Empirical Modelling with Cadence, DOSTE and EDEN

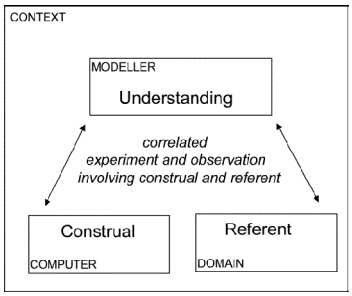
Key idea behind EM

Concerned with the process of sense-making through incremental construction

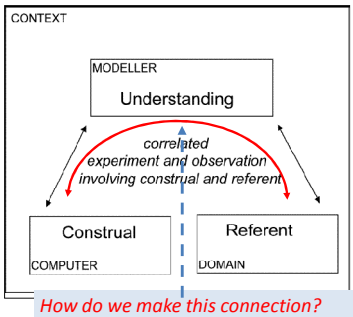
This doesn't have to involve using a computer, but the computer has liberated construction

Key problem: meaning has to be attached to the current state of the construction

Attaching meaning to the current state



Connection in experience



How do we make this connection?
How is this connection made?

William James
Essays in Radical Empiricism (1910)

See the appendix to Lecture 2 on Concurrency on CS405 website:
Empirical Modelling for the Single Agent

Principles of Radical Empiricism

Radical Empiricism consists in:

- a postulate
  the only things that shall be debatable among philosophers shall be things definable in terms drawn from experience
- a statement of fact
  the relations between things, conjunctive as well as disjunctive, are just as much matters of experience, neither more nor less so, than the things themselves
- a generalized conclusion
  the parts of experience hold together from next to next by relations that are themselves parts of experience. The directly apprehended universe needs, in short, no extraneous trans-empirical connective support, but possesses in its own right a concatenated and continuous structure.

Matches in experience

The "**Swan Vestas**" haircut

Matches in experience

Matches in experience

Matches in experience

Matches in experience

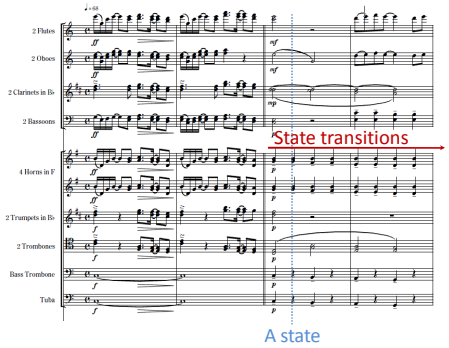
The Mouldy Shower Curtain

The Lava

Photographs from *Bad Hair*, ed. James Innes-Smith and Henrietta Webb, Bloomsbury

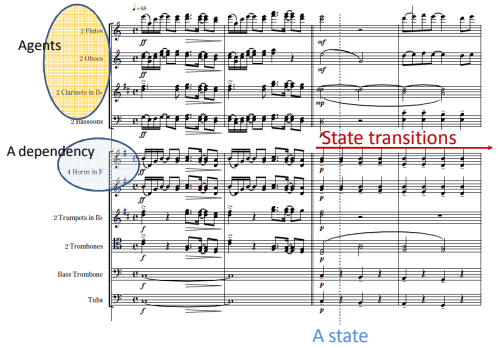
Orchestral Score Sample

*faculty.goucher.edu/eng215/**orchestra_score**.pdf*



Orchestral Score Sample

*faculty.goucher.edu/eng215/**orchestra_score**.pdf*



Three key concerns

How can something we construct …

- Represent state currently being experienced in a referent?
- Represent the direct transitions of state – those that can be experienced as meaningful in the referent?
- Enable the connection between construction and referent itself to be experienced?

In addressing these key concerns

What concepts can we use?

(People have always built construals – even before there were computers)

And what support can a computer model give to addressing each concern?

(The answer here will be more specific to computers – could ask same question for other technologies cf. analogue computers)

Three key concerns wrt MWDS

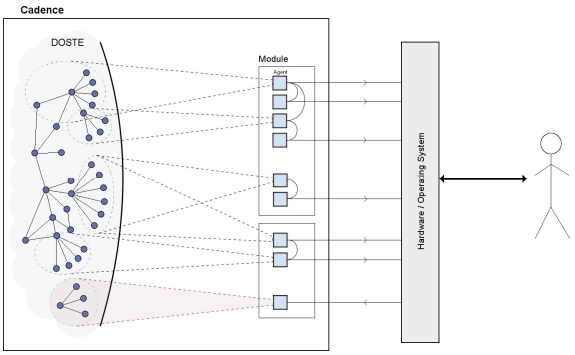
How can something we construct …

- Represent state currently being experienced in a referent?
  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? 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? Establishes a correspondence between patterns of observable, dependency and agency in construal and referent, as can be experienced through experimental interaction.

Three key concerns wrt DOSTE

How can something we construct …

- Represent state currently being experienced in a referent?
  Combinatorial state graph for evaluating expressions that defines values of observables together with extant processes that update observables
- Represent the direct transitions of state – those that can be experienced as meaningful in the referent? Can change the evaluation and updating mechanisms on-the-fly, by '=' , ':=' and 'is'
- Enable the connection between construction and referent itself to be experienced? High degree of realism, analogue observables, expressiveness, indirection in reference …



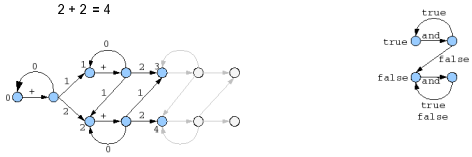
The CADENCE environment and the DOSTE engine

Models of integer arithmetic and boolean logic in DOSTE

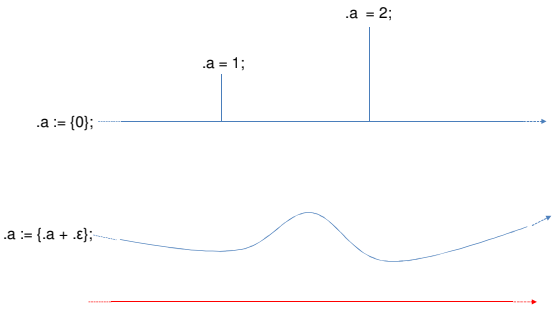
(from Nick Pope's PhD thesis – forthcoming)

```
0 + = (new);
0 + 0 = 0;
0 + 1 = 1;
0 + 2 = 2;
1 + = (new);
1 + 0 = 1;
1 + 1 = 2;
1 + 2 = 3;
2 + = (new);
2 + 0 = 2;
2 + 1 = 3;
2 + 2 = 4
```

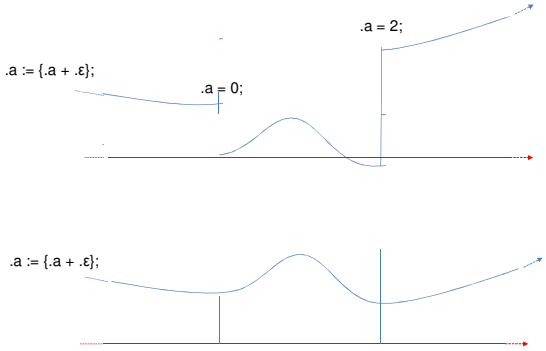
```
true and = (new);
true and true = true;
true and false = false;
false and = (new);
false and true = false;
false and false = false;
```



DOSTE semantics



"Agent intervention" in a process



Comparing DOSTE and EDEN syntax & semantics

DOSTE	=	Changes the configuration of the state graph as of now
	:=	Defines how the state graph is reconfigured from one instant to the next
DOSTE Eden / Eddi	is	Defines the value of an observable so that it is now and is thereafter unless and until redefined the current value of an expression
Scout / Donald	=	
Eden / Eddi	=	Assigns the value of an expression to an observable

Scout and Donald act as pre-processors to Eden but have their own symbol tables
Scout and Donald definitions translate to 'is' definitions in Eden
Scout observable names translate unchanged from Scout to Eden
Donald observable names, which in general refer to nesting within **openshapes**, translate into observable names including '_' symbols. For instance:
room/width in Donald maps to **_room_width**
Attributes of the Donald observable **X** are defined by the observable **A_X** in Eden

More about EDEN syntax

- Scout and Donald act as pre-processors to Eden but have their own symbol tables
- Scout and Donald definitions translate to 'is' definitions in Eden
- Scout observable names translate unchanged from Scout to Eden
- Donald observable names, which in general refer to nesting within **openshapes**, translate into observable names including '_' symbols. For instance: **room/width** maps to **_room_width**
- Attributes of the Donald observable **X** are defined by the observable **A_X** in Eden

Comparing DOSTE and EDEN semantics

DOSTE	**=**	**Changes the configuration of the state graph as of now**
	:=	Defines how the state graph is reconfigured from one instant to the next
DOSTE Eden / Eddi	is	Defines the value of an observable so that it is now and is thereafter unless and until redefined the current value of an expression
Scout / Donald	=	
Eden / Eddi	**=**	**Assigns the value of an expression to an observable**

MWDS in relation to Radical Empiricism

- modelling with definitive scripts is a means to creating conjunctive relations in experience
- BUT conjunctive relations need not be of this nature (e.g. my aversion to tuna in puff pastry)
- by realising dependencies using observables that appear to be recognisable in an objective sense, EM contributes to communication between the private and the public …

MWDS characteristics ….

- network of definitions with EDEN-style observables and dependencies
- definitions are based on a definitive notation:
  - standard operators on the RHS of defns
  - standard routines for visualisation
- generic communication based on objective artefacts
- relationships framed using 'abstract algebra'

Definitive notations and scripts …

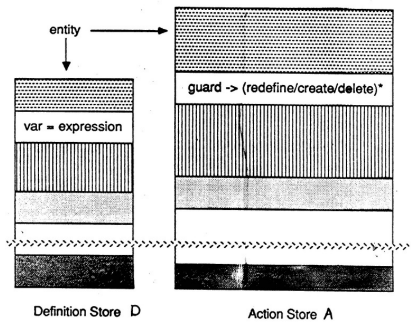
Simplifying assumptions, such as in spreadsheets:

- context not too rich (» *use scripts of definitions*)
- autonomous behaviour is limited (» *only use 'is' and '=' definitions*)
- perceived structure is primitive / embryonic (» *have a flat space of observables*)
- modeller not an expert in software development (» *modeller frames defns / programs functions*)
- modeller has domain-specific expertise (» *supply familiar types, operators, depictions*)

… basis for older EM tools more limited than Cadence

… try to address these problems in the ADM ….

The Abstract Definitive Machine: entity = definitions + actions



Many perspectives on ADM ….

- definitive parallel programming
- animation of LSD accounts
- conceptual framework for EM in EDEN

- machine-computing-oriented viewpoint
- human-computing-oriented viewpoint

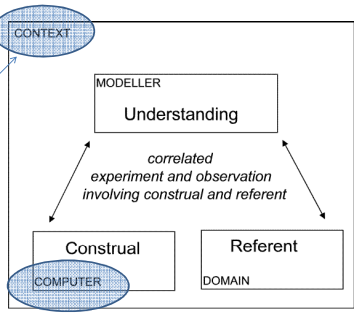
Aspirations for the ADM …

Catering for state-process duality in experience:

- continuity in the stream-of-consciousness
- State → process: cf. a stream of discrete snapshots experienced as motion
- Process → state: cf. cathode ray tube picture as static view (*human perception of illumination of a fluorescent scr*een) of a dynamic process (*motion of the electron gun*)

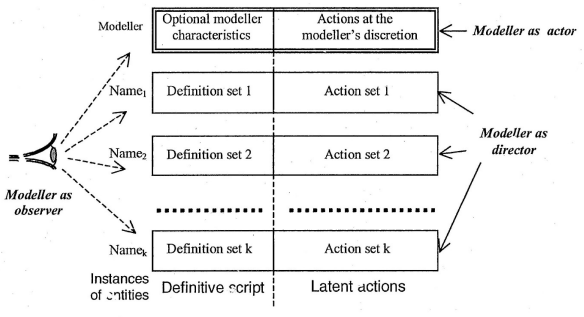
Empirical Modelling as *Construction*

Cadence has exceptional qualities in respect of *management of observational contexts* and the development of construals *on a computer*



Definitive parallel programming? … and more

- Computational model of Cadence can express this and more
- Can represent ADM-style entities
- The 'will be' feature of Cadence plays the role of the clock-based updating of the ADM
- Have super-agent privileges
- Potential for adding agents on-the-fly?



The ADM from the human perspective (Rungrattanaubol, 2002)

Conceptual framework for EM with EDEN

Two places where procedural programming invades EDEN …

- function and procedure definitions
- actions that invoke external packages

Instead use the richer constructs in Cadence e.g.

using functional-style definitions for operators

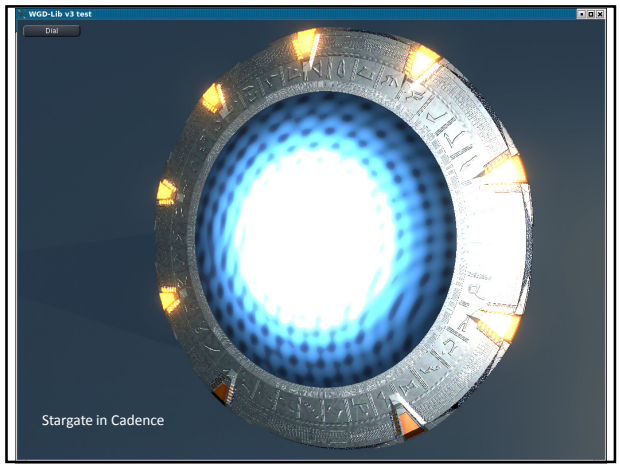
using EM-style modelling to mould the interface to external devices etc

Beyond the definitive script …

What the virtuoso programmer does is more than can be abstractly described without reference to the computer …

… cf. bloom in Stargate

Cf. musical performance …



Brödner, P. (1995). The Two Cultures in Engineering. In Goranzon, B. (Ed.), Skill, Technology and Enlightenment, Berlin: Springer-Verlag, 249-260.

Brödner, P. (1995). The Two Cultures in Engineering. (1)

One position, ... the *closed world* paradigm, suggests that all real-world phenomena, the properties and relations of its objects, can ultimately, and at least in principle, be transformed by human cognition into objectified, explicitly stated, propositional knowledge.

Brödner, P. (1995). The Two Cultures in Engineering. (2)

The counterposition,... the open development paradigm .. contests the completeness of this knowledge. In contrast, it assumes the primary existence of practical experience, a body of tacit knowledge grown with a person's acting in the world. This can be transformed into explicit theoretical knowledge under specific circumstances and to a principally limited extent only ...

Brödner, P. (1995). The Two Cultures in Engineering. (2)

Human interaction with the environment, thus, unfolds a dialectic of form and process through which practical experience is partly formalized and objectified as language, tools or machines i.e. form the use of which, in turn, produces new experience (i.e. process) as basis for further observation.

David Gooding: Experiment and the Making of Meaning, 1990

(as introduced by J.E.Tiles in his article Review: One Dimensional Experimental Science, in the British Journal for the Philosophy of Science,
Vol. 45(1), 1994, 341-352)

David Gooding, Experiment and the Making of Meaning, 1990 (p. xi)

[*By treating science as consisting entirely of declarative knowledge embodied in representations,* philosophy in general] ... bifurcates the scientist's world into an empirical world of pre-articulate experience and know-how and another world of talk, thought and argument.

David Gooding, Experiment and the Making of Meaning, 1990 (p. xi)

Most received philosophies of science focus so exclusively on the literary world of representations that they cannot begin to address the philosophical problems that arise from the interaction of these worlds: empirical access as a a source of knowledge, meaning and reference, and, of course, realism.

Richard Feynman:*The Pleasure of Finding Things Out*, p146

The scientist has a lot of experience with ignorance and doubt and uncertainty, and this experience is of very great importance, I think. When a scientist doesn't know the answer to a problem, he is ignorant. When he has a hunch as to what the result is, he is uncertain. And when he is pretty damn sure of what the result is going to be, he is in some doubt.