

Example: Using APE to Generate Complex Dialogues

10/17/2006

1. Trace

1.1

Put initial goal on agenda.

Agenda:

```
(did-tutor mechanism TPR)
```

1.2

Operator T7 is selected.

Agenda:

```
(did-elicited mechanism TPR)
<END>(did-tutor mechanism TPR)
```

KB:

```
(w-topic-is mechanism)
(w-level-is topic)
```

1.3

Operator T11 is chosen and performs the primitive action of asking “What controls TPR?”

Student gives a near miss response of “arteriolar radius” which causes (*did-handle-near-miss*) to be added to agenda.

The correct answer to the question in turn 1 is the nervous system. Although the student does not give that answer in turn 2, the student gives an answer that the tutor can build on in a collaborative fashion. We sometimes call an answer such as this, one that is not the desired answer but not wrong either, a *near miss*. Near misses are a subcategory of the category Woolf (1984) labels “grain of truth” answers.

Assume that a plan operator not shown, one of a series of input-handling operators, cause the goal *did-handle-near-miss* to be invoked whenever the input processor recognizes a near miss. (Note that “near miss” is as much a pragmatic distinction as a semantic one.)

Agenda:

```
(did-handle-near-miss)
<END>(did-elicited mechanism TPR)
<END>(did-tutor mechanism TPR)
```

1.4

Operator T1 is triggered. We note that *?w-variable-is = ?neural = TPR*.

There is a set of variables linked in a causal chain leading to the correct answer. We start at the

bottom of the chain. If the student gives an answer closer to the answer, then *?w-variable-is* will be set to that value, e.g. “arteriolar radius”.

?i-input-concept-is = “arteriolar radius”

Agenda:

```
(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicite mechanism TPR)
<END>(did-tutor mechanism TPR)
```

1.5

Operator T2 applies. We note that (*not (w-max-nearmiss-is ?max)*) is true since *?max* is a free variable, and (*e-on-leg "arteriolar radius" TPR none*) is true because arteriolar radius controls TPR.

Agenda:

```
(did-tutor-deep machanism "arteriolar radius" TPR t)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicite mechanism TPR)
<END>(did-tutor mechanism TPR)
```

KB:

```
(w-topic-is mechanism)
(w-level-is topic)
(w-max-nearmiss-is "arteriolar radius")
```

1.6

Operator T8 applies.

Agenda:

```
(did-utter "And ")
(did-elicite mechanism "arteriolar radius")
<END>(did-tutor-deep machanism "arteriolar radius" TPR t)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicite mechanism TPR)
<END>(did-tutor mechanism TPR)
```

KB:

```
(w-topic-is mechanism)
(w-max-nearmiss-is "arteriolar radius")
(w-level-is prompt)
```

1.7

Operator T12 outputs “And”.

1.8

Operator T11 outputs “What controls arteriolar radius?”.

Agenda:

```
<END>(did-utter "And ")
<END>(did-elicit mechanism "arteriolar radius")
<END>(did-tutor-deep mechanism "arteriolar radius" TPR t)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicit mechanism TPR)
<END>(did-tutor mechanism TPR)
```

1.9

Student gives synonym of desired answer.

Agenda:

```
(did-near-miss-synonym TPR "Radius of arterioles" "arteriolar radius")
<END>(did-utter "And ")
<END>(did-elicit mechanism "arteriolar radius")
<END>(did-tutor-deep mechanism "arteriolar radius" TPR t)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicit mechanism TPR)
<END>(did-tutor mechanism TPR)
```

1.10

Operator T3 applies.

Agenda:

```
<END>(did-near-miss-synonym TPR "Radius of arterioles" "arteriolar radius")
<END>(did-utter "And ")
<END>(did-elicit mechanism "arteriolar radius")
<END>(did-tutor-deep mechanism "arteriolar radius" TPR t)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicit mechanism TPR)
<END>(did-tutor mechanism TPR)
```

1.11

Tutor utters “No, radius of arterioles and arteriolar radius are the same” via operator T12.

1.12

Execute *prune-replace*, pruning until (w-level-is prompt) is false and pushing (did-tutor-deep mechanism "arteriolar radius" TPR post-error) on agenda

Agenda:

```
(did-tutor-deep mechanism
  "arteriolar radius" TPR post-error)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicit mechanism TPR)
<END>(did-tutor mechanism TPR)
```

KB:

```
(w-topic-is mechanism)
(w-max-nearmiss-is "arteriolar radius")
(w-level-is topic)
```

1.13

Operator T8 applies.

Agenda:

```
(did-utter "And ")
(did-elicit mechanism "arteriolar radius")
<END>(did-tutor-deep mechanism "arteriolar radius" TPR post-error)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicit mechanism TPR)
<END>(did-tutor mechanism TPR)
```

KB:

```
whatever was there before, plus:
(w-level-is prompt)
```

1.14

Operator T12 prints "And " on the screen.

1.15

Operator T11 prints "What controls arteriolar radius?".

1.16

Student answers.

1.17

Operator T4 applies (very similar to operator T3), pushes

```
(did-tutor-deep mechanism "arteriolar radius" TPR post-error)
```

on agenda to generate the rest of turn 7 as we did before.

Agenda:

```
(did-tutor-deep mechanism "arteriolar radius" TPR post-error)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicit mechanism TPR)
<END>(did-tutor mechanism TPR)
```

1.18

We get a near miss, so add (did-handle-near-miss) to agenda. Operator T1 applies, resulting in:

```
w-variable = ?neural = "arteriolar radius"
?partial = "arteriolar resistance"
```

Agenda:

```
(did-near-miss-specific "arteriolar radius" "arteriolar resistance")
<END>(did-handle-near-miss)
<END>(did-tutor-deep mechanism "arteriolar radius" TPR post-error)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicited mechanism TPR)
<END>(did-tutor mechanism TPR)
```

but operator T2 doesn't apply because (e-on-leg "arteriolar resistance" "arteriolar radius" none) is false. Thus an unseen but similar operator fires and states the radius controls resistance, and the question is re-asked as before probably with *prune-replace* resulting in:

Agenda:

```
(did-tutor-deep mechanism
  "arteriolar radius" TPR post-error)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicited mechanism TPR)
<END>(did-tutor mechanism TPR)
```

1.19

Linguistic near miss comes up

Agenda:

```
(did-handle-ling-near-miss)
<END>(did-tutor-deep mechanism
  "arteriolar radius" TPR post-error)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicited mechanism TPR)
<END>(did-tutor mechanism TPR)
```

The recipe is

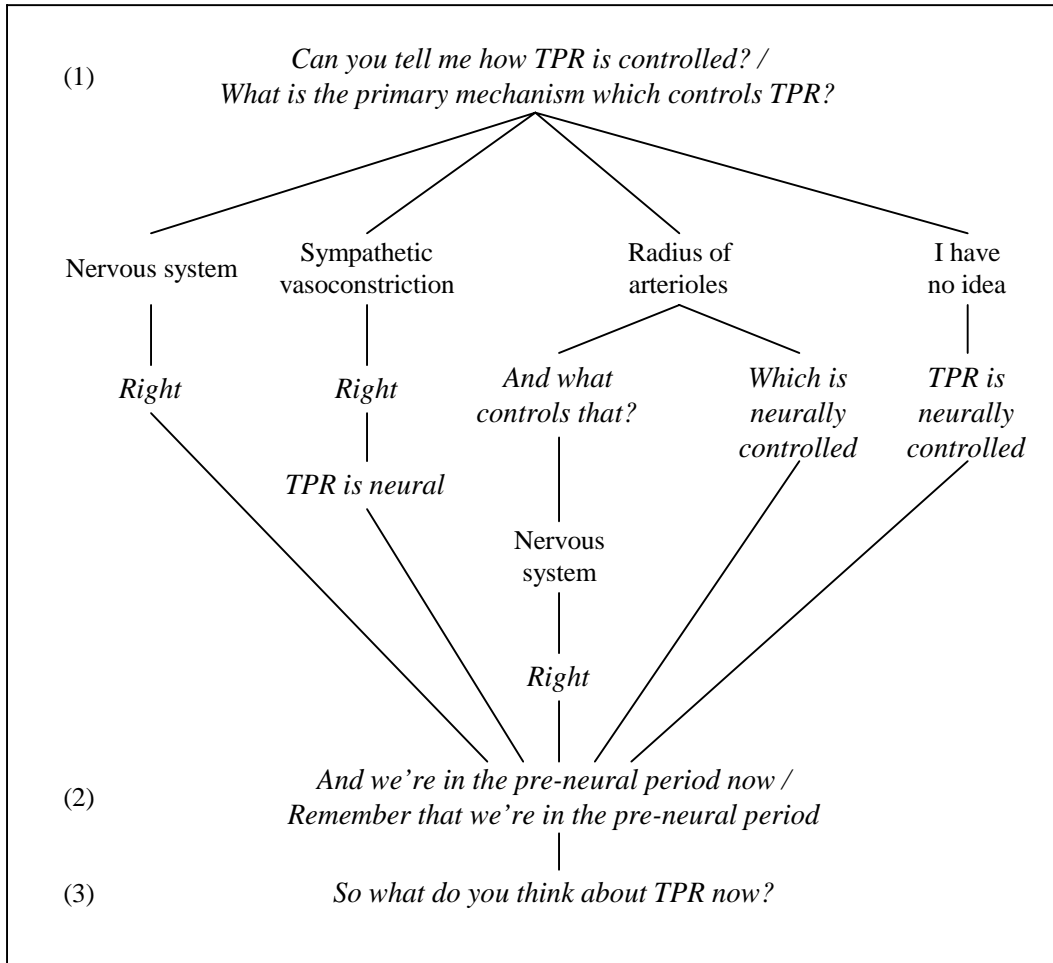
```
(retry-at (and (w-topic-is mechanism)
               (w-level-is topic)))
```

Retry-at is defined as: pop the agenda through the first <END> marker where the retry argument is false, the restore that entry's original goal.

Agenda:

```
<END>(did-tutor-deep mechanism
  "arteriolar radius" TPR post-error)
<END>(did-near-miss-specific TPR "arteriolar radius")
<END>(did-handle-near-miss)
<END>(did-elicited mechanism TPR)
<END>(did-tutor mechanism TPR)
```

2. Diagrams



- 1 T: What controls TPR?
- 2 S: Arteriolar radius. (*near miss*)
- 3 T: And what controls arteriolar radius?
- 4 S: Radius of arterioles. (*near miss*)
- 5 T: No, radius of arterioles and arteriolar radius are the same thing.
So what controls arteriolar radius?
- 6 S: Arteriolar diameter. (*near miss*)
- 7 T: Arteriolar diameter and arteriolar radius are the same thing in different units.
Try again. What controls arteriolar radius?
- 8 S: Arteriolar resistance. (*near miss*)
- 9 T: No, actually arteriolar radius controls arteriolar resistance.
So what controls arteriolar radius?
- 10 S: Sympathetic stimulation.
- 11 T: In other words, the nervous system.

Figure 1: Sample dialogue

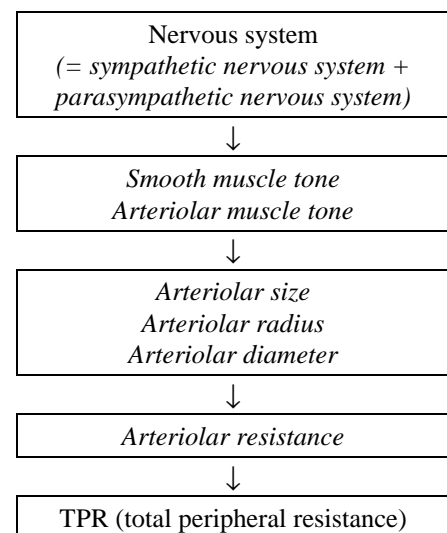


Figure 3: Domain knowledge

Operator T1:

```
(def-operator neural-partial-mechanism
  :goal (did-handle-near-miss)
  :precond ((w-variable-is ?neural)
            (is-neural ?neural)
            (w-topic-is mechanism)
            (w-level-is topic)
            (i-input-concept-is ?partial))
  :recipe ((goal (did-near-miss-specific ?neural ?partial)))
  :hiercx ())
```

Operator T2:

```
(def-operator neural-partial-mechanism-first
  :goal (did-near-miss-specific ?neural ?partial)
  :precond ((not (w-max-nearmiss-is ?max))
            (e-on-leg ?partial ?neural none))
  :recipe ((assert (w-max-nearmiss-is ?partial))
            (goal (did-tutor-deep mechanism ?partial ?neural t)))
  :hiercx ())
```

Operator T3:

```
(def-operator neural-partial-mechanism-later-synonym-execution
  :goal (did-near-miss-synonym ?neural ?partial ?max)
  :precond ()
  :recipe ((goal (did-utter ("No, " ?partial "and" ?max "are the same.")))
            (prune-replace ((w-level-is prompt)
                            (goal (did-tutor-deep mechanism ?max ?neural post-error))))))
  :hiercx ())
```

Operator T6:

```
(def-operator mech-ling-near-miss
  :goal (did-handle-ling-near-miss)
  :precond ((w-topic-is mechanism)
            (w-variable-is ?vbl)
            (is-neural ?vbl))
  :recipe ((retry-at (and (w-topic-is mechanism)
                          (w-level-is topic))))
  :hiercx ())
```

Operator T7:

```
(def-operator tutor-by-elicit
  :goal (did-tutor ?topic ?vbl)
  :precond ((not (i-input-catg-is ling-near-miss))
            ...)
  :recipe ((goal (did-elicit ?topic ?vbl)))
  :hiercx ((assert (w-topic-is ?topic))
            (assert (w-level-is topic))))
```

Figure 2: Sample discourse operators

Operator T8:

```
(def-operator tutor-deep-mechanism-by-elicite-cont
:goal (did-tutor-deep mechanism ?from-vbl ?to-vbl t)
:precond ()
:recipe ((goal (did-utter "And "))
(goal (did-elicite mechanism ?from-vbl)))
:hiercx ((assert (w-level-is prompt))))
```

Operator T9:

```
(def-operator tutor-by-inform ;; not used in example
:goal (did-tutor ?topic ?vbl)
:precond (...)
:recipe ((goal (did-inform ?topic ?vbl)))
:hiercx ((assert (w-topic-is ?topic))
(assert (w-level-is topic))))
```

Operator T10:

```
(def-operator tutor-by-rewrite
:goal (did-tutor mechanism ?vbl)
:precond ((is-neural ?vbl)
(i-input-catg-is ling-near-miss))
:recipe ((goal (did-utter "In other words, the nervous system.")))
:hiercx ((assert (w-topic-is mechanism))
(assert (w-level-is topic))))
```

Operator T11:

```
(def-operator elicite-mechanism-default
:goal (did-elicite ?topic ?vbl)
:precond ((w-topic-is mechanism))
:recipe ((prim-interactive (ask ("What controls" ?vbl "?"))))
:hiercx ())
```

Operator T12:

```
(def-operator utter
:goal (did-utter ?utterance)
:precond ()
:recipe ((primitive (say ?utterance)))
:hiercx ())
```

Figure 2 (continued): Sample discourse operators