TR-415

Cognitive Model of Conscious/Unconscious Processing and Its Simulation in a Parallel Logic Programming Language

by

N. Oka

July, 1988

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

意識処理と無意識処理の相互作用のモデル化

岡 夏樹

(財) 新世代コンピュータ技術開発機構
oka@icot.junet

意識処理と無意識処理の認知モデル (C/U モデル) を提案する。モデルは記号処理部と非記号処理部とからなる。記号処理部において並列に行なわれる処理のうち、ある時点では最大で一つが意識され、その他の処理 (たとえば熟練した処理) は自動的に行なわれる。記号処理は決定的に進むが、そのうち意識処理では近時記憶による近似的なバックトラッキングが可能である。非記号処理部ではネットワーク上での活性化拡散処理が無意識的に行なわれる。記号処理部からアクセスできる知識は無意識処理により文脈と関係したものに絞られるので、記号処理は効率的に行なわれる。並列論理型言語の特徴を生かした C/U モデルのシミュレーション法といくつかの例題が示される。

Cognitive Model of Conscious/Unconscious Processing and Its Simulation in a Parallel Logic Programming Language

Natsuki Oka

ICOT Research Center
Institute for New Generation Computer Technology
4-28, Mita 1-chome, Minato-ku, Tokyo 108, JAPAN
csnet: oka%icot.jp@relay.cs.net
uucp: {enea,inria,kddlab,mit-eddie,ukc}!icot!oka

A cognitive model of conscious/unconscious processing (the C/U model) is proposed. The model consists of two parts: symbolic processing and nonsymbolic processing. In symbolic processing, at most one of parallel processes is executed consciously and the others, for example, experienced processes, are executed automatically. Although conscious processing follows choice nondeterminism, recent memory enables pseudo-backtracking. Nonsymbolic processing is spreading activation in a network, which is executed unconsciously. It associatively narrows down knowledge that can be accessed from symbolic processing, enabling effective execution of symbolic processing. A simulation method of the C/U model which utilizes the characteristics of a parallel logic programming language is shown.

1 Introduction

The cognitive architecture of human beings has been modeled mainly in two ways: modeling based on network architecture analogous to the neural network (for example, [7]), and modeling based on serial symbolic processing. The former can explain automatic sensory and motor systems, and associative cognitive processing. It has the advantages of inherent parallelism, tolerance to faults and noise, and graceful degradation. However, it is not suitable for executing symbolic processing.

The latter, however, can explain symbolic problem-solving and inference, but it is not sufficient for modeling cognitive architecture as a whole. For example, it is difficult to extract related knowledge dynamically; this is one of the causes of the frame problem. It is also difficult to make a correspondence between input signals and concepts which are used by the system. The difficulty with symbol-based modeling appears as the bottleneck of knowledge acquisition in building expert systems.

In section 2, we propose a cognitive model that utilizes the benefits of both a network-based and a symbol-based model. We also discuss the other viewpoint of the model, that is, the distinction between what one can be conscious of and what one cannot be conscious of. Section 3 describes a simulation method that utilizes the characteristics of a parallel logic programming language and briefly introduces the language. Section 4 shows some examples to explain the model.

2 C/U model

Here, we propose a cognitive model of conscious/unconscious processing (C/U model) which explains various features of mental processes as follows:

- Mental processes occur in parallel. For example, sensory, cognitive and motor systems can work at the same time.
- There are two kinds of processes: what one can be conscious of, and what one cannot be conscious of. For example, a person can be conscious of logical inferences and the results of recognition, but cannot be conscious of how he recognizes human faces or how he comes up with an idea.
- A person is usually conscious of one thing at a time.
- A person can tell what he was recently conscious of.
- As a person becomes experienced at a task, it becomes automatic.
- The knowledge of which a person is conscious is automatically limited to items which relate to the current context. Therefore, he can think efficiently.
- The earlier stage of input processing is not affected by conscious processing, but the later stage is affected.

Fig. 1 shows the structure of the C/U model. The model consists of two parts: symbolic processing, of which a person can be conscious, and nonsymbolic processing (or pattern processing), of which a person cannot be conscious. Both parts are executed in parallel.

We assume that symbolic processes are executed in parallel in symbolic processing. At most one of these processes is executed consciously (conscious processing) and the others are executed automatically (symbolic unconscious processing). One example of symbolic unconscious processing is an experienced skill. A state that has no conscious process is expressed as losing one's consciousness or sleep without dreaming.

One of the characteristics of conscious processing is that it utilizes recent memory which keeps the contents of recent conscious processes. A person can report the contents of recent memory. Recent memory has limited capacity, and it does not keep the candidates which are not selected and of which the person is not conscious, but keeps what the person was conscious of. Thus, it does not enable strict backtracking such as Prolog offers, but enables pseudo-backtracking. It also enables return from a short interrupt. For example, if a person is interrupted while doing a task, he can resume the task if the interruption is short.

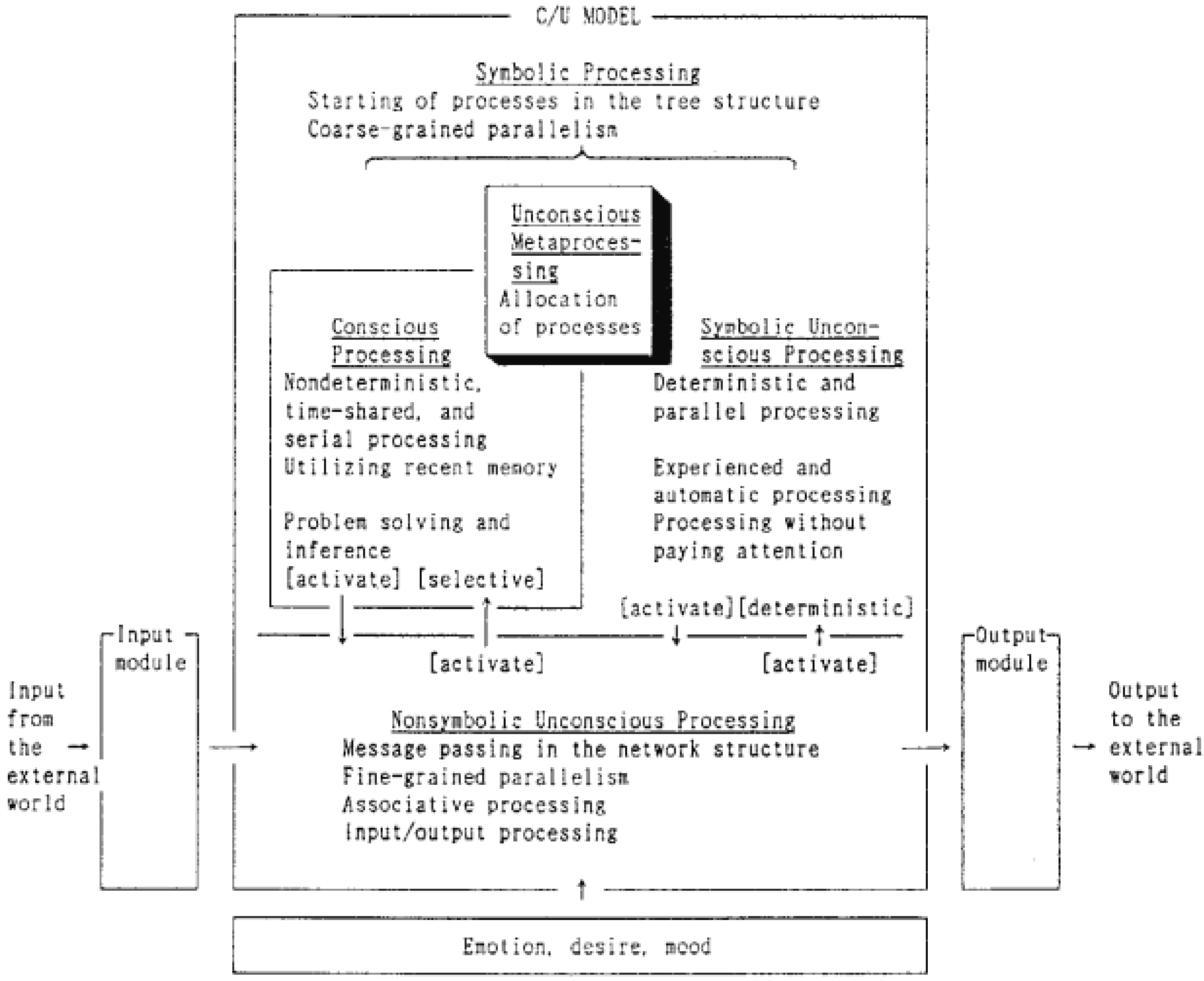


Figure 1: Structure of the C/U model

We suppose that there is an unconscious process that decides which process a person is conscious of (unconscious metaprocessing). For example, a person is usually forced to pay attention to changes in input from the environment, and tends to do tasks to which he is accustomed without paying close attention to them.

In nonsymbolic processing, nonsymbolic (or pattern) processes are executed in parallel and unconsciously (nonsymbolic unconscious processing) by spreading activation. The functions of this part are later processing of input signals, and associatively narrowing down knowledge that can be accessed from the symbolic processing part. The processing of the nonsymbolic part goes on receiving activation from the input modules and from the symbolic processing part, and according to the weights of the connections which reflect the statistical properties of past processing.

Early processing of input signals is done in input modules which are the outside of the C/U model, and the functions of input modules are not affected by symbolic processing. Emotion, desire, and mood are also the outside of the model.

The words "conscious" and "unconscious" are used in various ways. (See, for example, [6].) Representations such as losing or recovering consciousness correspond to the absence or presence of conscious processing in the C/U model. Consciousness as a monitor of the current process is represented in the model, too. Epistemic awareness about the contents of memory, such as what one knows or how one obtains knowledge is not handled in the model. Consciousness of self is not handled, either.

The term "unconscious processes" refers to various kinds of processes: symbolic unconscious processes as experienced skills or processes without focal attention, nonsymbolic unconscious processes such as nonverbal processes hiding under symbolic processes, processes in input/output modules, or emotion, desire, or mood.

3 Simulation in a Parallel Logic Programming Language

This section describes a simulation method that utilizes the characteristics of parallel logic programming languages. In this paper, it is explained using Guarded Horn Clauses (GHC) [11], a parallel logic programming language, but the basic idea is common to simulation in other parallel logic programming languages, such as Concurrent Prolog [10] and PARLOG [2].

3.1 Overview of GHC

GHC was designed as a general parallel programming language based on Horn-clause logic. It can be used for describing concepts proper to parallel programming such as parallelism, processes, communication, and synchronization. Its framework is suitable for describing open systems which interact with the outside world.

A GHC program is a set of guarded Horn clauses of the following form:

```
H :- G1,...,Gm | B1,...,Bn.     (m>0, n>0)
```

where `H`, `Gi`'s, and `Bi`'s are atomic formulas. `H` is called a clause head, the `Gi`'s are called guard goals, and the `Bi`'s are called body goals. The connective `:-` means 'is implied by', and `,` means conjunction. The only difference from an ordinary Horn clause is that one of the conjunctive operators is replaced by a commitment operator, `|`. The part of a clause before `|` is called a guard, and the part after `|` is called a body. Declaratively, the commitment operator denotes conjunction, and the above guarded Horn clause is read as "`H` is implied by `G1,...,Gm` and `B1,...,Bn`".

AND-connected goals are solved in parallel. This kind of parallelism is called AND-parallelism in this paper. Clauses which can be resolved with a goal are searched in parallel, and one of the clauses whose guard has succeeded is selected; at that time, other candidates are discarded, therefore, the selection cannot be made again. This kind of nondeterminism is called choice nondeterminism. Choice nondeterminism is not suitable for gathering multiple solutions, but it is suitable for describing the progress of things in the real world. AND-parallelism and choice nondeterminism are common characteristics of parallel logic programming languages.

GHC introduces partial order on bindings by the suspension rules of unification:

- Guard goals cannot instantiate the caller.
- Body goals of a clause cannot instantiate the caller until the clause is selected.

  Unification that can succeed only by causing such instantiation is suspended until it can succeed without causing such instantiation.

GHC can be viewed as a process description language. That is, a goal can be viewed as a process that observes input bindings and generates output bindings according to them. A program clause can be viewed as a process rewrite rule. Interprocess communication is done using shared variables, and synchronization is realized by the above suspension rules.

3.2 Simulation Method

This section explains the correspondence between the concepts in the C/U model and the descriptions in GHC. Symbolic processing in the C/U model is described as successive starting of processes

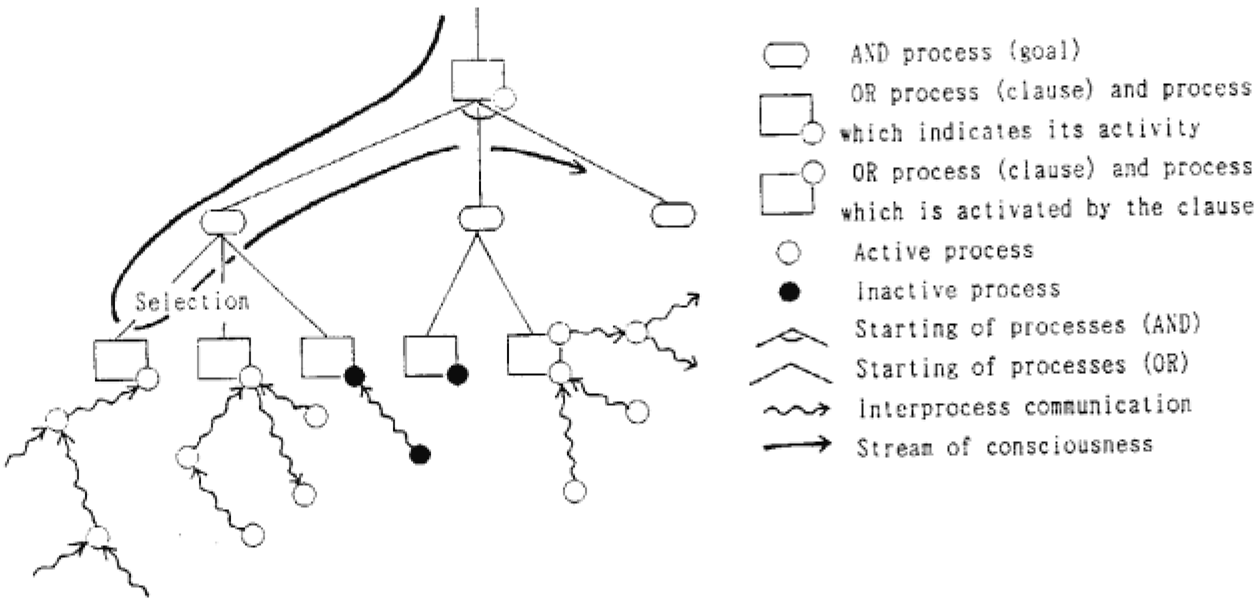


Figure 2: Computation in the C/U model

in the AND/OR-tree structure, and nonsymbolic processing in the model is described as interprocess communication in the network structure (Fig. 2).

A template of the program which simulates the C/U model is shown below. In GHC, we follow the syntactic convention that begins variables with uppercase letters. In the following templates of programs, most arguments are omitted for simplicity.

```
mental_processes(  ):- true|
        symbolic_processing(  ),
        nonsymbolic_processing(  ).
```

This clause says that symbolic processing and nonsymbolic processing are executed in AND-parallel.

A template of clauses which represent symbolic processing is as follows:

```
head(  ):- State1=active|
        goal11(  ),...,goal1n(  ).              (1)
head(  ):- State2=active|
        goal21(  ),...,goal2m(  ).              (2)
head(  ):- State3=active|
        goal31(  ),...,goal3l(  ).              (3)
...                                             ...
```

`State1`, `State2`, and `State3` are variables which denote the activity of each clause, (1), (2), and (3). Each variable is kept uninstantiated while the corresponding clause is inactive, and it is instantiated by nonsymbolic processing when the corresponding clause turns active. If the variable is not instantiated, execution of the guard goal is suspended according to the suspension rules of GHC.

Clauses of this form represent the following concepts:

- A kind of parallel processing

Body goals can be solved in parallel (one type of AND-parallelism). This represents one kind of parallelism in human information processing, that is, one can do many things at the same time; one thing is done consciously, while the others are done automatically.

- Arrival order dependency

  Which clause is chosen among clauses whose guard goals have succeeded is not decided deductively. It typically depends on the order of success. As Hewitt points out, the behavior of concurrent systems is often critically affected by the order of arrival of communications, and in general, the arrival order decisions are not deductively derivable [4]. We suppose that the arrival order dependency is common to human information processing.

- Stream of consciousness

  Once a clause is chosen, the selection cannot be made again (choice nondeterminism). This represents the nature of nondeterminism of human information processing, that is, processes proceed as time passes and basically do not backtrack. For the conscious process, this represents a stream of consciousness.

- Knowledge with scope

  Only a clause whose guard has succeeded can be chosen, and one of the guard goals of each clause tests the activity of the clause. This represents knowledge with scope which enables efficient inferences according to the context. That is, symbolic processing can access only knowledge which is activated by unconscious processing.

- Input-driven processes

  While the variables which denote activity of the clauses are not instantiated, executions of the guard goals are suspended. This represents input-driven processes, that is, one is conscious only of the presence of some stimulus, but not conscious of its absence.

An example of clauses which represent nonsymbolic processing in the C/U model is partly shown below. This part of the C/U model can be viewed as simulating the connectionist model [7], that is, a node in the connectionist model is represented by a process in GHC program, and spreading activation is represented by interprocess communication.

```
process(In1,In2,Out1,Read,Activity,State1):- In1=[fire|Tail]|
        calculate(Activity,NewActivity),
        set_Out1_if_necessary(  ),
        instantiate_State1_if_necessary(  ),
        process(Tail,In2,Out1,Read,NewActivity,State1).
process(In1,In2,Out1,Read,Activity,State):-
        Read=[(Threshold,State1)|Tail], Activity>=Threshold|
        State1=active,
        process(In1,In2,Out1,Tail,Activity,State).
process(In1,In2,Out1,Read,Activity,_):-
        Read=[(Threshold,State1)|Tail], Activity<Threshold|
        process(In1,In2,Out1,Tail,Activity,State1).
...
```

The process programmed here has two input streams, `In1` and `In2`, and one output stream, `Out1`, for interprocess communication. The process has a corresponding clause, (1), for the symbolic processing shown above. The activity of the process can be read through a stream, `Read`. The first clause of the program means that if the `In1` stream is instantiated, that is, if the process which is connected through `In1` fires, then `NewActivity` is calculated, and so on. The second clause shows that if a message, `(Threshold,State1)`, comes through the `Read` stream, and the process is active, then variable `State1` in the message is instantiated. Variable State1 is used for judging the activity of clause (1) as shown above.

To sum up, from the viewpoint of expansion of the basic function of GHC, the function of the C/U model is described as follows: the model utilizes AND-parallelism and choice nondeterminism of the language as a basis, narrowing down OR candidates with nonsymbolic unconscious processing, and enabling pseudo-backtracking with recent memory in conscious processing.

The unconscious process that decides which process a person is conscious of and the conscious process with recent memory have not been implemented. They are planned to be realized as an expansion of the GHC interpreter; the basic idea is that processes which can be executed deterministically by narrowing down OR candidates tend to be executed automatically.

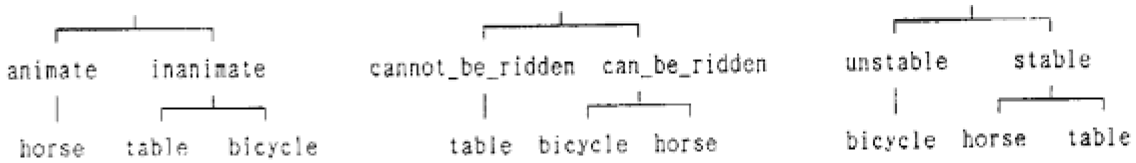
4 Examples

This section describes three examples to explain the advantages of the C/U model. Example 1 shows dynamic extraction of knowledge which relates to the current context. Example 2 describes pseudo-backtracking utilizing recent memory. Example 3 is an example of AND-parallelism, input-driven processes, and skill acquisition.

4.1 Example 1: Knowledge with scope

In conventional knowledge representation such as frame or semantic networks, knowledge is statically represented for particular problems; thus, it is difficult to use the knowledge for other problems. If knowledge is added for further use, another problem of specifying related knowledge occurs.

For example, the hierarchical relation among horse, table, and bicycle must be represented differently depending on the problem as follows:



In the C/U model, these relations are all represented together as shown below, and related knowledge is dynamically extracted by nonsymbolic unconscious processing according to the current context. For example, in a context which relates to a trip, the property, can-be-ridden, may be found. Nonsymbolic processing is not shown here; it is written as spreading activation as shown in section 3.2.

```
property(horse,X):- Life=active| X=animate.
property(horse,X):- Riding=active| X=can_be_ridden.
property(horse,X):- Stability=active| X=stable.
...                 ...
property(table,X):- Life=active| X=inanimate.
property(table,X):- Riding=active| X=cannot_be_ridden.
property(table,X):- Stability=active| X=stable.
...                 ...
property(bicycle,X):- Life=active| X=inanimate.
property(bicycle,X):- Riding=active| X=can_be_ridden.
property(bicycle,X):- Stability=active| X=unstable.
...                   ...
```

4.2 Example 2: Pseudo-backtracking

Waltz et al. explained the interpretation of semantic garden path sentences by an activation network [13]. Initially ambiguous and unstable, the network settles on a single interpretation, using a parallel, analog relaxation process. An example is the sentence, "The astronomer married the star." "Star" means both movie star and celestial body. The meaning, celestial body, is initially highly preferred because "astronomer" primes celestial body, but eventually, since celestial body conflicts with "married", the meaning, movie star, wins.

They argue that, if activation level is taken as a prime determinant of the contexts of consciousness, then their model captures a common experience of people when hearing the sentence. However, they do not argue further about consciousness and backtracking.

In the C/U model, the interpretation of semantic garden path sentences is explained as intentional pseudo-backtracking using recent memory. In the following program, nonsymbolic processing is not shown, and some arguments are omitted for simplicity.

```
            ...
            meaning(star,Meaning,none),
            consistency_check(Meaning,Result,..),
            branch(Meaning,Result,..),
            ...

branch(Meaning,consistent,..):- true| true.
branch(RecentMemory,inconsistent,..):- true|
            meaning(Word,Meaning,RecentMemory),
            consistency_check(Meaning,Result,..),
            branch(Meaning,Result,..).

meaning(star,Meaning,RecentMemory):-
            CelestialBody=active, CelestialBody\=RecentMemory|
            Meaning=celestial_body.
meaning(star,Meaning,RecentMemory):-
            MovieStar=active, MovieStar\=RecentMemory|
            Meaning=movie_star.
```

The second clause of branch shows that if the meaning was inconsistent, another meaning is found and its consistency is checked again. The first clause of `meaning` says that if `CelestialBody` is active and it is not in `RecentMemory`, that is, it was not tried before, set the meaning of "star" to `celestial_body`. The capacity of recent memory is set to 1 for simplicity.

4.3 Example 3: Simulation of a robot

This example has not been implemented, but here, we briefly point out the advantages of the C/U model for representing this kind of problem. A robot should do many things at the same time, such as problem solving, speaking, and moving its arms and fingers. The C/U model can describe this kind of AND-parallelism. Consciousness enables serial communication with the parallel system, the robot, at an appropriate level, that is, the level of consciousness is appropriate for communication with man. Recent memory is necessary for problem solving. Since a robot receives stimuli from the outside world, input-driven processes are necessary, too. The C/U model can also explain skill acquisition as shown below.

If a robot is not experienced in a task, it consciously executes the task in detail. For example, it moves its arm above a block, opens its hand, lowers its arm, and closes its hand. If the robot masters the task, it only pays attention to more abstract levels, such as moving a block, and detailed actions are done automatically. In the C/U model, skill acquisition of this kind is explained as narrowing down OR candidates so that there is no need for selection, that is, so that the task can be done automatically.

5 Related Research

Pylyshyn [9] hypothesized three autonomous levels of description: biological, symbolic, and semantic. He also insisted on the distinction between functional architecture and representation-governed cognitive processes. Cognitive processes are realized by symbol systems. Cognitive processes are said to be cognitively penetrable, which means that the function of the architecture can be altered in a semantically regular way by changing the subject's goal or beliefs, and functional architecture is said to be cognitively impenetrable.

Symbolic processing in the C/U model corresponds to cognitively penetrable cognitive processes, and nonsymbolic processing in the C/U model corresponds to cognitively impenetrable functional architecture.

Fodor and Pylyshyn [3] insist that implicit rules are hard-wired; here, being implicit means that a rule cannot be modified or examined. In the C/U model, implicit rules are not necessarily hard-wired. Nonsymbolic unconscious processing is implicit, and symbolic unconscious processing is usually implicit.

Johnson-Laird proposed the following parallel processing model [5]. There is a division in the mind between a high-level operating system and a hierarchical organization of parallel processors. The evidence for this division comes from a clue to human mentality: the distinction between what one can be conscious of and what one cannot be conscious of. A high-level processor monitors and controls the overall goals of lower-level processors, which in turn monitor and control the processors at a still lower level, and so on in a hierarchy of parallel processors, which at the lowest level govern sensory and motor interactions with the external world. There are interactions between processors at the same or different levels. Interactions are limited to the exchange of information so that one processor is unable to gain access to the internal operations of other processors.

Although conscious processing in the C/U model superficially corresponds to the processing in the high-level operating system, conscious processing does not control nonsymbolic unconscious processing, and nonsymbolic unconscious processing is not hierarchical. That is, the principle of the parallelism in the C/U model is different from that of Johnson-Laird's model.

Anzai et al. proposed a model which combines a pattern recognition mechanism represented in a network-based model, and an inference mechanism represented in a rule-based system [1]. Roughly speaking, the pattern recognition mechanism corresponds to nonsymbolic unconscious processing of the C/U model, and the inference mechanism corresponds to conscious processing. However, those two mechanisms seem to be less closely connected to each other than in the C/U model, and they use a network-based model only for pattern recognition, and not for hidden support for conscious inferences.

In research of artificial intelligence as engineering, that is, in research which aims at making a useful computer program, they have tried to represent implicit knowledge explicitly. However, the aim of the C/U model is to simulate human information processing, that is, to represent implicit knowledge as it is.

There is a similar argument about backtracking. For example, Ueda showed a method for transforming exhausted search programs in pure Prolog into deterministic programs in GHC which generate the same answer [12]. However, our aim is not to realize exhaustive search in the C/U model, but to simulate imperfect backtracking of human beings as shown in section 4.2.

6 Conclusions

This paper proposed a cognitive model of conscious/unconscious processing (the C/U model), and showed a simulation method of the model which utilizes the characteristics of a parallel logic programming language. The characteristics of the model are:

- Knowledge with scope
- Inherent AND-parallelism
- Serial communication utilizing consciousness
- Choice nondeterminism
- Pseudo-backtracking utilizing recent memory

Simulation programs of some examples are under development in order to verify the validity and the efficiency of the model. Some of the open problems are the mechanism of deciding which process a person is conscious of, and the learning mechanism in the model.

Acknowledgments

I would like to thank the researchers of ICOT, the members of the ICOT-FAI working group, and the members of AIUEO for their helpful discussions and comments.

References

[1] Anzai, Y., Mori, H., Ito, M., and Hayashi, Y., A Serial-Parallel Integrated Information Processing Model for Complex Human Problem Solving, in G. Salvendy (ed.), *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*, pp. 175-182, Elsevier Science Publishers, 1987.

[2] Clark, K. L. and Gregory, S., PARLOG: Parallel Programming in Logic, *ACM Trans. Prog. Lang. Syst.*, Vol.8, No.1, 1-49, 1986.

[3] Fodor, J. A. and Pylyshyn, Z. W., Connectionism and Cognitive Architecture: A Critical Analysis, *Cognition*, Vol.28, No.1-2, 3-71, 1988.

[4] Hewitt, C., Metacritique of McDermott and the Logicist Approach, *Comput. Intell. 3*, 185-189, 1987.

[5] Johnson-Laird, P. N., *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness*, Cambridge University Press, 1983.

[6] Klatzky, R. L., *Memory and Awareness: An Information-Processing Perspective*, W. H. Freeman and Company, 1984.

[7] McClelland, J. L., Rumelhart, D. E., and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press, 1986.

[8] Oka, N., A Cognitive Model of Conscious/Unconscious Processing, in *Proc. 4th Conference of Japan Society for Software Science and Technology*, 459-462, 1987 (in Japanese).

[9] Pylyshyn, Z. W., *Computation and Cognition: Toward a Foundation for Cognitive Science*, MIT Press, 1984.

[10] Shapiro, E. Y., A Subset of Concurrent Prolog and Its Interpreter, ICOT Tech. Report TR-003, Institute for New Generation Computer Technology, 1983.

[11] Ueda, K., Guarded Horn Clauses, *ICOT Tech. Report TR-103*, Institute for New Generation Computer Technology, 1985 (revised in 1986). also in E. Wada (ed.), *Proc. Logic Programming '85*, Lecture Notes in Computer Science 221, pp. 168-179, Springer-Verlag, 1986.

[12] Ueda, K., Making Exhaustive Search Programs Deterministic, *New Generation Computing*, Vol.5, No.1, 29-44, 1987.

[13] Waltz, D. L. and Pollack, J. B., Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation, *Cognitive Science*, Vol.9, No.1, 51-74, 1985.