Software Design, Modelling and Analysis in UML

Errata for the Course Slides

February 15, 2017

Contents

Lecture 10: Modelling Behaviour
Slide 22, ‘From UML to Core State Machines: By Example’ . . . . . . . . . . . . . 1

Lecture 13: Core State Machines III
Slide 5, ‘Discarding an Event’ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

Lecture 14: Hierarchical State Machines I
Slide 11, ‘Create Transformer Example’ . . . . . . . . . . . . . . . . . . . . . . . . 1

Lecture 15: Hierarchical State Machines II
Slide 19, ‘Scope’ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

Lecture 16: Hierarchical State Machines III
Slide 12, ‘Initial Pseudostate’ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

Lecture 18: Live Sequence Charts II (NEW)
Slide 43, ‘Words over Signature’ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

Tutorial 5: State Machines
Slide 5, ‘Exercise 4.(i)’ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

Lecture 10: Modelling Behaviour

Slide 22, ‘From UML to Core State Machines: By Example’

The initial state component of a core state machine is not a set, so the ‘maps to’ notes at the bottom of the slide should read:

\[ M(S,M) = (\{s_1, s_2\}, s_1, \{(s_1, ev, gd, act, s_2)\}) \]
Lecture 13: Core State Machines III

Slide 5, ‘Discarding an Event’

Checking whether any transition is enabled should also use \( \tilde{\sigma} \) as defined on Slide 6 (which sets up the corresponding, transient ‘param’ link). A transition may in particular not be enabled because the parameters of an event do not satisfy the transition’s guard.

Lecture 14: Hierarchical State Machines I

Slide 11, ‘Create Transformer Example’

The annotations of the transition arrows should read

\[
\frac{(\emptyset, \{(\ast, 2_C)\})}{u}
\]

e etc. (First component on top is cons (‘consumed set’), second component is \( Snd \) (‘sent set’, which includes creation and destruction), and below is the object which does this step.)

Lecture 15: Hierarchical State Machines II

Slide 19, ‘Scope’

The scope of transition \( t \) is the union of the transitive and reflexive children of the states in the least common region of \( source(t) \cup target(t) \) (not just the least common region).

Otherwise, some transitions in Exercise Sheet 6.A would unintentionally become consistent.

Lecture 16: Hierarchical State Machines III

Slide 12, ‘Initial Pseudostate’

The description of the principle is maybe a bit too much ‘natural-language’. A more precise description may be:

- when taking a transition (with multiple target states),
- consider the least common ancestor of the target states and
- for each region of the lca for which the transition does not have a basic state target,
- use the region’s initial state as indicated by a transition from an initial pseudo-state.
- Continue to choose initial states of all child states of the considered region.

Lecture 18: Live Sequence Charts II (NEW)

Slide 43, ‘Words over Signature’

The language of a UML model is a set of words over the signature named on the slide (not just one word).
Tutorial 5: State Machines

Slide 5, ‘Exercise 4.(i)’

Somebody proposed the step from Line 0 to Line 1 as an example for a run-to-completion (RTC) step. Sticking to our definition, this is wrong: an RTC-step needs to begin with a ‘dispatch’ (in this case an event is only discarded).