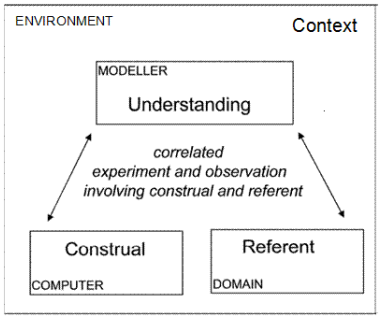
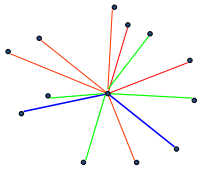
Empirical Modelling as *Construction*



Definitive scripts as "furry blobs"



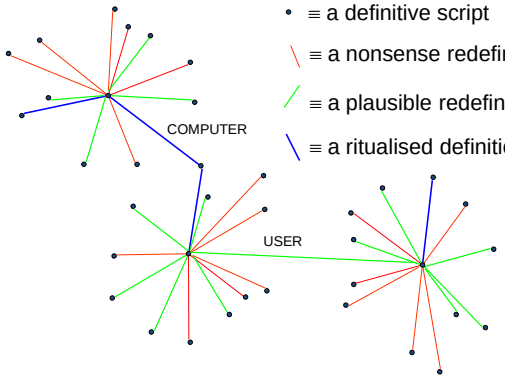
≡ a definitive script

≡ a nonsense redefinition

≡ a plausible redefinition

≡ a ritualised definition

Plausible : *could* open the desk drawer
– note continuous spectrum of redefinitions
Ritualised : door *automatically* closes after being opened
Nonsense : opening the drawer makes the room smaller



≡ a definitive script

≡ a nonsense redefinition

≡ a plausible redefinition

≡ a ritualised definition

Traditional programming

Requirements capture and specification	specification	Program design implementation maintenance	user interface	**Use** affordances interface culture
Identifying agency in the machine-like components and in the human context for use *Framing goals for the design protocols for interaction and interpretation* *e.g. devise UML*		*constructing and programming the machine-like components* *designing program by identifying objects and functions* *technical interface development* *e.g. writing Java code*		*human factors study* *interface design* *empirical studies of use* *prototyping* *e.g. goals, operators, methods (GOMS) evaluation*

Empirical Modelling

Requirements capture and specification	Program design implementation maintenance	**Use** affordances interface culture
develop scripts in isolation as "furry blobs" that represent the observables and dependencies associated with putative machine-like components and human interactions and interpretations	*identify and document reliably reproducible sequences of redefinition / chains of "furry blobs" that correspond to programmable automatable machine behaviours and ritualisable human behaviours and interfaces*	*exercise, explore, customise, revise and adapt sequences of redefinition and interpretation to reflect emerging and evolving patterns of interaction and interpretation; extend and augment observables to support additional functionalities combining scripts*

private experience / empirical / concrete

interaction with artefacts: identification of persistent features and contexts

practical knowledge: correlations between artefacts, acquisition of skills

identification of dependencies and postulation of independent agency

identification of generic patterns of interaction and stimulus-response mechanisms

non-verbal communication through interaction in a common environment

directly situated uses of language

identification of common experience and objective knowledge

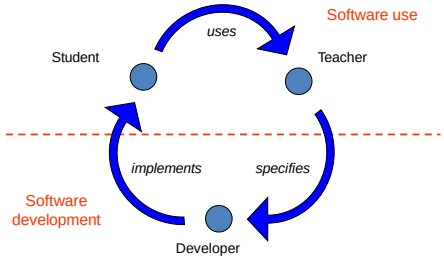
symbolic representations and formal languages: public conventions for interpretation

public knowledge / theoretical / formal

An Experiential Framework for Learning (EFL)

TEDC 2006

Developing educational software



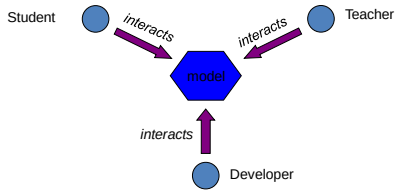
Perspectives of educational software

- Student vs teacher vs developer
- Mind-centred vs reality-centred
- Software development vs software use

➢ How can we bring together these different perspectives? Why?

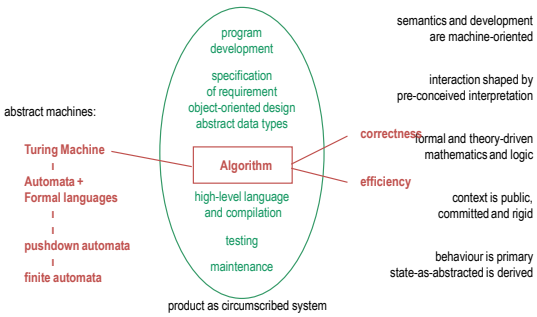
Empirical Modelling (EM)

- Offers a set of principles for model building in any of the student, teacher and developer roles:



Focus of conventional Computer Science

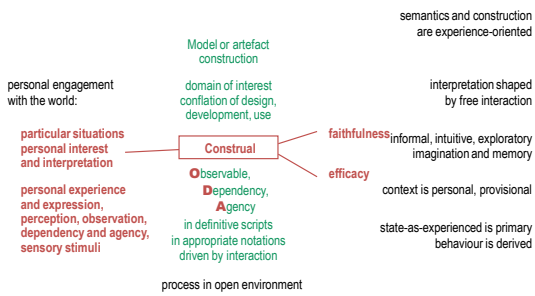
computation = execution of algorithm (cf. mechanism + automation)



Empirical Modelling: a broader view of computing

computation = making sense of phenomena *and* information processing (human computing)

observation and experiment



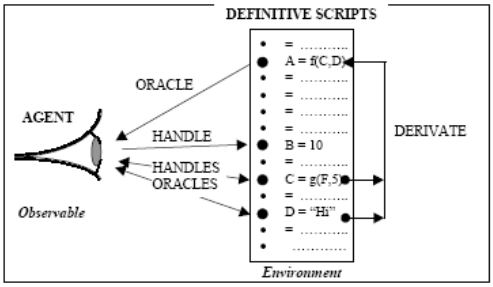


Figure 2-18: Definitive script as observer's model of state ('one-agent' modelling)

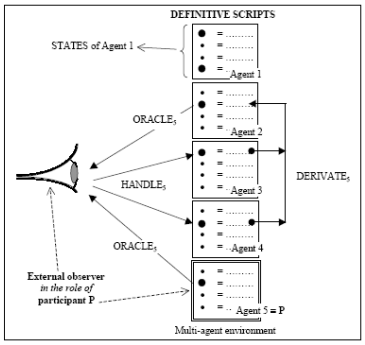


Figure 2-19: Definitive script as observer's model of state ('multi-agent' modelling)