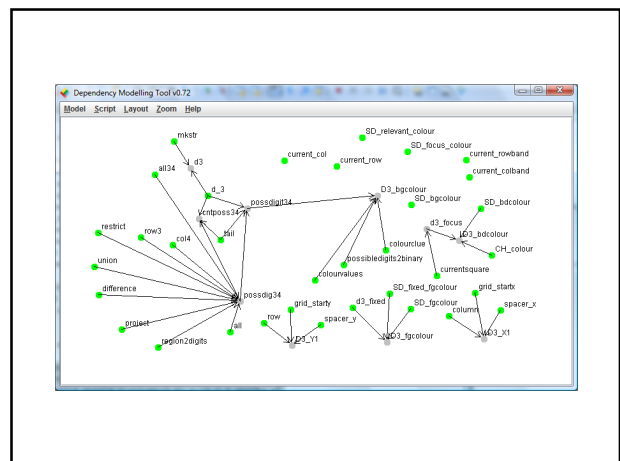
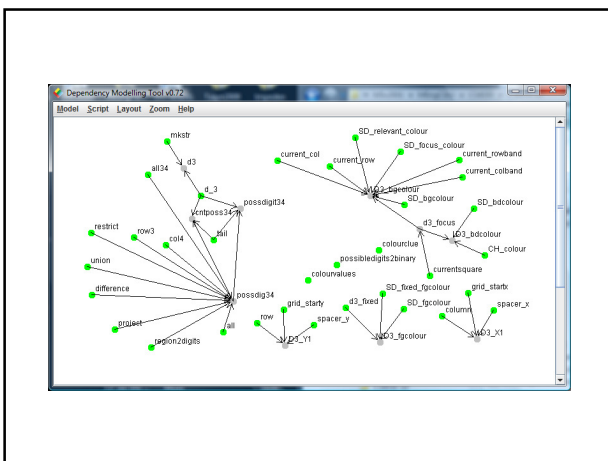
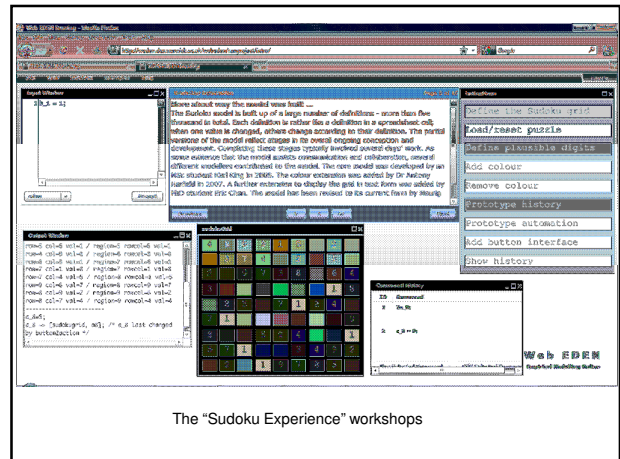
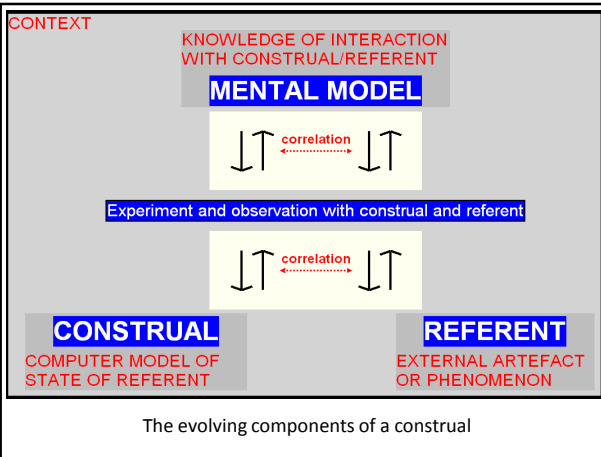
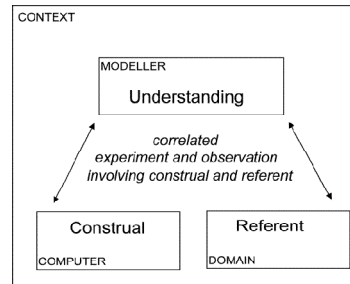
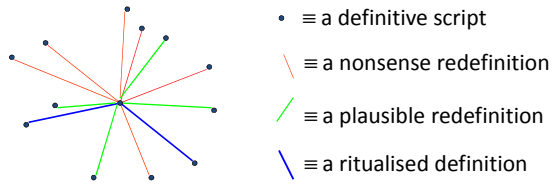


Empirical Modelling Key concepts

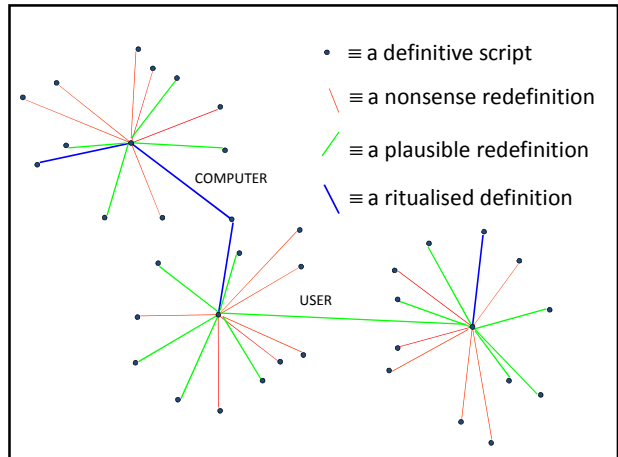
Meurig Beynon



Definitive scripts as “furry blobs”



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



Traditional programming

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

Empirical Modelling

Requirements capture and specification	Program design implementation maintenance	Use affordances interface culture
<i>develop scripts in isolation as “furry blobs” that represent the observables and dependencies associated with putative machine-like components and human interactions and interpretations</i>	<i>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</i>	<i>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</i>