

ISO 2022 를 따르는 한 바이트 한글 부호계를 지원하는 X-Window 용 한글 문서 편집기 개발 연구

조 충래, 김 경석, 부산대학교 전자계산학과
clcho@asadal.cs.pusan.ac.kr

A Study of developing Hangul text editor for X-Window
using a one-byte Hangul code supporting ISO 2022

Chunglae Cho, Kyongsok Kim
Dept. of Computer Science, Pusan National University

요약

현재 정보 관련 응용 분야 가운데 ISO 2022 를 따르는 분야가 아주 많은데, 현재 쓰고 있는 한글 부호계들은 ISO 2022 를 제대로 지원하지 못하거나, ISO 2022 를 지원하더라도 한글을 제대로 지원하지 못하는 문제가 있다. 이러한 문제를 해결하기 위해서 ISO 2022 를 지원하면서 한글을 제대로 지원하는 새로운 한 바이트 한글 부호계를 만들었다. 새로운 한 바이트 한글 부호계는 요즘 한글 11,172 소리마디를 모두 표현할 수 있으며, 불안정한 소리마디를 나타내는 방법으로 기존의 채움 글자 방식을 버리고 자연스럽게 한글의 특성에 맞는 땀 글자 방식을 택하였다.

본 연구에서는 새로운 한 바이트 한글 부호계를 지원하는 X-Window 용 한글 문서 편집기를 개발해 봄으로써 그 운용 가능성을 검증하였다. 한 바이트 한글 부호계를 지원하는 문서 편집기의 운용 환경으로 유닉스 운영체제 하에서 돌아가는 X-Window 시스템을 택하였고, 한글 입출력 부분을 모티프 (Motif) 의 위젯 (widget) 형태로 구현하여 다른 응용 프로그램에서도 쉽게 한 바이트 한글 부호계를 지원할 수 있게 하였다.

1. 들머리

현재 정보 관련 응용 분야 가운데 ISO 2022 를 따르는 분야로는 Telematic Services (보기, Videotex, Teletex, Digital Fax, Message Handling Service), Teletext (문자 다중 방송), Digital VCR (영상, 음성, 문자 등의 모든 정보를 디지털로 저장하는 차세대 VCR), EUC (Extended UNIX Code: 유닉스에서 여러 나라 글자를 지원하는 방안), EDI (Electronic Data Interchange), Directory Service 등 아주 다양하다. 특히 유닉스 운영 체제는 아주 많은 컴퓨터에서 쓰고 있으므로, 유닉스 분야의 한글 지원 문제는 아주 중요하다. 하지만, 현재 쓰고 있는 한글 부호계들은 ISO 2022 를 제대로 지원하지 못하고 있다. KS C 5601 은 ISO 2022 를 따르고는 있지만, 2,350 소리마디밖에 지원하지 못하고, 국제 표준 한글 부호계인 ISO 10646-1 한

글 부호계와 현재 많이 사용하고 있는 상용 조합형 한글 부호계는 ISO 2022 를 따르지 않는다 [김 경석 93, 김 경석 94 가, 김 경석 94 나].

이러한 문제를 해결하기 위하여 [김 경석 94 나] 에서 ISO 2022 를 따르면서 모든 요즘 한글 소리마디를 쓸 수 있는 새로운 한 바이트 한글 부호계를 제안하였다. 제안한 한 바이트 한글 부호계는 첫소리 글자 19 개, 가운뎃소리 글자 21 개, 끝소리 글자 27 개와 땀 글자 (syllable-break) 1 개로 이루어져 있으며, 요즘 한글 11,172 소리마디를 모두 지원한다. 특히 불안정한 소리마디를 나타낼 때 기존에 사용했던 채움 글자 방식을 버리고 땀 글자 방식을 택하였는데, 여기에 대해서는 뒤에 자세히 설명하겠다.

본 연구에서는 새로운 한 바이트 한글 부호계를 지원하는 X-Window 용 한글 문서 편집기를 개발해 봄으로써 그 운용 가능성을 검증하였다. 한 바이트 한글 부호계를 지원하는 문서 편집기의 운용 환경으로 유닉스 운영체제 하에서 돌아가는 X-Window 시스템을 택하였고, 한글 입출력 부분을 모티프 (Motif) 의 위젯 (widget) 형태로 구현하였다. 모티프에서는 텍스트 자료를 처리하거나 문자열을

* 본 연구는 시스템 공학 연구소 국어 공학 센터의 연구비 지원으로 이루어졌습니다.

화면에 나타내는 등의 기능을 위젯 (Widget) 이라는 개념을 통하여 사용자에게 제공한다. 사용자는 단지 위젯을 호출함으로써 원하는 문자열을 화면에 출력하거나 텍스트 자료를 편집할 수 있는 응용 프로그램을 만들 수 있다. 이러한 점을 고려하여 먼저 ISO 2022 를 따르는 한 바이트 한글 레이블 위젯 (label widget) 과 한글 텍스트 위젯 (text widget) 을 개발하였고 이를 이용하여 문서 편집기를 만들었다.

본 연구에서 개발한 문서 편집기는 한 바이트 한글 부호계를 실제 사용하기 위한 준비 작업으로서, 제안된 부호계의 운용 가능성을 검증하는데 그 의의가 있다.

2. ISO 2022 를 따르는 분야를 위한 새 한글 부호계

이때까지 ISO 2022 를 따르려면 두 바이트 완성형 말고는 대안이 없는 것처럼 생각하는 듯 한데, 이는 사실과 다르다. [김 경석 94 나] 에서 제안한 한 바이트 한글 부호계는 10646-1 의 첫가끝 한글 부호계의 기본 정신은 그대로 따르되, 238 글자 가운데서 옛글자를 뺀 요즘한글 67 글자로 구성되어 있으며, 요즘 한글 11,172 소리마디를 모두 표현할 수 있으면서도 ISO 2022 를 따른다. 제안한 새로운 한 바이트 한글 부호계가 [그림 1] 에 있다. 이 부호계의 특징을 아래에서 자세히 살펴 보겠다.

2.1 10646-1 의 기본 정신을 그대로 따른다.

새로 만든 한 바이트 한글 부호계는 10646-1 의 첫가끝 한글 부호계와 마찬가지로 첫소리 글자, 가운뎃소리 글자, 끝소리 글자 각각에 대해 부호를 할당한 세벌식 체계이다. 이런 글자를 조합하여 소리마디를 나타내며, 한 소리마디는 끝소리 글자가 없을 때는 두 바이트이며 끝소리 글자가 있으면 세 바이트를 차지하는 가변 크기이다.

2.2 채움 글자 대신 땀 글자를 사용한다.

10646-1 의 첫가끝 한글 부호계는 불완전한 소리마디를 나타내기 위해서 첫소리 채움 글자와 가운뎃소리 채움 글자를 사용한다. 보기를 들어 “ㄱ”, “ㄴ”, 또는 “ㅇ” 을 따로 나타내기 위해서는, “ㄱ”은 “ㄱ + 가운뎃소리 채움 글자” 로, “ㄴ”은 “첫소리 채움 글자 + ㄴ” 로, “ㅇ”은 “첫소리 채움 글자 + ㅇ” 으로 나타내게 되어 있다. 여기서

끝소리 글자만 있는 경우 (첫소리 채움 글자 + 끝소리 글자) 로 나타내었는데 이는 자연스럽지 못하다. 끝소리 글자만 있는 경우, 자연스러운 방식은 (첫소리 채움 글자 + 가운뎃소리 채움 글자 + 끝소리 글자) 일 것이다. 그런데, 현재 첫가끝 조합형에서는 단순 비교를 했을 때 한글 자료가 가나다순에 맞게 나오도록 하기 위해 이를 억지로 끼워 맞추어, 끝소리 글자만 있는 경우에 (첫소리 채움 글자 + 끝소리 글자) 라는 어색한 표현 방식을 쓰고 있으며, 이는 상식이나 직관에 벗어나서 전혀 한글답지 않다.

이러한 채움 글자 방식의 문제점을 해결하는 방안으로 두 가지가 있다. 첫째는 가운뎃소리 채움 글자를 두 개로 나누어 쓰는 방안이며, 둘째는 땀 글자 방식을 쓰는 방법이다 [김 경석 93, 김 경석 94 다] .

2.2.1 채움 글자의 개선 방안 (1): 가운뎃소리 채움 글자를 두 개로 나누어 쓰는 방안

현재 첫가끝 조합형에는 가운뎃소리 채움 글자가 하나만 있으며, 그 값의 위치는 첫소리 채움 글자 바로 뒤이고,

		X(b7b6b5)							
Y(b4b3b2b1)		00	01	02	03	04	05	06	07
00						ㅈ	ㅊ	ㅋ	ㆁ
01				ㄱ	ㅋ	ㆁ	ㄴ		
02				ㄴ	ㄷ	ㄹ	ㅇ	ㅇ	
03				ㄷ	ㅌ	ㄹ	ㄱ	ㄴ	
04			syl break	ㄴ	ㅌ	ㄹ	ㄱ	ㄴ	ㅈ
05				ㄴ	ㅎ	ㅌ	ㄹ	ㅈ	ㅊ
06				ㄴ		ㄱ	ㄴ	ㅈ	ㅊ
07				ㄴ	ㅌ	ㄹ	ㄱ	ㄴ	
08				ㅂ		ㄱ	ㄴ	ㅈ	ㅊ
09				ㅂ	ㅌ	ㄹ	ㄱ	ㄴ	ㅇ
10				ㅂ	ㅌ	ㄹ	ㄱ	ㄴ	
11				ㅂ	ㅌ	ㄹ	ㄱ	ㄴ	
12					ㄱ	ㄴ	ㅈ	ㅊ	
13				ㅇ	ㅎ	ㄱ	ㄴ	ㅈ	ㅊ
14					ㄱ	ㄴ	ㅈ	ㅊ	
15				ㅅ	ㅎ	ㅌ	ㄹ	ㅈ	ㅊ

[그림 1] ISO 2022 를 따르는 한 바이트 한글 부호계

가운뎃소리 글자 가운데 처음으로 나오는 “ㄴ” 바로 앞이다.

그런데, 현재 가운뎃소리 채움 글자 하나로, 용법이 전혀 다른 두 가지 경우에 같이 쓰고 있기 때문에, 앞에서와 같은 채움 글자의 문제가 발생한다. 보기를 들어, “ㅎ”을 (ㅎ + 가운뎃소리 채움 글자)로 나타낼 때 쓰는 가운뎃소리 채움 글자와, 끝소리 글자 “ㄴ”을 (첫소리 채움 글자 + 가운뎃소리 채움 글자 + ㄴ)으로 나타낼 때 쓰는 가운뎃소리 채움 글자는 그 용법이 근본적으로 다르다. “ㅎ” 뒤에 따라 나오는 가운뎃소리 채움 글자는, 첫소리 글자 뒤에 가운뎃소리가 뒤따르지 않는다는 것을 나타내는데 반해서 (이를 “뒤따르는” 가운뎃소리 채움 글자라고 부르겠다), “ㄴ” 앞에 나오는 가운뎃소리 채움 글자는, 끝소리 글자 앞에 가운뎃소리 글자가 앞서지 않는다는 것을 나타낸다 (이를 “앞서는” 가운뎃소리 채움 글자라고 부르겠다). 그런데도, 이 두 가지 다른 가운뎃소리 채움 글자를 하나로 뭉쳤기 때문에, 첫가끝 조합형의 채움 글자 용법이 현재 아주 이상하게 되어버렸다.

따라서, 뒤따르는 가운뎃소리 채움 글자의 부호값은 모든 가운뎃소리 글자의 부호값보다 작은 부호값을 주고, 앞서는 가운뎃소리 채움 글자의 부호값은 모든 가운뎃소리 글자의 부호값보다 큰 부호값을 주어야, 단순 비교로 가나다순대로 간추릴 수 있다.

2.2.2 채움 글자의 개선 방안 (2): 채움 글자 대신에 뎀 글자를 쓰는 방안

한글에서 나타낼 수 있는 가장 작은 단위가 글자가 아니라 소리마디라고 생각하여, 불완전한 소리마디를 나타낼 때 꼭 채움 글자를 써야 한다고 많은 사람들이 생각하는데 이는 잘못된 생각이다. 한글의 가장 작은 단위는 글자이며, 채움 글자를 쓰지 않고도 불완전한 소리마디를 나타낼 수 있는데 굳이 채움 글자를 쓸 이유가 없다 [김 경석 93, 김 경석 94] .

현재의 채움 글자 방식에 따르면, 완전한 소리마디가 아니면 반드시 채움 글자가 들어가게 되어 있다. 보기를 들어, “ㄱ”이 따로 나오는 것을 나타내려면, “ㄱ + 가운뎃소리 채움 글자”로 나타내어야 한다. 이에 비해서 뎀 글자 방식에서는 이 경우 그냥 “ㄱ”으로 나타내기 때문에, 직관적으로 이해하기도 쉽고, 또 뎀 글자를 꼭 쓰지 않아도 된다. 일반적으로 말해서, 완전한 소리마디가 아닌 때, 채움 글자 방식에서는 채움 글자를 반드시 써야 하는 데 비

해서, 뎀 글자 방식에서는 보통의 경우 거의 뎀 글자를 쓰지 않고도 한글 정보를 다 나타낼 수 있다는 장점이 있다.

그러면 어떤 경우에 뎀 글자를 꼭 써야 하는지 살펴 보겠다. 보기를 들어, “가ㄴ”을 나타내려면 (ㄱ + ㄴ + 뎀 글자 + ㄴ)과 같이 나타낸다. 뎀 글자가 없는 (ㄱ + ㄴ + ㄴ)은 “간”을 나타낸다. 다른 보기로, “가”가 아니라 “ㄱ ㄴ”라는 두 글자를 나타내고자 한다면, (ㄱ + 뎀 글자 + ㄴ)라고 하면 된다. 그런데, 이와 같이 뎀 글자를 쓰는 경우는 아주 드물게 나오므로, 보통의 경우에는 뎀 글자를 거의 쓸 필요가 없다. 예를 들어, “ㄱㄴㄷ”을 10646-1 첫가끝 조합형의 채움 글자 방식으로 나타내면 (ㄱ + 가운뎃소리 채움 글자 + ㄴ + 가운뎃소리 채움 글자 + ㄷ + 가운뎃소리 채움 글자)인 반면, 뎀 글자 방식으로 나타낼 때는 그냥 (ㄱ + ㄴ + ㄷ)으로 나타내면 된다.

뎀 글자 방식은 새로운 것이 아니며, 현재 많은 다른 나라에서도 뎀 글자 방식을 쓰고 있다 [김 경석 94 다, ISO 93] . 다른 나라의 해결 방법을 무조건 따르는 것은 옳지 않지만, 지금까지 살펴 본 바로는 뎀 글자 방식이 채움 글자 방식보다는 더욱 자연스럽고 한글의 특성에 맞다.

2.3 한 바이트 한글 부호계에 맞는 자판

ISO 2022 를 따르는 한 바이트 한글 부호계를 위해서는 기존의 두벌식 혹은 세벌식 자판에서 뎀 글자를 위한 글쇠만 추가하면 된다. 뎀 글자를 위한 글쇠를 두는 것은 꼭 필요하지는 않지만, 불완전한 소리마디를 입력할 때 보다 쉽게 입력할 수 있게 하기 위함이다. 예를 들어 “가ㄴ”을 입력하고자 할 때, 기존의 자판에서는 “가”를 입력한 후 띄움 글쇠 (space bar) 를 눌렀다가 다시 뒤지움 글쇠 (backspace key) 를 누른 후 “ㄴ”을 입력하여야 하지만, 뎀 글쇠가 있으면 “가”, 뎀 글쇠, “ㄴ”의 식으로 입력을 하면 되므로 편리하다.

두벌식, 세벌식 문제에 있어서는 어느 것도 상관은 없지만, 제안하는 부호계가 세벌식 체계이므로 자판도 세벌식 자판을 사용하는 것이 바람직하다 [KimK 92, KimK 94] .

3. 새로운 한 바이트 한글 부호계를 지원하는 문서 편집기 개발

본 연구에서는 새로운 한 바이트 한글 부호계를 지원하는 X-Window 용 한글 문서 편집기를 개발하여 그 운용

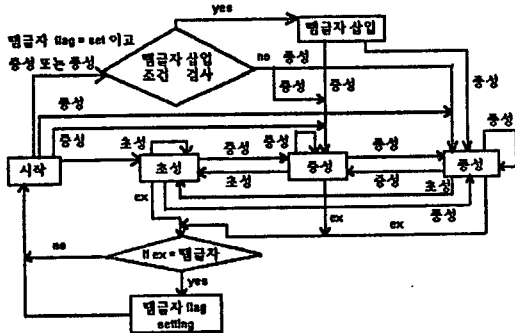
가능성을 시험해 보았다. 한 바이트 한글 부호계를 지원하는 문서 편집기의 운용 환경으로 유닉스 운영체제 하에서 돌아가는 X-Window 시스템을 택하였고, 한글 입출력 부분을 모티프 (Motif) 의 위젯 (widget) 형태로 구현하였다.

모티프에서는 텍스트 자료를 처리하거나 문자열을 화면에 나타내는 등의 기능을 위젯이라는 개념을 통하여 사용자에게 제공한다. 사용자는 단지 위젯을 호출함으로써 원하는 문자열을 화면에 출력하거나 텍스트 자료를 편집할 수 있는 응용 프로그램을 만들 수 있다. 이러한 점을 고려하여 먼저 ISO 2022 를 따르는 한 바이트 한글 레이블 위젯 (label widget) 과 한글 텍스트 위젯 (text widget) 을 개발하였고 이를 이용하여 문서 편집기를 만들었다.

3.1 절에서는 뎀 글자를 처리하기 위한 새로운 한글 입력기 개발에 대해 살펴 보겠다. 그리고, 3.2 절에서는 ISO 2022 를 따르는 한 바이트 한글 부호계를 지원하는 레이블 위젯에 대해 살펴보고, 3.3 절에서는 한 바이트 한글 텍스트 위젯에 대해 살펴 보며, 3.4 절에서는 한 바이트 한글 레이블 위젯과 텍스트 위젯을 사용하여 만든 문서 편집기 개발에 대해서 살펴 보겠다.

3.1 뎀 글자를 처리하기 위한 새로운 한글 입력기 개발

새로운 한 바이트 한글 부호계는 기존의 조합형 부호계에서 불완전한 소리마디를 처리하기 위해 사용하던 채움 글자 방식을 버리고 뎀 글자 방식을 택했다는 것은 앞에서 이미 설명하였다. 채움 글자가 없어지고 뎀 글자를 사용하게 됨으로써, 완전한 소리마디인 경우에는 예전의 방식으로 한 소리마디가 조합되지만 불완전한 소리마디의 경우에는

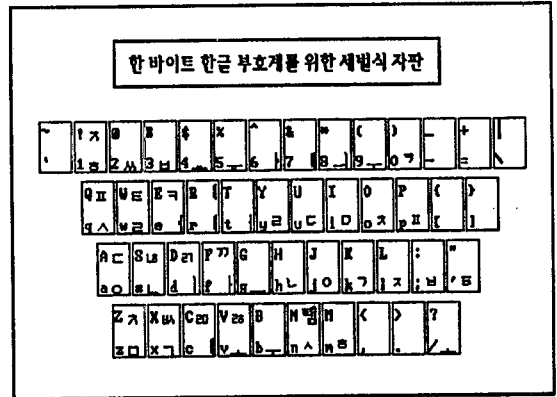


[그림 2] 한 바이트 한글 부호계를 위한 한글 입력 오토마타

다소 차이가 있다. 이로 인해 새로운 한 바이트 한글 부호계를 위한 한글 입력기를 새로 개발하게 되었다 [그림 2].

3.1.1 뎀 글쇠의 지정

뎀 글자를 위한 글쇠로 "shift + n" 을 지정하였고 나머지 글쇠는 기존의 세벌식 자판의 것을 그대로 사용하였다. 아래의 [그림 3] 은 본 문서 편집기에서 지원하는 한 바이트 한글 부호계용 세벌식 자판이다.



[그림 3] 한 바이트 한글 부호계용 세벌식 자판

3.1.2 뎀 글자의 처리

뎀 글자는 뎀 글쇠가 눌러지면 무조건 삽입되는 것이 아니라 뎀 글자가 꼭 들어가야 할 상황에서만 삽입되도록 설계해서, 사용자가 뎀 글자의 용법을 잘 모르더라도 사용할 수 있도록 했으며, 뎀 글쇠가 중복적으로 눌러지더라도 한 번 눌러졌을 때와 똑같이 처리하도록 하였다. 참고로, 뎀 글자가 사용되어야 하는 경우는 다음 세가지 경우 뿐이다.

- 1) 앞소리마디의 끝글자가 첫소리 글자이고, 뒷소리마디의 첫글자가 가운뎃소리 글자일 때, 예) "ㄱ ㅏ", "ㄴ ㅓ"
- 2) 앞소리마디의 끝글자가 첫소리 글자이고, 뒷소리마디의 첫글자가 끝소리 글자일 때, 예) "ㄱ ㄴ", "ㄴ ㄴ"
- 3) 앞소리마디의 끝글자가 가운뎃소리 글자이고, 뒷소리마디의 첫글자가 끝소리 글자일 때, 예) "가 ㄴ", "처 ㅏ"

위의 세가지 경우 외에는 뎀글자가 없어도 자연적으로 소리마디의 구분을 할 수가 있다. 예를 들어, 첫소리 글자 다음에 다시 또 첫소리 글자가 오게되면 앞의 소리마디는 첫소리 글자 하나만으로 이루어진 것으로 처리하게 된다.

뎀 글자에 대한 처리는 크게 두 가지로 나누어진다. 첫

제는 뎀 글쇠가 눌러졌을 때 뎀 글자를 처리하는 경우이고, 둘째는 뎀 글쇠를 누르지 않았더라도 뎀 글자를 처리해야 하는 경우이다. 뎀 글쇠가 눌러졌더라도 뎀 글자를 넣어야 한 상황에서만 뎀 글자를 넣어야 하며, 뎀 글쇠를 누르지 않았더라도 문자열의 중간에서 글자를 새로이 넣거나 지우는 경우 등에는 소리마디의 구분을 위해서 뎀 글자를 넣어야 한 때가 있다. 위의 세가지 조건 중 어느 하나가 맞다면 소리마디의 구분을 위해서 뎀 글자를 넣도록 한글 입력기를 만들었다.

3.2 ISO 2022 를 따르는 한 바이트 한글 레이블 위젯 개발

ISO 2022 를 따르는 한 바이트 한글 부호계를 지원하는 레이블 위젯을 개발하기 위해서는 기존의 영문 레이블 위젯을 위한 자료구조와 함수들 중에서 몇가지 부분을 수정하여야 한다. 레이블을 위한 구조체에서 한글 폰트를 위한 구조체를 추가하고, 문자열을 화면에 보여주는 함수에서 한글 처리를 위한 부분을 추가하였다 [김 경석 95] .

3.3 ISO 2022 를 따르는 한 바이트 한글 텍스트 위젯 개발

ISO 2022 를 따르는 한 바이트 한글 부호계를 지원하는 텍스트 위젯을 개발하기 위해서는 레이블 위젯보다 훨씬 더 많은 작업이 필요하다. 특히 한글 입력을 위한 한글 오토마타를 구현하여야 하는데 이에 대한 설명은 3.1 에서 이미 하였다. 그외에 텍스트 위젯에서 새로운 한 바이트 한글을 지원하게 하기 위해서 수정한 주요 자료구조들과 함수들이 [그림 4] 에 있다 [김 경석 95] .

3.4 ISO 2022 를 따르는 한글 문서 편집기 개발

본 연구에서는 ISO 2022 를 따르는 한 바이트 한글 부호계에 대한 운용 가능성을 검증하기 위해서 한 바이트 한글 부호계를 따르는 간단한 X-Window 용 문서 편집기를 만들었다. 이 문서 편집기를 통해서 한 바이트 한글 부호계로 표현되는 한글 문서를 작성하고 편집할 수 있다.

3.4.1 한 바이트 한글 부호계를 위한 한글 입출력기

한글 문서 편집기를 만드는 데 가장 중요한 것이 한글 입출력기이다. 특히, 제안한 한 바이트 한글 부호계가 뎀 글자 방식을 취하므로 기존의 입출력기와는 조금 다르다. 따라서 이를 위한 새로운 한글 입출력기를 만들었다는 것

자료구조 이름	하는 일
XnSBHCTextPart	한글 텍스트 위젯의 리소스 (resource) 들을 정의하는 구조체
XnSBHCTSeg	세그먼트를 위한 구조체
XnSBHCText	텍스트를 위한 구조체

함수 이름	하는 일
getStringDimension()	현재 세그먼트의 문자열의 폰트 길이를 구하는 함수
displaySimpleOneLine()	한 줄을 구성하는 문자열을 화면에 찍는 함수
displaySimpleOneHBounded()	화면상에서 정해진 시작 위치에서 끝 위치까지 문자열을 화면에 찍는 함수
InsertChar()	키보드에서 글쇠를 누르면 불러지는 함수
InsertHangul()	키보드에서 한글을 입력하면 불러지는 함수
RemoveChar()	뒤짐 글쇠 (backspace key) 를 눌렀을 때 불러지는 함수
DeleteChar()	지움 글쇠 (delete key) 를 눌렀을 때 불러지는 함수
hangulAutomata()	새로운 한 바이트 한글 부호계를 지원하기 위한 한글 오토마타를 구현한 함수

[그림 4] 텍스트 위젯에서 새로운 한 바이트 한글 부호계를 지원하기 위해 고친 주요 자료 구조와 함수들

은 이미 앞에서 말하였다.

구현된 입출력기는 모티프의 위젯 형태로 제공되기 때문에 문서 편집기를 만들 때는 단지 위젯을 선언하고 그에 대한 자원을 설정함으로써, 새로운 한글 입출력기의 기능을 그대로 사용할 수가 있다.

3.4.2 레이블 위젯과 텍스트 위젯의 등록과 자원 설정

본 연구에서 개발한 위젯을 사용하기 위해서는 응용 프로그램에서 위젯을 등록하고 이에 대한 자원을 설정해야 한다. 위젯을 등록하고 그 자원을 설정하는 일은 기존의 모티프 프로그램에서의 위젯 등록, 자원 설정과 똑같기 때문에 아주 쉽다. 또한 문서 편집기를 구현할 때 필요한 기본적인 편집 기능들은 텍스트 위젯 내에 이미 구현이 되어 있으므로 응용 프로그램에서는 위젯 등록과 자원 설정만으로

```

/* create a text widget and attach it to the form. */
/* hangul text widget version */
text = XtVaCreateManagedWidget("text",
    XnSBHCTextWidgetClass, form,
    XmNtopAttachment, XmATTACH_WIDGET,
    XmNtopWidget, menu_bar,
    XmNleftAttachment, XmATTACH_FORM,
    XmNrightAttachment, XmATTACH_FORM,
    XmNbottomAttachment, XmATTACH_POSITION,
    XmNbottomPosition, 92,
    XmNeditMode, XmMULTILINE_EDIT,
    XmNshadowThickness, 2,
    NULL);
XtAddCallback(text, XnNfocusCallback, text_changedCB,
    NULL);
XtSetSensitive(text, True);

ac = 0;
XtSetArg(al[ac], XnNeditable, True); ac++;
XtSetValues(text, al, ac);

```

[그림 5] 위젯 등록과 자원 설정의 예

로 기본적인 편집 기능을 사용할 수 있다. [그림 5]는 응용 프로그램에서 위젯을 등록하고 자원을 설정하는 예이다.

3.4.3 문서 편집기의 기능

본 연구에서 개발한, 새로운 한 바이트 한글 부호계를 지원하는 X-Window 용 한글 문서 편집기는 사용자들이 한 바이트 한글 부호계로 저장되어 있는 문서를 편집할 수 있으며, 부호계의 운용 가능성을 살펴 보는 것이 목적이므로 기본적인 기능만을 제공한다. 새로운 파일을 작성하거나, 이미 작성한 파일을 열어서 수정할 수 있으며, 현재 작성중인 파일을 현재 이름 그대로 저장하거나 세이름으로 저장할 수 있다. 그리고, 마우스를 이용하여 블록을 복사하거나 이동시키거나 삭제할 수 있으며, 지정된 줄로 커서를 옮기는 기능도 제공한다. 문서 편집기의 주요 기능을 나타낸 그림이 [그림 6]에 있다.

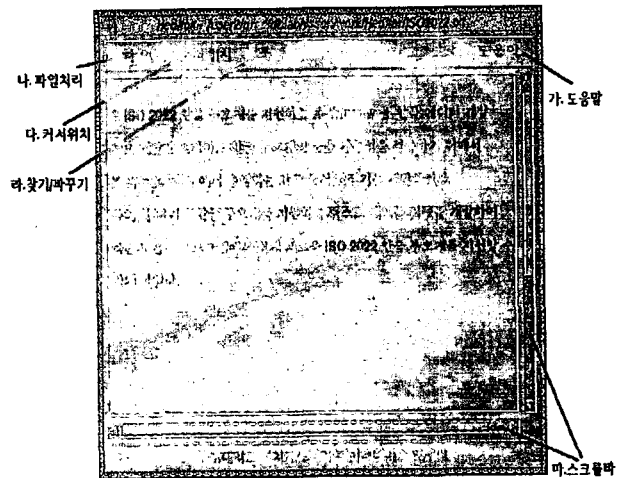
4. 맺음말

새로운 한 바이트 한글 부호계는 ISO 2022 를 따르면 서, 요즘한글 11,172 소리마디를 모두 쓸 수 있다. 또한 불완전한 소리마디를 나타내는 방법으로 뎀 글자 방식을 사

용하여 기존의 채움 글자 방식보다 훨씬 자연스럽고 한글의 특성에 맞게 불완전한 소리마디를 처리한다.

한 바이트 한글 부호계의 운용 가능성을 시험해 보기 위해서 이를 지원하는 X-Window 용 문서 편집기를 만들어 보았다. 이 문서 편집기에서 사용하는 한글 입출력기는 모티프의 위젯 형태로 구현하여 다른 응용 프로그램에서도 쉽게 사용할 수 있도록 하였다.

새로운 부호계로 인하여 현재 많이 사용하고 있는 한글 부호계와의 혼란이 일어날 수도 있을 것이다. 하지만, 이것은 일시적인 현상일 것이고, 또한 현재 ISO 2022 를 따르는 응용 분야에서 한글이 제대로 지원되고 있지 않은 현실을 볼 때 ISO 2022 를 따르면서 한글을 제대로 지원하는 한 바이트 한글 부호계는 꼭 필요하다. 새로이 개발한 한 바이트 한글 부호계를 이룬 시일 안에 국제 등록부에 등록하여 ISO 2022 를 따르는 응용 분야에서 쓸 수 있도록 해야 할 것이다.



[그림 6] 한 바이트 한글 부호계를 지원하는 X-Window 용 문서 편집기

5. 참고 문헌

[김 경석 93] 고속 중형 컴퓨터 (주전산기 III) S/W 환경에서의 표준 한글 처리 연구, 김 경석 (책임 연구원), 부산 대학교 정보 통신 연구소, 1993.12.

[김 경석 94 가] 관계형 데이터베이스에서 국제 표준 한글 부호계 수용 방안 기초 연구 최종 보고서, 김 경석 (책임

연구원), 부산대학교, 1994.3.

[김 경석 94 나] 국제 표준 틀 안의 한글 부호계 운용 방안 및 한 바이트 한글 부호계 안, 김 경석, 한글 및 한국어 정보 처리, 1994.11, 253-258 쪽.

[김 경석 94 다] 유니버설 코드셋 국제 표준화 대응 방안 연구, 김 경석, 한국 산업 표준원, 1994.12.

[김 경석 95] 한글 표준화 기반 기술 종합 연구 및 개발, 김 경석, 시스템 공학 연구소, 1995.8.

[변 정용 94] 한글 문자의 음소 및 음절 문자 특성의 구현 방안, 변 정용 외, 한글 및 한국어 정보처리 학술대회, 1994.11.

[금 대식 95] 훈민정음이 살아있는 hunterm의 구현, 금 대식, 변정용, 함 경수, 한국 정보 과학회 '95 봄 학술발표논문집, 1995.4.

[KimK 92] A Common Approach to Designing the Hangeul Code and Keyboard, Kyongsok Kim, Computer Standards & Interfaces, 1992.

[KimK 94] An Implementation of an Old-Hangeul Editor supporting Syllable-Initial-Peak-Final Hangeul Code in ISO/IEC 10646-1, Kyongsok Kim, Uido Lee, Hongwon Yun, Junghwa Lee, International Conference on Computer Processing of Oriental Languages, 1994, p377-382.

[ISO 93] ISO/IEC 10646-1:1993(E). International Standard. 1st edition. Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, May 1, 1993. ISO (the International Organization for Standardization).