distance along a link (identified by a *L*ink *I*dentifier) *f*rom an *i*dentified *h*ub *t*o an *i*dentified *h*ub.

50 A *Frac*tion is a **Real** properly between 0 and 1.

51 The *Journies* must be *well_f*ormed in the context of some net.

52 A set of journies is well-formed if

  53 the bus stops are all different,

  54 a bus line is embedded in some line of the net, and

  55 all defined bus trips of a bus line are equivalent.

(3. Domain Engineering 3.5. Scripts, Licenses and Contracts 3.5.1. Scripts)

```
type
 [43] T, BLId, BId
 [44] TT = BLId →m Journies
 [45] Journies′ = BusRoute × BusRides
 [46] BusRoute = BusStop × BusStop* × BusStop
 [47] BusRides = BId →m BusSched
 [49] BusSched = T × T* × T
 [50] BusStop == mkBS(s_fhi:HI,s_ol:LI,s_f:Frac,s_thi:HI)
 [51] Frac = {|r:Real·0<r<1|}
 [45] Journies = {|j:Journies′·∃ n:N · wf_Journies(j)(n)|}
value
 [52] wf_Journies: Journies → N → Bool
 [52] wf_Journies((bs1,bsl,bsn),js)(hs,ls) ≡
 [53]   diff_bus_stops(bs1,bsl,bsn) ∧
 [54]   is_net_embedded_bus_line(⟨bs1⟩⌢bsl⌢⟨bsn⟩)(hs,ls) ∧
 [55]   commensurable_bus_trips((bs1,bsl,bsn),js)(hs,ls)
```

■ End of Example 10

- Timetables are used in Example 11 on the following page.

3.5.2. Licenses and Contracts

- By a **license** (a **contract**) language we understand a pair of languages
  - of licenses and
  - of the set of actions allowed by the license
  - such that non-allowable license (contract) actions
    - incur moral obligations
    - (respectively legal responsibilities).

Example 11 – Public Bus Transport Contracts

- An example contract can be ‘schematised’:

cid: **contractor** cor **contracts sub-contractor** cee
to perform operations
{"conduct","cancel","insert","subcontract"}
with respect to timetable tt.

We assume a context (a global state) in which all contract actions (including contracting) takes place and in which the implicit net is defined.

- Concrete examples of actions can be schematised:

(a) cid: **conduct bus ride** (blid,bid) **to start at time** t
(b) cid: **cancel bus ride** (blid,bid) **at time** t
(c) cid: **insert bus ride like** (blid,bid) **at time** t

The schematised license shown earlier is almost like an action; here is the action form:

(d) cid: **contractor** cnm′ **is granted a contract** cid′
to perform operations
{"conduct","cancel","insert",sublicense"}
with respect to timetable tt′.

- All actions are being performed by a sub-contractor in a context which defines
  - that sub-contractor *cnm*,
  - the relevant net, say *n*,
  - the base contract, referred here to by *cid* (from which this is a sublicense), and
  - a timetable *tt* of which *tt′* is a subset.
- contract name *cnm′* is new and is to be unique.
- The subcontracting action can (thus) be simply transformed into a contract as shown on Slide 84.

type

Action = CNm × CId × (SubCon | SmpAct) × Time
SmpAct = Conduct | Cancel | Insert
Conduct == μConduct(s_blid:BLId,s_bid:BId)
Cancel == μCancel(s_blid:BLId,s_bid:BId)
Insert = μInsert(s_blid:BLId,s_bid:BId)
SubCon == μSubCon(s_cid:CId,s_cnm:CNm,s_body:body)
    where body = (s_ops:Op-**set**,s_tt:TT)

■ End of Example 11

3.6. Management and Organisation

- By management we shall understand
  - the set of behaviours which perform
    - strategic,
    - tactical and
    - operational

    actions.

- By organisation we shall understand
  - the decomposition of these behaviours into, for example, clearly separate
    - strategic,
    - tactical and
    - operational

    "areas",
    - possibly further decomposed
    - by geographical and/or
    - "subject matter" concerns.

- To explain differences between strategic, tactical and operational issues we introduce notions of
  - *strategic, tactical* and *operational funds*, $\mathbb{F}_{\mathcal{S},\mathcal{T},\mathcal{O}}$,
  - and other *resources*, $\mathbb{R}$,
  - a notion of *contexts*, $\mathbb{C}$,
  - and a notion of *states*, $\mathbb{S}$.
- Contexts bind resources to bindings from locations to disjoint time intervals (allocation and scheduling),
- states bind resource identifiers to resource values.

- Simplified types of the strategic, tactical and operational actions are now of the following types:
  - executive functions apply to contexts, states and funds and obtain and redistribute funds;
  - strategic functions apply to contexts and strategic funds and create new contexts and states and consume some funds;
  - tactical functions apply to resources, contexts, states tactical funds and create new contexts while consuming some tactical funds;
  - etcetera.

type

$\mathbb{R}, \mathbb{RID}, \mathbb{RVAL}, \mathbb{F}_{\mathcal{S}}, \mathbb{F}_{\mathcal{T}}, \mathbb{F}_{\mathcal{O}}$

$\mathbb{C} = \mathbb{R} \overrightarrow{m} ((\mathbb{T} \times \mathbb{T}) \overrightarrow{m} \mathbb{L})$

$\mathbb{S} = \mathbb{RID} \overrightarrow{m} \mathbb{RVAL}$

value

$\omega\mathbb{RID}: \mathbb{R} \to \mathbb{RID}$

$\omega\mathbb{RVAL}: \mathbb{R} \to \mathbb{RVAL}$

Executive_functions: $\mathbb{C} \times \mathbb{S} \times \mathbb{F}_{\mathcal{S},\mathcal{T},\mathcal{O}} \to \mathbb{F}_{\mathcal{S},\mathcal{T},\mathcal{O}}$

Strategic_functions: $\mathbb{C} \times \mathbb{F}_{\mathcal{S}} \to \mathbb{F}_{\mathcal{S}} \times \mathbb{R} \times \mathbb{C} \times \mathbb{S}$

Tactic_functions: $\mathbb{R} \times \mathbb{C} \times \mathbb{S} \times \mathbb{F}_{\mathcal{T}} \to \mathbb{C} \times \mathbb{F}_{\mathcal{T}}$

Operational_functions: $\mathbb{C} \times \mathbb{S} \times \mathbb{F}_{\mathcal{O}} \to \mathbb{S} \times \mathbb{F}_{\mathcal{O}}$

Example 12 – Public Bus Transport Management We relate to Example 11:

56 The **conduct, cancel** and **insert bus ride** actions are operational functions.

57 The actual **subcontract** actions are tactical functions;

58 but the decision to carry out such a tactical function may very well be a strategic function as would be the acquisition or disposal of busses.

59 Forming new timetables, in consort with the contractor, is a strategic function.

We omit formalisations. ■ End of Example 12

3.7. Human Behaviour

- By human behaviour we shall understand
  - those aspects of the behaviour of domain stakeholders
  - which have a direct bearing on the “functioning” of the domain
- Behaviours “fall” in a spectrum
  - from diligent
  - via sloppy
  - to delinquent and
  - outright criminal neglect

  in the observance of maintaining
  - entities,
  - carrying our actions and
  - responding to events.

Example 13 – **Human Behaviour** Cf. Examples 11–12:

60 no failures to conduct a bus ride must be classified as diligent;

61 rare failures to conduct a bus ride must be classified as sloppy if no technical reasons were the cause;

62 occasional failures $\cdots$ as delinquent;

63 repeated patterns of failures $\cdots$ as criminal.

We omit showing somewhat "lengthy" formalisations.
. ■ End of Example 13

3.8. Discussion

- We have briefly outlined six concepts of domain facets and we have exemplified each of these.
- Real-scale domain descriptions are, of course, much larger than what we can show. Typically, say for the domain of logistics, a basic description is approximately 30 pages; for "small" parts of railway systems we easily get up to 100–200 pages – both including formalisations.
- You should now have gotten a reasonably clear idea as to what constitutes a domain description.

- As mentioned, in the introduction to this lecture, we shall not cover post-modelling activities such a validation and domain theory formation. The latter is usually part of the verification (theorem proving, model checking and formal testing) of the formal domain description.
- Final validation of a domain description is with respect to the narrative part of the narrative/formalisation pairs of descriptions.
- The reader should also be able to form a technical opinion about what can be formalised, and that not all can be formalised within the framework of a single formal specification language, cf. Sect. .

End of Lecture 3: DOMAIN ENGINEERING