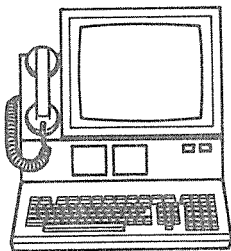


吳 吉 祿  
韓國電子通信研究所  
주전산기개발본부장 / 工博

# 지능형 에디터 개발을 위한 기본 에디터 개발



## 1. 서론

지능형 에디터의 개발의도는 인간이 컴퓨터를 사용하여 문서나 프로그램을 작성할 경우, 그 작업을 원활하게 하고 능률적인 작업 환경을 제공하는 데 있다. 그에 대한 최근의 중심기술은 지식 습득, 지식관리 시스템, 창문관리 시스템, 지식기반 에디터 시스템 [5, 12] 등이 있다. 문서 및 프로그램 작성환경에 일련의 지능 제공으로 넓게는 자연어처리 기술과 문서 작성 지식 및 프로그래머의 지식을 기존 에디터에 적용한 응용 시스템의 개념을 가진다.

본 연구는 일차 연도에 지능형 에디터를 정의하고 그 그릇이 될 기본 에디터를 개발하는 것이다. 개발의 기본원칙 [16]은 일반 터미널에 이른 바 「보는 것이 곧 얻는 것이다.」라는 개념을 가지고 호환성 및 우리의 문서 작성환경에 고유성을 가지는 것이다.

## 2. 지능형 에디터의 구조

### 가. 구조

지능형 에디터는 사용자에게 문서작성과 프로그램 작성에 편리한 환경을 제공하는「사용자 정의가능 지식기반 에디터」로서 개념적으로는 그림 1 과 같은 지능을 가진 에디터로서 다음의 특성을 제공한다.

- (1) 기존 에디터의 기능 : 다중화면, 다중 버퍼 및 화면에서 대화 형식에 의한 문서구성 가능
- (2) 문서작성 및 프로그램 작성환경 제공 기능 : 철자오류수정, 단어 번역, 형식 문서 / 프로그램 작성의 지적화 가능, 지적인 도움
- (3) 사용자가 정의한 지식 제공 기능 : 어떤 특수분야의 전문가인 사용자가 자신의 지식

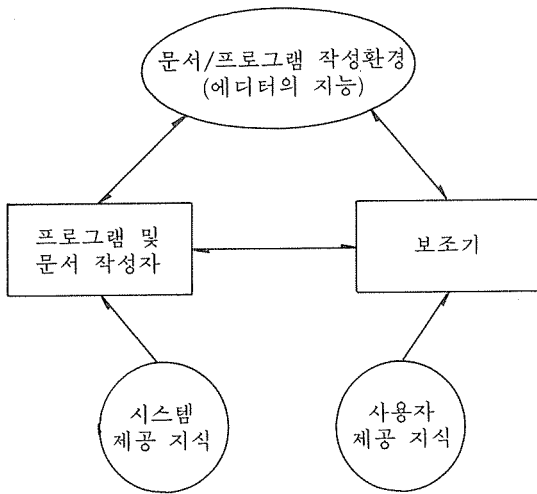


그림 1. 문서 및 프로그램 작성 보조기

을 시스템에 제공, 지식의 공유 가능

- (4) 사용자가 특정 분야의 전문적 지식의 습득이 가능하여 응용분야의 기술개발이 가능하다. 구성요소

지능문제를 지원하기 위한 기본적인 접근방법으로서 인간전문가의 지식을 이용하는 데, 프로그램 및 문서작성을 위한 지능부분은 실시간 모드에서 기존의 방법에 의한다. 지능의 사용자 정의란 사용자는 에디터를 이용하면서 자신의 분야에 편리한 사용법을 추적함을 의미한다. 지능형 에디터 시스템은 그림 2와 같고 각 구성요소를 설명한다.

- (1) 기본 에디터

이 부분은 문서 에디터를 말하며 EMACS(9)

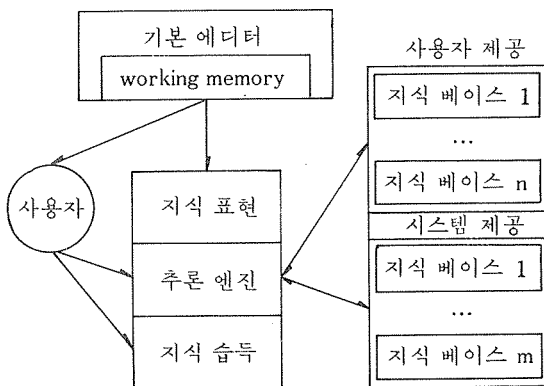


그림 2. 지능형 에디터 시스템 구조

와 같은 기존하는 에디터를 사용할 수도 있지만 우리는 지능형 에디터라는 관점에서 기본 에디터의 기능을 정의할 것이며 다중화면 및 화면상 문서구성 기능, 사람에 친숙한 도움말기능 등을 중점적으로 고려할 것이다.

- (2) 지식표현

지식표현은 효과적인 처리를 위한 표현과 사용하기 쉽게 연결해 주는 외부표현이 있다. 특히 사용자와 전문가 사이에 어떠한 지식기술자도 없기 때문에 외부표현방식은 마치 지식서술 언어와 같다. 그래서 이에 대한 번역기가 있어야 할 것이다.

- (3) 지식 습득

사용자가 정의할 수 있도록 하려면 이 부분은 매우 중요하다. 이는 지식습득을 위해서 우수한 보조자가 되어야 한다. 즉 기존 지식기술자의 역할이 이 모듈에 집약되어야 하며, 또한 수집된 지식의 일치성이나 완전성 등을 검사할 수도 있어야 한다.

- (4) 추론 엔진

이 부분은 어떤 전문가 시스템 개발 도구의 전통적인 추론 엔진일 수도 있다. 그리고 문서나 프로그램 처리를 위한 특수목적의 추론장치로 만들 수 있다.

- (5) 시스템 제공 지식

수집된 많은 종류의 지식, 즉 문서의 형식 및 적합한 내용에 관한 지식, LISP 프로그램의 문법오류 검사를 위한 지식, 지능적인 도움말에 관한 지식 등이 이 분야 전문가로부터 수집되어 지능형 에디터에 축적된다.

- (6) 사용자 정의가능한 지식

사용자가 사용할 목적에 따라서 필요한 정의를 하거나 사용 목적의 특수성에 관한 지식을 사용자가 필요에 따라서 지식베이스를 만들거나 지식 습득 기능을 조성한다.

### 3. 기본 에이더

어떤 소프트웨어를 설계하는 과정은 일반적으로 개개인의 주관적인 판단과 개인적인 취향에 따라 많이 유동적이다. 비록 이러한 주관적인 요소가 설계 과정에서 완전히 배제되더라도 설계 기준이 명백하게 선언되고 전체 시스템 설

계 및 구현 과정을 통하여 이러한 기준을 지켜나가지 않는다면 더 좋은 설계를 기대할 수 없을 것이다. 본 연구팀이 지능형 에디터 'IE'의 기본이 되는 기본 에디터를 설계하는 데 있어서 가장 큰 비중을 둔 설계 기준은 사용자의 입장에서 문서 작성기를 사용할 때 배우고, 기억하고, 사용하기 쉬운 관점에서 설계 구현하였다. 또한 시스템에 새로운 기능을 확장하거나 사용자 개개인이 자신의 환경을 구축하여 좀 더 효율적으로 문서를 작성하는 데 초점을 두었다. UNIX 하에서 개발된 에디터 시스템을 [4, 9, 14] 비교 검토한 결과 여러 측면에서 부족한 점을 발견할 수 있었다. 그래서 사용자 측면에서 이러한 기준을 가진 기본 에디터를 설계하였으며 상세한 데이터 구조는 부록에 첨가되어 있다. [18]

#### 가. 배우기 쉬운 에디터

에디터 프로그램은 숙달되게 잘 사용할 수 있는 사용자 뿐 아니라 초보적인 사용자같은 여러 계층의 사용자를 가지기 때문에 특히 사용자 인터페이스는 새로운 사용자가 빨리 배울 수 있고 다른 사용자들도 쉽게 복잡한 기능을 사용할 수 있게 설계되어야 한다. 지능형 에디터의 기본 에디터는 여러 계층의 사용자들이 명령어들을 배우기 쉽게 계층적인 명령어구조를 가진다. 초보적인 사용자는 첫번째 계층의 명령어만 배우므로써 기본적인 에디터의 명령을 사용하여 문서를 작성하고, 다른 계층의 명령어들은 사용자의 또 다른 요구나 사용자가 보다 에디터에 익숙하게 될 때 배울 수 있게 되어 있다. 기본 에디터의 명령어는 그 명령어의 사용 빈도와 난이도에 따라 표 1 과 같은 네개의 계층으로 구성하였다.

표 1. 계층적인 명령어 구조

계 층	명령어	설 명
네번째 층	추 가	Escape 및 Escape Control 키
세번째 층	기능 확장	Control (B, F, T, V, W, X) 키
두번째 층	단위 확장	Escape 키
첫번째 층	기 본	Control 키 및 커서이동 키

이러한 네개의 층을 가지는 명령어 구성은 지능형 에디터 사용자 안내서 [14]에서 명확하게

나타나 있다. 이렇게 명령어를 계층적으로 구분함으로써 초보적인 사용자라도 쉽게 사용자 메뉴얼이 없어도 사용할 수 있게 되어 있다.

#### 나. 기억하기 쉬운 에디터

일치성을 가지는 명령어 구조와 mnemonic 한 명령어 이름이 에디터 명령어를 기억하기 쉽게 하여 준다. 에디터가 특수한 경우의 명령어 이름을 가능한 한 적게 가지고 또한 비슷한 기능의 명령어가 비슷한 유형의 명령어 이름을 가질 때 일치성을 가지는 명령어 구조를 갖게 된다. 이러한 조건이 사용자가 명령어를 배우고 기억하는데 쉽게 해주며 사용자 메뉴얼의 필요성도 없애 준다.

지능형 에디터가 가지는 명령어의 일치성의 예를 들면, 모든 명령어들은 문서의 단위에 대한 갯수를 지정하는 인수를 가질 수 있는 데 이들은 정수 값으로서 그 숫자의 배 만큼 명령을 수행하여 준다. 이러한 의미에서 인수가 0 은 어떤 동작도 하지 않는 의미이며 -1 인수는 기능의 음수적인 효과를 기대하는 것으로 단어 단위로 커서를 순방향으로 이동하는 명령에 -1 인수를 사용하면 단어 단위로 커서를 역방향으로 이동시키는 것을 의미한다.

사용자가 기억하기 쉬운 명령어를 위해 명령어 이름은 가능한 한 mnemonic 하여야 한다. 그러나 모든 명령어를 mnemonic하게 설계하기는 명령어들간의 중복되는 이름들 때문에 사실상 불가능하다. 그래서 지능형 에디터의 명령어 이름을 설계하는 과정에서 가능한 한 그 명령어의 의미에 맞는 두문자를 제어 키에 지정하여 명령어를 쉽게 기억할 수 있게 하였다.

#### 다. 사용하기 쉬운 에디터

사용자에게 명령어를 사용하기 쉽게 하기 위해서는 먼저 그 기능적인 면에서 단순하면서도 강력한 기능을 제공하여야 한다. 이러한 기능의 단순성과 강력성 사이의 적절한 비중을 유지하는 것은 사용자와의 인터페이스를 설계하는 데 있어서 가장 어려운 일이라 할 수 있다. 명령어들은 사용자가 배우고 기억하기 쉽게 단순성을 가져야 하지만, 한편 예외적이거나 인위적인 노력없이 대부분의 업무를 수행할 수 있게 명령어의 기능이 충분히 강력하여야 한다. 앞에서 서술한 계층적인 명령어 구조에서 첫번째 층

의 명령어는 기능의 강력성보다는 기능적인 단순성이 우선적으로 고려되었고 위의 층으로 갈수록 특별한 기능을 수행하기 위한 강력한 명령이 추가되었다. 다음에서 지능형 에디터의 기본 에디터 시스템이 제공하는 강력한 기능을 보여 준다.

#### (1) 다중 버퍼 기능

다중 버퍼 명령어는 에디터 시스템을 빠져 나가지 않고 동시에 여러개의 화일을 편집할 때 유용한 기능으로서 주로 한 화일의 특정한 내용을 다른 화일의 임의의 위치로 이동하거나 혹은 복사할 때 사용된다. 이러한 다중 버퍼 기능은 에디터 내에서 mail 기능이나 에디터 시스템의 사용자, 메뉴얼 혹은 디렉토리 에디터를 위해 사용될 수 있다.

#### (2) 다중 창문 기능

다중 창문 기능은 다른 기능들을 단순화시킨 강력한 기능으로서 한 화일에서 특정한 문서의 부분을 다른 화일로 이동시키는 명령을 화일 내에서 블록 이동처럼 쉽게 할 수 있는 기능을 제공한다. 다중 창문은 다중 버퍼 기능이 한 순간에 한 화면이 한개의 버퍼만 보여주는 데 비해 두개 혹은 여러개의 버퍼를 한 화면에 여러개의 창문으로 분리되어 사용자가 한 화면상에서 여러개의 문서를 동시에 보면서 편집할 수 있는 기능을 제공하여 준다.

#### (3) 셸 창문 기능

에디터를 벗어나지 않고서 UNIX의 shell 명령을 수행하고 그 수행 결과를 에디터 내에 특정한 버퍼로 저장하여 그 내용을 이용할 수 있게 shell 버퍼 기능을 제공한다. 이러한 기능은 사용자 환경을 종합하여 프로그램이나 문서작성의 효율을 높이기 위해 제공되는데 에디터 내에서 shell 명령을 수행하면 shell 버퍼를 만들고 버퍼로부터의 입력을 받아 shell을 수행시키고 그 결과를 다시 버퍼로 출력하여 준다.

#### (4) 화면상에서 문서 구성 기능

화면상에서 작성한 문서가 프린터를 통하여 출력될 때 가능한 한 같은 형태를 유지하기 위해 지능형 에디터에서는 화면상에서 문서를 구성, 재구성, 인쇄효과 지정에 대한 기능을 제공하는데, 이들 기능은 화면상에서 그 기능에 대한 효과가 바로 반영되지는 않으나 화면상에서

구성된 어떤 행이나 페이지가 출력시에 한개의 행 혹은 페이지로서 출력되는 효과를 제공하여 사용자는 전체 문서의 형태를 짐작할 수 있다. 또한 문서의 형태를 제어하는 여러개의 기능자를 제공하여 화면상에서 사용자가 작성할 문서의 윤곽을 대략 짐작할 수 있게 하는 기능을 제공한다. 인쇄 효과에 대한 명령은 일반 터미널의 제한된 특성을 고려하여 화면상에서는 그 명령에 대한 숨은 코드를 정의하여 화면에 보여준 다음 출력시에 프린터에 적절한 코드로 변환하여 출력함으로써 입출력 장치인 터미널과 프린터에 가능한 한 독립적으로 사용될 수 있게 설계하였다. 문서를 편집한 후 재구성할 때에 발생하는 데이터의 과도한 이동을 방지하기 위해 기본 에디터는 일반적으로 에디터 시스템이 가지는 데이터 구조 외에 Unit라는 데이터 구조를 가짐으로서 재구성시에 행의 포인터만 변경함으로써 데이터 이동을 방지하여 시스템의 성능을 향상시켰다.

#### (5) 확장된 문서 단위에서 명령어 수행 기능

문서의 최소 단위인 문자 외에 단어, 문장, 문단, 화면 그리고 페이지 단위로 커서를 이동하거나 삭제 혹은 대치하는 기능을 제공하여 사용자의 요구에 따라 좀 더 효과적인 문서 작성을 기할 수 있다. 이러한 기능은 초보적인 사용자에게는 사용하기 어렵지만 점차적으로 에디터의 사용에 익숙하여지면 효과적인 문서작성을 위해 필수적인 기능이 된다.

#### (6) 한글/영문 변환 기능

한글과 영문을 같이 사용하기 때문에 자주 발생하는 문제로서 사용자가 현재의 모드가 한글인지 혹은 영문인지 정확하게 알지 못하고 입력하는 경우가 많이 발생한다. 이러한 경우에 많은 양의 문서가 잘못된 모드에서 입력하였기 때문에 새로이 입력하여야 하는 경우가 발생한다. 이러한 상태를 복구하기 위해 한글 혹은 영문 모드에서 잘못 입력된 부분을 원래 사용자가 입력하기를 원하였던 문자열로 변환하여 주는 '둔초두' 기능을 제공한다.

#### (7) 화면 좌우 스크롤 기능

사용자가 작성하는 문서의 좌우 길이가 터미널의 좌우 길이보다 클 경우에 일반적으로 에디터들은 다음 행으로 그 뒷 부분을 넘겨 주는데

비해 지능형 에디터에서는 소프트웨어적으로 화면을 좌우로 스크롤하는 기능을 제공하여 화면보다 좌우로 긴 문서를 쉽게 작성하게 하는 기능을 제공한다.

(8) 에러·메시지 및 버퍼상태 출력기능

지능형 에디터에서는 정의되지 않은 명령어나 잘못된 명령어 수행시에 사용자에게 이해하기 쉬운 에러 메시지를 화면의 아래 부분에서 보여준다. 이러한 기능은 사용자에게 명령어 수행에 따른 적절한 정보를 제공하여 사용자가 쉽게 배우고 더 효과적으로 명령어를 사용할 수 있는 기능을 제공한다.

라. 지식 베이스의 활용기능

(1) 명령어 확장성

사용자 요구에 의한 추가 명령어 구현기능으로써 일반적인 매크로의 기능으로는 구현이 어렵고 또한 기능이 “한글 띄어쓰기”와 같이 지식처리가 필요한 것이라면 부적합하다. 이를 위해서 기본 루틴으로 에디터와 정보를 교환하고 그 바탕에서 사용자가 원하는 기능을 표현할 수 있는 그림 3 과 같은 일련의 서술언어가 필요하다.

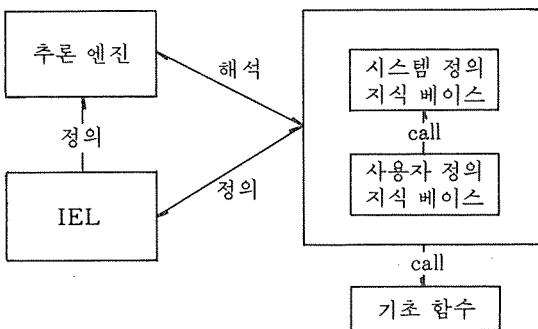


그림 3. IEL과 지식 베이스와의 관계

(2) 사용자 환경구축

사용자가 원하는 모드를 선택할 수 있게 하여 사용자의 기호대로 에디터가 동작하게 하는 기능과 특정 명령에 부여된 키를 다른 것으로 매치하는 기능이다. 이때 시스템 정의 지식 베이스는 모드선택을 위한 모드 결정 변수를 모아 놓은 것이다.

(3) 교육 기능

쉽고 능률적인 사용법 교육이 필요하다. 컴퓨

터를 이용한 교육은 인공지능의 한 연구분야로서 에디터를 그 교육대상으로 할 때 명령어를 그 사용빈도에 따라서 그리고 상황에 따라서 효율적인 명령어 또는 순서를 제시해 주는 기능이 필요하다.

(4) 계층 구조형 설계

기본 에디터 설계를 표 2 와 같은 계층구조로 했으며 일련의 서술언어를 이용해서 원하는 명령어를 손쉽게 구현할 수 있다. 계층 구조의 특징을 활용하면 모든 공통변수의 측면효과를 없앨 수 있다.

4. 결 론

지능형 에디터는 지식베이스를 가지고 사람에게 친숙한 기능을 제공하는 에디터를 만드는 관점에서 그 바탕이 될 기본 에디터를 만들어 있어서 앞에서 말한 설계의 개념과 방법론을 가지고 유닉스 환경의 일반 터미널에서 개발하였다.

표 2. 소프트웨어의 계층 구조

계층	내 용		
3	키 명령 수행 루틴		
2	공용 루틴		
1	기본 자료구조 취급 루틴	device 취급 루틴	display library
0	system call		

주기능으로써 일반적인 문서처리 기능 및 창문관리 기능과 「보는 것이 곧 얻는 것이다」 라는 개념 등이 반영되었고, 또한 자료구조 설계 및 사람에게 친숙한 명령어 설계에 있어서 한국형 에디터로서 가치를 가질 것이며 다음에 계속 될 지능형 에디터의 기본 그릇으로써 손쉽게 다룰 수 있음이 부가가치로써 의미를 가질 것이다. 앞으로의 연구로서는 지능형 에디터의 나머지 모듈 구현, 지식표현, 추론 문제와 지식 베이스 구성 방법론 등이다.

참고문헌

1. Bentley, Jon, "Programming Pearls", CACM

- May V. 28, 1985.
2. Cherry, Lorinda, "Writing Tools", IEEE Trans. on Comm. V. 30, Jan. 1982.
  3. Fairly, R., Software Engineering Concepts, McGraw Hill, 1985.
  4. W. Joy, An Introduction to Display Editing with Vi. Reference manual, Computer Science Div., Dept. Elec. Eng. Computer Science, Univ. of California, Berkeley, Feb. 1980.
  5. Jacob, R., Window Manager설계서, 1980.
  6. Macdonald N. H., Frase Law. T., Gingrich P., and Keenan S. A., The Writer's Workbench: Computer Aids for Text Analysis", IEEE Trans. on Comm., Jan. 1982.
  7. Peterson, James L., "Computer Programs for detecting and correcting Spelling Errors", CACM Dec. 1980.
  8. Peterson, James L., "A Note on Undetected typing Errors", CACM July, V. 29, 1986.
  9. R. M. Stallman, "EMACS: The extensible, customizable, self-documenting display editor", Artificial Intelligent Lab., Messachusetts Inst. of Technol., Cambridge, AI memo 519, June. 1979.
  10. R. M. Stallman, "GNU Emacs Manual", 5 th Edition, V. 18. Dec. 1986.
  11. Waters, Richard C., "The Programmer's apprentice: A session with KBEmacs". IEEE Trans on Software Engineering, 1985.
  12. Waters, Richard C., "KBEmacs: where's AI?," The AI Magazine, 1986.
  13. 한국 전자 통신 연구소, 미니 컴퓨터 설계 기술 개발 중 Man-machine Interface 개발에 관한 연구(2차년도), 최종 연구 보고서 5 NG 0012031240, 1986.
  14. 신동하, 김권양, "KOAWORD 사용자 안내서," KIET 컴퓨터 개발부, 1984.
  15. 변정용, 신동하, 임영환, "지능형 에디터 계획서," ETRI 컴퓨터 개발부 Memo, 1986.
  16. 김권양, 김상철, 임영환, "지능형 에디터 시스템 요구서," ETRI 컴퓨터 개발부 Memo, 1986.
  17. 김상철, 김권양, 변정용, 임영환, "지능형 에디터 사용자 안내서," ETRI 컴퓨터 개발부 Memo, 1986.
  18. 변정용, 김상철, 김권양, 임영환, "지능형 에디터 설계서," ETRI 컴퓨터 개발부 Memo, 1986.

