CS405 Introduction to Empirical Modelling 2009-2010

More about DOSTE and EDEN in relation to EM

The ADM is a useful framework in which to see how EDEN supports EM. It doesn't give a good account of how DOSTE currently supports EM. This motivates a deeper comparison of how DOSTE and EDEN support EM. What I've termed the *three key concerns of EM* are:

- representing state as experienced in a referent,
- representing direct transitions as experienced in interaction with a referent,
- how the connection between artefact and referent is itself made in experience

This agenda is different in a subtle way from the agenda we considered when first discussing DOSTE. Specifically, if we refer to the previous lectures on DOSTE for EM, we considered:

1. How **the computer model** captures state
2. What sort of direct state transition **it** affords
3. How the [i.e. **its**] states and state transitions correspond to external states and state transitions

In that setting, the discussion of DOSTE made explicit reference to the role of the computer. But in fact DOSTE and EDEN can both be interpreted as idealised perspectives on the three key concerns - both are ways to represent state and transition that are meaningful in isolation from the computer. In fact, it's best to think of them in isolation from the computer. Can regard them both as ways of expressing / conceptualising what we experience.

Idealised conceptualisations of experience

To recap, these are the idealised characteristics that **D**OSTE and **E**DEN offer:

How can something we construct …

Represent state currently being experienced in a referent?

- D: Combinatorial state graph for evaluating expressions that define values of observables together with extant processes that update observables
- E: Set of observables and dependencies expressed in the form of a ‘definitive script’

Represent the direct transitions of state – those that can be experienced as meaningful in the referent?

- D: Can change the evaluation and updating mechanisms on-the-fly, by = and 'is'
- E: Make redefinitions that change the current values and dependencies between observables on-the-fly, by = and ‘is’

Enable the connection between construction and referent itself to be experienced?

- D: High degree of realism, analogue observables, expressiveness, indirection in reference
- E: Establishes a correspondence between patterns of observable, dependency and agency in construal and referent, as can be experienced through experimental interaction

---

Relating the two conceptualisations ...

The state-transition-interpretation characteristics of DOSTE and EDEN reflect different perspectives on

experience and the nature of observables (what I referred to previously as *horizontal and vertical views of 'now'* respectively).

Observables of both kinds have a role in construal, and conceptually they can be blended ("is it most natural to see an observable as having a *static* value 'now' or as always having a *dynamic* value 'now'?")

In presenting EM in the past, we have argued for EDEN observables as *more primitive* than identities/objects and processes - the associations of observables in relation to identity and over time being made by experience.

You can equally well argue the reverse - that the static view of an observable is a construct derived from the dynamic view, and that EDEN-like dependencies reflect potential relationships and stabilities that may or may not emerge.

In keeping with the tradition of EM, it would be inappropriate to favour either presumption about what is primitive.

The role of the computer ...

In this context of comparing the static and dynamic views of experience, it is also worth remarking that how well something can be realised on a computer also has a lot to do with the nature of the computer itself.

An illustrative example: Consider "I sign this paper, and the house is mine" - but can this degree of indivisibility / conjunction be realised in a computer implementation? Perhaps if an EDEN "a is b+c;" is evaluated in such a way that 'a' is evaluated from the current values of b and c whenever it is required (rather than updated eagerly when they change). More plausibly in the DOSTE state graph when the edge "a x = .y" is redefined, then the values of all expressions that are computed by navigation through the edge x from node a will be immediately changed.

A significant difference between DOSTE and EDEN is that DOSTE is far better oriented towards present-day computer implementation. EDEN has no object hierarchy, makes quite unrealistic assumptions about the indivisibility of function evaluation, exploits metaphors that don't map directly onto the computer architecture (tables, points and lines, displays), supports a plethora of notations all with very different characteristics, taking many different syntactic forms and exceedingly messy to integrate. But DOSTE also has some idealisation: it treats navigation as instantaneous, optimises away the navigation involved in representing arithmetic, and treats instants as registered by the machine as in some sense infinitessimal (though they are necessarily finite).

From the EM perspective ...

Argue in EM for the conceptual need for all manner of subtlety in observables and dependencies - can't realistically expect to express using a computer-based artefact all the relationships that we experience - shadows, nanotechnology [scale], continuity [lines demo] .. but then we can't display a mathematical line either. So our construals may often [will always?] be imperfect, but this imperfection doesn't necessarily obstruct our sense-making ("I can regard a set of pixels as an idealised line"). Machine and human observer dependence is another facet of this.

A closer look at EDEN and the static observable perspective

The idealised view

First give "idealised" responses to the three key concerns, then consider these from a more "realistic" perspective.

1. **Capturing state:** The current state is at all times recorded in "a definitive script" - a family of definitions. A definition typically takes the form:

   ```
   a is f(b,c,d);
   ```

   where f is a function $U \times U \times U \rightarrow U$. The process of updating the values of observables so that the relationship is at all times valid is called 'dependency maintenance'. The dependency maintenance activity is not interpreted - it is treated as if it involved instantaneous update.

2. **Direct transition:** A direct change of state is expressed by a set of (re)definitions. Provided that two redefinitions don't conflict (e.g. both redefine the same observable, or they introduce a cyclic definition), these can typically be performed concurrently.

3. **Connection with experience:** For each script, there will be many possible families of redefinitions that we may wish to interpret. Interpretation relates to recognising that the pattern of change in values to observables in the script is matched to a pattern of change observed in an external referent. Will also typically identify these with particular kinds of agency in the situation. The meanings that can be attached to a state are determined by the correlation between direct interactions with the script and with its referent.

There is an implicit "observing / acting" agent in the above context. Perception of state relates to what is deemed to be concurrent - to relate to observations made *at the same point in time*. (For instance, the experimental scientist may measure the values of several observables at actually different times, but presumes that they can be regarded as *simultaneous* observaions. A simple example might relate to measurements of the load and extension of a string, such as might be used to verify Hooke's Law. An engineer would remark that these measurements are not in fact time-independent, as a loaded string is in fact subject to extension due to 'creep' if measurement of its length is postponed after loading.) Potential actions that might probe the dependencies between observables may or may not be such that they can be carried out by the agent. (Consider how observation of shadows made by the sun might disclose dependencies that cannot be verified by physically moving the sun.)

Taking a more realistic view of how EDEN addresses the three concerns ...

1. **More about capturing state**

   EDEN definition is in spirit similar to definition in a spreadsheet

   ```
   A1 = B1 * 2
   ```

   This describes a static relation to be dynamically maintained - specifying a relation that pertains between the **current** values of observables. Semantics of an EDEN definition:

   ```
   a is b+c;
   ```

   If b or c changes in value, the value of a is updated, *as if in one and the same atomic transition.*

   Why is this different from the DOSTE model? It's more dishonest - at least as far as computer implementation is concerned!

   ```
   .a is {.b+.c};
   ```

in DOSTE means that value of .a *at the next instant* will be what is the current value of .b+.c

May nonetheless be the case that the human observer detects no distinction between the one relation and the other, at least if the values of .b and .c are subject to change only at a rate that is matched (in slowness) to human perception.

What is the semantic difference? - if we have chains of DOSTE-style definitions, there may be a "hiaton" issue. For instance, if the definition `.a is {.b+.c}` is interpreted in the presence of the definition `.c is {.c+1}` then the current value of .a is never the current value of .b+.c.

Of course, in EDEN evaluation, there will also be an instant (and potentially much much longer periods of time) in which the values of a, b and c are inconsistent. Indivisibility relates to the notion that the observing agent cannot access such inconsistent states, either to observe or act. Nor shall we attempt to interpret these intermediate states.

2. **More about direct transition** What is deemed to be reachable by direct transition is linked to the existence of particular kinds of agency. In designing a new building our construals must take account of all the changes that the architect might make, and the how people will use the building, but doesn't consider meteorite strikes.

3. **More about connection with experience** The basic mechanism for establishing a connnection in experience is illustrated in the digit-cabinet example. The use of definitive notations based on standard underlying algebras affords familiar functions that the trained modeller will be accustomed to recognising - cf. the nonsense functional relationship example (A1 is the number of 1s in the binary representation of the value in cell A2 etc)

Construals in conception and realisation

As far as sense-making is concerned, the benefit of a construal doesn't depend entirely on whether it can be realised physically in a way that is faithful to its referent in every aspect. Mathematicians had means of computing the effect of a billiard ball collision long before there was the technology to simulate such a collision as if in real-time. Likewise, Faraday surely didn't build physical artefacts to represent every specific situation he encountered in investigating electromagnetic phenomena. In that sense construal can be linked to "thought experiment" - subject of course to making the construal using objects with which we are very familiar.

It is when we want to realise a system on the basis of our construal that the issues of physical embodiment in real-time and with realistic interfaces for response, visualisation and manipulation become most significant. For instance, it's when we have to automate a system component through computer implementation that the limitations of EDEN and the merits of DOSTE are most apparent.
