Start of Lecture 5: DOMAIN ENTITIES

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B. Domain Entities

B.1. Entities

- The reason for our interest in 'simple entities'
  - is that assemblies and units of systems
  - possess static and dynamic properties
  - which become contexts and states of
  - the processes into which we shall "transform" simple entities.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.1.1. Observable Phenomena

- We shall just consider 'simple entities'.
  - By a simple entity we shall here understand
    - ∗ a phenomenon that we can designate, viz.
    - ∗ see, touch, hear, smell or taste, or
    - ∗ measure by some instrument (of physics, incl. chemistry).
  - A simple entity thus has properties.
  - A simple entity is
    - ∗ either continuous
    - ∗ or is discrete, and then it is
      - · either atomic
      - · or composite.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.1.1.1. Attributes: Types and Values

- By an attribute we mean a simple property of an entity.
  - *A simple entity has properties* $p_i, p_j, \ldots, p_k$.
- Typically we express attributes by a pair of
  - a type designator: *the attribute is of type* V, and
  - a value: *the attribute has value* v (of type V, i.e., $v : V$).
- A simple entity may have many simple properties.
  - A continuous entity, like 'oil', may have the following attributes:
    - ∗ type: *petroleum*,
    - ∗ kind: *Brent-crude*,
    - ∗ amount: *6 barrels*,
    - ∗ price: *45 US $/barrel*.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

– An *atomic* entity, like a 'person', may have the following attributes:

∗ gender: *male*,
∗ name: *Dines Bjørner*,
∗ birth date: *4. Oct. 1937*,
∗ marital status: *married*.

– A *composite* entity, like a railway system, may have the following attributes:

∗ country: *Denmark*,
∗ name: *DSB*,
∗ electrified: *partly*,
∗ owner: *independent public enterprise owned by Danish Ministry of Transport.*

B.1.1.2. Continuous Simple Entities

- A simple entity is said to be continuous
  - if, within limits, reasonably sizable amounts of the simple entity, can be arbitrarily decomposed into smaller parts
  - each of which still remain simple continuous entities
  - of the same simple entity kind.
- Examples of continuous entities are:
  - oil, i.e., any fluid,
  - air, i.e., any gas,
  - time period and
  - a measure of fabric.

B.1.1.3. Discrete Simple Entities

- A simple entity is said to be discrete if its immediate structure is not continuous.
  - A simple discrete entity may, however, contain continuous sub-entities.
- Examples of discrete entities are:
  - persons,
  - rail units,
  - oil pipes,
  - a group of persons,
  - a railway line and
  - an oil pipeline.

B.1.1.4. Atomic Simple Entities

- A simple entity is said to be atomic
  - if it cannot be meaningfully decomposed into parts
  - where these parts has a useful "value" in the context in which the simple entity is viewed and
  - while still remaining an instantiation of that entity.
- Thus a 'physically able person', which we consider atomic,
  - can, from the point of physical ability,
  - not be decomposed into meaningful parts: a leg, an arm, a head, etc.
- Other atomic entities could be a rail unit, an oil pipe, or a hospital bed.
- The only thing characterising an atomic entity are its attributes.

(B. Domain Entities B.1. Entities B.1.1. Observable Phenomena B.1.1.4. Atomic Simple Entities)

B.1.1.5. Composite Simple Entities

- A simple entity, *c*, is said to be composite
  - if it can be meaningfully decomposed
  - into sub-entities that have separate
  - meaning in the context in which *c* is viewed.
- We exemplify some composite entities.
  - (1) A *railway net* can be decomposed into
    - a set of one or more *train lines* and
    - a set of two or more *train stations*.
  - Lines and stations are themselves composite entities.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.1. Entities B.1.1. Observable Phenomena B.1.1.5. Composite Simple Entities)

- (2) An *Oil industry* whose decomposition include:
  - one or more *oil fields*,
  - one or more *pipeline systems*,
  - one or more *oil refineries* and
  - one or more *one or more oil product distribution systems*.
- Each of these sub-entities are also composite.

- Composite simple entities are thus characterisable by
  - their attributes,
  - their sub-entities, and
  - the mereology of how these sub-entities are put together.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.1. Entities B.1.1. Observable Phenomena B.1.1.5. Composite Simple Entities)

B.1.2. Discussion

- In Sect. 3.2 we interpreted the model of mereology in six examples.
- The units of Sect. 2
  - which in that section were left uninterpreted
  - now got individuality —
    - in the form of
      - aircraft,
      - building rooms,
      - rail units and
      - oil pipes.
  - Similarly for the assemblies of Sect. 2. They became
    - pipeline systems,
    - oil refineries,
    - train stations,
    - banks, etc.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.1. Entities B.1.2. Discussion)

- In conventional modelling
  - the mereology of an infrastructure component,
    - of the kinds exemplified in Sect. 3.2,
  - was modelled by modelling
    - that infrastructure component's special mereology
    - together, "in line", with the modelling
    - of unit and assembly attributes.
- With the model of Sect. 2 now available
  - we do not have to model the mereological aspects,
  - but can, instead, instantiate the model of Sect. 2 appropriately.
  - We leave that to be reported upon elsewhere.
- In many conventional infrastructure component models
  - it was often difficult to separate
    - what was mereology from
    - what were attributes.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.1. Entities B.1.2. Discussion)

B.2. Examples of Composite Structures

- Before a semantic treatment of the concept of mereology
  - let us review what we have done and
  - let us interpret our abstraction
    - ∗ (i.e., relate it to actual societal infrastructure components).

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures)

B.2.1. What We have Done So Far ?

- We have
  - presented a model that is claimed to abstract essential mereological properties of
    - ∗ machine assemblies,
    - ∗ railway nets,
    - ∗ the oil industry,
    - ∗ oil pipelines,
    - ∗ buildings with installations,
    - ∗ hospitals,
    - ∗ etcetera.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures B.2.1. What We have Done So Far ?)

B.2.2. Six Interpretations

- Let us substantiate the claims made in the previous paragraph.
  - We will do so, albeit informally, in the next many paragraphs.
  - Our substantiation is a form of diagrammatic reasoning.
  - Subsets of diagrams will be claimed to represent parts, while
  - Other subsets will be claimed to represent connectors.
- The reasoning is incomplete.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations)

B.2.2.1. Air Traffic

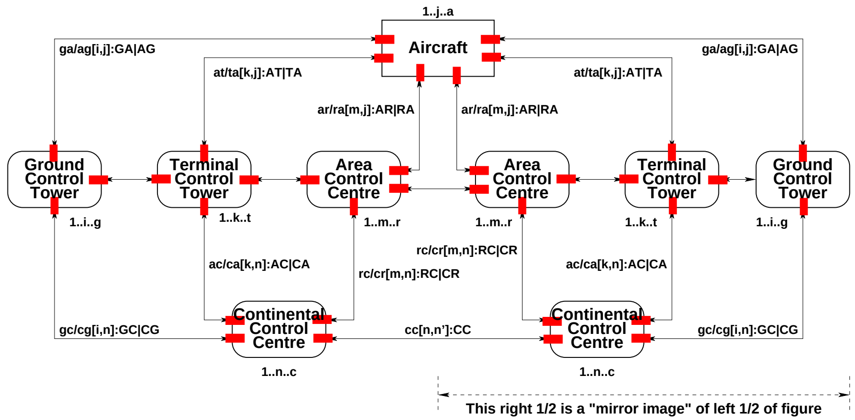

Figure 2: An air traffic system. Black (rounded or edged) boxes and lines are units; **red filled** boxes are connections

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

- Figure 2 on the previous page shows nine (9) boxes and eighteen (18) lines.
  - Together they form an assembly.
  - Individually boxes and lines represent units.
    - ∗ The rounded corner boxes denote buildings.
    - ∗ The sharp corner box denote an aircraft.
    - ∗ Lines denote radio telecommunication.
  - Only where lines touch boxes do we have connections.
    - ∗ These are shown as red horisontal or vertical boxes at both ends of the double-headed arrows,
    - ∗ overlapping both the arrows and the boxes.
- The index ranges shown attached to, i.e., labelling each unit,
  - shall indicate that there are a multiple of the "single" (thus representative) unit shown.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

- Notice that
  - the 'box' units are fixed installations and that
  - the double-headed arrows designate the ether where radio waves may propagate.
  - We could, for example, assume that each such line is characterised by
    - ∗ a combination of location and
    - ∗ (possibly encrypted) radio communication frequency.
  - That would allow us to consider all line for not overlapping.
  - And if they were overlapping, then that must have been a decision of the air traffic system.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.2.2.2. Buildings

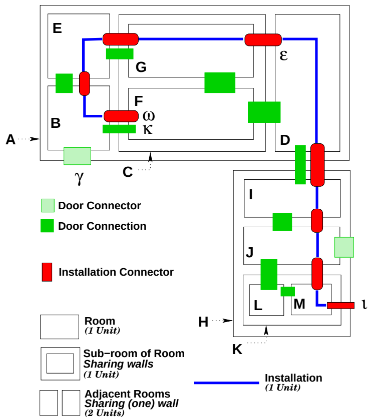

Figure 3: A building plan with installation

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

- Figure 3 on the preceding page shows a building plan — as an assembly
  - of two neighbouring, common wall-sharing buildings, A and H,
  - probably built at different times;
  - with room sections B, C, D and E contained within A,
  - and room sections I, J and K within H;
  - with room sections L and M within K,
  - and F and G within C.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.2. Buildings)

- Connector γ provides means of a connection between A and B.
- Connection κ provides "access" between B and F.
- Connectors ι and ω enable input, respectively output adaptors (receptor, resp. outlet) for electricity (or water, or oil),
- connection ϵ allow electricity (or water, or oil) to be conducted through a wall.
- Etcetera.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.2. Buildings)

B.2.2.3. Financial Service Industry

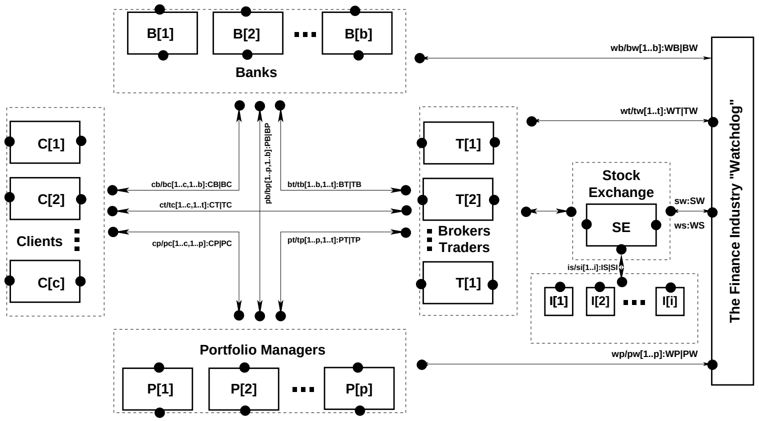

Figure 4: A financial service industry

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.3. Financial Service Industry)

- Figure 4 on the previous page shows seven (7) larger boxes [6 of which are shown by dashed lines] and twelve (12) double-arrowed lines.
  - Where double-arrowed lines touch upon (dashed) boxes we have connections (also to inner boxes).
  - Six (6) of the boxes, the dashed line boxes, are assemblies, five (5) of them consisting of a variable number of units;
  - five (5) are here shown as having three units each with bullets "between" them to designate "variability".
- People,
  - not shown, access the outermost (and hence the "innermost" boxes, but the latter is not shown)
  - through connectors, shown by bullets, •.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.3. Financial Service Industry)

B.2.2.4. Machine Assemblies

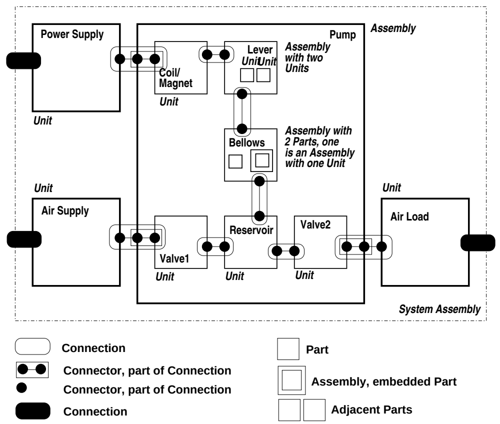

Figure 5: An air pump, i.e., a physical mechanical system

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.4. Machine Assemblies)

- Figure 5 on the preceding page shows a machine assembly.
  - Square boxes show assemblies or units.
  - Bullets, •, show connectors.
  - Strands of two or three bullets on a thin line, encircled by a rounded box, show connections.
  - The full, i.e., the level 0, assembly consists of
    - ∗ four parts
    - ∗ and three internal and three external connections.
  - The Pump unit
    - ∗ is an assembly
      - · of six (6) parts,
      - · five (5) internal connections
      - · and three (3) external connectors.

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.4. Machine Assemblies)

- Etcetera.
- One connector and some connections afford “transmission” of electrical power.
- Other connections convey torque.
- Two connectors convey input air, respectively output air.

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.4. Machine Assemblies)

B.2.2.5. Oil Industry

B.2.2.5.1. • *“The” Overall Assembly* •

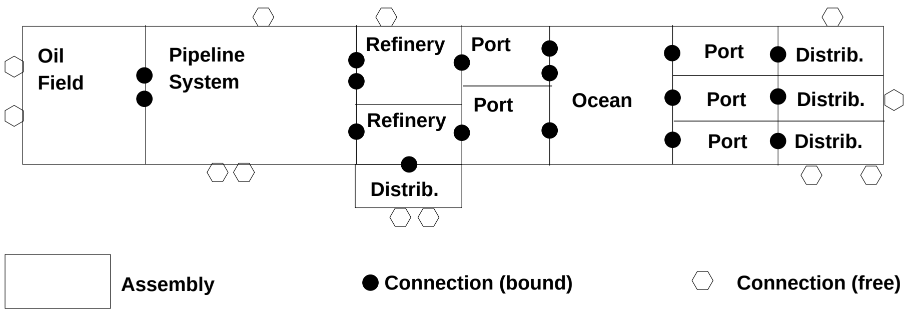

Figure 6: A Schematic of an Oil Industry

(B. Domain Entities B.2. Examples of Composite Structures B.2.2. Six Interpretations B.2.2.5. Oil Industry B.2.2.5.1. useboxA)

- Figure 6 on the previous page shows
  - an assembly consisting of fourteen (14) assemblies, left-to-right:
    - ∗ one oil field,
    - ∗ a crude oil pipeline system,
    - ∗ two refineries and one, say, gasoline distribution network,
    - ∗ two seaports,
    - ∗ an ocean (with oil and ethanol tankers and their sea lanes),
    - ∗ three (more) seaports,
    - ∗ and three, say gasoline and ethanol distribution networks.
  - Between all of the assembly units there are connections,
  - and from some of the assembly units there are connectors (to an external environment).
- The crude oil pipeline system assembly unit will be concretised next.

B.2.2.5.2. • *A Concretised Assembly Unit*•

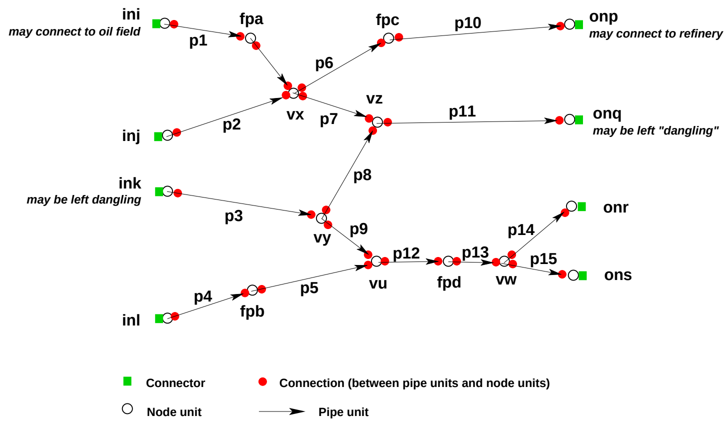

Figure 7: A Pipeline System

- Figure 7 on the preceding page shows a pipeline system.
- It consists of 32 units:
  - fifteen (15) pipe units (shown as directed arrows and labelled p1–p15),
  - four (4) input node units (shown as small circles, ∘, and labelled ini–inℓ),
  - four (4) flow pump units (shown as small circles, ∘, and labelled fpa–fpd),
  - five (5) valve units (shown as small circles, ∘, and labelled vx–vw), and
  - four (4) output node units (shown as small circles, ∘, and labelled onp–ons).

- In this example the routes through the pipeline system
  - start with node units and end with node units,
  - alternates between node units and pipe units,
  - and are connected as shown by fully filled-out **red**[4] disc connections.
  - Input and output nodes have input, respectively output connectors, one each, and shown with **green**[5]

[4] This paper is most likely not published with colours, so **red** will be shown as **darker colour**.
[5] Shown as **lighter colour**ed connections.

B.2.2.6. Railway Nets

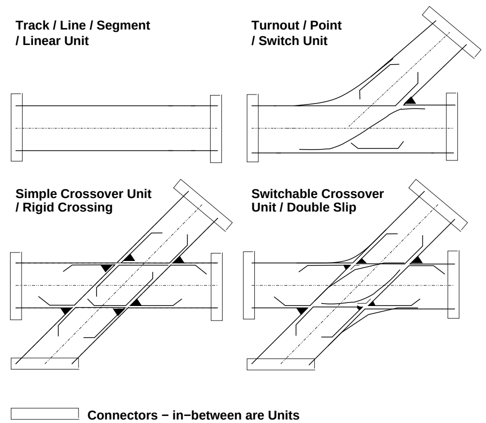

Figure 8: Four example rail units

- Figure 8 on the previous page diagrams
  - four rail units,
  - each with their two, three or four connectors.
- Multiple instances of these rail units
  - can be assembled
  - as shown on Fig. 9 on the following page
  - into proper rail nets.

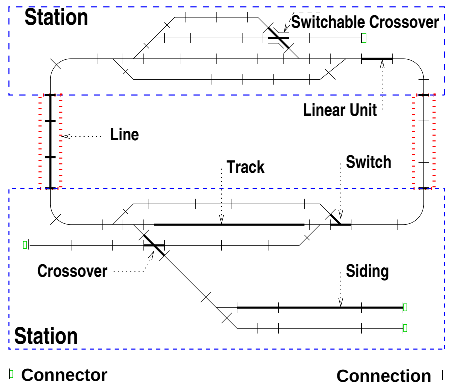

Figure 9: A "model" railway net. An Assembly of four Assemblies:
Two stations and two lines; Lines here consist of linear rail units;
stations of all the kinds of units shown in Fig. 8 on page 344.
There are 66 connections and four "dangling" connectors

- Figure 9 on the previous page diagrams an example of a proper rail net.
  - It is assembled from the kind of units shown in Fig. 8.
  - In Fig. 9 consider just the four dashed boxes:
    * The dashed boxes are assembly units.
    * Two designate stations, two designate lines (tracks) between stations.
    * We refer to to the caption four line text of Fig. 8 on page 344 for more "statistics".
    * We could have chosen to show, instead, for each of the four "dangling' connectors, a composition of a connection, a special "end block" rail unit and a connector.

B.2.3. Discussion

- It requires a somewhat more laborious effort,
  - than just "flashing" and commenting on these diagrams,
  - to show that the modelling of essential aspects of their structures
  - can indeed be done by simple instantiation
  - of the model given in the previous part of the talk.

- We can refer to a number of documents which give rather detailed domain models of
  - air traffic,
  - container line industry,
  - financial service industry,
  - health-care,
  - IT security,
  - "the market",
  - "the" oil industry[6],
  - transportation nets[7],
  - railways, etcetera, etcetera.
- Seen in the perspective of the present paper
  - we claim that much of the modelling work done in those references
  - can now be considerably shortened and
  - trust in these models correspondingly increased.

[6]http://www2.imm.dtu.dk/˜db/pipeline.pdf
[7]http://www2.imm.dtu.dk/˜db/transport.pdf

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.3. Attributes and Sub-entities of Sort Values

B.3.1. General

- Entities are defined in terms of
  - either sorts, that is, abstract types for whose values we do not define mathematical models,
  - or concrete types whose values are sets, Cartesians, lists, maps, functions or other.
- Entities are
  - either atomic,[8] in which case they are characterised solely in terms of all their attributes (types and values),
  - or are composite, in which case they are characterised in terms of all their attributes (types and values) and all their sub-entities.

[8]As dealt with elsewhere (Appendix Sect. , Pages 315–324) in these lecture notes: attributes of atomic or composite entities are (type and value) properties of entities (save those of being a composite entity and of such composite entities sub-entities). Atomic entities are atomic in that they have no sub-entities. Sub-entities of composite entities are proper entities.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

- For both atomic and composite sorts
  - we introduce, as need be, observer functions,
  - whether of attributes or (possibly, if composite) of sub-entities.[9]
- In this section we shall introduce and define an equality operator that compares entities modulo some attribute:
  - the name of the equality operator is $\simeq_{\omega_{\mathcal{A}_{attr}}}$,
  - and application of the equality operator to a pair of entities to be compared and the attribute for which comparison is left is expressed: $\simeq_{\mathcal{A}_{attr_A}} (a', a'')(\omega_\alpha)$.
- To explain this "modulo attribute" equality operator we first $\iota\ell\ell$ustrate[10] the concepts of functions that observe attributes and sub-entities.

[9]Till now, in these lecture notes, we have used "the same kind" of observer functions (ωB_i, ωC_j) for observing attributes (B_i) of atomic or composite entities and for observing sub-entities (C_j) of composite entities. In this section we shall distinguish between ωbserving αttributes ($\omega_\alpha B$) and ωbserving sub-ϵntities ($\omega_\epsilon C$). Maybe we shall have an opportunity to do so in a next version of these lecture notes.
[10]In this section we distinguish between $\iota\ell\ell$ustrations (formally marked with $\iota\ell\ell$s) and $\delta\epsilon\phi$initions (read: definitions, marked with $\delta\epsilon\phi$s). $\iota\ell\ell$ustrations are like schematic examples, but they are just that: rough-sketched generic examples. $\delta\epsilon\phi$initions are valid throughout these lecture notes.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.3.2. Constant and Variable Valued Attributes

- There are two kinds of attributes to be considered.
  - constant valued attributes, and
  - variable valued attributes.
- Attributes with variable values are also called entity state components.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

- Let A be (the type name of) a set of entities,
- let $B_1, \ldots, B_m$ be all the (distinct names of) types of constant valued attributes of A and
- let $\Sigma_1, \ldots, \Sigma_n$ be all the (distinct names of) types of variable valued attributes of A.
- We $\iota\ell\ell$ustrate these:

type
[$\iota\ell\ell$] A, B_1, ..., B_m, Σ_1, ..., Σ_n, C_1, ..., C_k

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.3.3. Sub-Entities

- Let $C_1, \ldots, C_k$ be all the (distinct names of) types of sub-entities of A.
- We $\iota\ell\ell$ustrate these:

type
[$\iota\ell\ell$] C_1, ..., C_k

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.3.4. Attribute and Sub-Entity Observers

- Let $\{\omega_\alpha B_1, \ldots, \omega_\alpha B_m\}$ be the corresponding set of all the constant valued observers of A,
- Let $\{\omega_\alpha \Sigma_1, \ldots, \omega_\alpha \Sigma_n\}$ be the corresponding set of all the variable valued observers of A and
- let $\{\omega_\epsilon C_1, \ldots, \omega_\epsilon C_k\}$ be the corresponding set of all the sub-entity observers of A.
- We $\iota\ell\ell$ustrate these:

value
[$\iota\ell\ell$] $\omega_\alpha B_1$: A $\rightarrow$ B_1, ..., $\omega_\alpha B_n$: A $\rightarrow$ B_m
[$\iota\ell\ell$] $\omega_\alpha \Sigma_1$: A $\rightarrow$ Σ_1, ..., $\omega_\alpha \Sigma_n$: A $\rightarrow$ Σ_n,
[$\iota\ell\ell$] $\omega_\epsilon C_1$: A $\rightarrow$ C_1, ..., $\omega_\epsilon C_k$: A $\rightarrow$ C_2

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.3.5. Attribute and Sub-entity Meta-Observers

- Let $\mathcal{A}ttr_A$ name the general type of a attribute observer function for sort A.
- Let $\mathcal{E}subs_A$ name the general type of a sub-entity observer functions for sort A.
- We $\iota\ell\ell$ustrate, with respect to the above $\iota\ell\ell$ustrations, these general types:

type
[$\iota\ell\ell$] $\mathcal{A}ttr_A = \omega_\alpha B_1 \mid \ldots \mid \omega_\alpha B_m \mid \omega_\alpha \Sigma_1 \mid \ldots \mid \omega_\alpha \Sigma_n$
[$\iota\ell\ell$] $\mathcal{E}subs_A = \omega_\epsilon C_1 \mid \ldots \mid \omega_\epsilon C_k$

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

- Let $\omega\mathcal{A}attr_A$ denote the function which from a type (A) observes all it attribute observer functions.
- Let $\omega\mathcal{E}_{subs}$ denote the function which from a type observes all it possible sub-entity observer functions.
- We $\delta\epsilon\varphi$ne these:

value

$[\,\delta\epsilon\phi\,]$ $\omega\mathcal{A}{ttr_A}s$: A $\rightarrow$ $\mathcal{A}{ttr_A}$-**set**

$[\,\delta\epsilon\phi\,]$ $\omega\mathcal{E}{subs_A}s$: A $\rightarrow$ $\mathcal{E}{subs_A}$-**set**

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.3.6. Meta-Observer Properties

- Let $\mathbb{A}{ttr_A}$ $\iota\ell\ell$ustrate the set of all attribute observers for type A, and
- let $\mathbb{E}{subs_A}$ $\iota\ell\ell$ustrate the set of all sub-entity observers for type A,
- then the two axioms $\iota\ell\ell_{attr}$ and $\iota\ell\ell_{subs}$ holds for the $\iota\ell\ell$ustrated type A and its observer functions:

value

$[\,\iota\ell\ell_{attr}\,]$ $\mathbb{A}{ttr_A}$:$\mathcal{A}{ttr_A}$-**set** $= \{\omega_\alpha B_1,...,\omega_\alpha B_m,\omega_\alpha\Sigma_1,...,\omega_\alpha\Sigma_n\}$,

$[\,\iota\ell\ell_{subs}\,]$ $\mathbb{E}{subs_A}$:$\mathcal{E}{subs_A}$-**set** $= \{\omega_\epsilon C_1,\ldots,\omega_\epsilon C_k\}$

axiom

$[\,\iota\ell\ell_{attr}\,]$ $\forall$ a:A $\cdot$ $\omega\mathcal{A}{ttr_A}s(a) = \mathbb{A}{ttr_A}$ $\wedge$

$[\,\iota\ell\ell_{subs}\,]$ $\forall$ a:A $\cdot$ $\omega\mathcal{E}{subs_A}s(a) = \mathbb{E}{subs_A}$

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.3.7. Sort Value Equality

- Now to register a possible change in but one attribute of A we meta-linguistically $\delta\epsilon\phi$ine the following equality operator:

value

$[\,\delta\epsilon\phi\,]$ $\simeq_{\mathcal{A}{attr_A}}$: A$\times$A $\rightarrow$ $\mathcal{A}{ttr_A}$ $\rightarrow$ **Bool**

$[\,\delta\epsilon\phi\,]$ $\simeq_{\mathcal{A}{attr_A}}$(a′,a″)($\omega_\alpha$) $\equiv$

$[\,\delta\epsilon\phi\,]$ $\quad\forall\, \omega F{:}\omega\mathcal{A}{ttr_A}s(a')\backslash\{\omega_\alpha\}\Rightarrow\omega F(a')=\omega F(a'') \wedge \forall\, \omega'_\epsilon{:}\mathcal{E}{subs_A}\Rightarrow\omega'_\epsilon(a')=\omega'_\epsilon(a'')$

$[\,\delta\epsilon\phi\,]$ $\quad$ **pre** $\omega\mathcal{A}{ttr_A}s(a') = \omega\mathcal{A}{ttr_A}s(a'')$

- The $\simeq_{\omega_{\mathcal{A}{attr}}}$ 'equality' operator
  - applies to two values a′,a″:A and an attribute observer function, ωB_i (given as ω_α),
  - and yields **true** if a′ and a″
    - ∗ have all but the same attribute values except for attribute B_i, and
    - ∗ have all exactly the same and equal sub-entities.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

Example 50 – Equality of Hubs Modulo Hub States:

- Please review Examples 2 on page 50 and 3 on page 53.
  - In Example 3 on page 53 on Page 361, formula line item [17], a comparison is made between two values of a sort: $\omega H\Sigma(h') {=} (\prod\{h\sigma' | h\sigma'{:}H\Sigma{\cdot}h\sigma' \in \omega\Omega(h)\backslash\{h\sigma\}\})_{\overline{p}}\prod_p h\sigma$.
  - We now redefine this comparion – which really does not capture all the value aspects of the compared hubs!

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

value
p:**Real**, **axiom** 0<p≤1, typically p$\simeq 1 - 10^{-7}$
$\overline{p}$:**Real**, **axiom** $\overline{p}$=1−p

[12] set_HΣ: H $\times$ HΣ $\rightarrow$ H
[13] set_HΣ(h,hσ) **as** h′
[14] **pre** h$\sigma \in \omega$HΩ(h)
[15] **post** $\simeq_{\omega_{\mathcal{A}_{attr_H}}}$(h,h′)($\omega$H$\Sigma$) $\wedge$
[17] ωHΣ(h′)=($\prod$\{hσ'|hσ':HΣ•h$\sigma'\in \omega\Omega$(h)\\{hσ}})$_{\overline{p}}\prod_p$hσ

■ End of Example 50

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

B.4. Unique Entity Identifiers

- In many domain and requirements modelling situations we make use of the concept of *unique entitiy identifiers.*
  - For any type A for which we introduce unique identifiers of all a:A values
  - we consider such unique identifiers as of sort AI[11].
  - The AI attribute shall be considered a constant-valued attribute.
  -
  -

[11] We may, in some immediate future, decide to instead of using the sort name AI using, for example, the sort name $\Im$A or $\Im_A$.

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark

End of Lecture 5: DOMAIN ENTITIES

© Dines Bjørner 2010, Fredsvej 11, DK–2840 Holte, Denmark