CS405 Introduction to Empirical Modelling 2009-2010

DOSTE and EDEN in relation to EM

Our primary concern is for how the relationship between a construal and its referent is apprehended

cf. *grounding* of symbols in logic, where the association can typically be made in many ways, and the generic quality of the abstract symbol is a virtue (*because* it is not subjective, not particular)

Compare 'the Bad Hair Book', where the association between a specific head of hair and an independent image is something to be experienced (if you have the right cultural familiarity etc)

Can mix both kinds of representation, but Empirical Modelling is primarily about making direct associations.

NB the connections we make in experience can be learnt - what doesn't at first connect, eventually does (cf reading words in a foreign language)

NB because of the dependence on the viewer, and the viewer's familiarity etc, the quality and expressive potential of a computer model cannot be assessed at first glance (e.g. who would guess that musical notation could be interpreted so readily by a skilled musician? - and that they could be able to interpret one notation but not another similar one?)

Three factors that are of interest:

1. How the computer model captures state
2. What sort of direct state transition it affords
3. How the states and state transitions correspond to external states and state transitions

So, for a musical extract: state expressed by vertical alignment, transition by horizontal progression, each musical line corresponds to a different instrument, and dependencies link the actual notes as sounding to the notes as written

Note the way in which many agents are represented, each with a different view of the music - the closest to the modeller might be the composer, or the conductor (contrast their agency, and consider whether the music is being composed, being rehearsed, or being performed)

Useful analogy:

- Can capture "the same" information using a synthesiser, but instrumentalists wouldn't play from it
- With a synthesiser, the individual oddities of instruments wouldn't be captured
- With a synthesiser, you wouldn't have to have conventional instruments at all

The semantics of DOSTE

1. State representation

In DOSTE, the current state is represented by a combinatorial directed graph

Each node in this graph corresponds to 'an object'
  ["burdened with additional meaning that is not appropriate here" - see Pope's DOSTE documentation]
  an object may be very broadly interpreted ...

Each edge is labelled by a node in the graph - one way to conceive this is as current state is a binary function F: U * U --> U which is subject to be redefined through the addition of more elements to U, and modifications to F which may be made directly by the modeller or effected through programs.

[Aside: Curiously, the only commonly studied external referent that precisely matches this that I know of is "the Cayley diagram" - and ARCA - the first product of research in EM at Warwick - was designed with "manipulating Cayley Diagrams" in mind]
[Compare and contrast the value/status associated with a conventional variable/observable as in Eden, where all indications of organisation and structure have to be conceived in terms of associations of observables that aren't supported by the interpreter, and exist only in the mind of the modeller, possibly mediated to some degree by naming conventions, and use of list structures etc
- in DOSTE "the entire graph" represents something, as do the individual nodes, and the relationships - which themselves are associated with nodes]

2. State-transition in DOSTE

The state-transition model for DOSTE is straightforward and maps directly to the state-as-experienced by the human agent interacting with the model

This can be construed as a sequential flow of states, perceived as a sequence of moments, in which there is a next state

An interesting and distinctive feature (cf. declarative programming) is that transition in DOSTE makes no reference to agency

[Yet the effects of what can be construed as intervention by some external agency can be emulated - the modeller can at any point directly change the graph structure, or introduce some recipe for change, such as was illustrated in the rotation / re-orientation of the Stargate]

[The DOSTE interpreter reflects this in its updating mechanism: at each moment in time the current status of the graph is updated

The speed of this updating is 'as fast as the computer allows', in keeping with the idea that are perception of state change is a sequence of transitions from one discrete state to another that occurs 'as fast as possible' (as when we speak of "the very next moment, ...")

Though this update can be seen as creating a transition from one state to another, it is not construed as attributable to any specific agent - cf. the way in which we interpret the moment-by-moment transitions in our stream-of-consciousness ("things happen")]

[DOSTE exploits the fact that the timescale on which computer moments operate is orders of magnitude smaller than that of human-perceived moments

So as long as the state model that DOSTE maintains is being interpreted by humans it hardly matters (? at least in many many contexts) that 'is' should be understood as 'becomes', and relates future values to present values]

[A remarkable feature of DOSTE is that it exploits the parallel between the experiencing mind, as associated with "the stream of consciousness", and the computer as a device that keeps on refreshing its state instant-by-instant]

The semantics of '=' and 'is' in DOSTE is such that

'=' specifies the graph structure at the moment of interpretation - it specifies the present only

'is' specifies potential changes to the structure of the graph - it specifies the future (only).

Examples of this

```
a b = c

a b is { x y z }
```

So for instance:

```
a is {b}
b is {a}
```

repeatedly swaps the values of a and b, expressing the idea that "a will be what b is", and "b will be what a is"
[Aside: Note the many possible interpretations of *dependency* that we might adopt - as in

"at each update x acquires the value of the RHS"

"when a value on the RHS changes, x is updated"]

3. Making the semantic connection with an external referent

Consider the distinction between how structures in the graph serve to

- define the boolean operators
- to express the hierarchical decomposition associated with the division of Stargate into components
- to specify dynamically changing "analogue" quantities using "is" and self-referent nodes

This shows that a great diversity of modes of interpretation is being invoked.

In the spirit of EM, the interpretation of an object is typically evolving as the model is being built

A major issue for an empirical approach:

- since there is no declared upfront specification, the association of components/features in the computer model - ideally "as experienced by the modeller" - with features of a real referent has to be apprehended in experience according to some principles (?)

How is this interpretation made in experience?

Both the skill of the human interpreter and the nature and quality of the computer model are significant

Two different modes of assisting interpretation

- what is explicitly encoded in the model (cf. how a logical specification is connected with its referent using natural language)
- what is implicit in the latent behaviour of the model and can be inferred by exploratory interaction with the model

Explicit encoding of meanings ...

- structure is significant - association of observables [cf. identification of Stargate from components - cf. stargate.d]

- names we give to components

- visual expression that is realistic and recognisable (cf. "that's a Stargate")

Implicit sources of meaning

Meanings latent in how observables depend on one another and what agency is possible

- **observe** how the values of observables are perceived to change autonomously, and
- **explore** how they change interdependently in response to intervention
  visual/perceptual expression is again significant (but some scalars etc are abstract and generic?)

  nature of this dependency? - more difficult to interpret if cyclic, and dynamic

  cf. "recognising booleans" and "recognising integers" as cases to consider here

- Relevant exercises are ...

  identifying an anonymous object from a dasm script

  identifying how the observables in a dasm script correspond to a known object

  [Here also like to consider the possibility that the manipulation of the graph might be made much more direct, eliminating names]

Another aspect of this semantic connection is the quality of the experience that the computer can offer

quality good for DOSTE in many aspects, since

- using the quasi-analogue quality of the computer processing effectively

- the "computation as navigation" model eliminates the lumpiness of traditional expression evaluation

  - at the primitive level all computational steps are uniformly costly in some sense

  [at least as far as identifying - if not visualising a value are concerned]

  good potential here for concurrency - consider multi-core architectures

- switching references can change the references and values of many observables simultaneously

  [contrast gross change of state by importing a file of definitions with "switching a pointer"]

... at same time clear that the level of primitive update of state (by 'navigation') is not always well-matched to perceived dependencies in the referent, nor adequate for performance

- hence the need to make "abstractions"

  - e.g. like types and operators (as in case of integers)

  or prototype objects like buttons etc

... realisation of these then presents another challenge to implementation in light of need for indivisibility

Clearly an excellent match:

- with analogue observables

- with object-like generic features
- with references that are established by indirection ("the colour of my mother-in-law's car")

More about the *directness* of the state-transition

Directness is concerned with what *doesn't* get to be - or need to be - explicitly interpreted

E.g. in a spreadsheet don't interpret the computation that does the updating of definitions

[Natural enough when the formula that defines the value in a cell is "profit is expenditure-income",

... but not so clear if "A1 is the-number-of-non-zero-bits-in-the-binary-representation-of A2"

- especially if "A2 is the-number-of-non-zero-bits-in-the-ternary-representation-of A3"]

[cf concrete aspects of the orchestral score that aren't musically significant

- what size the notes are, how they are spaced out, where the page breaks are, the quality of the graphics]
The need for explicit interpretation relates to whether relationships are recognisable in direct experience, which presumes familiarity.

What we expect to be able to see and interpret depends on our role and purpose

- Programmer - what can be represented on the computer?

- Analyst / Domain expert
  - what are the potential capabilities/privileges/roles of relevant agents in the environment?
  - what applications do these afford?

- Application user - how can a specific objective be most conveniently and efficiently achieved?

In relation to DOSTE, the programmer, analyst and user engage in interaction that is the same **nature**
- but it admits different interpretations, and modes of observation and interaction

The term "expertise" that can be applied to analyst, programmer and user reflects the nature of their familiarity with the artefact
[Closely parallel to the roles of the analyst, composer and performer of a piece of music]

By way of example, in the DOSTE context, we may not be concerned with

- how integers are computed
- how objects are specified from a prototype (e.g. is any particular chevron 'the one' from which all others are cloned?)

A key message: *How successfully the connection between construal and referent is made reflects both the medium and the way in which, and by whom, it is being used*.

[Cf. Playing Beethoven's Hammerklavier piano sonata involves generating a succession of sounds from the score in a way that makes sense of the music. You can't of course play the Hammerklavier on a mouth organ, but you can't necessarily play it on the piano either. Somebody else *can* play it on the piano. Beethoven composed it, but probably *couldn't* play it.]

Contrast the idea that adopting some particular programming paradigm will make everything transparent.

The relevance of human cognition, culture, disposition etc

Laws? or Agents?

- Bertrand Russell applauding the substitution of law-like "scientific" explanation for agency

- The status of phenomena that can't be readily predicted

  earthquakes (cf. when will I pour boiling water on an ant colony?)

  chaos theory

- Margaret Boden *explaining* the creation of poetry and music
  cf. sources for Coleridge's "The Rhyme of the Ancient Mariner")
  mechanising the artistic playing of Chopin

- Not that there isn't a pattern, but we can't recognise it
  [cf. the "mystery" of the telephone debunked by the 19th century rationalist, who says
  "just look at the wires"... and adds:
  "if you *couldn't* see any wires, now *that* would be a mystery"]

For more discussion, consult the analysis of the story of the "ficts" in EM paper #037, as summarised here.

DOSTE (at least as so far developed) is suited to the expression of particular kinds of construal, in which state-as-experienced is matched to continuous processes and agent interrupts, mediated by analogue observables, fluid object-like structures and dynamic references established by indirection.

- all observables are perceived as "living" and evolving instant by instant
- 'is' describes the relations that determine the next values of observables
- can't introspect about intermediate states in the evaluation of expressions by navigation
- no direct counterpart of the spreadsheet-like "is" linking "static" observables
  - have relationships in which "input" and "output" observables are in transition, possibly from one stable state to another
  [consider the way the gearbox makes a transition from one gear to the next, not as a discrete action]

Isn't everything like this - aren't all our construals conceived in terms of processes?
cf. "understanding forwards" vs. "understanding backwards" [William James] ...

Two complementary construals

- everything changes all the time according to laws e.g. of physics (and others undiscovered)

- all change is attributable to agency: character of change - e.g. whether it can be deemed process-like - is what is construed by the observer

Want to support both kinds of construal ...

- DOSTE as a 'new way to program', specifying processes and free agent intervention
  EDEN as not about programming in the narrow sense of specifying rule-based processes
  [programming can be taken in a wider sense to embrace what precedes writing the program]

- focus of attention on both DOSTE and EDEN is on "the present moment"
  DOSTE provides concepts to enable exceptionally dynamic and fluid crafting of processes as they are being experienced
  in EDEN there is no presumption of process: emphasis is on the way in which the present moment speculatively offers affordances for interaction and interpretation to many agents - processes may be crafted out of these to suit specific purposes if required