TR-88-001

EVALUATION OF REDUCED-RADICAL CHINESE INPUT METHODS FROM PARSING VISUAL ROUTINES

TR-88-001

Evaluation of Reduced-Radical Chinese Input Methods from Parsing Visual Routines

K. Y. Cheng, M. S. Hwu, S. Y. Hsu, D. H. Lu, G. T. Cheng, J. P. Hwang

Institute of Information Science
Academia Sinica

ABSTRACT

A theoretic basis for the evalution of reduced-radical Chinese input methods is discussed. A reduced-radical Chinese input method is defined in terms of a set of derivation diagrams, each describes the decomposition process of a Chinese character morphologically. Thus the derivation diagrams constitute visual routines in an operator's perception mind. In this paper, we show that derivation diagrams are parses of a set of web production rules such that the evalution to a reduced-radical Chinese input method can be performed based upon the number of production rules and their difficulty in the decomposition process required in the method.

I. INTRODUCTION

A Reduced-Radical Chinese Input (RRCI) method specifically means the one that uses keys on keyboard as the external code symbols in the decomposition of Chinese characters morphologically. Among them, Tsang-Chi, Three-Corner, and Simplest are most often mentioned methods [1]. To obtain a string of external code symbols from decomposing each Chinese character, each method has its own decomposition rules which are normally an order of spatial relations in a square block and its exceptions. For example, normal order of the decomposition rule of Three-Corner method is from upper-left then to upper-right then to lower-left then to lower-right and so the decomposition of 矮 :: = 人 , 禾 , 大 = 87,29,42, the external codes of the method. Similarly, Tsang-Chi is from left-to-right then up-to-down then outside-to-inside and so 腦 ::= 月 , 女 , 女 , 田 . Thus, in using an RRCI method, an operator has to memorize a table of the external code symbols as well as to be trained to have the capability in creating some visual routines to extract the proper codes in order from each square-blocked character.

In this paper, we are interested in the derivation of grammatical production rules from an operator's visual routines so that based on these rules an evaluation procedure for measuring an RRCI's efficiency such as rates of depressed keys required, collisions, and decomposition speed are obtained. Recently, the Information Industrial Institute of R.O.C. published a report saying that according to its survey any RRCI method is worthwhile to promote due to its compactness and ease of use, however, the report also pointed out a necessity to develop better approaches than current RRCI's to enhance especially the speed of decomposition [2]. Therefore, our purpose here is to present a theoretic basis for a machine which by itself can do the evaluation analysis directly from the input of visual routines. With this, a designer can implement his/her ideas of what a new method looks like and describe it visually to the machine to perform an immediate

evaluation and to see the result of what being sketched, hoping that better approaches can be reached. This idea had been proposed by the first author in a meeting on Chinese Computers sponsored by the Institute of Information Science and the Institute for Information Industry several years ago.

For convenience in our following discussion, some useful notations are given below:

$K(s)$ = number of depressed keys (or external code symbols) required for the decomposition of the character s.

K_a = average number of keys depressed.

$C(s)$ = m implies there are m other characters ($\neq s$) whose external code symbols are exactly the same as that of s, or m collisions.

R_c = rate of collision.

$T_d(s)$ = normalized delay time required for the decomposition of s,

$0 \leq T_d(s) \leq 1.$

$S_d(s)$ = the decomposition speed of s, i. e. the facile degree of decomposition of s, $0 \leq S_d(s) \leq 1.$

R_d = average of the decomposition speed.

Let $P = \{ p_i \}$, i=1,2,...,k denote the set of a RRCI's external code symbols (or primitives) and $S = \{ s_j \}$, j=1,2,...,n be the collection of Chinese characters. Then

$$K_a = \frac{1}{n} \sum_{j=1}^{n} K(s_j)$$

$$R_c = \frac{1}{n} \sum_{j=1}^{n} C(s_j)$$

$$R_d = \frac{1}{n} \sum_{j=1}^{n} S_d(s_j)$$

In the above, $T_d(s)$ is used to measure the delay time between start-to-watching and start-to-keying the character s. If $T_d(s) = 0$ then there is no delay time or it means no ambiguity at all in the decomposition of s. On the other hand, $T_d(s) = 1$ means the operator does not know how to decompose s. Similarly, $S_d(s)$ is used to measure the decomposition speed of s, $S_d(s) = 1$ means it is extremely easy to decompose s, on the contrary, $S_d(s) = 0$ means in no way to decompose s. Thus, for simplicity, we let

$$S_d(s) = 1 - T_d(s) \quad , \quad 0 \leq S_d(s), T_d(s) \leq 1.$$

In the following, we divide our discussion into sections. Section II describes two graph representations for the decomposition process of the Chinese character and the conversion of the graph into a derivation diagram. Section III shows that a derivation diagram is a context-free web grammar whose production rules can be obtained from parsing the derivation diagram. Section IV gives a measure of the decomposition speed by fuzzy logic. Section V describes an evaluation machine (or a simulator) and a visual programming approach to interface with RRCI methods.

II. DECOMPOSITION GRAPH

In actual operation, any Chinese character input to an operator's visual mind can be regarded as a sensory icon being processed by a perception mechanism to

form a decomposition graph as a recognition of the visual routines concerning a particular method . Although different methods produce different decomposition graphs for each Chinese character, but all of the graphs can be defined uniquely as $G_d = (V, E)$, where V is the set of nodes and E is the set of edges. For example, Fig.1 shows the bipartite graph expression for the decomposition of 腦. in Tsang-Chi method, where the left vertices are sensory icons and the right vertices are some spatial relations and part-of predicates.

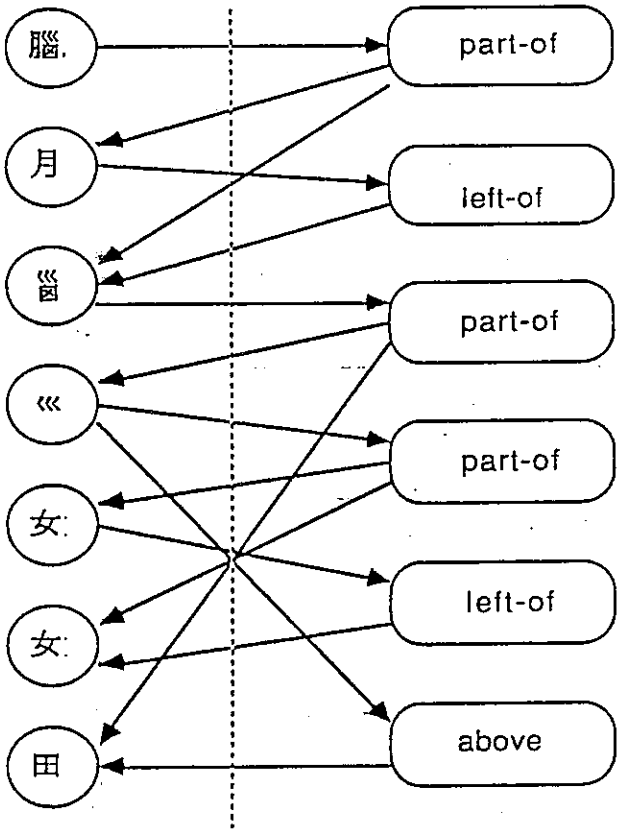


Fig.1 The bipartite graph expression for the decomposition of 腦.

More convenient expression for the representation of decomposition process would be the directed graph rather than that of the bipartite graph. Fig. 2 shows a directed graph expression of 腦. Now the vertices are icons and their interconnecting arcs are each assigned with either a spatial relation or a part-of

predicate. Notice that if only the primitive icons (external code symbols) are regarded as terminal nodes then the visited terminal nodes in a counterclockwise traverse of the graph are the external code symbols corresponding to the characters i.e., 月 , 女, 女, 田 in Fig. 2 .

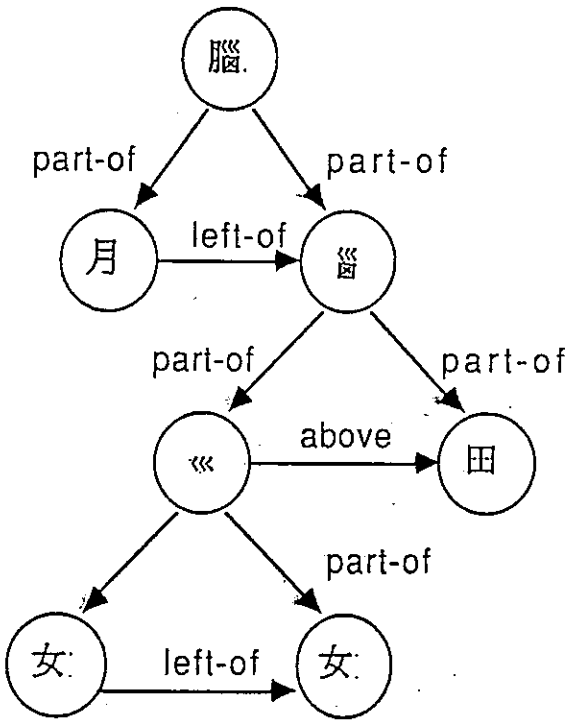


Fig.2 The directed graph expression for the decomposition of 腦.

As can be seen from above, we need to distinguish the part-of predicate from other spatial relations. Hence we describe a new graph called a derivation diagram as follows. A derivation diagram is obtained from a directed graph by using a dash-line arrow to denote a part-of predicate and a solid-line arrow to denote a spatial relation. Fig.3(a) shows the derivation diagram of 腦. Notice that the derivation of a node inherits the spatial relationship from the father node to form a complete conceptual graph [11]. Also, the process to decompose a Chinese character into a sequence of external code symbols has been embedded in the derivation diagram. Then, there are three derivation steps for the decomposition of 腦 in

Tsang-Chi method although each derivation step may involve some other rules such as the limitation of length of the external code symbols . For example, in Fig. 3(b) 腦. is decomposed according to the separation operator "left-right",㐫 is decomposed according to the separation operator "up-down", «« is decomposed into 女 ,女,女 but limited to only two codes and so 女,女 instead (i.e. this derivation step involves two rules in the decomposition, one is left-right separation and the other is the reduction of code length.). Fig. 3(c) shows a partial set of rules of Tsang-Chi method.

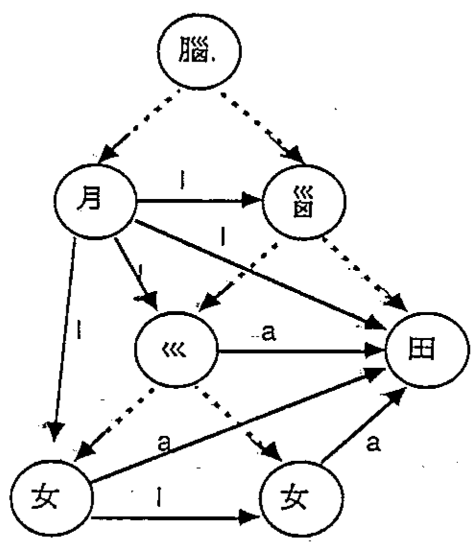
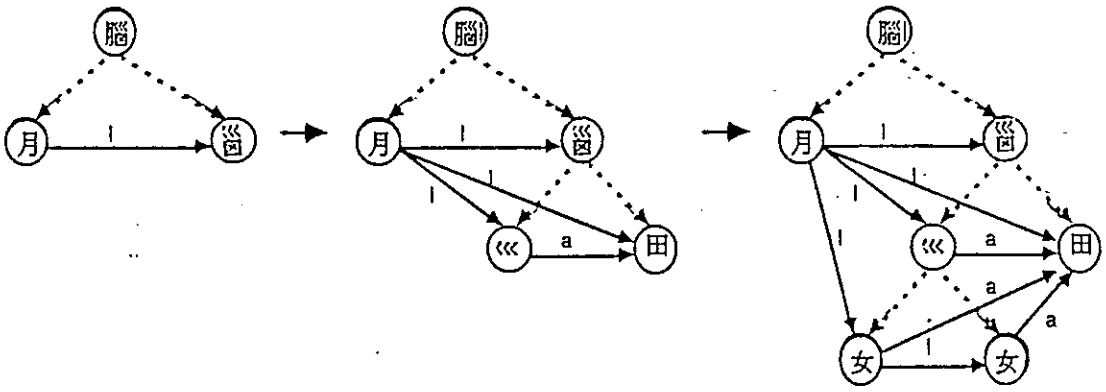


Fig.3(a) The derivation diagram of 腦.



rules used: i1 | rules used: i2 | rules used: i1, j1

Fig.3(b) the derivation steps for the decomposition of 腦.

:	:
rule i. separation rules	1. "left-right".
	2. "up-down".
	:
rule j. reduction rules	1. if symbols obtained from separation rules are more than two, take only the first and the last symbol.
:	:
:	:

Fig. 3(c) Table of Tsang-chi decomposition rules

III. CONTEX-FREE WEB GRAMMAR

From formal language theoretic point of view, a derivation diagram is a web grammar which can be used to produce sentential forms [3]. We shall show that a

derivation diagram is a context-free web grammar in the following. Sentences generated by a web grammar can be depicted as directed graphs with symbols at their nodes [4]. Due to this fact, the derivation diagram described above can be generated by a web grammar. The definition of a web grammar is as follows

$$G = (V_N, V_T, P, S)$$

where V_N is a set of nonterminals, V_T is a set of terminals, S is a set of initial webs, and P is a set of web production rules. A web production rule is defined as

$$\alpha \rightarrow \beta, \quad E$$

where α and β are webs and E is an embedding of β . In order to substitute a subweb α of a web ω by another subweb β , we have to specify how to embed β in ω in place of α . In the case of RRCI approaches, the specifications of E are some spatial relations. For example, to generate the web 腦. in Tsang -Chi method, V_N = { 腦., 㘝 , ⺍ }, V_T = { 月 , 女: , 田 }, S = { 腦. }, and P = production rules shown in Fig. 4. As can be seen, 月 , 女: , 女: , 田 is a sentence generated by the web production rules. Also, we see an immediate fact : a derivation diagram embeds a set of RRCI's production rules, or the RRCI's web grammar can be easily obtained from parsing the derivation diagram.

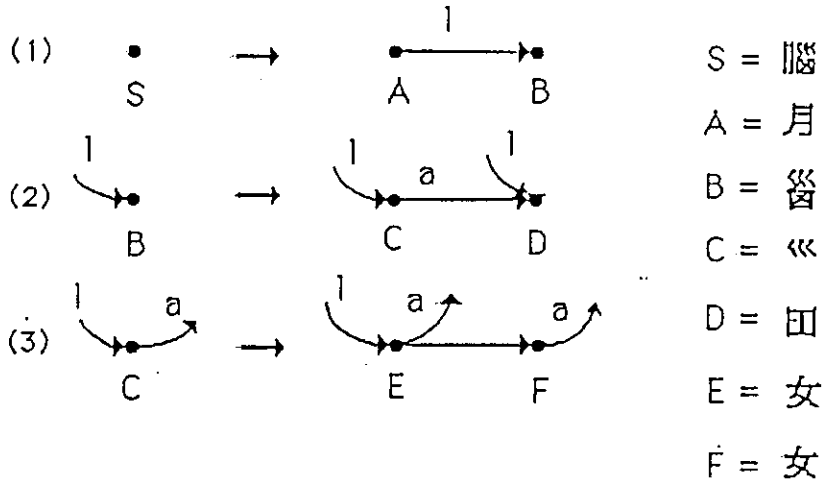


Fig.4 The production rules for the generation of Fig.3.

To derive web production rules from derivation diagram, we shall consider two subgraphs, one is called the skeleton and the other is called the section. A skeleton is a subdiagram obtained from a given derivation diagram by removing all of its spatial relation edges and leaving only part-of predicate edges and their connecting nodes. For example, Fig. 5(a) shows a skeleton of the diagram of Fig. 3(a). Notice that the skeleton is a tree structure whose root or subroot corresponds to a web α and its decendents to another web β and thus a rewritting (or production) rule $\alpha \rightarrow \beta$ where $\alpha \in V_N$ and $\beta \in V_N \cup V_T$ is obtained. Obviously, the grammar thus obtained is context-free because of the nature of tree structure.

A section is a subdiagram that corresponds to a sentential form of the language generated by the web grammar. Suppose $n_0,.....m_i$ is a path from root node n_0 to a frontier node (or leave node) m_i of the skeleton. Let $m_1,m_2,.....,m_k$ be all the frontier nodes. Then a set C of nodes of the derivation diagram is a cross-cut set if $C \cap [\, n_0,.....,m_i \,]$ is a singleton for all $1 \leq i \leq k$. A

cross-cut set C, together with all the edges of the derivation diagram between nodes of C, is called a section. We are only interested in the terminal section that shows the external code symbols in order. For example, Fig. 5(b) shows the terminal section of Fig. 3(a).

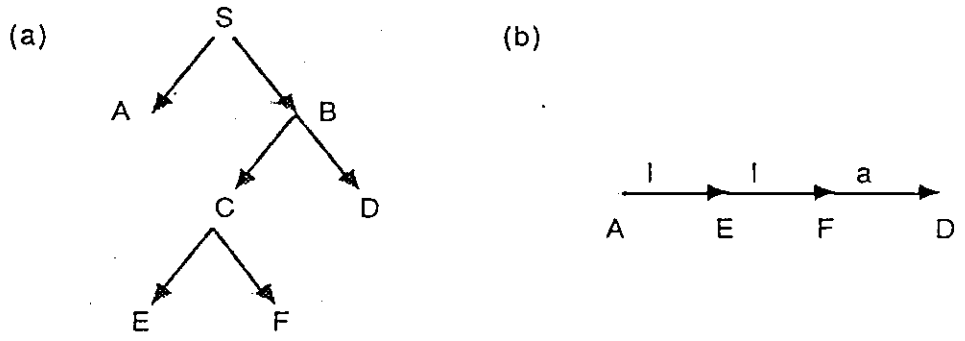


Fig. 5 Two subgraphs of Fig. 3(a): (a) the skeleton, (b) the terminal section.

To obtain the web grammar from a given derivation diagram, a skeleton is first obtained to derive the production rules with $\alpha \rightarrow \beta$, $\alpha \in V_N$ and then a section is obtained to describe the embedding relationship of a production rule. Notice that K (s) is the number of production rules with $\alpha \rightarrow \beta$ required in the decomposition of s. And, C(s) $=k_i$ if there are k_i other characters whose external code symbols are identical to the external code symbols of s. Also notice that the number of production rules with $\alpha \rightarrow \beta$, $\alpha \in V_N$ is equal to the number of internal nodes in the skeleton. We shall use this number as a factor to measure the decomposition speed in the following section.

IV. MEASURING DECOMPOSITION SPEED IN FUZZY LOGIC

As stated previously, decomposing each Chinese character into a sequence of reduced-radical symbols is through some visual routines, here a routine specifically

means one application of a rule. Thus, the decomposition speed depends on the number of rules being applied, the more rules are applied the slower the decomposition speed is reached.Also, the frequency of each rule in use is a factor that affects the decomposition speed, the more often applied rules are used the easier the decomposition is reached. The measurement of decomposition speed may be expressed in terms of fuzzy logic [5].

Let U be the universal set of P or $U = 2^P$, then the Chinese character set $S = \{s_i\}$, $i = 1, ..., n$ is a proper subset of U. Let $\mu(s_i)/s_i$ denote that each s_i in S is assigned with a function $\mu(s_i)$, called the membership function of s_i, to indicate its decomposition speed. Then the set of Chinese characters under consideration is a fuzzy subset of U.

$$S = \sum \mu(s_i)/s_i$$

where μ is a mapping

$$\mu : U \rightarrow [0,1]$$

Although there are many (human) factors that affect the decomposition speed, for simplicity, we assume the only factor is associated with production rules required for the decomposition. Without loss of generality, we also assume $S_d(\omega) = 1$ for applying the rule $\alpha \rightarrow \omega$ where ω is an external code symbol, and $S_d(\omega) < 1$ for applying a rule $\alpha \rightarrow \beta$, $\alpha \in V_N$. In order to calculate the values of μ at each node, we introduce the node height in the skeleton as follows

$$h(p_i) = 1 , \qquad p_i \in P .$$

$$h(q_j) = \max\{h(q_j)+1\} , \qquad q_j \in \text{child}(q_i).$$

The value of membership function at each node depends on its height in the skeleton and which rules being used in the decomposition process. Since each terminal node has height one, its value $\mu_1(p_i) = 1$. For each internal node, we use the following membership function

$$\mu_h(q_i) = [\ (\sum_j (\mu_{h-1}(q_j)) * f) / \sum_j h(q_j)\ ]^{1/2}$$

where f is to measure the weight associated with the frequency of each production rule being used

$$f = [\ 0.8 + 0.2 * (a_i / \sum_j a_j)]^{1/2},$$

where a_i is the number of occurrence of i^{th}-rule in the decomposition for the whole set S and K is the total number of decomposition rules in the input method.

V. SIMULATOR AND VISUAL PROGRAMMING

A Chinese input simulator is a machine that accepts user-defined decomposition graphs and corresponding decomposition rules as the input and produces the evaluation result as the output. Thus a simulator can be defined as a six tuple

$$CIS = (Q, S, \Gamma, \delta, q_0, F)$$

where:

1. Q is a finite set of states.
2. S = G (P, E) is a finite set of decomposition graphs, the node set P is a finite set of external code symbols and the edge set E consists of two types of edge, a dash-line arrow specifies a part-of predicate and a solid-line arrow specifies a spatial relation.
3. Γ is a finite set of output values.
4. δ is a mapping from $Q \times (S \cup \{\lambda\})$ to a finite subset of $Q \times \Gamma^*$, and λ denotes a null element.
5. $q_0 \in Q$ is the initial state.
6. $F \subseteq Q$ is the set of final states.

It is worthwhile to note that the values of K_a, R_c, and R_d are updated at each intermediate state. In other words, to update the value of K_a at the state q_x, the values x-1 and the sum of $K(s_i)$ from q_1 to q_{x-1} were accumulated at the state q_{x-1}. Also, the finite state q_f can be set by a RRCI designer. For example, say, if he/she decides to test 100 samples then $q_f = q_{100}$.

The input of decomposition graphs to the simulator may be through a visual programming environment [6,8,9]. The visual program allows a designer to specify the decomposition graphs on the screen window step-by-step. There is a definition language for defining each element of P , the external code symbol, as an icon. Usually, icons are defined in the right portion of the screen window and some manual selection operation commands are on the upper-most of the screen, the rest area is the working area for the construction of decomposition graphs and a table for the selection of the decomposition rules. Fig. 6 shows part of the screen design for

the input of decomposition graph and decomposition rules.

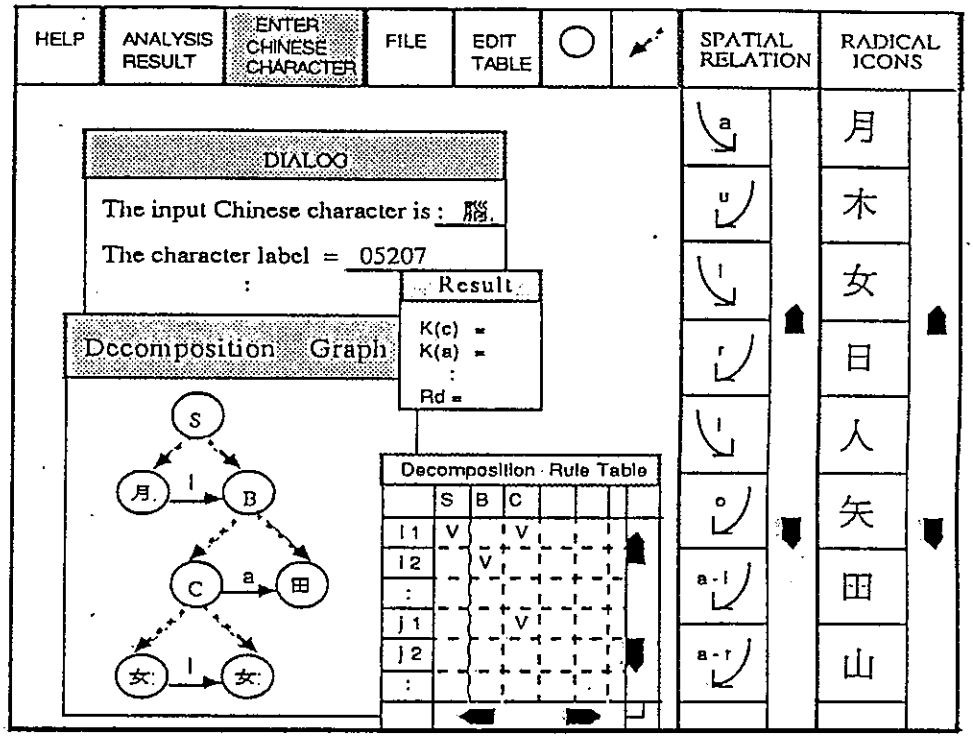


Fig. 6 An example of screen design.

As shown in Fig. 6, we can specify the subweb ω ($\omega \in V_T$) with radical icons , the subweb α ($\alpha \in V_N$) with some capital letters, and the selection of the decomposition rules from a table. The dash-line arrow that connects skeleton (tree) nodes indicates a part-of predicate, while the solid-line arrows are distinguished from each other by the spatial relation assigned on it respectively. Thus, through the visual programming approach, an RRCI designer can easily implement his idea in visual and in a manner of what-he-sketch-is-what-he-get.

VI. CONCLUDING REMARKS

In this paper, a theoretic basis for the evaluation of reduced-radical Chinese input methods or a simulator is presented. The virtual language of the simulator is

through a visual programming approach to define a designer's input method in terms of decomposing Chinese characters into decomposition graphs which generate derivation diagrams. A parser converts such derivation diagrams into a set of web production rules which are derived from two types of subgraphs of the derivation diagram, the skeleton and the section. The values of each evaluation parameter are performed based upon the number of production rules and their difficulty in the decomposition process required in the input method. The number of production rules determines the decomposition steps required and the number of external code symbols determines the number of depressed keys required.

In the Chinese phonetic input methods, each derivation diagram has one level (depth = 1) skeleton tree structure and considerable few terminal section nodes seem superior to RRCI's in the consideration of speed. However, current phonetic input methods have a common problem of high collision rate that hurts their superiority. Thus the issue in the design of Chinese phonetic input methods is in the elimination of ambiguity due to the collision rather than in the respect of speed [7].

More parameters may be added to the simulator for other use. For example, a keyboard design requires not only the information of the appearence frequency of every external code symbols, but also that of their appearence order and spatial relationship. This information can be deduced directly from the computation of terminal sections in sequence.

With the advent of morden personal computer technology, it is not difficult to implement a visual programming environment described in this paper. Though programming in visual is user-friendly but still it takes time to define huge amount of derivation diagrams manually. Thus an automatic acquisition of the derivation diagrams will be helpful. However, such a method shall be able to recognize every decomposed patterns as well as their spatial relations. For a detail discussion of the method, interested readers are encouraged to consult [10] .

REFERENCES

[1] C.K.Chen and R.W.Gong, " Evaluation of Chinese input methods," Computer Processing of Chinese & Oriental Languages, 1984, 1, P.236.

[2] ——— , " A guide for use of Chinese computers," (in Chinese) distributed by Institute for Information Industry, Nov.1983.

[3] K.S.Fu, " Syntactic pattern recognition and applications," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[4] J.M.Brayer and K.S.Fu, " Some multidimensional grammar inference methods," in Pattern Recognition and Artificial Intelligence, Academia Press, New York, 1976.

[5] L.A.Zadeh, " Fuzzy sets, " Inf. and Control, 1965, 8, P.338.

[6] S.K.Chang, " Icon Semantics - a formal approach to icon system design," proceedings of International Conference on Chinese Computing, Singapore, 1986, P.329.

[7] K.Y.Cheng and F.K.Yu, " On disambiguous Chinese phonetic input," proceedings of International Conference on Chinese and Oriental Language Computing, Chicago, 1987, P.2.

[8] K.Y.Cheng, et. al., "VIPS: A visual programming synthesizer", IEEE Workshop on Visual Languages, 1986, P.92.

[9] K.Y.Cheng, et. al., "An Extended Visual Programming Synthesizer for Computer Aided Instruction applications" , IEEE Workshop on Visual Languages, 1987

[10] G.T.Cheng and K.Y.Cheng, " Evaluation of radical Chinese input methods through recognition and description of characters in dot matrix," Technical Report, Inst. of Information Science, Academia Sinica, 1988.

[11] J. F. Sowa, "Conceptual Structures," Addison-Wesley Publishing Company.