

Terminal을 이용한 양식 편집에 관한 연구

이 철수 박사, 이 장희, 이 용무
한국데이터통신(주) 정보통신연구소 OA개발연구부

The Study of Form Editing Using the Dumb Terminal

Ph. D. Lee, Chul Soo, Lee, Jang Hee, Lee, Yong Moo

Data Communications Corp. of Korea

1. 개요
2. 그레이프 편집 기능의 구현
 - 가. 그레이프 패턴
 - 나. 그레이프 table 의 구조
 - 다. 그레이프 입력 및 삭제
3. 자료편집의 구현
 - 가. 자료 table 의 구조
 - 나. 자료 입력 및 삭제
4. 결론

1. 개요

각종 보고서와 양식을 비롯한 문서를 computer로 처리하는데는 문서의 편집 기능과 더불어 양식 자체를 편집하는 기능이 필수 불가결하다. 현재 이러한 사무업무의 처리를 위하여 word processor가 일부 이용되고는 있으나 현재의 word processor 또는 host 와의 통신방법(protocol)이나 부호개의 상이로 문서 전송에 제한이 있고, 종래의 수동 타자기의 비해 가격이 비싸 일반화하기에는 아직 실용성이 엄마하겠다. 따라서 본 연구는 현재 일반적으로 사용되고 있는 dumb terminal을 이용하여 그레이프 편집과 문자편집을 동시에 구현하는 방법을 제안하고자 한다.

일반적으로 dumb terminal은 가격이 저렴하여 computer의 단말기로 널리 보급되어 있고 그 자체에 양문 부호(code)와 함께 한글 부호 및 일부 semi-graphic 부호가 내장되어 있어서 이를 semi-graphic 부호를 사용하여 수직선과 수평선으로 구성된 문서 양식의 표현이나 초보적인 도표 및 그림표시에 유용하게 이용할 수가 있다.

여기서는 이러한 점에 착안하여 일반 문자입력 모드와 그레이프 문자입력모드를 설정하고 모드의 번환

만으로 손쉽게 그레이프 편집이나 문서의 편집이 가능한 방법을 제안하였다.

2. 그레이프 편집 기능의 구현

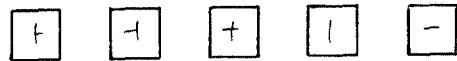
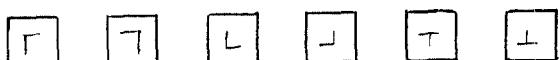
각종수치를 통계 table이나 글로 표현하는 것보다 그림이나 도표로 표시하는 것이 문서의 이해에 도움을 줄 수 있기 때문에 그레이프 편집은 OA system 구현에 필수 불가결한 요소이다.

그레이프의 구현에는 화면의 해상도(resolution)가 높아야 정확한 그림이나 도표의 표현이 가능하기 때문에 일반적으로 그레이프 단말기나 PC를 이용하여 다양한 그레이프 기능을 구현한다. 또 그레이프 부호는 현재 NAPLPS(The North American Presentation Level Protocols)에서 제정한 보완부호(mosaic) 부호가 사용되고 있다.

그러나 현실적으로 그레이프 단말기는 그 가이기 때문에 여기서는 일반 단말기에 내장되어 있는 semi-graphic을 이용하여 수평선과 수직선으로 구성된 문서양식의 제작에 활용하여 이는 모든 문서 양식이 90% 이상을 절유하고 있다.

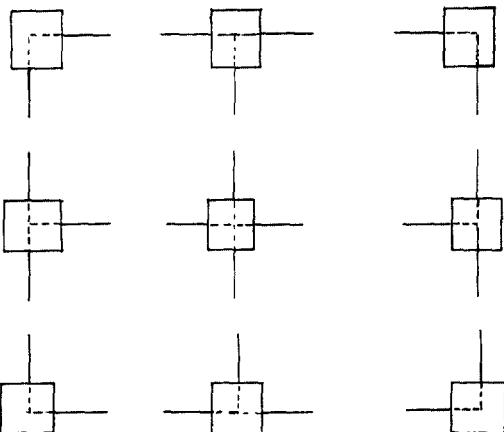
가. 그레이프 패턴(pattern)

수평선과 수직선으로 이루어진 양식을 제작하는 데는 아래에 일기한 패턴의 상호 연결로 가능하고 1개의 그레이프 패턴은 1 byte로 표시할 수 있어 확장과 1:1로 대응된다.



(또 7 패턴종류)

또 1에 일거울 그래픽 패턴으로 아래의 그림과 같은 선의 연결을 구현할 수 있다.



(그림 1 그래픽 패턴의 연결표현)

위와같이 다양한 선의 연결을 위해서는 현재 위치의 진행방향과 그 주위의 그래픽 유무를 판정하여 적당한 그래픽 패턴(pattern)을 선택해야 한다. 그래픽 패턴선택을 위해 그림 2와 같이 그래픽 판정도를 이용하여 symbol table에서 원하는 그래픽 패턴을 선택한다.

	A	
B	현재 커서	D
	C	

(그림2 그래픽 판정도)

** 각 자리에 그래픽이 존재하면 각각

'1'의 값을 부여하고 없으면 '0'값을 부여한다.

만일 현재 커서위치에서 네 방향중 'A'와 'B'에 그래픽이 있고 'C'와 'D'에는 선이 있으며 커서의 진행방향이 우측 혹은 아래로 하고싶다면 선의 연결을 위해서는 그래픽 패턴이 필요하게되고 이것이 갖는 symbol 수는 '0011'이 된다.(수의 조합은 '상자하우'의 방법)

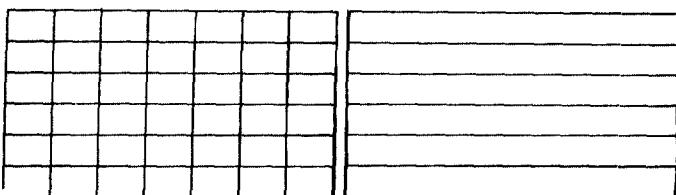
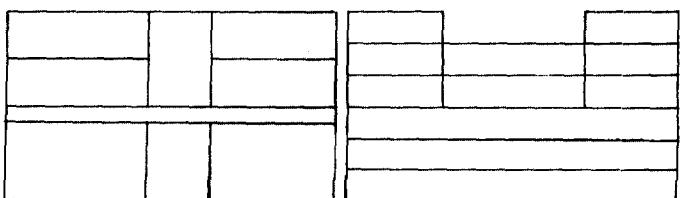
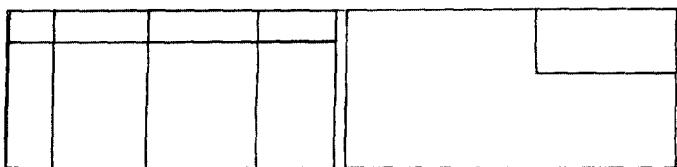
즉 와 '0011'을 1:1로 대응 하게 symbol table을 작성한다.

아래의 표 2는 선택할 수 있는 모든 그래픽패턴을 symbol table로 표현한 것이다.

순서 제공	1	2	3	4	5	6	7	8	9	a	b
symbol 주의 조합	Γ	Τ	L	Ι	Τ	Ι	Τ	Τ	+	-	

(표 2 그래픽 symbol table)

이와같은 방법으로 구현 가능한 그림의 형태는 일반문서에 쓰이는 각종 양식이며 아래그림 3과 같이 다양하게 표현된다.



(그림3 각종 문서양식)

나. 그래픽 table의 구조

표 1을 이용하여 확장에 다양한 사각형의 조합을 그래픽형태로 표현할때 작성되어져야 하는 그래픽table의 구현방법은 아래와 같다.

1) 그래픽 table의 크기

가로 88, 세로 88 즉 사각형의 전체크기는 88 line, 88 column 으로 한정한다.

2) 그래픽 table의 초기화

Table의 각 cell 을 '0'으로 초기화한다.

```
for i --- 1 to 88
```

```
for j --- 1 to 88
```

```

grp-tab[i][j]=0 ; /* grp-tab은 graphic table 8/
    end for
end for

```

4. 그레픽 입력 및 삭제

1) 그레픽 입력

현재 커지의 진행방향과 현재 커지위치에서 상하좌우의 그레픽 유무를 판정하여 표 2에서 해당하는 code 값을 그레픽 table에 입력시킨다. 해당 코드를 선택하여 그레픽 table에 입력하는 algorithm은 아래와 같다.

algorithm)

sym-tab --- value of symbolic table

/* 2차원 array */

line --- cur-line

/* 커지의 현재 line */

col --- cur-col

/* 커지의 현재 column */

code-tab --- value of code

/* 1차원 array */

array [0] --- grp-tab[line-1][col] ;

/* 현커지의 외측상백 */

array [1] --- grp-tab[line][cd-1] ;

/* 현커지의 좌측상백 */

array [2] --- grp-tab[line+1][col] ;

/* 현커지의 아래상백 */

array [3] --- grp-tab[line][col+1] ;

/* 현커지의 우측상백 */

/* symbolic table 이어서 해당 code 값 부여 */

for i --- 1 to 11

if sym-tab[i]=array

then grp-tab[line][col]---code-tab[i]

else continue ;

end if

end for

상기와 같은 algorithm으로 학연이 그림4를 그리면 그림 5와 같은 그레픽 table의 작성된다.

	10	20	30
*			
*			

(그림4 학연작성)

	10	20	30	40	50	60	70	80	90	100
1	a	a	a	a	a	a	a	a	a	a
2	b	b	b	b	b	b	b	b	b	b
3	b	b	b	b	b	b	b	b	b	b
4	b	b	b	b	b	b	b	b	b	b
5	b	b	b	b	b	b	b	b	b	b
6	b	b	b	b	b	b	b	b	b	b
7	b	b	b	b	b	b	b	b	b	b
8	b	b	b	b	b	b	b	b	b	b
9	b	b	b	b	b	b	b	b	b	b
10	b	b	b	b	b	b	b	b	b	b
11	b	b	b	b	b	b	b	b	b	b
12	b	b	b	b	b	b	b	b	b	b
13	b	b	b	b	b	b	b	b	b	b
14	b	b	b	b	b	b	b	b	b	b
15	b	b	b	b	b	b	b	b	b	b
16	b	b	b	b	b	b	b	b	b	b
17	b	b	b	b	b	b	b	b	b	b
18	b	b	b	b	b	b	b	b	b	b
19	b	b	b	b	b	b	b	b	b	b
20	b	b	b	b	b	b	b	b	b	b

→ C. 7.1.1

(그림5 그레픽 table)

2) 그레픽 삭제

그레픽 삭제는 이미 그려진 그레픽을 지우는 기능으로 커지의 움직임에 따라 그 위치의 그레픽을 학연과 그레픽 table의 해당 그레픽을 삭제하는 과정이다.

이 경우 선과 선이 교차되는 부분의 처리는 표 3, 표 4와 같이 미리 준비된 pattern table의 위치가 교차 부분을 '-' 혹은 '1' 형태의 그레픽 패턴으로 미리 이주거나 혹은 날려두기도 한다.

(가) 커지를 상하로 움직이면서 삭제할 경우

현재 커지위치의 좌측과 우측에 대한 그레픽 패턴을 표 3에서 찾아 flag이 1로 'on' 되어 있으면 '-' 그레픽 패턴을 그려주고 flag이 0로 off 되어 있으면 현재 그레픽만 삭제한다.

row	1	2	3	4	5	6	7	8	9	10	11	우측
col	1	2	3	4	5	6	7	8	9	10	11	
1	/											
2	/	/	/	/	/	/	/	/	/	/		
3	/	/	/	/	/	/	/	/	/			
4												
5	/											
6	/											
7	/											
8												
9	/											
10	/											
11												
12												

(표 3 가로선(=) 그리는 pattern table)

(4) 커지를 좌우로 움직이면서 삭제할 경우
현재 커지위치의 위와 아래의 그림
그림을 표 4에서 찾아 flag 이 1로 on 되어 있으면
그 그림을 그려주고 flag 이 0로 off 되어
있으면 현재 그림만 삭제한다.

row	1	2	3	4	5	6	7	8	9	10	11	아래
col	1	2	3	4	5	6	7	8	9	10	11	
T 1	/	/	/	/	/	/	/	/	/	/	/	
T 2	/	/	/	/	/	/	/	/	/	/	/	
L 3												
J 4												
T 5	/	/	/	/	/	/	/	/	/	/	/	
L 6												
H 7	/	/	/	/	/	/	/	/	/	/	/	
I 8	/	/	/	/	/	/	/	/	/	/	/	
+ 9	/	/	/	/	/	/	/	/	/	/	/	
- 10												
1 11	/	/	/	/	/	/	/	/	/	/	/	
위												

(표 4 세로선('1')을 그리는 pattern table)

3. 자료편집의 구현

자료의 편집에는 일반적으로 line 단위로 문자를 입력하거나 삭제하는 line editing 방식과 화면상에서 커지를 상하좌우로 이동시켜 문자이나 문자를 삭제하거나 수정 삽입하는 화면단위의 screen editing 방식의 편집을 구현할 수 있다.

여기서는 ARS 사용자가 컴퓨터에 전문가임을 강조하여 몇 가지 명령어로 간단히 자료를 편집할 수 있는 line editing 방식의 편집을 활용했다.

가. 자료 table 의 구조

그래픽 table 이 생성될 때 자료 table 이 함께 생성되며 그 자료 table 은 그래픽 table 이에서 지정해놓은 자료 입력위치를 갖게되며 자료 내용은 입력된 자료 내용크기 만큼 파일을 할당해준다.

자료 입력위치 결정은 그래픽 table 이 완성되면 그것이 의해 내용입력이 시작될 위치를 찾아 아래와 같이 만들 이전다.

예를 들어 그림 5와 같은 그래픽 table 을 작성하면 아래그림 6과 같은 자료 table 이 자동적으로 생성된다.

기초 입력 위치			내용 입력
col	row	size	
2	2	4	X
2	3	9	X
2	4	9	X
.	.	.	
.	.	.	
.	.	.	
11	2	F	X
11	3	F	X
.	.	.	
.	.	.	
.	.	.	

(그림6 자료 table 의 구조)

자료와 입력위치의 구성은 4 byte 로 되어 있으며 각각의 정의는 다음과 같다.

- .col — 커지의 column 위치 (ASCII 값)
- .row — 커지의 row 위치 ("")
- .size — maximum 입력자리수
- .종류 — data 의 종류

data 종류

구 분	그정 data	비그정 data
normal	F	X
가로 2배	G	y
세로 2배	H	z
가로 세로 2배	I	
가로 세로 1.5배	J	:

* 그정 data : 내용을 전부 삭제할 시 삭제되지 않는 부분

(표5 data 의 종류)

나. 문자입력 및 삭제

자료입력은 화면으로 부터 입력된 내용을 이미 작성된 자료 입력위치에 덮불인다. 내용입력은 입력을 받을 때마다 내용크기만큼의 메모리(memory)를 할당받아 현재 커지의 위치를 column 과 row로 계산한 후 해당 자료입력 위치에 덮불인다.

자료입력 algorithm은 아래와 같다.

(자료입력 algorithm)
repeat
line --- cur-line ;

```

/* 현재 커지 line 위치 */
col --- cur-col ;
/* 현재 커지 column 위치 */
getdata(tmp) ;
    /* 한 line 입력을 받아 의사 array에
       담는다 */
/* 입력된 문자를 한글 automata를 통하여 계산
   하여 그 숫자를 total에 저장 */
automata(tmp, total) ;
grp-find(xline, gcol) ; /* graphic table에서 현재
   커지 위치인 column과 row를 찾아 저장 */
space-cnt --- col - xline : /* 입력 초기 위치와
   실제 입력 위치와의 간격을 계산 */
x-tmp --- malloc(total+space-cnt) ;
    /* 입력을 위한 memory 할당 */
for i --- 1 to space-cnt /* x-tmp space 갯수 만큼
   x-tmp++ --- ' ' ;   space를 채움 */
/* 의사 file에 담겨있는 내용을 x-tmp와 합하여
   text-table을 만들 */
add(x-tmp, tmp, xline, ycol, text-table) ;
until 입력 끝

```

자료입력을 마친후의 자료 table의 구조는 아래
그림 7과 같다.

2 2 9 X		M	\	b/A	J	lo
2 3 9 X	m a	m a G E R				
2 4 9 X	p r o	J E C T				
.						
.						
.						
"/ 2 8 X M \ b w i	E h G	N				
"/ 3 8 X						
:						
:						

(그림7 자료 table 구조도)

4. 결 론

본 연구의 목적은 dumb terminal을 이용하여 사무에 필요한 각종 양식을 제작하고 그를 이용하여 문서를 작성하고자 하는데 있다. 따라서 그래픽과 문자의 입력 및 수정편집이 동시에 이루어질 수 있도록 하는 것이 특징이라 하겠다.

본 연구에서 사용한 자료입력 방식은 사용자 가

computer에 전문 가하는 편을 강조하여 menu-driven 방식의 line editing을 1차적으로 제공하였으나 사용자의 속도도 향상에 따라 screen editing 방식으로 발전되어야 한다.

본 논문에서는 dumb terminal을 이용하여 초보적인 graphic 편집 시스템을 구현하였으나 dumb terminal의 semi-graphic 부호로는 각종 서류 작성에 요구되는 다양한 그림형태를 표현하기에는 한계가 있다. 또 향후 pc가 단말기와 stand alone格局으로 사용될 것으로 예상되므로 pc의 bit-map display 방법을 이용하여 다양한 graphic 표현을 구현하는 것이 바람직하다.

효율적인 사무자동화를 추구하기 위해서는 graphic 편집 이외에 전자우편, word processing, spreadsheet, DBMS 등이 서로 통합되어야 하고 WP, PC, computer 등이 LAN을 통해 연결된 total 시스템이 되어야 한다.

이미한 total 시스템에서 dumb terminal이 유용한 OA 도구가 되기 위해서는 다음과 같은 문제가 해결되어야 한다.

- 가. 각종 단말기와 computer 간의 한글부호 및 graphic 부호 계와 처리방법에 관한 표준화 내지는 emulator 방법 개발
- 나. File format의 호환성 제공 방법의 강구
- 다. Data level의 공유를 위한 방안 연구
- 라. Man-machine interface 방안 연구

(참 고 문 헌)

1. 이철수 UNIX 시스템과 C 언어 1985. 5
2. Western electronic UNIX system 1982
3. S.R Bourne The UNIX system 1982
4. Brian W.kernighan.Dennis M.Ritdie The C programming language 1978
5. 이철수 사무자동화 1984. 1
6. David Barcomb Office automation 1981
7. 과학기술처 사무자동화 시스템 개발에 관한 연구 1983
8. OSI phoenix3 terminal user's manual
9. QNIX hq300 terminal user's manual