WELCOME BACK

Begin of Lecture 3: First Session — Continuous Endurants

Materials, States

FM 2012 Tutorial, Dines Bjørner, Paris, 28 August 2012

Tutorial Schedule

- **Lectures 1–2** 9:00–9:40 + 9:50–10:30

1 **Introduction** Slides 1–35

2 **Endurant Entities: Parts** Slides 36–110

- **Lectures 3–5** 11:00–11:15 + 11:20–11:45 + 11:50–12:30

√ 3 **Endurant Entities: Materials, States** **Slides 111–142**

4 **Perdurant Entities: Actions and Events** Slides 143–174

5 **Perdurant Entities: Behaviours** Slides 175–285

Lunch **12:30–14:00**

- **Lectures 6–7** 14:00–14:40 + 14:50–15:30

6 **A Calculus: Analysers, Parts and Materials** Slides 286–339

7 **A Calculus: Function Signatures and Laws** Slides 340–377

- **Lectures 8–9** 16:00–16:40 + 16:50–17:30

8 **Domain and Interface Requirements** Slides 378–424

9 **Conclusion: Comparison to Other Work** Slides 428–460

Conclusion: What Have We Achieved Slides 425–427 + 461–472

5. Continuous Endurants: Materials

- Let us start with examples of **materials**.

Example: 18 Materials. Examples of endurant continuous entities are such as

- coal,
- air,
- natural gas,
- grain,
- sand,
- iron ore,
- minerals,
- crude oil,
- solid waste,
- sewage,
- steam and
- water. ■

The above materials are either

- liquid materials (crude oil, sewage, water),
- gaseous materials (air, gas, steam), or
- granular materials (coal, grain, sand, iron ore, mineral, or solid waste).

- **Endurant continuous entities**, or **materials** as we shall call them,
  - ⟐ are the **core endurants** of process domains,
  - ⟐ that is, **domains** in which those **materials** *form the basis* for their *"raison d'être"*.

Example: 19 Material Processing.

- Oil or gas materials are ubiquitous to pipeline systems.
- Sewage is ubiquitous to, well, sewage systems.
- Water is ubiquitous to systems composed from reservoirs, tunnels and aqueducts which again are ubiquitous to hydro-electric power plants or irrigation systems. ■

- Ubiquitous means 'everywhere'.
- A continuous entity, that is, a material
  - is a core material,
  - if it is "somehow related"
  - to one or more parts of a domain.

5.1. "Somehow Related" Parts and Materials

- We explain our use of the term "somehow related".

Example: 20 "Somehow Related" Parts and Materials.

- Oil is pumped from wells, runs through pipes, is "lifted" by pumps, diverted by forks, "runs together" by means of joins, and is delivered to sinks – *and is hence a core endurant*.
- Grain is delivered to silos by trucks, piped through a network of pipes, forks and valves to vessels, etc. – *and is hence a core endurant*.
- Gravel, minerals (including) iron ore is mined, conveyed by belts to lorries or trains or cargo vessels and finally deposited. For minerals typically in mineral processing plants – *and is hence a core endurant*.
- Iron ore, for example, is conveyed into smelters, roasted, reduced and fluxed, mixed with other mineral ores to produced a molten, pure metal, which is then "collected" into ingots, etc. – *and is hence a core endurant* ■

5.2. Material Observers

- When **analysing domains** a key question,
  - in view of the above notion of **core continuous endurants** (i.e., materials)

  is therefore:
  - does the **domain** embody a notion of **core continuous endurants** (i.e., materials);
  - if so, then identify these "early on" in the **domain analysis**.
- Identifying materials —
  - their types and
  - attributes —

  is slightly different from identifying **discrete endurants**, i.e., **parts**.

Example: 21 Pipelines: Core Continuous Endurant.

- The core continuous endurant, i.e., material,
- of (say oil) pipelines is, yes, oil:

type
  O **material**
value
  obs_Materials: PLS → O

- The keyword **material** is a pragmatic. ■

- Materials are "few and far between" as compared to parts,
  - we choose to mark the type definitions which designate materials with the keyword **material**.
  - In contrast, we do not mark the type definitions which designate parts with the keyword **discrete**.

- First we do not associate the notion of atomicity or composition with a material. Materials are continuous.
- Second, amongst the attributes, none have to do with geographic (or cadestral) matters. Materials are moved.
- And materials have no unique identification or mereology. No "part" of a material distinguishes it from other "parts".
- But they do have other attributes when occurring in connection with, that is, related to parts, for example,
  - volume or
  - weight.

Example: 22 Pipelines: Parts and Materials. We refer to Example 14 on page 90.

4. From an oil pipeline system one can, amongst others,
   (a) observe the finite set of all its pipeline bodies,
   (b) units are composite and consists of a unit,
   (c) and the oil, even if presently, at time of observation, empty of oil.
5. Whether the pipeline is an oil or a gas pipeline is an attribute of the pipeline system.
   (a) The volume of material that can be contained in a unit is an attribute of that unit.
   (b) There is an auxiliary function which estimates the volume of a given "amount" of oil.
   (c) The observed oil of a unit must be less than or equal to the volume that can be contained by the unit.

type

4. PLS, B, U, O, Vol

value

4(a). obs_Bs: PLS → B-**set**

4(b). obs_U: B → U

4(c). obs_O: B → O

5. attr_PLS_Type: PLS → {"oil"|"gas"}

5(a). attr_Vol: U → Vol

5(b). vol: O → Vol

axiom

5(c). $\forall$ pls:PLS,b:B•b $\in$ obs_Bs(pls)$\Rightarrow$vol(obs_O(b))$\leq$attr_Vol(obs_U(b))

- Notice how bodies are composite and consists of
  - a discrete, atomic part, the unit, and
  - a material endurant, the oil.
- We refer to Example 23 on page 123.

5.3. Material Properties

- These are some of the key concerns in domains focused on materials:
  - transport, flows, leaks and losses, and
  - input to systems and output from systems,
- Other concerns are in the direction of
  - dynamic behaviours of materials focused domains (mining and production), including
  - stability, periodicity, bifurcation and ergodicity.
- In this tutorial we shall, when dealing with systems focused on materials, concentrate on modelling techniques for
  - transport, flows, leaks and losses, and
  - input to systems and output from systems.

- Formal specification languages like
  - `Alloy` [alloy],
  - `Event B` [JRAbrial:TheBBooks],
  - `CASL` [CoFI:2004:CASL-RM]
  - `CafeOBJ` [futatsugi2000a],
  - `RAISE` [RaiseMethod],
  - `VDM` [e:db:Bj78bwo,e:db:Bj82b,jf-pgl-97] and
  - `Z` [m:z:jd+jcppw96]

  do not embody the mathematical calculus notions of
  - continuity, hence do not "exhibit"
  - neither differential equations
  - nor integrals.
- Hence cannot formalise **dynamic systems** within these **formal specification languages**.
- We refer to Sect. 9 where we discuss these issues at some length.

Example: 23 Pipelines: Parts and Material Properties. We refer to Examples 14 on page 90 and 22 on page 119.

6. Properties of pipeline units additionally include such which are concerned with flows (F) and leaks (L) of materials:
   (a) current flow of material into a unit input connector,
   (b) maximum flow of material into a unit input connector while maintaining laminar flow,
   (c) current flow of material out of a unit output connector,
   (d) maximum flow of material out of a unit output connector while maintaining laminar flow,
   (e) current leak of material at a unit input connector,
   (f) maximum guaranteed leak of material at a unit input connector,
   (g) current leak of material at a unit input connector,
   (h) maximum guaranteed leak of material at a unit input connector,
   (i) current leak of material from "within" a unit,
   (j) maximum guaranteed leak of material from "within" a unit.

type

6. F, L

value

6(a). attr_cur_iF: U → UI → F

6(b). attr_max_iF: U → UI → F

6(c). attr_cur_oF: U → UI → F

6(d). attr_max_oF: U → UI → F

6(e). attr_cur_iL: U → UI → L

6(f). attr_max_iL: U → UI → L

6(g). attr_cur_oL: U → UI → L

6(h). attr_max_oL: U → UI → L

6(i). attr_cur_L: U → L

6(j). attr_max_L: U → L

- The maximum flow attributes are static attributes and are typically provided by the manufacturer as indicators of flows below which laminar flow can be expected.
- The current flow attributes as dynamic attributes.

7. Properties of pipeline materials may additionally include
   (a) kind of material[24],
   (b) paraffins,
   (c) naphtenes,
   (d) aromatics,
   (e) asphatics,
   (f) viscosity,
   (g) etcetera.

- We leave it to the student to provide the formalisations. ■

[24] For example `Brent Blend` Crude Oil

5.4. Material Laws of Flows and Leaks

- It may be difficult or costly, or both
  - to ascertain flows and leaks in materials-based domains.
  - But one can certainly speak of these concepts.
  - This casts new light on **domain modelling**.
  - That is in contrast to
    - incorporating such notions of flows and leaks
    - in **requirements modelling**
  - where one has to show implementability.
- Modelling flows and leaks is important to the modelling of materials-based domains.

Example: 24 Pipelines: Intra Unit Flow and Leak Law.

8. For every unit of a pipeline system, except the well and the sink units, the following law apply.
9. The flows into a unit equal
   (a) the leak at the inputs
   (b) plus the leak within the unit
   (c) plus the flows out of the unit
   (d) plus the leaks at the outputs.

axiom

```
8.    ∀ pls:PLS,b:B\We\Si,u:U ·
8.       b ∈ obs_Bs(pls)∧u=obs_U(b)⇒
8.       let (iuis,ouis) = mereo_U(u) in
9.       sum_cur_iF(iuis)(u) =
9(a).          sum_cur_iL(iuis)(u)
9(b).       ⊕ attr_cur_L(u)
9(c).       ⊕ sum_cur_oF(ouis)(u)
9(d).       ⊕ sum_cur_oL(ouis)(u)
8.       end
```

10. The **sum_cur_iF** (cf. Item 9) sums current input flows over all input connectors.
11. The **sum_cur_iL** (cf. Item 9(a)) sums current input leaks over all input connectors.
12. The **sum_cur_oF** (cf. Item 9(c)) sums current output flows over all output connectors.
13. The **sum_cur_oL** (cf. Item 9(d)) sums current output leaks over all output connectors.

10. sum_cur_iF: UI-**set** $\rightarrow$ U $\rightarrow$ F
10. sum_cur_iF(iuis)(u) $\equiv \oplus$ {attr_cur_iF(ui)(u)|ui:UI•ui $\in$ iuis}
11. sum_cur_iL: UI-**set** $\rightarrow$ U $\rightarrow$ L
11. sum_cur_iL(iuis)(u) $\equiv \oplus$ {attr_cur_iL(ui)(u)|ui:UI•ui $\in$ iuis}
12. sum_cur_oF: UI-**set** $\rightarrow$ U $\rightarrow$ F
12. sum_cur_oF(ouis)(u) $\equiv \oplus$ {attr_cur_iF(ui)(u)|ui:UI•ui $\in$ ouis}
13. sum_cur_oL: UI-**set** $\rightarrow$ U $\rightarrow$ L
13. sum_cur_oL(ouis)(u) $\equiv \oplus$ {attr_cur_iL(ui)(u)|ui:UI•ui $\in$ ouis}
    $\oplus$: (F|L) $\times$ (F|L) $\rightarrow$ F

- where $\oplus$ is both an infix and a distributed-fix function which adds flows and or leaks.

Example: 25 Pipelines: Inter Unit Flow and Leak Law.

14. For every pair of connected units of a pipeline system the following law apply:
    (a) the flow out of a unit directed at another unit minus the leak at that output connector
    (b) equals the flow into that other unit at the connector from the given unit plus the leak at that connector.

14. $\forall$ pls:PLS,b,b$'$:B,u,u$'$:U$\cdot$
14. {b,b$'$}$\subseteq$obs_Bs(pls)$\wedge$b$\neq$b$'$$\wedge$u$'$=obs_U(b$'$)
14. $\wedge$ **let** (iuis,ouis)=mereo_U(u),(iuis$'$,ouis$'$)=mereo_U(u$'$),
14. ui=uid_U(u),ui$'$=uid_U(u$'$) **in**
14. ui $\in$ iuis $\wedge$ ui$'$ $\in$ ouis$'$ $\Rightarrow$
14(a). attr_cur_oF(us$'$)(ui$'$) $-$ attr_leak_oF(us$'$)(ui$'$)
14(b). = attr_cur_iF(us)(ui) + attr_leak_iF(us)(ui)
14. **end**
14. **comment:** b$'$ precedes b

- From the above two laws one can prove the **theorem:**
  - what is pumped from the wells equals
  - what is leaked from the systems plus what is output to the sinks.
- We need formalising the flow and leak summation functions. ■

Continuous Endurant Modelling

As one of the first steps

- in domain analysis
- determine if the domain is materials-focused.

If so, then determine

- the material types,

  type M1, M2, ... Mn **material**

- the parts, that is, the part types, with which the materials are "somehow related",

  value obs_Mi: Pi → Mi, obs_Mj: Pj → Mj, ..., obs_Mk: Pk → Mk

- the relevant flow or transport and/or leak or loss attributes, if any,
- and the possible laws related to these attributes.

6. A Final Note on Endurant Properties

- The properties of **parts** and **materials** are fully captured by
  - (i) the **unique part identifiers**,
  - (ii) the **part mereology** and
  - (iii) the full set of**part attributes** and **material attributes**
- We therefore postulate a **property function**
  - when when applied to a **part** or a **material**
  - yield this triplet, (i–iii), of properties
  - in a suitable structure.

type

$$\text{Props} = \{|\text{PI}|\textbf{nil}|\} \times \{|(\text{PI-}\textbf{set}\times\ldots\times\text{PI-}\textbf{set})|\textbf{nil}|\} \times \text{Attrs}$$

value

$$\text{props: Part}|\text{Material} \rightarrow \text{Props}$$

- where
  - Part stands for a **part type**,
  - Material stands for a **material type**,
  - PI stand for **unique part identifiers** and
  - **PI-set** × ... × **PI-set** for **part mereologies**.
- The {|...|} denotes a proper specification language sub-type and **nil** denotes the empty type.

7. States

7.1. General

- The above Wikipedia characterisation of the concept of **perdurant**
  - mentioned **time**,
  - but implied a concept that we shall call **state**.
- In this version of this tutorial
  - we shall not cover the modelling of **time phenomena** —
  - but we shall model that some actions occur before others.

- By a **state** we shall understand a collection of parts
  - such that each of these parts have dynamic attributes.
- We can characterise the state
  - by giving it a type,
  - for example, Σ, where the **state type definition**
  - $\Sigma = \mathsf{S}_1 \times \mathsf{S}_2 \times \cdots \times \mathsf{S}_s$
  - assembles the types of the parts making up the state —
  - where we assume that types S_1, S_2, . . . , S_s
    - are types of parts
    - such that no S_i is a sub-part (of a subpart, . . .) of some S_j,
    - and such that each part has **dynamic attribute**s.

Example: 26 Net and Vessel States.

- We may consider a transport net, n:N, to represent a state (subject to the actions of maintaining a net: adding or removing a hub, adding or removing a link, etc.).
- We may also consider a hub, h:H, to represent a state (subject to the changing of a hub traffic signal: from red to green, etc., for specific directions through the hub).
- We may consider a container vessel to represent a state (subject to adding or removing containers from, respectively onto the top of stacks). ■

Thus the context determines how wide a scope the domain designer chooses for the state concept.

7.2. State Invariants

- States are subject to invariants.

Example: 27 State Invariants: Transport Nets. Nets, hubs and links were first introduced in Example 3 on page 16 – and were and will be prominent in this tutorial, to wit, Examples 6–15 and 28– 36 on page 188.

- Net hubs and links may be inserted into and removed from nets.
- Thus is also introduced changes to the net mereology.
- Yet, the axioms, as illustrated in Example 11, must remain invariant.
- Likewise changes to dynamic attributes may well be subject to the holding of certain well-formedness constraints.
- We will illustrate this claim.

With each hub we associate a hub [link] state and a hub [link] state space.

15. A hub [link] state models the permissible routes from hub input links to (same) hub output links [respectively through a link].
16. A hub [link] state space models the possible set of hub [link] states that a hub [link] is intended to “occupy”.

type
15. $H\Sigma = (LI \times LI)$**-set**, $L\Sigma = HI$**-set**
16. $H\Omega = H\Sigma$**-set**, $L\Omega = L\Sigma$**-set**
value
15. attr_HΣ: $H \rightarrow H\Sigma$, attr_LΣ: $L \rightarrow L\Sigma$
16. attr_HΩ: $H \rightarrow H\Omega$, attr_LΩ: $L \rightarrow L\Omega$

17. For any given hub, h, with links, $l_1, l_2, ..., l_n$ incident upon (i.e., also emanating from) that hub, each hub state in the hub state space
18. must only contain such pairs of (not necessarily distinct) link identifiers that are identifiers of $l_1, l_2, ..., l_n$.

value

17. wf_HΩ: H → **Bool**
17. wf_HΩ(h) ≡ ∀ hσ:HΣ · hσ ∈ attr_HΩ(h) ⇒ wf_HΣ(h)

17. wf_HΣ: H → **Bool**
17. wf_HΣ(h) ≡
18. ∀ (li,li′):(LI×LI)·(li,li′)∈ attr_HΣ(h) ⇒ {li,li′} ⊆ mereo_H(h)

- This well-formedness criterion is part of the state invariant over nets.

  - We never write down the full state invariant for nets.

  - It is tacitly assume to be the collection of all the axioms and well-formedness predicates over net parts. ■

End of Lecture 3: First Session — Continuous Endurants

Materials, States

FM 2012 Tutorial, Dines Bjørner, Paris, 28 August 2012


Software
Engineering 1


Software
Engineering 2


Software
Engineering 3

MINI BREAK