Sun, R.: Integrating Rules and Connectionism for Robust Commonsense Reasoning. John Wiley and Sons, New York (1994)

Sun, R.: Cognitive science meets multi-agent systems: A prolegomenon. Philosophical Psychology 14(1), 5–28 (2001)

Sun, R.: Duality of the Mind. Lawrence Erlbaum, Mahwah (2002)

Sun, R.: A Tutorial on CLARION. Technical report, Cognitive Science Department, Rensselaer Polytechnic Institute (2003), `http://www.cogsci.rpi.edu/~rsun/sun.tutorial.pdf`

Sun, R.: Desiderata for cognitive architectures. Philosophical Psychology 17(3), 341–373 (2004)

Sun, R.: Prolegomena to integrating cognitive modeling and social simulation. In: Sun, R. (ed.) Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation, Cambridge University Press, Cambridge (2006)

Sun, R.: Introduction to computational cognitive modeling. In: Sun, R. (ed.) The Cambridge Handbook of Computational Cognitive Modeling, Cambridge University Press, Cambridge (2007)

Sun, R., Coward, L.A., Zenzen, M.J.: On levels of cognitive modeling. Philosophical Psychology 18(5), 613–637 (2005)

Sun, R., Naveh, I.: Simulating organizational decision-making using a cognitively realistic agent model. Journal of Artificial Societies and Social Simulation 7(3) (2004), `http://jasss.soc.surrey.ac.uk/7/3/5.html`

Sun, R., Peterson, T.: Autonomous learning of sequential tasks: experiments and analyses. IEEE Transactions on Neural Networks 9(6), 1217–1234 (1998)

Sun, R., Peterson, T.: Multi-agent reinforcement learning: Weighting and partitioning. Neural Networks 12(4-5), 127–153 (1999)

Watkins, C.: Learning with Delayed Rewards. Ph.D Thesis, Cambridge University, Cambridge, UK (1989)

ACT-R Meets fMRI

Yulin Qin[1,2], Daniel Bothell[2], and John R. Anderson[2]

[1] Department of Psychology and Behavior Science, Zhejiang University
Hangzhou, Zhejiang, China
[2] Department of Psychology, Carnegie Mellon University
Pittsburgh, PA, 15213, USA

Abstract. ACT-R (Adaptive Control of Thought - Rational) is a theory and computational model of human cognitive architecture. It consists of a set of modules with their own buffers, each devoted to processing a different kind of information. A production rule in the core production system can be fired based on the chunks in these buffers and then it changes the chunks in the buffer of the related modules or the state of the related modules, which may leads to fire a new production rule and so on to generate the cognitive behavior. It has successfully predicted and explained a broad range of cognitive psychological phenomena and found applications in the human-computer interface and other areas (see http://act-r.psy.cmu.edu) and may have potential applications in Web intelligence.

In recent years, a series of fMRI experiments have been performed to explore the neural basis of cognitive architecture and to build a two-way bridge between the information processing model and fMRI. The patterns of the activations of brain areas corresponding to the buffers of the major modules in ACT-R were highly consistent across these experiments; and ACT-R has successfully predicted the Blood Oxygenation Level-Depend (BOLD) effect in these regions. The approach of ACT-R meets fMRI may shed light on the research of Web Intelligence (WI) meets Brain Informatics (BI).

1 ACT-R

ACT-R (Adaptive Control of Thought - Rational) is a theory and computational model of human cognitive architecture. As a theory, it proposes the systematical hypothesis on the basic structure of human cognitive system and functions of these structures in information processing to generate the human cognitive behavior; as a computational model, it offers a computer software system for the development of computational models to quantitatively simulate and predict the human behavior for a wide range of cognitive tasks.

1.1 Two Kinds of Knowledge

There are two kinds of knowledge represented in ACT-R – declarative knowledge and procedural knowledge. Declarative knowledge corresponds to things we are aware we know and can usually describe to others. Examples of declarative

N. Zhong et al. (Eds.): WImBI 2006, LNAI 4845, pp. 205–222, 2007.

knowledge include "George Washington was the first president of the United States" and "An atom is like the solar system". Procedural knowledge is the knowledge which we display in our behavior but which we are not conscious of. For instance, no one can describe the rules by which we speak a language and yet we do. In ACT-R declarative knowledge is represented in structures called chunks whereas procedural knowledge is represented as rules called productions. Thus chunks and productions are the basic building blocks of an ACT-R model.

Chunks in ACT-R. In ACT-R a chunk is defined by its type and its slots. One can think of chunk-types as categories (e.g., birds) and slots as category attributes (e.g., color or size). A chunk also has a name which can be used to reference it, but the name is not considered to be a part of the chunk itself. Below is a representation of a chunk that encodes the fact that $4+3=7$. The name of the chunk is Fact3+4. The isa slot is special and specifies the type of the chunk which is addition-fact in this example, and the other slots are addend1, addend2, and sum.

```
Fact3 + 4
   isa addition-fact
   addend1 three
   addend2 four
   sum seven
```

Productions in ACT-R. A production rule is a statement of a particular contingency that controls behavior. An example might be

IF the goal is to add two digits d1 and d2 in a column
  and $d1 + d2 = d3$
THEN set as a subgoal to write d3 in the column

The condition of a production rule (the IF part) consists of a specification of the chunks in various buffers. The action of a production rule (the THEN part) consists of modifications of the chunks in the buffers, requests for other chunks to be placed into the buffers, and/or requests for other actions to be taken.

Representations of Chunks and Productions in ACT-R. In previous sessions there are informal English specifications of chunks and production rules. This session shows examples of them represented in ACT-R. The examples are from an ACT-R model for counting numbers.

All ACT-R commands are Lisp functions and therefore are specified in parentheses. The first term after the left parenthesis is the command name. That is followed by the details of the item and then a right parenthesis. In this section we will show how to use the commands to create the knowledge representations in ACT-R.

An ACT-R model begins from creating the chunk type. To create a new type of chunk, one needs to specify a frame for the chunk using the chunk-type command. This requires specifying the name of the chunk type and the names of the slots that it will have. The general chunk type specification looks like this:

```
(chunk-type name slot-name-1 slot-name-2 ... slot-name-n)
```

To model count, the chunk types can be created as:

```
(chunk-type count-order first second)
(chunk-type count-from start end count state)
```

The command to create a set of chunks and automatically add them to declarative memory is add-dm. It takes any number of chunk specifications as its arguments. If a slot is without initial value, the value of this slot is nil (as in Lisp), and this slot can be missing in the chunk, as the slots of count and state in the goal chunk here.

```
(add-dm
 (b ISA count-order first 1 second 2)
 (c ISA count-order first 2 second 3)
 (d ISA count-order first 3 second 4)
 (e ISA count-order first 4 second 5)
 (f ISA count-order first 5 second 6)
 (first-goal ISA count-from start 2 end 5))
```

The general form of a production rule is

```
(P Production-Name
   condition part
==>
   action part
)
```

The production that counts from one number to the next is as follows:

```
(P counting-example              English Description
 =goal>                          If the goal is
    ISA     count                to count
    state   incrementing         the state slot is incrementing
    count   =num1                the number in count slot is called =num1
 =retrieval>                     and the chunk in the retrieval buffer
    ISA     count-order          is of type count-order
    first   =num1                the first slot has the value =num1
    second  =num2                and the second slot has a value
                                 we will call =num2
==>                              Then
```

```
   =goal>                        change the goal
      count   =num2              to continue counting from =num2
   +retrieval>                   and request a retrieval
      ISA     count-order        of a count-order chunk to
      first   =num2              find the number that follows =num2
)
```

The condition of the preceding production specifies a pattern to match to the goal buffer (indicated by =goal>) and a pattern to match to the retrieval buffer (indicated by =retrieval>). The slot values start with "=", such as =num1 and =num2, are variables. +retrieval> means requesting a retrieval in declarative memory.

1.2 Modules in ACT-R

ACT-R consists of a set of modules with their own buffers, each devoted to processing a different kind of information. Besides the declarative memory module mentioned above, there is a visual module for identifying objects in the visual field, a manual module for controlling the hands, a goal module for keeping track of current goals and intentions and other modules. The information in these modules is largely encapsulated and the modules communicate only by firing the productions based on the chunks in their buffers. Within a module and among different modules, the information processing can go on in parallel and asynchronously. However, there are two sequential processing restrictions in ACT-R: Only one chunk can be in the buffer of any module at any time and there is only one production rule can be fire at each processing cycle.

The Vision Module. Many tasks involve interacting with visible stimuli and the vision module provides a model with a means for acquiring visual information. It is designed as a system for modeling visual attention. It assumes that there are lower level perceptual processes that generate the representations with which it operates, but does not model those perceptual processes in detail. It includes some default mechanisms for parsing text and other simple visual features from a window and provides an interface that one can use to extend it when necessary.

The vision module has two buffers. There is a visual buffer that can hold a chunk that represents an object in the visual scene and a visual-location buffer that holds a chunk which represents the location of an object in the visual scene. As with all modules, it also responds to queries of the buffers about the state of the module. One should always check to make sure that the visual buffer is free before making any requests to avoid jamming it.

```
?visual>
    state free
```

is true when the buffer is available.

The Motor Module. When we speak of motor actions in ACT-R we are only concerned with hand movements. The hand movement can be the finger presses at a keyboard or other devices. The hands can also be used to move a mouse or other device.

The buffer for interacting with the motor module is called the manual buffer. Unlike other buffers, however, the manual buffer never holds a chunk. It is used only to issue commands and to query the state of the motor module. The manual buffer is used to request actions be performed by the hands. As with the vision module, one should always check to make sure that the motor module is free before making any requests to avoid jamming it.

1.3 A Simple Example of ACT-R Model

The example shown in this session includes the processes of reading the screen to get the task of finding the answer of $3 + 4$, retrieving the arithmetic fact, $3 + 4 = 7$, to solve the problem, and then moving the hand to press the key of 7 in the keyboard to give the output of the system. This example is, of cause, very simple, but involves the major steps of the interaction between human and the Web. We will give the ACT-R model with explanations in this session, but put other detailed issues in the appendix.

```
;; Define Chunk-types (Comments in Lisp begin with ;)

(chunk-type addition-fact addend1 addend2 sum)
(chunk-type arithmetic-task argu1 operator argu2 state)

;; Create Chunks in declarative memory

(add-dm
   (fact34 ISA addition-fact addend1 "3" addend2 "4" sum "7")
   (first-goal ISA arithmetic-task state start)
)
```

Note again that if a slot is without initial value, this slot can be missing in the chunk definition (as of slots argu1, argu2, and operator in the first-goal chunk here).

```
;; Productions for receiving visual stimulus
```

The first three productions form a typical process of receiving visual stimulus in ACT-R models. The first production is to find the location of an object in the visual scene, the second one is to switch attention to the object, and the third is to harvest the object.

```
(P find-unattended-txt
  =goal>
    ISA          arithmetic-task
    state        start
```

```
==>
  +visual-location>
    ISA           visual-location
    screen-x      lowest
    :attended     nil
  =goal>
    state          find-location
)
```

This production makes a request of the visual-location buffer and changes the goal state to find-location. The visual-location request asks the vision module to find the location of an object in its visual scene (which is the experiment window for this model) that meets the specified requirements (in this production it is the leftmost (with lowest x coordinate) unattended (:attended nil) object), to build a chunk to represent the location of that object if one exists, and place that chunk in the visual-location buffer.

```
(P attend-txt
  =goal>
    ISA           arithmetic-task
    state         find-location
  =visual-location>
    ISA           visual-location
  ?visual>
    state         free
==>
  +visual>
    ISA           move-attention
    screen-pos    =visual-location
  =goal>
    state         attend
)
```

If the goal state is find-location, there is a visual-location in the visual-location buffer, and the vision module is not currently active (state free), then a query is made of the visual buffer which isa move-attention and it specifies the screen-pos[ition] as the visual location in the visual-location buffer. A request of the visual buffer for a move-attention is a request for the vision module to move its attention to the specified location, encode the object that is there, and place that object into the visual buffer as a chunk.

```
(P encode-first-txt
  =goal>
    ISA           arithmetic-task
    state         attend
```



```
    arg1          nil
  =visual>
    ISA           text
    value         =letter
==>
  =goal>
    arg1          =letter
    state         start
)
```

After a visual object has been placed in the visual buffer, it can be harvested by a production like this one, which stores the object that was read into the arg1 slot of the goal and sets the state slot to start to fire the first production to find the location of the next object.

```
(P encode-second-txt
  =goal>
    ISA           arithmetic-task
    state         attend
    arg1          =t1
    operator      nil
  =visual>
    ISA           text
    value         =letter
==>
  =goal>
    operator      =letter
    state         start
)

(P encode-third-txt
  =goal>
    ISA           arithmetic-task
    state         attend
    arg1          =t1
    operator      "+"
    arg2          nil
  =visual>
    ISA           text
    value         =letter
==>
  =goal>
    arg2          =letter
    state         respond
  +retrieval>
```

```
    ISA          addition-fact
    addend1      =t1
    addend2      =letter
)
```

After reading all characters in the screen, the task is to do the addition of 3 + 4, which can be done by retrieving the arithmetic fact chunk from the declarative memory.

```
(P respond-answer
  =goal>
    ISA          arithmetic-task
    state        respond
    argu1        =num1
    operator     "+"
    argu2        =num2
  =retrieval>
    ISA          addition-fact
    addend1      =num1
    addend2      =num2
    sum          =sum
  ?manual>
    state        free
  ==>
  +manual>
    ISA          press-key
    key          =sum
  -goal>
)
```

After the arithmetic fact chunk 3 + 4 = 7 has been placed into the buffer of declarative memory, and the motor module is available (the state of motor buffer is free), a request is made to press the key corresponding to the value of the sum slot of the chunk in the retrieval buffer, and the goal is deleted to finish the task. The type of action requested of the hands is specified in the isa slot of the manual buffer request. The press-key request assumes that the hands are located over the home row and the fingers will be returned there after the key has been pressed.

```
(setf *actr-enabled-p* t)
```

The global variable *actr-enabled-p* is set to t to run the ACT-R model. If it is set to nil, the human subject, instead of the model, will perform this task.

```
(goal-focus first-goal)
```


The chunk first-goal is declared to be the current goal (placed into the goal buffer).

Typing (do-experiment) to run this model. The output, called the trace of the model, is as follows. Each line of the trace represents one event in the model and shows the time in seconds, the module that generated the event and the details of that event.

```
> (do-experiment)
0.000 GOAL          SET-BUFFER-CHUNK GOAL FIRST-GOAL REQUESTED NIL
0.000 VISION        SET-BUFFER-CHUNK VISUAL-LOCATION LOC0 REQUESTED NIL
0.000 PROCEDURAL      CONFLICT-RESOLUTION
0.050 PROCEDURAL      PRODUCTION-FIRED FIND-UNATTENDED-TXT
0.050 PROCEDURAL      CLEAR-BUFFER VISUAL-LOCATION
0.050 VISION        Find-location
0.050 VISION        SET-BUFFER-CHUNK VISUAL-LOCATION LOC1
0.050 PROCEDURAL      CONFLICT-RESOLUTION
0.100 PROCEDURAL      PRODUCTION-FIRED ATTEND-TXT
0.100 PROCEDURAL      CLEAR-BUFFER VISUAL-LOCATION
0.100 PROCEDURAL      CLEAR-BUFFER VISUAL
0.100 PROCEDURAL      CONFLICT-RESOLUTION
0.185 VISION        Encoding-complete LOC1-0 NIL
0.185 VISION        SET-BUFFER-CHUNK VISUAL TEXT3
0.185 PROCEDURAL      CONFLICT-RESOLUTION
0.235 PROCEDURAL      PRODUCTION-FIRED ENCODE-FIRST-TXT
0.235 PROCEDURAL      CLEAR-BUFFER VISUAL
0.235 PROCEDURAL      CONFLICT-RESOLUTION
0.285 PROCEDURAL      PRODUCTION-FIRED FIND-UNATTENDED-TXT
0.285 PROCEDURAL      CLEAR-BUFFER VISUAL-LOCATION
0.285 VISION        Find-location
0.285 VISION        SET-BUFFER-CHUNK VISUAL-LOCATION LOC3
0.285 PROCEDURAL      CONFLICT-RESOLUTION
0.335 PROCEDURAL      PRODUCTION-FIRED ATTEND-TXT
0.335 PROCEDURAL      CLEAR-BUFFER VISUAL-LOCATION
0.335 PROCEDURAL      CLEAR-BUFFER VISUAL
0.335 PROCEDURAL      CONFLICT-RESOLUTION
0.420 VISION        Encoding-complete LOC3-0 NIL
0.420 VISION        SET-BUFFER-CHUNK VISUAL TEXT4
0.420 PROCEDURAL      CONFLICT-RESOLUTION
0.470 PROCEDURAL      PRODUCTION-FIRED ENCODE-SECOND-TXT
0.470 PROCEDURAL      CLEAR-BUFFER VISUAL
0.470 PROCEDURAL      CONFLICT-RESOLUTION
0.520 PROCEDURAL      PRODUCTION-FIRED FIND-UNATTENDED-TXT
0.520 PROCEDURAL      CLEAR-BUFFER VISUAL-LOCATION
0.520 VISION        Find-location
0.520 VISION        SET-BUFFER-CHUNK VISUAL-LOCATION LOC5
0.520 PROCEDURAL      CONFLICT-RESOLUTION
0.570 PROCEDURAL      PRODUCTION-FIRED ATTEND-TXT
0.570 PROCEDURAL      CLEAR-BUFFER VISUAL-LOCATION
0.570 PROCEDURAL      CLEAR-BUFFER VISUAL
0.570 PROCEDURAL      CONFLICT-RESOLUTION
0.655 VISION        Encoding-complete LOC5-0 NIL
0.655 VISION        SET-BUFFER-CHUNK VISUAL TEXT5
0.655 PROCEDURAL      CONFLICT-RESOLUTION
0.705 PROCEDURAL      PRODUCTION-FIRED ENCODE-THIRD-TXT
0.705 PROCEDURAL      CLEAR-BUFFER VISUAL
0.705 PROCEDURAL      CLEAR-BUFFER RETRIEVAL
0.705 DECLARATIVE     START-RETRIEVAL
0.705 PROCEDURAL      CONFLICT-RESOLUTION
0.755 DECLARATIVE   RETRIEVED-CHUNK FACT34
0.755 DECLARATIVE   SET-BUFFER-CHUNK RETRIEVAL FACT34
0.755 PROCEDURAL      CONFLICT-RESOLUTION
0.805 PROCEDURAL      PRODUCTION-FIRED RESPOND-ANSWER
0.805 PROCEDURAL      CLEAR-BUFFER GOAL
0.805 PROCEDURAL      CLEAR-BUFFER RETRIEVAL
0.805 PROCEDURAL      CLEAR-BUFFER MANUAL
```

```
0.805 MOTOR              PRESS-KEY 7
0.805 PROCEDURAL             CONFLICT-RESOLUTION
1.055 PROCEDURAL             CONFLICT-RESOLUTION
1.105 PROCEDURAL             CONFLICT-RESOLUTION
1.205 MOTOR              OUTPUT-KEY $(7 2)
1.205 PROCEDURAL             CONFLICT-RESOLUTION
1.355 PROCEDURAL             CONFLICT-RESOLUTION
1.355 ---          Stopped because no events left to process
```

The trace of the model could be used to predict the human subjects' behavior (such as the reaction time, RT) and the BOLD effect of the related brain areas when a subject is performing this task. We will discuss this issue in the coming sections.

2 ACT-R Meets fMRI

2.1 Regions of Interest

Our original efforts in ACT-R meeting fMRI (Anderson et al, 2003) used an exploratory study to help with the identification of the parietal, prefrontal, and motor regions. Based on this study we identified three regions, each 5 voxels wide, 5 voxels long, and 4 voxels high, about 16x16x13 mm^3 (The voxel in our experiments is 3.125 mm long and wide and 3.2 mm high). These regions were subsequently used in a number of studies, and several new regions have also been identified in these studies. Figure 1 shows the location of the 8 regions of interest (ROI) we have identified.

The detailed information of these regions, including the corresponding buffers in ACT-R (inside the parentheses), the Talairach coordinates of the center, the Brodmann Area and the size (if not 5x5x4), is as follows:

1. **Motor1 (Manual):** Centered at $x = +/-37$, $y = -25$, $z = 47$. This includes parts of Brodmann Areas 2 and 4 at the central sulcus. The activation in the left hemisphere of this region will be much stronger than that in the right if moving the right hand.
2. **Anterior Cingulate (Goal):** Centered at $x = +/-5$, $y = 10$, $z = 38$. This includes parts of Brodmann Areas 24 and 32. This is a 5-voxel-long, 3-voxel-wide, and 4-voxel-high region.
3. **Motor2 (Vocal):** Centered at $x = +/-44$, $y = -12$, $z = 29$. This includes parts of Brodmann Areas 2 and 4 at the central sulcus.
4. **Parietal (Problem State or Imaginal):** Centered at $x = +/-23$, $y = -64$, $z = 34$. This includes parts of Brodmann Areas 7, 39, and 40 at the border of the intraparietal sulcus. The activation in the left hemisphere of this region was stronger than that in the right in most of our experiments.
5. **Prefrontal(Retrieval):** Centered at Talairach coordinates $x = +/-40$, $y = 21$, $z = 21$. This includes parts of Brodmann Areas 45 and 46 around the inferior frontal sulcus. The activation in the left hemisphere of this region was stronger than that in the right in most of our experiments.
6. **Caudate (Procedural):** Centered at $x = +/-15$, $y = 9$, $z = 2$. This is a subcortical structure, with 4-voxel-long, 4-voxel-wide, and 4-voxel-high.

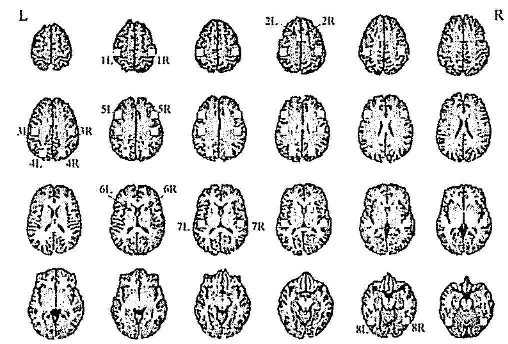

Fig. 1. Brain regions corresponding to the buffers of 8 ACT-R modules. Each slice is with 3.2 mm thick, The AC-PC line is on slice 19. The ROI number is shown in the first slice each ROI, L means the left hemisphere, R means the right hemisphere.

7. **Auditory Cortex (Aural):** Centered at $x = +/-47$, $y = -22$, $z = 4$. This includes parts of Brodmann Areas 21,22, and 42 in the region known as the auditory cortex. Note, however, this region excludes Brodmann Area 41, which is the primary auditory cortex.
8. **Fusiform Gyrus (Visual):** Centered at $x = +/-42$, $y = -60$, $z = -8$. This includes parts of Brodmann Area 37.

This completes the mapping of existing ACT-R modules onto brain regions. Of course, there are many brain regions not included above and cognitive functions not yet represented in ACT-R.

Declarative memory retrieval and problem state representation are involved in most of the cognitive tasks, including working with the Web. As an example of the consistency of the activation patterns across the cognitive tasks of above identified regions, figure 2-4 show the activation patterns in the left prefrontal (ROI 5) and left posterior parietal areas (ROI 4) across the event related fMRI experiments of algebra equations solving (Anderson et al., 2003), artificial equations solving (Qin et al, 2003), and a simple task related to memory retrieval and mental manipulation with different input and output modality (Anderson et al., 2007). We can see that the left parietal region was consistently sensitive to the task load of mental representation and mental manipulation in these tasks, and the left prefrontal region was consistently sensitive to the task load of memory retrieval in these tasks. These figures also show that the predictions of ACT-R theory are very highly correlated with the scanned data. The methods to make the predictions will be discussed in next session.

Before going ahead, we need to state a few qualifying remarks about these brain-module associations. It might seem that we are claiming that the cortical regions

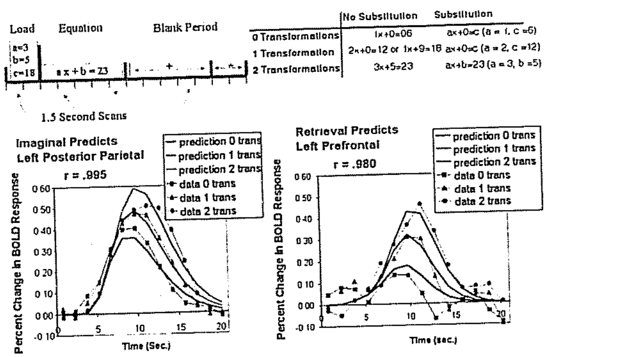

Fig. 2. The design and result of the Event-Related fMRI experiment on algebra equation solving. To solve an equation, one needs to retrieve related algebra rules and arithmetic facts, and to change the representation of the equation. One of the two factors in this design is the complexity of the equation. There are three levels of this factor. 0-transformation means no transformation needed to get the solution. 1-transformation means only one transformation (e.g., in 2x+0=12, only division is needed) needed to get the solution, and 2-transformation means 2 transformation operations needed to get the solution. The result shows the effect of this factor in both left posterior parietal cortex (ROI 4) and left prefrontal cortex (ROI 5) and the prediction of the ACT-R model based on the activation time course of imaginal buffer and retrieval buffer respectively.

listed perform the function associated with each module. While this is a plausible inference, it is not necessary to the logic of this approach. It is only necessary that the activity of the brain region reliably reflect a particular information-processing function. Even if the function is performed in that region, there is no reason to suppose that its activity will only reflect that function. Finally, there is no claim that the ascribed function is restricted to a specific region.

2.2 Using ACT-R to Predict fMRI Data

A number of researchers (e.g., Boyton et al., 1996; Cohen, 1997; Dale and Buckner, 1997) have proposed that the Blood Oxygenation Level-Depend (BOLD) response to an event in an fMRI experiment varies according to the following function of time, t, since the event

$$B(t) = t^a e^{-t}$$

where estimates of the exponent, a, have varied between 2 and 10 (and we will constrain our estimates within these bounds). This is essentially a gamma function which will reach maximum at t = a time units after the event.

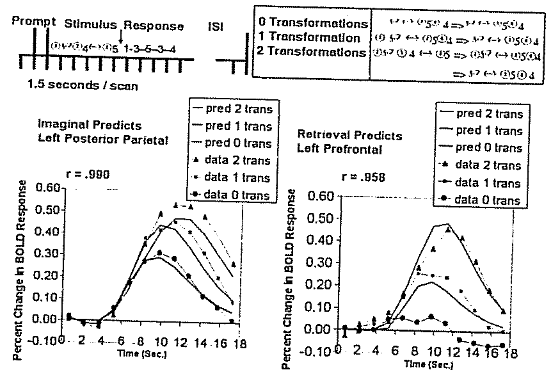

Fig. 3. The design and result of the Event-Related fMRI experiment on artificial equation solving. The result also shows the effect of the complexity of the equation in both left posterior parietal cortex (ROI 4) and left prefrontal cortex (ROI 5) and the prediction of the ACT-R model based on the activation time course of imaginal buffer and retrieval buffer respectively.

As we have seen, the output trace of an ACT-R model is the time course of the activations of the modules involved in the cognitive task and can be used to predict the behavior data such as RT. Further more, with the information of when and how long of each activation of the modules, one can predict the BOLD effect of the brain regions corresponding to the ACT-R module listed above using the gamma function.

Figure 5 illustrates the general idea about how we map from events in an information-processing model onto the predictions of the BOLD function in our early efforts. Basically, one can think of the observed BOLD function in a region (e.g., Total line in Figure 5) as reflecting the sum of separate BOLD functions for each period of time the buffer is active (e.g., the First, Second, and Third events in Figure 5). Each period of activity is going to generate a BOLD function according to a gamma function as illustrated. The peak of the BOLD functions reflects roughly when the buffer was active but is offset because of the lag in the hemodynamic response. The height of the BOLD function reflects the duration of the event since the integration makes the height of the function proportional to duration over short intervals.

In Anderson et al. (2003) it was proposed that while a buffer is active it is constantly producing a change that will result in a BOLD response according above gamma function. The observed fMRI response is integrated over the time that the buffer is active. Therefore, the observed BOLD response will vary with time as

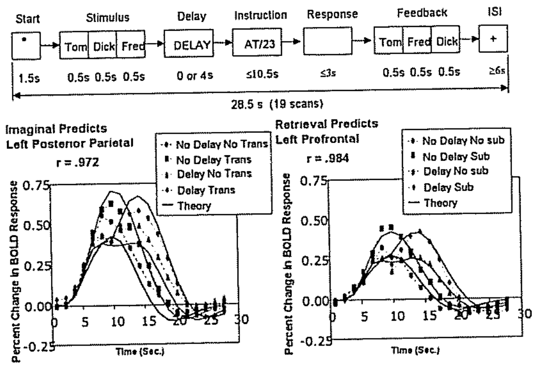

Fig. 4. The design and result of the third Event-Related fMRI experiment. The task is to change the order of the three names according to the instruction. The instruction can be a two digit number. For example, if it is 23, it means shifting the order of the second name with the third name. This is the case of with transformation. There are 1/3 trials no transformation required (e.g., the number is 24, there is no fourth name in stimulus). If the instruction is a word, such as AT, this is a trial with substitution (needs to retrieve the associated number, 23 in this example. The participants were trained well for this kind of associations before scan). There are 1/2 trials with no substitution needed. The result shows the effect of the transformation in left posterior parietal cortex (ROI 4) and the effect of the substitution in the left prefrontal cortex (ROI 5) and the prediction of the ACT-R model based on the activation time course of imaginal buffer and retrieval buffer respectively.

$$CB(t) = M \int_0^t i(x) B\left(\frac{t-x}{s}\right) dx$$

where M is the magnitude scale for response, s is the latency scale, and $i(x)$ is 1 if the buffer is occupied at time x and 0 otherwise. Note because of the scaling factor, the prediction is that the BOLD function will reach maximum at roughly $t = a * s$ seconds.

Figure 6 shows an example of the activation time course of the retrieval and imaginal modules formed from the trace of the ACT-R model for the algebra equation solving (Anderson et al., 2003). Figure 2 shows the prediction of the BOLD effect based on this kind of time course and CB(t). For the left prefrontal region (corresponding to the retrieval buffer), the parameters of scale (s) is estimated 0.691, exponent (a), 8.180, and magnitude $M\Gamma(a+1)$, 0.933 (This is a more meaningful measure since the height of the function is determined by the exponent as well as M.), the result correlation coefficient between the scan data and the prediction among the condition of the complexity of the equations is 0.980; For the left posterior parietal region (corresponding to the imaginal buffer), s is 1.647, a is 3.054, and $M\Gamma(a+1)$ is 3.486, the result correlation coefficient between the scan data and the prediction among the condition of the complexity of the equations is 0.995. Three parameters for the left motor region has also estimated and the result correlation coefficient between the scan data and the prediction among the condition of the complexity of the equations is 0.955, but the detail will not be given here.

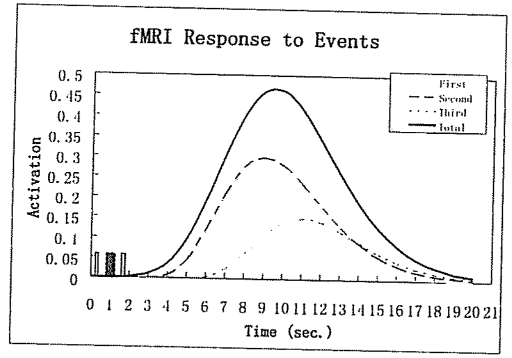

Fig. 5. The observed BOLD effect (total) in a ROI is the sum of the BOLD effects of three events in this ROI with different starting time and length of duration

These parameters were estimated by trying to minimize the following quantity:

$$\sum_{i \in ROIs} \sum_{j \in Conditions} \sum_{k \in Scans} (\bar{B}_{ijk} - \hat{B}_{ijk})^2 / S_i^2$$

where $\bar{B}_{ijk}$ is the mean BOLD response, $\hat{B}_{ijk}$ is the predicted response, and S_i^2 is the mean error in the BOLD response for ROI i calculated by the interaction between the 84 values (6 conditions x 14 scans) by 8 participants interaction term. Under the hypothesis that all deviations are normally distributed noise, this quantity is distributed as a chi-square with degrees of freedom equal to the number of observations (252) minus parameters (9)-that is 243 degrees of freedom. The value of this quantity is 340.83, which is quite significant indicating, not surprisingly, that there are things in the data not predicted by the model. On the other hand, it is not radically different from its expected value (the degrees of freedom), indicating that we are capturing most of the systematic variance (Anderson et al, 2003).

The methods of making the prediction of BOLD effect have been improved in the later studies (e.g., Anderson et al, 2007), but we will not go ahead to cover the detailed issues here.

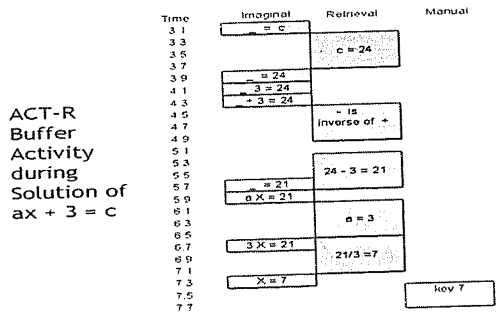

Fig. 6. The time course of the activations in the imaginal and retrieval buffers in solving the equation of ax+3=c, where a=3, c=24

3 Discussion

The central points discussed in this article are:

1. ACT-R is a theory and computational model on the components of information processing in human brain and how the interaction among these components generates human cognitive behavior. This theory and computational model may have potential application in Web intelligence;
2. The empirical findings on the brain regions corresponding to these components;
3. The approach of how the detailed processing of an information-processing theory like ACT-R can make precise predictions about the BOLD response.

This approach can be seen as a two way bridge. From one direction, it allows us to interpret the significance of the fMRI data in terms of precise information-processing operations rather than in terms of relatively diffuse concepts. On the other hand, we can use fMRI experiments to test the predictions of the theory and to help us to find ways to improve the theory. Therefore, this approach may shed light on the research of Web Intelligence meets Brain Informatics.

An important issue to make a precise prediction of fMRI data is to make a precise prediction of behavior data, which may need to estimate some parameters in an ACT-R model. For the detail, one can check the tutorial in http://act-r.psy.cmu.edu/actr6.

We have only introduced the symbolic level of ACT-R in this article. There is a subsymbolic level of ACT-R. For example, the activation of a chunk i is a sum of a base-level activation B_i, reflecting the recency and frequency of practice of the chunk, and an associative activation, reflecting its relevance to the current context:

$$A_i = B_i + \sum_k \sum_j W_{kj} S_{ji} + \varepsilon \quad (1)$$

where, W_{kj} is the amount of activation from source j in buffer k, S_{ji} is the strength of association from source j to chunk i. ε is the noise value, and the base-level activation

$$B_i = \ln(\sum_{j=1}^{n} t_j^{-d}) \quad (2)$$

where n is the number of presentations for chunk i. t_j is the time since the jth presentation. d is the decay parameter which is almost always set to 0.5.

The probability of recall the chunk i depends on the expected activation, A_i, the retrieval activation threshold τ , and the amount of noise in the system which is controlled by the parameter s:

$$recall\ probability_i = \frac{1}{1 + e^{\frac{\tau - A_i}{s}}} \quad (3)$$

Also the time to retrieve chunk i is depended on A_i:

$$Time = Fe^{-A_i} \quad (4)$$

The detailed information on the subsymbolic level of ACT-R can be found in Anderson et al (2004) and the ACT-R 6 tutorial in the ACT-R website shown above.

References

1. Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., Qin, Y.: An integrated theory of Mind. Psychological Review 111, 1036–1060 (2004)
2. Anderson, J.R., Qin, Y., Sohn, M.-H., Stenger, V.A., Carter, C.S.: An information-processing model of the BOLD response in symbol manipulation tasks. Psychonomic Bulletin and Review 10, 241–261 (2003)
3. Anderson, J.R., Qin, Y., Yung, K.-J., Carter, C.S.: Information-Processing Modules and their Relative Modality Specificity. Cognitive Psychology 54, 185–217 (2007)
4. Boyton, G.M., Engel, S.A., Glover, G.H., Heeger, D.J.: Linear systems analysis of functional magnetic resonance imaging in human V1. Journal of Neuroscience 16, 4207–4221 (1996)
5. Cohen, M.S.: Parametric analysis of fMRI data using linear systems methods. NeuroImage 6, 93–103 (1997)
6. Dale, A.M., Buckner, R.L.: Selective Averaging of Rapidly Presented Individual Trials Using fMRI. Human Brain Mapping 5, 329–340 (1997)
7. Qin, Y., Sohn, M.-H., Anderson, J.R., Stenger, V.A., Fissell, K., Goode, A., Carter, C.S.: Predicting the practice effects on the blood oxygenation level-dependent (BOLD) function of fMRI in a symbolic manipulation task. Proceedings of the National Academy of Sciences of the United States of America 100, 4951–4956 (2003)

Appendix

Here is the whole model of solving the problem of 3+4 with the Lisp functions simulating the visual stimulus in the experiment window. This experiment can be performed by a human subject or by an ACT-R 6 model.

```
(clear-all)
   (defvar *response* nil)
   (defmethod rpm-window-key-event-handler ((win rpm-window) key)
   (setf *response* (string key)))
        ;; define the experiment window

(defun do-experiment ()

(reset)

(let* ((window (open-exp-window "addition")))

(add-text-to-exp-window :text "3" :x 75 :y 175)
(add-text-to-exp-window :text "+" :x 125 :y 175)
(add-text-to-exp-window :text "4" :x 175 :y 175)
        ;;To simulate the stimulus of the experiment
(setf *response* nil)

(if *actr-enabled-p*
    (progn
     (install-device window)
     (proc-display)
     (run 10 :real-time t))
        ;; if *actr-enabled-p* = t, ACT-R model performs the task
(while (null *response*)
   (allow-event-manager window)))))
        ;; otherwise, a human subject will perform the task

(define-model addition34

     (sgp :v t :show-focus t :needs-mouse nil)
        ;; set :v to t is to show the trace, set :show-focus to t is to show the red
        ;; focus circle on the experiment window
     (sgp :esc t :lf .05 :trace-detail medium)
        ;; to set retrieval time as 0.05 seconds

     ;; Define Chunk-types
     < The ACT-R model shown in the text >
 )
```

The Neural Mechanism of Human Numerical Inductive Reasoning Process: A Combined ERP and fMRI Study

Peipeng Liang[1], Ning Zhong[1,2], Shengfu Lu[1], Jiming Liu[1,3], Yiyu Yao[1,4], Kuncheng Li[5], and Yanhui Yang[5]

[1] The International WIC Institute, Beijing University of Technology, China
ppliang@emails.bjut.edu.cn, lusf@mail.bjut.edu.cn
[2] Dept. of Life Science and Informatics, Maebashi Institute of Technology, Japan
zhong@maebashi-it.ac.jp
[3] Dept. of Computer Science, Hong Kong Baptist University, Hong Kong
jiming@comp.hkbu.edu.hk
[4] Dept. of Computer Science, University of Regina, Canada
yyao@cs.uregina.ca
[5] Dept. of Radiology, Xuanwu Hospital, Capital Medical University, China
likuncheng1955@yahoo.com.cn, yanhui826@gamil.com

Abstract. Inductive reasoning is one of the most important higher level cognitive functions of the human brain, and we still know very little about its neural mechanism. In the present study, event-related potential (ERP) and event-related fMRI are used to explore the dynamic spatiotemporal characteristics of inductive reasoning process. We hypothesize that the process of numerical inductive reasoning is partially dissociable over time and space. A typical task of inductive reasoning, function-finding, was adopted. Induction tasks and calculation tasks were performed in the experiments, respectively. ERP results suggest that the time course of inductive reasoning process is partially dissociable as the following three sub-processes: number recognition (the posterior P100 and N200), strategy formation (P300) and hypothesis generation and verification (the positive slow waves). fMRI results show many activations, including prefrontal gyrus (BA 6), inferior parietal lobule (BA 7, 40), and occipital cortex (BA 18). After the respective discussions, the two kinds of data are combined qualitatively, then the dynamic spatiotemporal characteristic of inductive reasoning process are depicted using a conceptual figure. This study is a preliminary effort towards deeply understanding the dynamic information processing mechanism of human inductive reasoning process.

1 Introduction

1.1 Background

Web intelligence (WI) is a new direction for scientific research and development that explores the fundamental roles as well as practical impacts of Artificial

N. Zhong et al. (Eds.): WImBI 2006, LNAI 4845, pp. 223–243, 2007.