Lecture 5: Empirical Modelling for concurrency: the roles of LSD and the ADM

The preceding lectures relating to EM for concurrent systems supply the context for most of the practical research on principles and tools. As has been emphasised, these lectures in fact motivate a radically different perspective on 'systems', and shift the focus to the exploratory and explanatory activities in the commonsense world that must precede system identification and specification. These activities reflect the characteristic movement in EM: from provisional towards assured knowledge; from the private to the public domain; from the subjective to the objective; from the experiential to the propositional; from the concrete to the abstract.

The lecture Concurrent Systems Modelling: Agentification, Artefacts, Animation identifies the three principal ingredients in EM. *Artefacts* are construals developed using definitive scripts. *Agentification* involves classifying the observables associated with a construal into clusters that have some integrity ("they come and go together") and to which some state-changing potential is attributed. The nature of this attribution is clarified in an LSD account. *Animation* involves automating interactions that are typically first conceived and tested through manual interactions on the part of the modeller. The environment in which this animation can most naturally be conducted is supplied by the ADM, which - see below! - can be interpreted as referring either to the Abstract Definitive *Modelling framework* (as, for instance, in Rungrattanaubol's PhD thesis) or to the Abstract Definitive *Machine* (as, for instance, in Slade's MSc thesis).

In making sense of an EM model, it is essential to situate it in the conceptual space between 'human' and 'computer' / 'manual' and 'automated'. As has been illustrated, an EM model typically begins as a fragment of script with no capacity for autonomously changing state, and only at a later stage incorporates automatic behaviours. The script may also evolve for a considerable time before it begins to take on entirely distinctive meanings in the modeller. That is to say, even though the modeller may have a specific referent or situation in mind, the initial forms of the construal will admit many interpretations, and might not invoke the 'intended' referent or situation for an uninformed observer. It is questionable whether it makes sense to think of a construal converging to the point where it has a single unambiguous interpretation. Certainly, a prerequisite for placing constraints on interpretation is that the interactions with the construal come to be restricted in such a way that it cannot be totally remodelled through over-writing key definitions etc. What does happen as the construal "takes shape" is that the modeller's interactions with it fall into more and more stereotypical patterns - usually patterns that have some particular significance in the referent. Some may also get to be automated, though only in such a way that state-change is effected by simulated redefinition rather than by actions optimised for a specific function. This means that, if automated operation is suspended, this suspension leaves the construal in a state that is conceptually similar to any other state, comprising observables and dependencies that are meaningful to the modeller.

What distinguishes a mature EM model from an embryonic one is the expressiveness of the meaningful patterns of interaction that surround it. This can only be fully appreciated by rehearsing many possible interactions from many different states rather than contemplating the script in any particular state. Notice that how far the intrinsic qualities of a model can be appreciated has a lot to do with characteristics of the human interpreter (e.g. *what interactions are you familiar with, both in the model and the referent?*). Notice also that the interpretation of a state of the construal, of each interaction with it, and of any programmed state-changing behaviours it exhibits, is also in the modeller's mind (cf. A Perspective on Concurrent Systems). There is an underlying semantic stance in EM, epitomised by the questions, *when does the evolving EM artefact become a model? when does it serve as a construal?* (As a relevant exercise, follow the development of the `vimodesBeynon2006` model as documented in the associated 'tour of the model' [access from the `~empublic/projects` webpage], and see at what stages you see evidence of its having the status of a model of `vi`, or serving a purpose in construing how `vi` works.) In the transition from initial conception to 'final' form, everything in Figure 1 gets to be shaped in some way (though not necessarily frozen or even closed): the EM artefact itself, its referent, the modeller's understanding and the context. This also affects the

role of the modelling tool, which may at first serve the purpose of 'a modelling framework' and subsequently resemble 'a machine'. This potential evolution in the role of the tool is part of the larger evolution of context that is hard to recognise because it cannot be made explicit (cf. the frame problem). All we can say about the context is that (on the empirical evidence) - provided we restrict interactions to certain kinds - some degree of integrity of observables and predictability in interactions can be expected. In this connection, note that "a mature EM model" doesn't necessarily have to be a monolithic artefact with a very specific referent - it might be that understanding is served as well as it can be by having many loosely connected EM artefacts that gives insight into independent aspects of a situation. The maturity of such a model relates to the capacity of the modeller to demonstrate deep understanding of the situation through making rich interactions and interpretations.

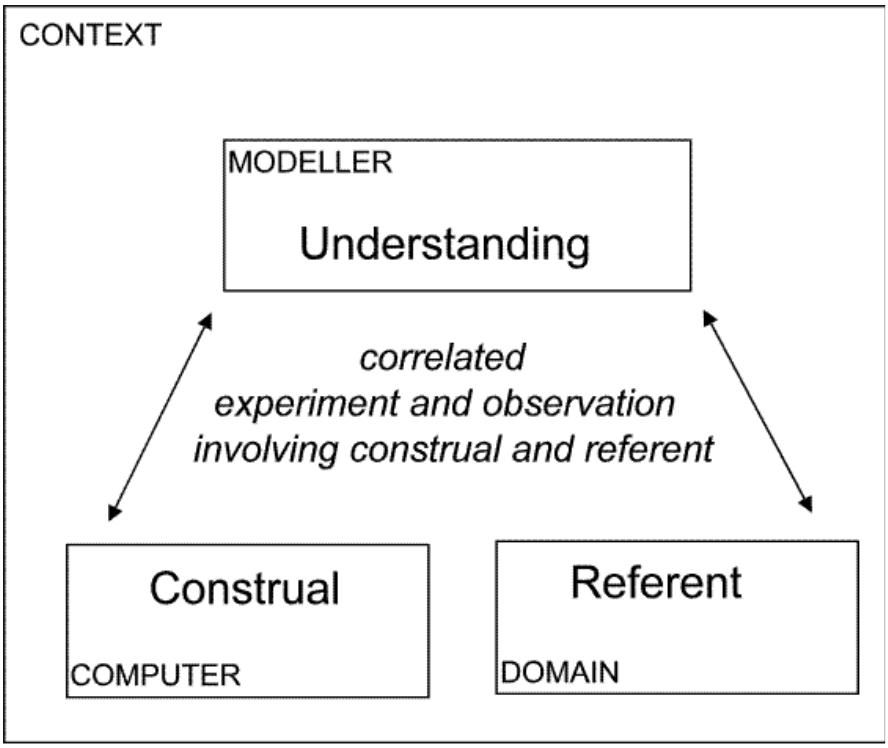

Figure 1: Empirical Modelling as sense-making or "construing"

LSD and the ADM

The thrust of lectures 1,2 and 3 relating to concurrent systems is to put in context the idea that:

- modelling with definitive scripts is an effective way to mediate understanding where there is no other agent at work apart from the modeller;
- that it is appropriate to use a similar approach to express the modeller's construal of a complex situation in which there are other agents present; *but*
- that exploring and documenting the patterns of interaction demands additional concepts.

This is the motivation behind LSD and the ADM.

The primary role for LSD is in documenting the modeller's interaction, and connecting this with what is construed to be happening by way of actions and interactions of agents in the referent. There is a simple underlying premise in LSD to the effect that any programmed synchronisation of concurrent action has to be mediated in some way by observables, and a primary purpose of the construal is to identify these. In any meaningful use of LSD, there must always be some experience in mind that supplies the observables to which the account refers. With reference to Figure 1, this experience may stem from a referent in the domain, or the modeller's construal. The LSD account itself relates

primarily to the modeller's understanding, and is closely tied up with the experimental interactions that inform this (cf. the document on "Empirical Principles for LSD specification").

It's important to appreciate that when the modeller follows the guiding principle of 'thinking anthropomorphically' about the agents construed to operate within the domain, a whole range of viewpoints on observables get to be represented (cf. the discussion of "Projection of Agency" in section 3 of Concurrent Systems Modelling: Agentification, Artefacts, Animation). In that sense, far from making sense of concurrent interaction in the domain, it potentially serves to problematise it, and with it the whole notion of an 'external observer'. This is why some way to shape the context is quite essential - this entails experimental activity concerned with the responses and interactions of agents that is intended to bring coherence to the views of the individual agents. (You can regard this as somewhat similar to the empirical research involved in building the hardware and interfaces that underlie a computer system so that they work effectively together. The term 'identification' is used to refer to this in a paper "Rethinking Programming" [087] to be discussed later in the module.) The ADM is the appropriate environment within which this search for coherence can take place.

When we consider LSD in the context of a system-like activity, such as the Arrival-Departure Protocol in the Railway Animation, it seems quite plausible that it can be a specification notation from which it is easy to derive behaviours. The reason for this is quite simple: the Arrival-Departure Protocol is an activity that has been empirically validated, and no doubt operates in much the way that we construe it to operate. That is to say, for most practical purposes, it does not appear to be necessary to invoke any other agents apart from the stationmaster, the guard and the driver, or to take account of any other observables, or to refine the context in any way. As the LSD account of the railway animation shows, there are nevertheless contextual issues that are implicit in the scenario - for instance, the reference to the two observables "whistling" and "has whistled" is an acknowledgement that a situation in which the stationmaster blew the whistle for eternity would be hugely problematic. There is nothing in the so-called "LSD specification" of the situation to reflect the fact that fortunately this is physically impossible.

It's when we investigate other contexts, such as the scenarios that arise in railway accidents (especially those prior to the advent of modern regulations and more mature communication technologies - but even to the present-day), that the difficulties of reaching closure in the search for stability and coherence in interaction amongst agents become apparent. Studying these accidents exposes the difficulty of identifying a suitable closed context in which to construe plausible operation, and indicates that in general we can only hope to make models that fall short of "being system-like". The Clayton Tunnel model illustrates what we *can* do, but it is a model that is most appropriately interpreted within the Abstract Definitive *Modelling framework*, in so far as essentially human, rather than robotic, agency has to enter the account.

More generally, when the same scenario is being viewed from different perspectives there is an issue about engineering a coherent common context. In the railway animation, the combination of the 'model railway', 'electronic track segments' and 'arrival-departure protocol' ingredients illustrates some of the key issues. Here the role of human agency in conflict resolution and the negotiation of conceptual integrity is again prominent.

Many of the above issues are clearly topical in applying object-oriented methods to reactive systems software development (see e.g. Harel's *Biting the Silver Bullet* and subsequent research on *Play-in Scenarios*). In Empirical Modelling that is in the near vicinity of such systems in the conceptual framework outlined above, there are some significant points of connection. For instance, there are many contexts in which the object-like integrity of clusters of observables is not in question, and their specific agency is patent. In such contexts, issues surrounding such clusters with object-like status include:

- managing their instantiation and deletion (invocation and dismissal),
- associating roles with these clusters according to context.

A model of a digital watch and associated statechart has been developed to exemplify how this might be specified in LSD.

Sources relating to the LSD notation

LSD was first introduced in 1986. There has been no attempt to give a comprehensive report on its design and use since the original report "The LSD notation for communicating systems" [CS-RR-87]. More thinking about LSD is implicit in Slade's MSc thesis, though the ghost of a fictional formal operational semantics features there. The original LSD report is no longer a good source, even though many of the key ideas touched on above are to be found in it, albeit couched in inappropriate terms. Until the planned revision of / sequel to this report is available, it is about the only place where some of the points discussed above get to be elaborated. However, to read it with any insight, it is essential to note that:

- The discussion is throughout conducted as if LSD were really much more to do with systems and potential formal specification of concurrent system behaviour than it turned out to be. The words 'specification' and 'system' should really be replaced by 'description' or 'account' and 'situation' or 'phenomenon'. Also 'variables' are 'observables'.
- The term 'definitive script' doesn't appear: the term 'dialogue state' is instead used to convey the kind of 'interface' the modeller that such a script provides. The intuition associated with all agents interacting through such a 'dialogue' remains sound, even though the word 'dialogue' itself is too limited to convey what interaction with definitive scripts can express.
- The realisation that LSD was concerned with *agents* rather than *processes* hadn't emerged clearly. This was the legacy of the early SDL influence, though in fact there doesn't seem to be too much problem about substituting 'agent' or in some contexts 'agent-role' for 'process' throughout the report.
- The original classification of observables for an agent was different from what it became soon after: what is now called a *handle* was originally called a 'state' observable, and what is now called a *state* observable was identified in an LSD account by putting a hash symbol (#) in front of the observable name.

The most informative sources for LSD are otherwise to be found in standard examples (catflap, telephone, cricket-related fragments) and in various final year projects (of which `footballTurner2000` is the most accessible).

Sources relating to the ADM

The first discussion of the ADM is in Slade's thesis (1990). A careful study of the subsequent development of the ADM concept exposes some discrepancies between Slade's conception (which is somewhat oriented towards a "machine" rather than "modelling framework" interpretation) and later applications of the ADM concept. Ward's thesis (2004) discusses the most significant issues (which concern the operational semantics) in detail; it includes a clear description of the 'Authentic' Abstract Definitive Machine that is conceptually best suited to its human-centred sense-making modelling role. Chapter 3 of Rungrattanaubol's thesis (2002) is probably the best place to find the key ideas behind the 'A'ADM explained and illustrated. There are also one or two interesting examples of using the ADM in its machine-like mode (see e.g. Parallel Computation in Definitive Models [010]).