

# 연필 컴퓨터의 구현

변정용  
동국대학교 전자계산학과

## An Implementation of Pencil-like Computer

Byun, Jeongyong  
Department of Computer Science, Dongguk University  
*Byunjy@mail.dongguk.ac.kr*

### 요 약

정보사회의 주 도구인 컴퓨터에서 한글처리의 불완전함으로 인하여 한국인의 사고 체계는 부분적으로 제약을 받고 있다. 이것은 지적 노동 생산성을 저해할 뿐만 아니라 컴퓨터로 표현할 수 없는 글자는 장기적으로 사용이 뜸해져서 결국 사라지게 될 수도 있다. 본 논문은 컴퓨터가 한국인의 사고력을 높여주고 지적 노동 생산성을 극대화할 도구가 될 수 있도록 마치 연필로 글을 쓸 때처럼 자유스럽게 글자를 임의로 조합할 수 있게 하는데 역점을 가지고, 이를 위하여 훈민정음 원리를 적용하여 천지자연의 글자를 표현한다는 원리를 구현함으로써 연필 컴퓨터에 관한 한 가지 구현을 하였다.

#### 1. 서론

컴퓨터 시스템은 정보사회의 주된 도구이다. 산업 사회가 손과 발의 속도화에 주력했었다면 정보사회는 두뇌의 속도화에 노력하고 있다고 할 수 있다. 즉 전자가 육체 노동 생산성 향상을 위한 것이라면 후자는 정신노동의 생산성 향상을 위한 것이다. 또한 현재 컴퓨터는 정치, 경제, 사회, 문화 등 사회 전 분야에서 그 사용이 보편화되고 있으며 1998년 초 국내 PC 보급은 1천만 대, 국제적으로 1억대를 돌파했다.

워드프로세서를 이용한 문서 작성이 보편화되고 기계번역과 같은 자연언어처리 분야의 연구[9]가 매우 활발해 지고 있다. 이러한 노력의 결과는 컴퓨터가 인간의 정신 노동에 크게 도움을 줄 것으로 판단된다. 문서 작성에서 연필을 주로 사용하던 시대와 비교하면 입력과 편집과 출력에서 엄청난 편리성을 제공하고 있지만 한글 입력에서는 연필로 글을 쓸 때와는 반대로 제약이 많은 것이 사실이다. 무엇보다 정신 노동 생산성을 향상시키려면 우리의 사고력

을 확대시켜 주고 여러 가지 장애를 제거하는 방향으로 컴퓨터를 설계해야 한다. 특히 입력계에서 중요한 것은 한글 표현의 제약으로 우리의 사고력을 제한하거나 방해하지 않도록 하는 것이다. 그런데 현재 우리의 컴퓨터들은 '똥방각하'와 '시공뉴'에서 똥과 공을 표현하지 못한다. 이것은 두 가지 문제를 낳는다. 첫째, 표현할 문자의 제한으로 사고력에 방해로 끼친다. 둘째, 정보사회가 보편화되고 문서들이 주로 컴퓨터에 저장되어 있다면 컴퓨터에 표현되지 못한 글자들은 결국 사라지거나 사용빈도가 점차 줄어들게 될 것이다. 본 논문은 컴퓨터가 한국인의 정신 노동 생산성을 극대화하기 위하여 연필로 글을 쓸 때처럼 사고력을 방해하지 않도록 훈민정음 원리를 도입하여 현재 컴퓨터의 기능을 개선하는 방안을 제안하고 이를 구현하여 보이고자 한다.

#### 2. 연필 컴퓨터의 정의

##### 2.1 사고력과 표현

우리가 사용하고 있는 컴퓨터는 한글의 입력과 표현에서 연필로 쓸 때와는 많은 차이가 난다. 연필로 쓸 때는 임의로 한글 음절을 구성해 볼 수 있다. 그러나 컴퓨터에서는 컴퓨터가 정의해 둔 글자에 한해서만 입력과 표현이 가능하다. 컴퓨터가 인간의 두뇌 활동을 돕는다는 점에서 이것은 중요한 논의의 대상이 될 수 있다. 예를 들어서 '똥방각하'를 표현할 수 없다면 '또옴방각하'로 표현하거나 다른 어휘를 생각해 봐야 한다. '똥'을 다른 글자로 표현한다면 여러 가지 의미에서 본래의 의도와 의미로 표현하기 어려울 것이다. 우리의 표현은 이미 그 어떤 뜻을 함의하고 있는 어휘를 사용하였을 때 정확하게 된다. 그리고 어휘의 선택이 제한되어 있다면 사고 체계에 혼란이 올 것이다.

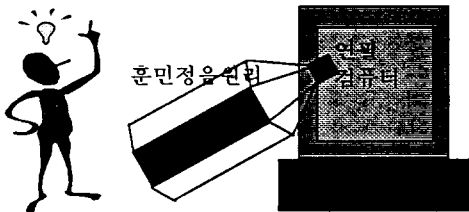


그림 1. 연필 컴퓨터의 개념도

## 2.2 연필과 자판의 물리적 특성

인터페이스 측면에서도 연필과 자판은 많이 다르다. 연필은 심이 있어서 쓰다가 심이 닳아 없어지면 연필을 깎거나 샤프 연필의 경우 심을 길게 하기 위하여 뒤 꼭지를 눌러야 한다. 연필은 한 손으로 잡고 쓰며 글을 쓸 때는 종이를 응시해야 하고, 간혹 힘을 많이 가하면 부러진다. 자판은 양손의 모든 손가락을 활용하며 자판에 익숙하면 원고를 보면서 칠 수 있다. 연필 심과 같은 불편은 없다. 그러나 공간이 필요하고 휴대성이 연필에 비하여 못하다. 또한 자판은 자판의 글쇠에 할당하여 둔 글자만을 출력할 수 있다.

## 2.3 연필 컴퓨터 정의

연필 컴퓨터란 한글 문서 입력 및 표현에 국한하여 논의한다. 2.1에서 논의하였듯이 현행 한글 부호계 KS C 5601-1987에서 정의한 한글 이외에도 우리가 연필로 쉽게 한글 음절을 구성하듯이 할 수 있는 컴퓨터를 말한다. 그렇지만 자유도가 로마자처럼

무한인 것은 아니다. 그래서 자판의 특성에 추가하여 글자의 입력과 표현을 함에 있어서 훈민정음 원리가 정의하는 약 399억 음절자를 표현[2]할 수 있도록 해 주는 것이다.

## 3. 연필 컴퓨터의 설계

### 3.1 훈민정음 원리의 적용

2장에서 요구는 우리가 이제까지 파악해 온 한글과 컴퓨터에 대한 개념을 바꾸는 것이다. 이제까지 한글 부호계 제정자들이 한글을 보는 관점을 "컴퓨터를 위한 한글"로 하였다면 본 논문의 관점은 "한글을 위한 컴퓨터"로 한다는 점이 크게 다르다. 컴퓨터는 한글을 처리하기 위한 도구로써 한글이 주라면 컴퓨터는 종의 입장에 있다. 종의 한계는 기술의 개발로 극복해야 한다. 그런데 현재처럼 주인 한글의 특성을 제약하면서 해결하려는 것은 공학의 이치에 크게 어긋난다. 컴퓨터 공학을 위한 훈민정음 원리는 많은 연구[2.5.8]가 이루어져 왔다. 훈민정음 해례본[1]에서 "천지자연의 소리를 표현할 문자를 만들겠다"는 요구에 더불어 "배우기 쉽고 사용하기 쉬운 문자를 만들겠다"는 요구를 만족시키기 위하여 집합의 조건제시법을 적용하여 한글 글자의 전체 집합을 정의하고 있다. 다시 말해서 399억 음절 글자를 표현[2]할 수 있는 한 가지 음절 글자 구성원리를 개발한 것이 곧 훈민정음이다. 해례를 요약하면 28자의 기본 글자와 중성부용초성, 연서법, 합용법, 부서법, 성음법 등 5가지 규칙으로 정의한 것이다. 이것을 기반으로 만든 정음형 부호계[2.5.8]를 사용한다. 정음형 부호계는 중성부용초성이라는 규칙을 없애므로써 기본 글자를 45자로 하고 있다.

### 3.2 입력 및 표현

연필 컴퓨터에서 중요한 사항은 연필처럼 글자 표현의 자유스러움이다. 그렇지만 로마자처럼 무한 길이의 문자열을 허용하지는 않는다. 그것은 훈민정음 해례에서 규정한 합용법과 성음법 때문이다. 합용법은 합자해에서 초성, 중성, 종성은 2 또는 3자를 합용한다고 정의한 부분을 말한다. 그래서 고어의 예에서 "ㄱ"처럼 석자 조합이 존재한다. 이것은 최대로 초성 3자, 중성 3자, 종성 3자로 된 글자 구성을 가능하게 한다.

기존의 한글 부호계 제정자들은 한글 자소와 음절자와의 관계를 정확하게 구분하지 못하였다. 그것은

한글 문자가 로마자와 가나와 함께 표음문자에 속하지만 로마자가 음소문자에 속하고 가나가 음절문자에 속하는 반면에 한글은 음소 및 음절 문자에 동시에 속한다[2]는 사실을 몰랐기 때문이다. 여기서 우리는 한글의 표현 방법으로써 내부 표현에서는 음소 문자 특성을 가지게 하고 외부 표현에서는 음절문자 특성을 가지게 함으로써 한글이 가지고 있는 특성을 모두 만족하게 한다. 흔히 이 경우 자모형 부호계에 익숙해 있는 사람은 음절 구분에 대하여 문제가 있음을 말한다. 하지만 정음형은 자음을 초성과 중성에 대하여 따로 부호계를 가지고 있기 때문에 그 정보가 음절구성 정보로 되어 음절 구분은 저절로 이루어지기 때문에 음절 구분을 위하여 특별히 어떤 처리를 할 필요는 없게 된다.

현재 자판에 있는 한글 자모의 수는 자음 19자와 모음 14자로 되어 있다. 여기서 훈민정음 창제 당시 글자에서 빠진 글자는 자음 ㅇ ㅅ ㅈ 3자와 모음 . 1자이다. 이것은 연구[2]에 따라서 자판의 가운데 단에서 왼쪽의 D F G와 오른쪽의 K에 할당한다. 옛한글 취급에 관한 기존의 인식은 1933년 조선어학회의 한글맞춤법에 두고 있음에도 불구하고 현대 한글을 바탕으로 해서 옛한글을 취급하려고 하기 때문에 발생한다. 이것의 해결은 훈민정음에 기반할 때 간단하게 해결된다. 훈민정음 당시에는 옛한글이 없었기 때문이다.

### 3.3 편집 및 출력

편집에서 요구되는 명령은 삽입, 삭제, 대치(삭제, 삽입), 찾기, 커서의 이동 등이다. 이러한 명령은 대상에 따라서 크게 두 가지 유형으로 분리될 수 있다. 그것은 자소와 음절로 대별된다. 따라서 자소 모드와 음절 모드로 정의한다. 앞에서 언급하였지만 한글은 내부에선 1차원의 풀어쓰기로 존재하기 때문에 자소 모드에서 명령에 대한 연산은 로마자와 같다. 예를 들어서 1) 커서 이동은 "맑"의 경우 4자로 구성되어 있기 때문에 커서를 다섯 번 옮겨야 완전히 지나갈 수 있다. 여기서 문제는 음소문자와 음절문자의 특성을 동시에 가지고 있기 때문에 외부 표현은 음절자로 되어 있어서 커서를 네 번 누르는 동안 화면상의 커서는 "맑"에 계속 머물러 있어야 한다. 2) 삭제의 경우 "맑"에서 커서가 "ㄹ"에 놓여 있다면 삭제 글쇠를 누르면 "막"이 된다. 그리고 이때 커서는 "ㄱ"에 놓이게 된다. 또한 "각"의 경우에 커서가 "ㄱ"에 놓여 있다면 삭제후 "ㄱ" 형태로

바뀐다. 3) 삽입의 경우는 삭제와 반대로 생각하면 된다. 4) 대치의 경우 "각"을 "곡"으로 바꾸려면 커서를 "각"의 "ㄱ"에 두고 대치모드에서 "ㅇ"를 치면 "곡"이 된다. 자소모드와 달리 음절모드는 현재의 한글 편집기나 워드프로세서에서 취급하는 형태가 된다. 문자열 비교는 자소 모드와 음절 모드 둘 다 자소 문자열의 비교를 적용하지만 실패하면 음절정보를 추가함으로써 보다 큰 효과를 거둘 수 있다. 예를 들어서 패턴 "학교"는 문자열 "한국민족주의..."에 대하여 자소문자열 비교에서 "하"까지는 일치하지만 "ㄱ"과 "ㄴ"이 다르기 때문에 실패한다. 이때 다음 비교는 패턴의 "ㅎ"이 문자열 "ㅏ"와 하지 않고 바로 "국"의 "ㄱ"과 비교한다.

출력에서 요구는 399억 글자에 대한 글자꼴의 지원이 중요하다. 훈민정음 원리에 따라 조합할 수 있는 글자가 399억에 달한다. 이들 글꼴을 모두 만든다는 것은 매우 어려운 일이다. 따라서 기존에 만들어진 글꼴이 있으면 이용하고 나머지 없는 경우에는 조합해서 쓰도록 한다. 현대 한글은 MS 윈도우95의 완성 음절 글꼴 11172자를 이용하면 된다. 이처럼 훈민정음 원리를 따르는 편집기 개발에서 글자꼴을 조합하는 방식에 대한 연구[4]가 있었다. 여기에 따르면 초성, 중성, 종성에 대하여 3자씩 조합은 현재의 글자꼴 크기로는 표현이 안되기 때문에 40x40으로 하였다. 그래서 "?"과 같은 석자 조합의 경우에 맞는 날자를 준비하고 이들 3자를 조합한 글자꼴을 구성하여 초성 또는 중성에 배치한다. 중성의 경우 역시 마찬가지로 한다. 이런 식으로 하는 글자꼴 베이스 모듈을 개발한다. 이 모듈은 기본 글자꼴과 기존의 글자꼴에 없는 글자꼴을 조합해 주는 조합루틴과 함께 구성되어 있다.

## 4. 구현

### 4.1 시스템 환경

프로세서에 관계없이 마이크로 소프트 윈도우95 또는 98를 사용하는 환경에서 비주얼 C++ 프로그래밍 언어[6]를 사용한다. 비주얼 C++는 MFC(MicroSoft Foundation Class) 라이브러리와 윈도우95 API를 결합한 것이다. 입력 모드에서 윈도우95는 한글을 입력하면 한글 IME를 통과하도록 하고 있는데 이것의 역할은 자판에서 들어오는 한글 자모 문자열을 음절 완성형 한글 부호계로 바꾸어 주는 역할을 한다. 정음형 부호계를 사용하기 위하여 이 IME 대신에 자모형 문자열을 정음형 부호계로 바꾸어 주는 모듈로 대체

한다. 이것은 UNIX 시스템에서 xterm과 vi에 구현한 바도 있다[8].

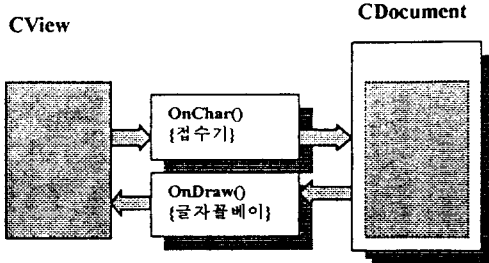


그림 2. 도큐먼트-뷰 모델

#### 4.2 입력과 내부 표현 및 출력

비주얼 C++는 도큐먼트-뷰 모델[6]을 제공한다. 이것의 역할은 프로그램에서 처리하는 데이터와 사용자의 입력을 받아서 처리하는 작업으로 분리되어 있다[4.5]. 문서 객체는 프로그램에서 사용하는 데이터의 관리와 처리 작업을 책임지고 있으며, 뷰 객체는 사용자의 입력을 처리하는 작업을 담당한다. 비주얼 C++의 통합 개발 환경은 입력된 자료의 ASCII를 반환하는 OnChar()와 OnKeyDown()이라는 뷰 클래스의 멤버 함수를 WizardApp를 통하여 쉽게 그 프로토타입을 부여 받는다. 함수 JungEumView::OnChar() 함수에 자모형-정음형 변환 루틴을 추가하면 된다. 비주얼 키는 OnKeyDown() 함수에서 인식된다. 한글 문자열 입력 과정에서 자음의 모호성 해결을 위하여 종성과 초성의 구별은 별도의 글쇠 L에 그 역할을 배정하여 구분한다. 이런 식으로 입력된 정음형 부호로 변환된 자료는 CString 타입의 오브젝트에 저장된다. 편집은 CString 멤버함수의 확장으로 이룩된다.

```

Void JungEumView::OnChar(UINT nChar,
UINT nRepCnt, UINT nFlags)
{
    Cview::OnChar(nChar, nRepCnt, nFlags);
}
    
```

출력은 뷰 클래스에 있는 글꼴 출력 취급 멤버 함수인 OnDraw() 함수를 활용한다. OnDraw() 함수는 디스플레이 콘텍스트(DC)의 포인터를 매개 변수로 받아서 텍스트 버퍼가 있는 문서 접근 포인트를 활용

해서 문서의 코드에 대응하는 글자꼴을 화면의 적절한 위치에 뿌려준다.

```

Void CjeongEumView::OnDraw(CDC* pDC)
{
    CjeongEumDoc* pDoc=GetDocument();
    ...
    pDC->TextOut(CaretPos.x, CaretPos.y,
HanFont[k]);
    ...
    SetCaretPos(CaretPos);
}
    
```

매개 변수 CDC\* pDC (화면버퍼)가 지시하는 곳 즉 화소로 구성된 출력 공간에 글자꼴을 뿌린다. 이런 식으로 문서의 내용을 적절하게 출력한다. 한 음절의 글자꼴은 같은 화면 공간에 초성자, 중성자, 종성자에 대하여 각자 조합된 글자꼴을 겹쳐서 투명하게 출력해야 한다. 이 때 디스플레이 콘텍스트 멤버함수인 SetBkMode(TRANSPARENT)로 하면 글자꼴 이미지를 정해진 공간에 대하여 OR 연산을 하게 하는 효과를 낸다.

#### 4.3 국제문자 부호계의 한계

국제문자 부호계[7]에는 240자 접자소형, 접자모형, 음절 완성형 등 세가지 한글 부호계를 포함하고 있다. 이제 윈도우NT, Java 등에서 이 부호계를 기본적으로 지원하고 있다. 하지만 아직 대부분의 자료가 기존의 KS C 5601-1987로 되어 있기 때문에 쉽게 옮겨 갈 수 없는 형편이다. 그런데 기존 연구[3]에서 지적한 바와 같이 세가지 부호계를 혼합하여 사용할 수 밖에 없다. 즉 한글 한 음절자를 표현하기 위하여 8가지 경우의 혼합 부호계 문자열 구성이 가능하다. 따라서 이 문제를 해결을 위하여 우선 방편으로 접자모형과 음절 완성형을 제거하고 240자 자소형만 사용하도록 하고, 궁극적으로 훈민정음 원리에 기반한 정음형 부호계로 개정을 서둘러야 한다.

#### 5. 결론

컴퓨터가 한국인의 지적 노동 생산성을 높여 주는 도구로 되려면 먼저 연필로 글을 쓰듯이 자유롭게 문자를 구성할 수 있어야 한다는 전제를 가지고 훈민정음 원리에 기반한 연필로 글을 쓰듯이 컴퓨터에서 문서를 작성할 수 있는 컴퓨터를 설계하고 구현

하여 보았다.

현재 컴퓨터가 가진 문제는 바로 한글 부호계의 제정에서 한글 문자의 음소 및 음절문자 특성을 제대로 파악하지 못하였을 뿐만 아니라 혼민정음의 기본 원리를 인식하지 못한 채로 단지 1933년 한글 맞춤법에 기반하여 부호계를 제정하였기 때문이다. 현재 우리가 가진 한글 처리에 관련된 모든 문제를 근본적으로 제거하려면 혼민정음 창제 원리를 적용해야 한다. 약 399억 음절자를 낱자의 조합으로 생성하는 원리를 담고 있기 때문에 적어도 "연필로 글을 쓰듯이..."를 가능하게 하는 연필 컴퓨터를 구현하였다. 연필 컴퓨터는 표현할 수 없는 글자가 없기 때문에 현재 우리가 사용하고 있는 컴퓨터들처럼 우리의 사고를 중단시키거나 방해하지 않을 것이다. 따라서 컴퓨터를 사용하는 목적인 인간 두뇌 활동의 속도화 즉 지적 노동 생산성을 극대화 할 수 있을 것이다. 또한 본 연구의 결과는 "컴퓨터를 위한 한글"이 아니라 "한글 위한 컴퓨터"라는 한글 구현의 기본 개념을 바르게 잡는 계기가 될 것이다.

#### 참고문헌

- [1]강신항, 혼민정음연구, 성균관대학교출판부, 1991
- [2]변정용, "혼민정음 창제원리의 공학화에 기반한 한글보호계의 발전방향", 한국정보과학회, V12, N2, 1994
- [3]변정용, "국제문자부호계에서 한글부호계의 개선 방안", 가을 학술발표논문집, 제25권 제2호, 한국정보과학회, p192-194, 1998
- [4]윤지현, 변정용, "정음형 한글코드를 위한 글꼴 편집기의 설계", 추계학술발표논문집, 제5권 제2호, 한국정보처리학회, p651-654, 1998
- [5]변정용, 한글정보처리를 위한 표준 입력라이브러리 연구:3차년도, 한국과학기술원, 1997
- [6]이동휘 역, 고급사용자를 위한 Advanced Visual C++, 삼각형, 1997
- [7]한국표준협회, KS C 5700-1995, 1995
- [8]변정용, 한글정보처리를 위한 표준 입력 라이브러리 연구:1차년도, 시스템공학연구소, 1995
- [9]최기선, "국어정보베이스의 현재와 미래", 정보과학회지:국어공학, p5-7, 1997