

## Unit1 Model Code Description

This document (and the corresponding documents for future units) will describe any ACT-R commands used in the models which are not described in the main unit text, the Lisp code that controls the experiments for the model, and how ACT-R is interfaced to those experiments.

For this unit there are no experiments and all of the models run without any interaction with a real or virtual world. They start with a predefined goal and a given set of declarative memories, and the result is a modification of the goal chunk representing the cognition that took place. Later units will have much more involved perception and action, but for this unit all of the models have the same basic structure as shown here in a solution to the multi-column addition model one was to write for unit1:

```
(clear-all)

(define-model tutor-model

(sgp :esc t :lf .05 :trace-detail medium)

;; Add Chunk-types here

(chunk-type addition-fact addend1 addend2 sum)
(chunk-type add-pair one1 ten1 one2 ten2 ten-ans one-ans carry)

;; Add Chunks here

(add-dm
 (fact17 isa addition-fact addend1 1 addend2 7 sum 8)
 (fact34 isa addition-fact addend1 3 addend2 4 sum 7)
 (fact67 isa addition-fact addend1 6 addend2 7 sum 13)
 (fact103 isa addition-fact addend1 10 addend2 3 sum 13)
 (goal isa add-pair ten1 3 one1 6 ten2 4 one2 7))

;; Add productions here

(p start-pair
 =goal>
   ISA add-pair
   one1 =num1
   one2 =num2
   one-ans nil
 ==>
 =goal>
   one-ans busy
 +retrieval>
   ISA addition-fact
   addend1 =num1
   addend2 =num2)
```

```
(p add-ones
=goal>
  ISA add-pair
  one-ans busy
  one1 =num1
  one2 =num2
=retrieval>
  ISA addition-fact
  addend1 =num1
  addend2 =num2
  sum =sum
```

```
==>
=goal>
  one-ans =sum
  carry busy
+retrieval>
  ISA addition-fact
  addend1 10
  sum =sum)
```

```
(p process-carry
=goal>
  ISA add-pair
  ten1 =num1
  ten2 =num2
  carry busy
  one-ans =ones
=retrieval>
  ISA addition-fact
  addend1 10
  sum =ones
  addend2 =remainder
```

```
==>
=goal>
  carry 1
  ten-ans busy
  one-ans =remainder
+retrieval>
  ISA addition-fact
  addend1 =num1
  addend2 =num2)
```

```
(p no-carry
=goal>
  ISA add-pair
  ten1 =num1
  ten2 =num2
  one-ans =ones
  carry busy
?retrieval>
  buffer failure
```

```
==>
```

```

=goal>
  carry nil
  ten-ans busy
+retrieval>
  ISA addition-fact
  addend1 =num1
  addend2 =num2)

(p add-tens-done
=goal>
  ISA add-pair
  ten-ans busy
  carry nil
=retrieval>
  ISA addition-fact
  sum =sum
==>
=goal>
  ten-ans =sum)

(p add-tens-carry
=goal>
  ISA add-pair
  carry 1
  ten-ans busy
=retrieval>
  ISA addition-fact
  sum =sum
==>
=goal>
  carry nil
+retrieval>
  ISA addition-fact
  addend1 1
  addend2 =sum)

(goal-focus goal)
)

```

It starts with a call to a function called **clear-all** followed by a call to the command **define-model**. Inside the call to **define-model** there is a call to a command called **sgp**, then all of the components of the model are defined (chunk-types, chunks, and productions) and finally the starting goal chunk gets set using the **goal-focus** command.

Most of these commands were described in the main unit text. Here we will provide some additional details for some of them and describe the **sgp** command.

## Clear-all

The **clear-all** command is used to set the ACT-R software to its initial state. It removes all models that may be defined and restores all settings to their default values. If a model (or group of models) is contained within a single file, then one probably should call **clear-all** in that file to make sure ACT-R is initialized before defining the model components.

**Clear-all** also records the name of the file that contains it when it is loaded. After that point, the ACT-R command **reload** can be used to load that file again, which is often faster than using the load command or the menu options to do so.

## Define-model

The **define-model** command takes one required parameter which is the name of a new model to create and an arbitrary number of other commands. The name should be a symbol and must be unique with respect to other models that are currently defined. When **reset** is called, all of the commands specified inside of the **define-model** are reevaluated for that model. An important feature of ACT-R is that more than one model can be defined simultaneously. Each call to **define-model** creates a new model which is independent of the other models, and all of those models will run in parallel when the **run** command is called.

## SGP

The **sgp** command is used to set or show the parameters for a model (it stands for set/show general parameters). If you call **sgp** at the Lisp prompt with no parameters it will print out all of the parameters for all of the modules in the current model each with their current value, the default value, and a short description. The parameters for the modules control many different things. Some are used in equations that control the performance of the model and others are there to help the modeler with debugging or extending a model by changing some of the general ACT-R mechanisms. The details of all the parameters for the modules can be found in the reference manual.

When using **sgp** to set parameters the syntax is to specify a parameter and then the new value you want to assign to that parameter. Any number of parameters and values may be specified in a single call to **sgp**. All of the parameters in ACT-R begin with a “:” (in Lisp syntax they are keywords). All of the unit1 models have a call to **sgp** similar to this:

```
(sgp :esc t :lf .05 :trace-detail medium)
```

That is setting three parameters: **:esc**, **:lf**, and **:trace-detail**. The first two together are specifying that retrieval requests will always take 50ms, but other than that are beyond the scope of this unit and will be discussed fully in later units. The third parameter being set, **:trace-detail**, controls how much information is shown in the trace when a model runs. The default value is **medium**, and that is also how it is being set in this model. The other values that it can have are **high** and **low**. When it is set to **high**, effectively every action the model does shows in the trace. Here is the trace of the two-digit addition model with **:trace-detail** set to **high**:

```
0.000    GOAL                SET-BUFFER-CHUNK GOAL GOAL REQUESTED NIL
0.000    PROCEDURAL          CONFLICT-RESOLUTION
```

0.000	PROCEDURAL	PRODUCTION-SELECTED START-PAIR
0.000	PROCEDURAL	BUFFER-READ-ACTION GOAL
0.050	PROCEDURAL	PRODUCTION-FIRED START-PAIR
0.050	PROCEDURAL	MOD-BUFFER-CHUNK GOAL
0.050	PROCEDURAL	MODULE-REQUEST RETRIEVAL
0.050	PROCEDURAL	CLEAR-BUFFER RETRIEVAL
0.050	DECLARATIVE	START-RETRIEVAL
0.050	PROCEDURAL	CONFLICT-RESOLUTION
0.100	DECLARATIVE	RETRIEVED-CHUNK FACT67
0.100	DECLARATIVE	SET-BUFFER-CHUNK RETRIEVAL FACT67
0.100	PROCEDURAL	CONFLICT-RESOLUTION
0.100	PROCEDURAL	PRODUCTION-SELECTED ADD-ONES
0.100	PROCEDURAL	BUFFER-READ-ACTION GOAL
0.100	PROCEDURAL	BUFFER-READ-ACTION RETRIEVAL
0.150	PROCEDURAL	PRODUCTION-FIRED ADD-ONES
0.150	PROCEDURAL	MOD-BUFFER-CHUNK GOAL
0.150	PROCEDURAL	MODULE-REQUEST RETRIEVAL
0.150	PROCEDURAL	CLEAR-BUFFER RETRIEVAL
0.150	DECLARATIVE	START-RETRIEVAL
0.150	PROCEDURAL	CONFLICT-RESOLUTION
0.200	DECLARATIVE	RETRIEVED-CHUNK FACT103
0.200	DECLARATIVE	SET-BUFFER-CHUNK RETRIEVAL FACT103
0.200	PROCEDURAL	CONFLICT-RESOLUTION
0.200	PROCEDURAL	PRODUCTION-SELECTED PROCESS-CARRY
0.200	PROCEDURAL	BUFFER-READ-ACTION GOAL
0.200	PROCEDURAL	BUFFER-READ-ACTION RETRIEVAL
0.250	PROCEDURAL	PRODUCTION-FIRED PROCESS-CARRY
0.250	PROCEDURAL	MOD-BUFFER-CHUNK GOAL
0.250	PROCEDURAL	MODULE-REQUEST RETRIEVAL
0.250	PROCEDURAL	CLEAR-BUFFER RETRIEVAL
0.250	DECLARATIVE	START-RETRIEVAL
0.250	PROCEDURAL	CONFLICT-RESOLUTION
0.300	DECLARATIVE	RETRIEVED-CHUNK FACT34
0.300	DECLARATIVE	SET-BUFFER-CHUNK RETRIEVAL FACT34
0.300	PROCEDURAL	CONFLICT-RESOLUTION
0.300	PROCEDURAL	PRODUCTION-SELECTED ADD-TENS-CARRY
0.300	PROCEDURAL	BUFFER-READ-ACTION GOAL
0.300	PROCEDURAL	BUFFER-READ-ACTION RETRIEVAL
0.350	PROCEDURAL	PRODUCTION-FIRED ADD-TENS-CARRY
0.350	PROCEDURAL	MOD-BUFFER-CHUNK GOAL
0.350	PROCEDURAL	MODULE-REQUEST RETRIEVAL
0.350	PROCEDURAL	CLEAR-BUFFER RETRIEVAL
0.350	DECLARATIVE	START-RETRIEVAL
0.350	PROCEDURAL	CONFLICT-RESOLUTION
0.400	DECLARATIVE	RETRIEVED-CHUNK FACT17
0.400	DECLARATIVE	SET-BUFFER-CHUNK RETRIEVAL FACT17
0.400	PROCEDURAL	CONFLICT-RESOLUTION
0.400	PROCEDURAL	PRODUCTION-SELECTED ADD-TENS-DONE
0.400	PROCEDURAL	BUFFER-READ-ACTION GOAL
0.400	PROCEDURAL	BUFFER-READ-ACTION RETRIEVAL
0.450	PROCEDURAL	PRODUCTION-FIRED ADD-TENS-DONE
0.450	PROCEDURAL	MOD-BUFFER-CHUNK GOAL

```

0.450 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.450 PROCEDURAL CONFLICT-RESOLUTION
0.450 ----- Stopped because no events left to process

```

That can be very useful when debugging a model but it can be a bit too much at other times. Here is the same model running with a **medium** level of **:trace-detail** (which is the default value):

```

0.000 GOAL SET-BUFFER-CHUNK GOAL GOAL REQUESTED NIL
0.000 PROCEDURAL CONFLICT-RESOLUTION
0.050 PROCEDURAL PRODUCTION-FIRED START-PAIR
0.050 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.050 DECLARATIVE START-RETRIEVAL
0.050 PROCEDURAL CONFLICT-RESOLUTION
0.100 DECLARATIVE RETRIEVED-CHUNK FACT67
0.100 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT67
0.100 PROCEDURAL CONFLICT-RESOLUTION
0.150 PROCEDURAL PRODUCTION-FIRED ADD-ONES
0.150 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.150 DECLARATIVE START-RETRIEVAL
0.150 PROCEDURAL CONFLICT-RESOLUTION
0.200 DECLARATIVE RETRIEVED-CHUNK FACT103
0.200 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT103
0.200 PROCEDURAL CONFLICT-RESOLUTION
0.250 PROCEDURAL PRODUCTION-FIRED PROCESS-CARRY
0.250 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.250 DECLARATIVE START-RETRIEVAL
0.250 PROCEDURAL CONFLICT-RESOLUTION
0.300 DECLARATIVE RETRIEVED-CHUNK FACT34
0.300 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT34
0.300 PROCEDURAL CONFLICT-RESOLUTION
0.350 PROCEDURAL PRODUCTION-FIRED ADD-TENS-CARRY
0.350 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.350 DECLARATIVE START-RETRIEVAL
0.350 PROCEDURAL CONFLICT-RESOLUTION
0.400 DECLARATIVE RETRIEVED-CHUNK FACT17
0.400 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT17
0.400 PROCEDURAL CONFLICT-RESOLUTION
0.450 PROCEDURAL PRODUCTION-FIRED ADD-TENS-DONE
0.450 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.450 PROCEDURAL CONFLICT-RESOLUTION
0.450 ----- Stopped because no events left to process

```

In that trace we no longer see the individual conditions and actions of the productions. Now, here is the same model run with a **low** setting for **:trace-detail**:

```

0.000 GOAL SET-BUFFER-CHUNK GOAL GOAL REQUESTED NIL
0.050 PROCEDURAL PRODUCTION-FIRED START-PAIR

```

```
0.100  DECLARATIVE      SET-BUFFER-CHUNK RETRIEVAL FACT67
0.150  PROCEDURAL       PRODUCTION-FIRED  ADD-ONES
0.200  DECLARATIVE      SET-BUFFER-CHUNK RETRIEVAL FACT103
0.250  PROCEDURAL       PRODUCTION-FIRED  PROCESS-CARRY
0.300  DECLARATIVE      SET-BUFFER-CHUNK RETRIEVAL FACT34
0.350  PROCEDURAL       PRODUCTION-FIRED  ADD-TENS-CARRY
0.400  DECLARATIVE      SET-BUFFER-CHUNK RETRIEVAL FACT17
0.450  PROCEDURAL       PRODUCTION-FIRED  ADD-TENS-DONE
0.450  -----         Stopped because no events left to process
```

At **low** we only see production firings and buffer settings. The setting of **:trace-detail** does not change how the model runs. It only affects how the trace is displayed to the modeler and what conditions are available to “step” to using the Stepper tool in the ACT-R Environment.