3 Entities 65

By a domain entity we mean a *fact*[11] which is either an endurant entity, a *part* or is a perdurant entity, that is, an *action* an *event* or a *behaviour*. In contrast to facts we have *concepts*, that is, abstractions derived from facts. Concepts can also be considered entities. Domain entities are the things, the tangible, spatial facts we observe and the concepts we abstract from these.

Examples: 66

Example 6 (Domain Entities) One example per each of four entity categories:

- *part:* transport net;
- *action:* insertion of link;
- *event:* disappearance of a link segment (that is, fraction of a link); and
- *behaviour:* movement of vehicles along net. •

Inquiry: Fact

We say that facts are what we can observe as being a part or an action or an event or a behaviour. And we say that we can observe these.

A part, to a first "approximation", is a spatial, manifest phenomenon, something that one can point to. Is a bank a part ? Well, in "ye olde times" [still relevant Anno 2011] a bank can be presented by one or more buildings by the "books" of various kinds (for demand/deposit accounts, for mortgage accounts, for savings & loan accounts, etc.), by the cash registers, by ATMs, etc. Even though some banks may have all of the books represented electronically, they still have some physical extent: cubic spaces of electronic memory.

What part concepts may be derived from banks ? MORE TO COME

An action seems, at first, to be a concept, but it can be observed by seeing the changes of the state of physically manifested parts: change of account balance, is an example.

An event also seems, at first, to be a concept, but it can be observed by seeing the changes of the state of physically manifested parts: the disappearance of a transportation net link.

A behaviour is likewise manifest observable through its actions, events and other behaviours. •

Inquiry: Concept

So which concepts appears to be more concepts than phenomena ? An example could be a timetable. Even though there may not be any form of timetable for some (say bus) traffic, one may be allowed to say" "the busses appear to run according to some timetable". MORE TO COME •

[11] We use the terms 'fact', 'entity', 'particular', 'thing' and 'individual' synonymously

3.1 Parts 67

By a part we understand a manifest, an endurant, that is, something we can point to, inert, possibly dynamic, i.e., animate, phenomenon or a concept thereof, something that we might (later on) represent as data by a computer.

68 **Examples:**

Example 7 (Parts) Five domain examples:

- *Container line:* container, container vessel, container terminal port, bill of lading, etc.
- *Financial service industry:* bank, bankbook, money (notes, coins), insurance policy, stock certificate, etc.
- *Transportation:* net, link, hub, vehicle, driver, etc. 69
- *Health care:* hospital, ward, bed, patient, medical staff, medical record, medicine, surgery instruments, health insurance policy, etc.
- *Pipeline system:* well, pump, pipe, valve, fork, join, sink, pipeline, etc. •

3.1.1 Atomic Parts 70

By an atomic part we shall understand a part which we, as observers, have decided form an indivisible whole, that is, one for which it is not, in a current context, relevant to speak of meaningful subparts.

71 **Examples:**

Example 8 (Atomic Parts) Five domain examples:

- *Container Line:* container, bill of lading, way bill.
- *Financial Service Industry:* bankbook, money; insurance policy; stock certificate.
- *Health Care System:* bed, patient, medical record, health insurance policy.
- *Pipeline System:* well, pump, pipe, valve, fork, join, sink.
- *Transportation System:* link, hub, vehicle, driver. •

Inquiry: Atomicity

It is the domain describer who decides, sovereignly, which parts are to be abstracted as being atomic, which parts are to be considered composite. But the domain describer does so carefully taking into consideration the scope and span of the domain description. If, for example, the domain is that of the personnel department of a company then a person may very well be considered atomic. That is, cannot be "taken apart" into head, limbs, intestinals, etc. If, instead, the domain is that of a hospital department concerned with donations of, say deceased human organs, then such a deceased may be considered composite. •

3.1.2 Composite Parts 72

By a composite part we shall understand a part which we, as observers, have decided consists of one or more proper parts also referred to as subparts.

Examples: 73

Example 9 (Composite Parts and Subparts) Five domain examples:

- *Container Line:* container vessel and its bays; bay and its rows; row and its stacks, stack and its containers.
- *Financial Service Industry:* bank and its accounts.
- *Health Care System:* hospital and its wards; ward and its bedrooms, bedroom and its beds.
- *Pipeline System:* pipeline and its wells, pumps, pipes, valves, forks, joins and sinks.
- *Transportation System:* net and its hubs and links. •

Inquiry: Composition

The remarks, above, Page 26, on atomicity, applies, inter alia, here: "It is the domain describer who decides, sovereignly, which parts are to be abstracted as being atomic, which parts are to be considered composite. But the domain describer does so carefully taking into consideration the scope and span of the domain description."

In this section, Sect. 3.1, we consider compositionality of only parts. We shall also inquire as to the compositionality of actions, events and behaviours, Sects. 3.2–3.4. •

3.1.3 Part Attributes 74

By an attribute we shall mean a pair: a type name and a value (of that type). Earlier we stated that we consider parts to be values and have types. Now we state that attributes are pairs of types and values. This must not be construed as attributes being parts. It is only that we use the concept of 'type' for two purposes: to characterise sets of parts, and to characterise individual properties of parts.

Atomic Part Attributes By the attributes of an atomic part we mean the set of properties (type names and values) that we have decided together characterise that atomic part (and all of the atomic parts of the same type).

75 **Examples:**

Example 10 (Atomic Part Attributes) Five domain examples:

- *Container line:* `container attributes:` length, width, height, weight, refrigerated or not refrigerated, physical location, contents, etc.
- *Financial service industry:* `account attributes:` interest rate (on loans), yield (on deposits), owner(s), maximum credit, current balance, etc.
- *Health care:* `patient attributes`: name, central personal registration identifier, gender, birth date, birth place, nationality, weight, height, insurance policies, medical record, etc.
- *Pipeline system:* `pipe attributes`: circular diameter, length, location, maximum laminar flow, current flow, guaranteed maximum leak (in volume/second), current leak, etc.
- *Transportation:* `link attributes`: length, location, link state (open in one direction or the other or open in both or closed in both directions), link type (road, rail, sea, air), etc. •

76 **Composite Part Attributes** By the attributes of a composite part we mean the set of properties (type names and values) (exclusive of all subparts of that composite part) that we have decided together characterise that composite part (and all of the composite parts of the same type).

77 **Examples:**

Example 11 (Composite Part Attributes) Five domain examples:

- *Container Line Attributes:* name of container line, for example MAERSK, legal residence (address), incorporated ?, responsible capital, organisational structure (explicated organigram), subsidiaries, budget, accounts, etcetera.
- *Financial Service Industry Attributes, Bank:* name of bank, kind of bank [whether a demand/deposit or a savings & loan or an investment bank or other], legal residence (address), responsible capital, organisational structure (explicated organigram), sub-
78 sidiaries, budget, accounts, etcetera.

- *Health Care System Attributes, Hospital:* name, kind of hospital [whether a general hospital or a specialised hospital, and then its speciality], legal residence (address), legal owner, organisational structure (explicated organigram), sources of financing, budget, accounts, etcetera.
- *Pipeline System Attributes:* name of pipeline system[12], legal residence (address), legal owner, sources of financing, geography, maintenance subcontractors, budget, accounts, etcetera. 79
- *Transportation System Attributes:* name of transport system, kind of transport system[13], legal residence (address), legal owner, sources of financing, geography, maintenance subcontractors, budget, accounts, etcetera. •

• • •

Inquiry: Attribute and Property

We shall use the concept of 'property' in a wider sense than that of 'attribute'. This broader concept of 'property' has been studied by philosophers [17, 19, 32]. MORE TO COME
•

80

Static Part Attributes A part attribute is static if that part never changes its value.

Examples

Example 12 (Static Part Attributes) Two examples:

- **Patients:** name, central personal registration identifier, gender, and birthplace.
- **Links:** length. •

81

Dynamic Part Attributes A part attribute is dynamic if that part can change its value.

Examples

Example 13 (Dynamic Part Attributes) Three examples:

- The height, weight, blood pressure, blood sugar, temperature, and (hence) patient medical record of a patient are dynamic attributes. 82
- A hub typically can connect a number of distinct links and thus can attain either one of number of hub states each hub state being a possibly empty set of pairs, $(\mathsf{li}_j,\mathsf{li}_k)$, of not necessarily distinct link identifiers (li) of the links connected to that hub.

[12] for example: Nabucco http://en.wikipedia.org/wiki/Nabucco_pipeline
[13] whether a road system or a rail/train system, or an airline, or a shipping company, etc.

The state of a hub is a dynamic attribute.

- Similarly for link states. •

83 **Indivisibility of Attributes** Given a part of some kind (i.e., having some set of attributes), whether atomic or composite, one cannot "remove" an attribute from that entity and still retain the entity as being of that kind.

Examples

Example 14 (Indivisibility of Attributes) Two examples:

- One cannot remove the attribute 'height' from an entity of kind person
- and one cannot remove the attribute 'kind of transport system' from an entity of kind 'transport system'. •

Inquiry: Atomic Parts and Highly Structured Attributes

Let us analyse an example domain: that of banking. A bank has clients; clients have accounts; a clients may have zero, one or more accounts; two or more clients may share accounts (i.e., multiple accounts); and accounts register which transactions (`open, deposit, withdraw, get_statements, transfer, close`*) have been performed on the account, its balance, its interest rate to be paid by clients if account balance is negative, its yield accrued to the client when the account balance is positive. Accounts have account numbers. Etcetera. The terms written in the sans serif font designate parts (and, as we shall see later, also behaviour — distinct from parts). The terms written in slanted, italic font denote attributes. The terms written in* `teletype font` *designate actions. Now to an analysis. In a financial service system there are many clients (seen as behaviours) with contexts and states (also named clients), and many banks (likewise seen as behaviours etcetera). The fact that a bank may have several branch offices must not be confused: these branch offices are not separate banks: they all share the same clients and the same accounts. We shall see a bank as a concept and as a set of one or more branch offices (including the one located at the head quarters of the bank). The bank concept is a part and the branch offices are a set of parts. The bank concept has a number of attributes: customers ("proxies" for, but not to be confused with, clients), accounts, account sharing, multiple customer accounts, etc., with accounts having sub-attributes: owners, balance, interest, yield, etcetera. The bank concept part is shared by all branch offices. The branch offices are usually physically embodied and distinct. The bank concept ism in principle, not physically embodied: one cannot point to it. It may be "implemented" physically in terms of "ye olde" ledgers (books), or in terms of a large, central database or in terms of a set of smaller, distributed databases, say one per branch office, or in terms of client-held smart cards, etcetera. The branch offices may be hard to identify physically: they could be physical buildings and offices, or they could be ATMs: automatic teller machines, or they could be the client-held smart cards. The essence of the above is that a bank is an atomic entity with highly structured attributes. A model of these attributes could be:*

1. *A bank has as attributes*
   a *a name,*
   b *a set of uniquely identified customers,*
   c *a set of uniquely identified accounts,*
   d *a "book" which records the accounts of each customer,*
   e *a "book" which records the owners of each account.*
   f *etcetera.*
2. *An account has*
   a *a balance,*
   b *a chronologically identified set of transactions hitherto performed on the account,*
   c *an interest on negative balances,*
   d *a yield on positive balances,*
   e *etcetera,*

```
type
1.   Bank ::
1a.      BankName
1b.      CustId-set
1c.      AccId →m Account
1d.      CustId →m AccId-set
1e.      AccId →m CustId-set
2.   Account ::
2a.      Balance
2b.      DateTime →m Transaction
2c.      Interest
2d.      Yield
```

Truly a highly structured set of attributes and sub-attributes. •

Inquiry: Parts, Behaviours (Agents) and Attributes

There is a commonly misunderstood dichotomy[14]*: parts as possibly dynamic, inert entities, and behaviours. Take the term 'train'. We may speak of "the train" as a composite (or even as an atomic) part — such as it is manifested on the train station platform at some time; we may speak of "the train" as a behaviour such as it is manifested when it speeds down the rails. We may even extend this dichotomy beyond two mutually exclusive or contradictory entities to also include part attributes. Thus we may speak of "the train" as an attribute — such as it is manifested by a specific entry in a timetable, or as in "she took the 4:50 from Paddington". Of course, the confusion arises from our use of the same term 'train' in all these cases. (We invite the reader to formulate appropriately distinct 'train terms'.) But the message should be clear: That is describing the train behaviour one needs refer to the train part(s). Thus the catchphrase: to some a "thing" is a part, to others it is a behaviour.* •

[14]Dichotomy: a division into two mutually exclusive or contradictory entities.

3.1.4 Subparts Are Parts 84

By a subpart, p', of a part, p, we thus mean an entity which is not the same as the part, that is $p \neq p'$. We say that a part, p', is a *proper part* of another part if it is a subpart of that part. So by proper part of p and subpart of p we mean the same.

Examples

Example 15 (Sets of Hubs and Hubs – Sets of Links and Links) From a net we observe sets of hubs and sets of links: A set of hubs is a value of the type sets of hubs. A hub is a value of type hub. A set of links is a value of the type sets of links. A link is a value of type link. •

3.1.5 Subpart Types Are Not Subtypes 85

Thus, by a subpart type we mean a part type but the type of the subpart cannot be the same as the type of the part of which it is a subpart.

Examples

Example 16 (Part and Subpart Types) We refer to Example 15. Let a part be a transportation net, *n:N*. A subpart of a transportation net, *n:N*, is, for example, the part *hs:HS*, which is the set of all hubs of the net, and a hub, *h:H*, which is a part of *hs:HS*, is a subpart of *hs:HS*. And all these subparts are of different types, to wit: *HS* and *H*, and, as we shall see, *LS* and *L*, are not subtypes of type *N*. •

86 By a 'union', A, of *disjoint types*, say B, C, ..., D, that is: A=$B|C|...|D$, we mean a type whose values are either of type B or of type C or ... of type D, and where every type value is of exactly one of the types B, C, ..., D. These types, B, C, ..., D, are subtypes of A.

Thus subpart types are not the same as subtypes of the part of which the subpart is a proper part.

To be consistent we rule out the possibility of defining types recursively.

3.1.6 Mereology of Composite Parts 87

By the mereology of a composite part we understand the number of subparts of respective kinds (types) of that composite entity and how the subparts are related to one another.

88 **Examples**

Example 17 (Mereology of Composite parts) Five domain examples:

- *Container Line System:* A container vessel contains a number of uniquely identified bays, bays consists of a sequentially indexed sequence of (usually several) rows, and rows consist of a sequentially indexed sequence of (usually several) stacks, and stacks consists of a sequence of zero or more containers — such that access to stacks are by identity of bay, number of row, number of stack and then to the top of this possibly empty stack. Etcetera.

- *Financial Service Industry:* A bank consists of (i) a set of uniquely identified demand/deposit accounts, (ii) a set of uniquely identified savings & loan accounts, (iii) a set of uniquely identified mortgage accounts. Etcetera. 89

- *Health Care System:* A hospital consists of (1) a set of uniquely identified wards of kind κ_1, (2) a set of uniquely identified wards of kind κ_2, ..., and (n) a set of uniquely identified wards of kind κ_n. Etcetera.

- *Pipeline System:* A pipeline system consists of a set of units — where units a either wells, pumps, pipes, valves, forks, joins or sinks — and such that (a) a well is connected to one or more pumps, (b) a pipe is input-connected to either a pipe or a pump or a valve or a fork or a join and is output-connected to either a pipe or a pump or a valve or a fork, (c) a pump is input-connected to a pipe and is output-connected to a pipe, (d) a valve is input-connected to a pipe and is output-connected a pipe or a sink, (e) a fork is input-connected to a pipe and is output-connected to two pipes, (f) a join is input-connected to two pipes and is output-connected to a pipe, and (g) a sink is input-connected to a valve. Etcetera. 90

- *Transportation System:* A transport net consists of a set of hubs and a set of one or more links such that links connect exactly two distinct hubs, and thus such that hubs are connected to zero or more distinct links. Etcetera.
  - The mereology of a net can be expressed in terms of unique identifiers associated with hubs, hij, hik, . . . , him, and links, lia, lib, . . . , lic. •

91

Mereologically two parts, e_i, e_j, may stand in the following relationships: (a) either e_i is identical to e_j, (b) or e_i is fully disjoint from e_j, (c) or e_i is adjacent (i.e., connects) to (disjoint from, but "touches") e_j, (d) or e_i is fully contained within e_j, (e) or e_i partially overlaps with e_j (that is, there are "areas" of e_i which are not overlapping with "areas" of e_j).

3.1.7 Part Descriptions 92

To describe an atomic part (type) it suffices to describe all the atomic part attributes: its type name, the attributes, and its possible contribution to the mereology of "a whole": own unique identification, and how it 'unique identifier'-relates to other parts.

To describe a composite part (type) it is necessary to describe these things: (i) all the composite part attributes, (ii) each of the subpart types (i.e., subparts), and (iii) their mereology.

Examples 93

Example 18 (Description of An Atomic Part) We continue our example of transport nets. A link is here considered an atomic part.

Type Name: link.

Attributes: length, location[15], current state[16] and state space[17], etc.

Unique Identification: unique Link identifier.

Mereology: a pair of unique hub identifiers. •

94

Example 19 (Description of A Composite Part) We continue our example of transport nets. A net is here considered a composite part.

Type Name: net.

Attributes: name, transport kind, legal address, legal owner, sources of financing, geographical area, maintenance subcontractors, budget, accounts, etc.

Unique Identification: `not applicable`.

Mereology: `not applicable`.

Subpart Type[s]: set of links, set of hubs. •

3.1.8 States 95

By a state we understand a specific set of parts such that for each of these parts some attributes are dynamic.

96

Examples

Example 20 (States) Five domain examples:

- *Container line:* container, container vessel, container terminal port.
- *Financial service industry:* bank (as a whole), account (as a subpart).
- *Health care:* hospital, ward, bed, patient.
- *Pipeline system:* well, pump, pipe, valve, pipeline.
- *Transportation:* net, link (open in one direction, open in the opposite direction, open in both directions; closed in all [two] directions), hub (open between a specific [possibly empty] set of pairs of links connected to the hub), vehicle. •

[15] The cartographic and cadestral location of a link may, amongst other components, include, for example, a Beziér curve description of how that link "traverses" a, or the landscape.

[16] in terms of sets of pairs of distinct identifiers of connecting hubs

[17] in terms of sets of possible link states

Inquiry: State

What is a state and what is not is sometimes an elusive issue. If the outcome of each of a set of operations on parts is independent, say over long time intervals, say years for a transport system, then it seems that the arguments of these operations do not contribute to a state notion. If, however, the time span being considered is such that the outcome of operations being carried out depend very much on argument values, then these do indeed contribute to a state notion. So the notion of a state has to do with whether part values are constant over very long periods or whether they vary quite often. Please note that we only very roughly referred to a notion of time interval without being specific. •

Inquiry: Environment

In 'inquiry' `State`*, above, we alluded to some operations being dependent as to their varying outcome on a state. In contrast these or other operations may require arguments whose value remain constant over "large" time intervals. We say that these arguments for an environment. Other than this mentioning we shall not deal with the notion of environment in these notes.* •

3.2 Actions 97

By an action we understand a state change resulting directly from the expected application of a specific function (one of several possible), that is, the specific function was performed deliberately, on purpose.

98

Examples

Example 21 (Actions) We give examples from five domains. The examples are not proper descriptions of actions. We basically just give their names. These names — and the familiarity of the domains — are such that the reader is "tricked into" thinking: *"oh yes, I see; but, of course."* Only a proper action description can reveal the action. 99

- *Container line:*
  - loading a container;
  - unloading a container;
  - moving a container from one location (say on-board a vessel) to another location (say in a container terminal port).
- *Financial service industry:*
  - open an account,
  - deposit money into an account,
  - withdraw money from an account,
  - obtain account statement,
  - close account.

100

- *Health care:*
  - admitting a person as a patient;
  - allocating a bed in a ward to a patient;
  - medicating a patient.
- *Pipeline system:*
  - opening a pump (for pumping);
  - closing a valve.
- *Transport Net:*
  - inserting a hub;
  - inserting a link;
  - removing a hub;
  - removing a link. •

Inquiry: Action, Operation, Function

We shall, perhaps somewhat arbitrarily, be making a distinction between the concepts of function, action and operation.

By a function we shall understand something which when applied to something called its arguments yield something called the result of that function for those arguments.

By invocation we mean the same as application.

By an action we shall mean a function application which potentially changes a state.

By an operation we shall understand a function application which is like an action, or does not change such a state. •

3.3 Events 101

By an event we understand a state change resulting indirectly from the unexpected application of a function, that is, the specific function was performed "surreptitiously", Events can be characterised by a pair of (before and after) states, a predicate over these and a time.

Events are thus like actions: change states, but are usually either caused by "previous" actions, or caused by "an outside action".

102

Example

Example 22 (Events) Five domain examples:

- *Container line:* A container falls overboard.

- *Financial service industry:* A bank goes bankrupt.
- *Health care:* A patient dies.
- *Pipeline system:* A pipe breaks.
- *Transportation:* A link disappears. •

Inquiry: Event

The characterisation of 'event' given above is far from satisfactory. The event concept characterisation is pragmatic. A more satisfactory characterisation might be:

> *An event can be described by a predicate, by a time (point), and a pair of states (the "before the event" and the "after the event") such that the predicate holds for these two states [and the time].*

In these notes we do not ascribe time points with the occurrences of actions. That should be done in subsequent work. Likewise we do not ascribe time points with the occurrences of events.

The philosophic concept of event is treated by, for example, [15, 35, 31, 23, 34, 9, 14].

•

3.4 Behaviours 103

By a behaviour we understand a set of sequences of actions, events and behaviours.

104

Example

Example 23 (Behaviours) Five domain examples:

- *Container line:* The transport of a container from it being fetched at the sender, via a sequence of one or more triplets of loadings onto a vessel, unloading at another container terminal port and possibly temporary storage at that port, to its final delivery at a receiver. 105
- *Financial service industry,* account handling: the opening of an account, a sequence of deposits, withdrawals and statements to the closing of that account. 106
- *Health care,* patient hospitalisation: the admission of a patient to a hospital, initial anamnese, analysis, diagnostics and treatment plan, via an alternating sequence of treatments (including surgical operations), repeated analyses, evaluations and possible reformulation of diagnostics and treatment plan, to a final discharge. 107
- *Pipeline system,* simple, day-to-day operations. The flow of gas (or a liquid) through a pipeline net: pumped from wells, fed through pipes, valves, forks and joins, to leaving the net at sinks. 108

- *Transportation:* The movement of a vehicle along a transport net:

  from positions at hub or link positions via a sequence of zero, one or more hub and link movements, to a final hub or link position.

  Example 4 (Pages 10–14) illustrated a transport behaviour. •

Inquiry: Behaviour

> *The notion of behaviour is not the same as the notion of process. We shall reserve the use of the term behaviour for "what goes on in the domain", and we shall use the term process for "what goes on 'inside' the computer".* •

3.5 Discussion 109

We have dealt, in some detail, with the concept of parts (Sect. 3.1, Pages 26–35). Our "corresponding" treatment of actions, events and behaviours (Sects. 3.2–3.4, Pages 35–38) have been far less detailed. The reason for this is the following. Types emerge (Sect. 3.1) as a means of describing parts. And types are indispensable in the description of action, event and behaviour signatures (Sects. 3.2–3.4). Types thus form the very basis for the description of all entities. And we have chosen to let the type concept emerge from our
110 treatment of parts. There is another reason for Sect. 3.1 being somewhat more detailed than Sects. 3.2–3.4. When studying parts we could, relatively easily, introduce such notions as atomic and composite parts, attributes of these, and mereologies of composite parts. These notions, under some disguise, can likewise be found for actions, events and behaviours, but they are not that easily introduced.