

훈민정음 표현을 위한 최적 폰트 설계

김경욱^o 변정용
동국대학교 컴퓨터멀티미디어학부
{kakaru36, byunjy}@dongguk.ac.kr

An Optimized Font Design for Hunminjeongeum Representation

Kim Kyoungwook^o Byun Jeongyong
School of Computer and Multimedia, Dongguk University

요 약

1987년 정보교환용 부호가 완성음절형 부호계로 개정되면서 한글 음절 표현은 극히 제한되어 컴퓨터상에서 한글의 표현은 제약을 받게 되었다. 1446년 훈민정음이 공포된 후로 목판, 활자시대를 지나서 타자기를 만나면서 한자와 다른 길을 걸어왔다. 특히 컴퓨터시대가 되면서 한글은 타자 이상의 적합성을 가지고 있지만 훈민정음 창제원리에 대한 이해부족으로 음절문자로 분류하여 완성음절형 부호를 지원하게 됨에 따라서 많은 문제를 야기하고 있다. 국제표준인 유니코드에는 세가지 종류의 부호가 반영되어 있는데 음절문자 중심의 부호화이므로 훈민정음의 특성을 해치는 일이며 또한 한글의 과학성을 부정하는 일이기도 하다. 본 논문은 훈민정음 창제원리에 입각하지 않음으로 생긴 문제에 대한 근본적인 해결책으로 훈민정음을 표현하는 최적 폰트 설계 방안을 검토하고 현행의 한글부코드표준의 개선 방향을 제시하고자 한다.

1. 서 론

현행 한글부호계인 KS X 1001 정보교환용 부호계는 2350개의 완성 음절에 대한 부호로서 1987년 조합 음절형 부호계를 개선한 내용이다. 한글부호계 표준에서 혼란의 핵심은 문자론 입장에서 한글문자를 기본적으로 음절문자로 이해함에 기인한다[1,2,3]고 본다.

15세기에 창제된 훈민정음은 표의문자인 한자와 달리 표음문자에 속하며 음소문자이면서 음절문자에 속한다. 음소문자는 훈민정음의 근간이며 음절문자는 가독성을 위한 것이다. 하지만 현행 정보교환용 부호계 KS X 1001과 KS X 1005는 음절문자를 중심으로 삼고 음소문자는 딸린 것으로 보고 있다. 이러한 관점의 차이로 현행 한글부호계는 훈민정음의 기본 원리를 크게 해치고 있다. 최근 KS X 1005에는 1992년 이후 새로 발견된 옛한글 자모 117개를 새로 추가하였다. 훈민정음해례의 “무릇 글자는 모아야 소리가 난다”[13]는 낱자소 기반의 조합음절생성원리를 뜻하므로 훈민정음 표현은 낱글자를 중심으로 음절폰트를 조합해야 한다. 여기서 낱글자는 음절 구성 요소인 초성, 중성, 종성의 기본 자모를 말하며 음절 구성요소를 음소라 할 때 음절 글자 구성요소는 자소(字素 grapheme)라 하고 초성, 중성, 종성 낱글자는 낱자소라 한다. 그런데 문서가 낱자소 코드일 경우 모음꼴에 따라 다른 초성 낱글자의 적합한 글자꼴을 선택하는 일이 별도의 모듈에서나 운영체제 차원에서 지원되어

야 한다.

본 논문은 이러한 문제를 근본적으로 해결하기 위하여 훈민정음이 가진 과학적 원리에 따라서 훈민정음이 표현 가능한 약 399억 음절자를 생성하는 조합폰트를 설계함에 있어서 최적 설계 방안을 기존 옛한글 표현법과 비교 분석하고 또한 현행 한글부호계를 개선하는 방향을 제시하고자 한다.

2. 기존연구

한글 부호화 대상[2]에서 보면 한글의 음절, 자모, 자소를 대상으로 하고 있다. 음절은 KS C 5601-1982, KS X 1001과 KS X 1005이고, 자모는 KS C 5601-1974와 KS X 1005이며, 자소는 정음형 코드와 KS X 1005이다. 즉



그림 74 마이크로소프트 워드2000과 한글과컴퓨터 한글

KS X 1005는 음절, 자모, 자소 코드를 모두 포함하고 있다. 음절은 완성 음절형(완성형)과 조합 음절형(조합형)이 있다. 현재 옛한글을 지원하는 방식을 살펴보면, 그림 1에서 보는 바와 같이 마이크로소프트 워드2000과 한글과 컴퓨터 한글에서 한양 사용자정의 영역코드

(PUA:private user area) 코드에 따라서 옛한글 약 160 만자를 지원하고 있다. 반면에 훈민정음 폰트 연구[8,9]에 따르면 훈민정음이 표현할 수 있는 약 399억 음절 폰트를 훈민정음 창제원리에 근거하여 조합 원리에 따라서 생성하고 있다. 여기서 특히 문자론적으로 한글은 음소 문자인 로마자, 음절문자인 가나자와는 달리 한글은 음소 및 음절문자의 특성을 가지고 있어서 코드화 대상이 음소와 음절 가운데 어느 것이 옳은 지에 대한 정당성을 갖지 못한 상태에서 완성 음절형 코드체계를 선택했음을 본다. 또한 옛한글¹⁰⁾을 입력하는 입력틀(IME: Input Method Editor)로는 [4,5,6] 등이 사용되고 있는데 모두 완성음절형 코드를 생성한다. 낱자소기반 음절조합 연구 [8,9,10]는 낱자소에 대한 코드인 정음형 코드[13]를 생성하여 문서에 출력된다. 그러면 모음의 글자꼴에 맞게 설계된 1651개의 낱자소 글꼴을 설계하여 399억 음절자를 조합한다.

3. 문서코드와 폰트 코드

현재 컴퓨터에서 로마자의 경우 코드는 문서표현과 폰트표현의 경우에 그 값이 일치한다. 한글은 자판을 거친 이후 완성음절형 코드가 만들어지면 이때부터 문서와 폰트 코드가 일치된다. 현재 유니코드[12]에는 자모, 호환자모, 한글이 배치되어 있는데, 여기서 자모는 자소이고, 호환자모는 자모, 한글은 음절을 부호화 한 것이다. 옛한글 음절 표현은 자모 코드를 이용하여 음절 조합을 지원하고 있다. 여기에 반영된 옛한글 자소는 현재까지 사용된 적이 있는 것 가운데 발견된 것이고 아직 더 발견될 여지도 있고 옛한글에서 사용된 것처럼 외국어의 표현을 위하여 더 많은 자모의 조합이 이뤄질 여지가 있다. KS X 1005에서 자모(자소) 코드를 이용한 조합음절의 경우 현대 한글음절인 ‘깍’과 ‘꼭’을 조합할 경우 ‘ㄱ’은 모음의 꼴에 따라서 모양이 달라야 하지만 같은 코드가 제공되므로 초성의 꼴이 같아서 음절글꼴이 예쁘지 못하다. 반면에 모음에 따라 초성 글꼴을 달리할 경우 다른 폰트 코드를 부여하게 되어 ‘ㄱ’이 서로 다른 글자로 인식되는 문제가 있다. 이는 그림2를 보면, 음절 ‘가’의 초성 ‘ㄱ’과 음절 ‘구’의 초성 ‘ㄱ’의 문서 코드는 같더라도, 폰트로 표현시 코드가 다르다는 것을 알 수 있다. 훈민정음은 하나의 원리로써 399억 음절의 표현을 정의하고 있으나 현행 코드체계는 현대한글과 옛한글을 구분하여 음절 중심의 코드체계를 구성하며 부분 표현을 하고 있어서

근본적 해결책이 되지 못하고 있다.

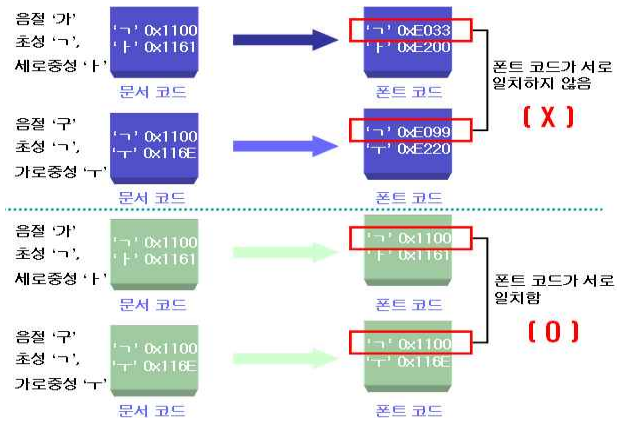


그림 75 한글의 문서 코드와 폰트 코드

이 점을 해결하려면 문서에는 정음형 코드로 표현하고 글자꼴을 출력하는 때에 한 음절의 낱자소열을 인식하여 중성의 모양에 따라서 적합한 글자꼴이 선택되어 음절글꼴을 조합하는 별도 모듈이 필요하다.

본 논문에서 가정은 일단 폰트코드로써 문서상에 직접 표현하고 훈민정음 창제당시의 모든 음절인 약 399억 글자꼴을 9개의 음절구조에 따라서 정음형 낱글자 폰트를 이용하여 조합음절글자를 만들어서 문서편집기나 웹브라우저에서 표현되도록 한다.

4. 한글 음절의 표현 범위

현행 정보교환용한글부호계 KS X 1001은 1933년 조선어학회가 정의한 초성, 중성, 종성의 조합에 따른 11172 음절을 표현하는 것을 목표로 하고 옛한글은 우선 필요한 것들만 지원하고 있다. 옛한글 처리에서 필요한 것은 입력틀과 표현 가능한 글자꼴 집합인데, 워드2000과 한글에서는 훈민정음의 표현 가능 음절인 약 399억 자의 극히 일부분만을 표현할 수 있을 뿐이다. 그 주된 이유는 훈민정음을 음소가 아닌 음절문자 중심으로 파악하고 있기 때문이다. 본 논문에서 강조하고 싶은 것은 훈민정음 창제원리는 옛한글을 정의한 바가 없다는 점이며, 28개 음소문자와 음절조합틀을 정의함으로써 “천지자연의 문자를 표현한다.”는 원리를 담고 있다. 여기서 현행 표준부호의 오류는 음절조합틀을 파악하지 못하고 단지 이들이 만들어낸 결과인 음절문자만을 인식하는데 따른다. 다시 말해서 훈민정음은 낱자소 글꼴을 조합하여 음절글꼴을 만들어내는 원리를 가지고 있는 것이다.

현재 옛한글을 지원하는 한글 코드 체계를 살펴보면, 유니코드, 한양PUA코드, 정음형 코드를 대표적으로 들 수 있다. 본 연구에서는 정음형 코드와 기존의 한글 코

10) 옛한글은 1446년에는 있었으나 1933년 한글맞춤법 통일안에서 제외 네글자(· △ ○ ㄱ)를 말함

드의 옛한글 표현 범위에 대해서 비교하고자 한다. 이에 대해서 다음 표1은 정음형 낱자소 집합을 보여주며, 표2에서는 한글 음절을 만들 수 있는 초성, 중성, 종성 개수를 보여주고 있다.

표 1 정음형 자소집합

초성	ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅈ, ㅊ, ㅋ, ㆁ, ㆁ, ㆁ, ㆁ, ㆁ, ㆁ, ㆁ (23자)
중성	ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ, ㅚ, ㅜ (11자)
종성	ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅈ, ㅊ, ㅋ, ㆁ, ㆁ, ㆁ (17자)

표 2 한글 코드별 낱자모음, 2, 3자 자모음 개수

코드 \ 자모	초성			중성			종성			합계
	낱자음	2자자음	3자자음	낱자음	2자자음	3자자음	낱자음	2자자음	3자자음	
유니코드2.0	23	59	8	11	44	11	17	54	11	238
유니코드5.1	23	86	15	11	61	22	17	91	29	355
한양PUA	23	88	13	11	65	18	17	98	26	359
정음형	23	-	-	11	-	-	17	-	-	51

위의 정음형 코드를 살펴보면, 초성, 중성, 종성이 자소 형태로만 되어 있다. 이는 복자모를 코드화 대상에 포함하지 않았기 때문이다. 훈민정음 창제원리에 따라서 합자해를 통한 낱자소만으로 나열하여 복자모를 만든다. 이에 비해 유니코드와 한양PUA코드에서는 복자모를 코드화 대상으로 삼고 있다. 이는 추가적인 자모가 나타났을 때 가나다순으로 추가할 수 없게 되는 문제점이 있다. 또한 정음형 코드가 다른 코드와 비교했을 때, 문자 집합이 작고 399억 음절자를 표현할 수 있다는 장점을 가지고 있다.

이에 대한 표3의 각 초성, 중성, 종성 개수에 따라 한글 음절을 조합할 수 있는 음절 개수를 산출할 수 있다. 산출법은 기존 연구 [7]을 참조하였다.

표 3 한글 코드 별 초성, 중성, 종성 개수

	초성	중성	종성	합계
유니코드2.0	90자	66자	82자	238자
유니코드5.1	124자	94자	137자	355자
한양PUA	124자	94자	141자	359자
정음형	23자	11자	17자	51자

표1,2에서 정음형 코드는 낱자소 초성, 중성, 종성의 개수만 제시하였다. 하지만 정음형 코드를 통해 한글 음절을 생성할 수 있는 초성 5,219자, 중성, 1,463자, 5,220자[7]를 제시하였는데(정치음, 치두음 ㅅ, ㅆ, ㅈ, ㅊ,

ㅊ, ㅊ 미추가시), 이는 코드는 낱자소만으로 지정하고, 복자모를 만들 시에는 코드를 지정하지 않았다. 단지 낱자소를 나열하여 복자모 형태로 조합할 수 있기 때문이다. 즉, 기존의 다른 코드와는 달리 최소한의 코드를 통해 모든 한글을 생성할 수 있다. 훈민정음 표현을 위한 한글코드들의 표현능력을 요약하면 표4와 같다.

표 4 코드별 옛한글 표현 개수 및 퍼센트(%)

	자소 총 개수	한글 음절 총 개수	%
유니코드2.0	238자	약 50만 음절	0.001%
유니코드5.1	355자	약 160만 음절	0.004%
한양PUA	359자	약 160만 음절	0.004%
정음형	51자	약 399억 음절	100%

정음형 코드의 낱자소만으로 모든 한글을 표현할 수 있다는 것을 입증하기 위해서는 이에 따른 폰트가 필요하다. 이전 연구 [8,9,10]에서 개발된 자소형 최적 폰트는 이를 뒷받침 해주고 있다.

그렇다면, 왜 유니코드와 한양PUA코드에서는 옛한글 표현이 부족한 것인가? 살펴보면, 우선 컴퓨터상에 표현할 폰트가 없기 때문이다. 이 말은 수용할 코드가 부족하다고 하는 말과도 같다. 유니코드에서 지정된 한글 코드는 모든 한글을 표현하기에는 턱없이 부족하다. 그렇기 때문에 한양PUA코드는 부족한 옛한글 부분을 유니코드영역 중 사용자 정의 영역 부분에 별도의 한글 코드를 지정하여 폰트 제작을 하였다. 그렇다고 한양PUA코드가 모든 한글을 표현하지 못한다. 완성형으로 모든 한글을 표현하기에는 코드와 폰트가 부족하다. 이에 대한 대책으로 조합형으로 코드와 폰트를 제작하여야 하는데, 현재 유니코드와 한양PUA코드에서는 초성, 중성, 종성의 복자모를 수용하고 있다. 이는 완성형 문제처럼 코드와 폰트의 부족 문제에 부딪힌다. 하지만 본 연구에서 제시하는 정음형 코드의 낱자소만으로 이용한 코드와 최적 폰트는 이러한 문제를 해결해 준다. 다음 5장의 자소형 최적 폰트 설계부분을 살펴보자.

5. 자소형 최적 폰트와 기존 폰트 비교

앞에서는 정음형 코드가 훈민정음 음절을 표현하는데 있어서 기존 코드보다 효율적이라는 것이다. 자소형 최적 폰트를 제작하는 데에 있어서 기존 폰트들과 비교함으로써 본 연구에서 제시한 자소형 최적 폰트가 효율적이라는 것을 입증하고자 한다.

우선 훈민정음 창제당시의 모든 한글 약 399억 음절을

표현하기 위해서는 음절 구성 형식과 형이 필요하다. 그림3에서 보는 바와 같이 기존 연구 [8,9]에서 제시한 9 가지 형식과 3가지 형을 살펴보면 다음과 같다.

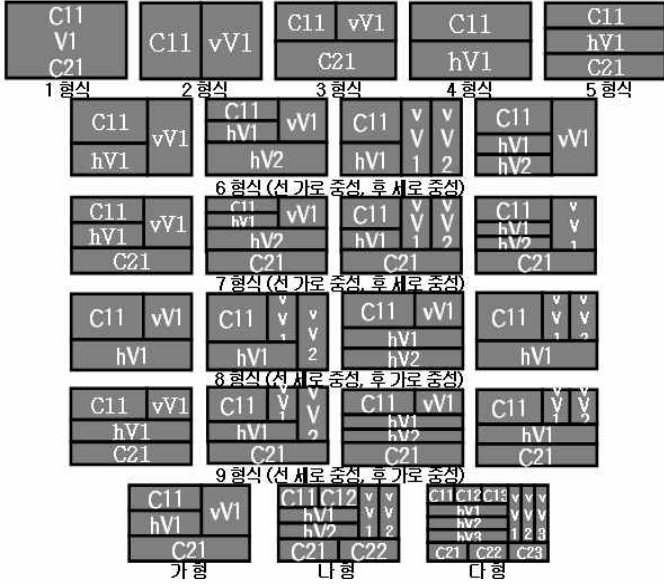
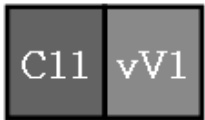


그림 76 한글 음절 9가지 형식과 3가지 형[8.9]

C11 : 초성 1, C12 : 초성 2, C13 : 초성 3
 hV1 : 가로중성 1, hV2 : 가로중성 2, hV3 : 가로중성 3
 vV1 : 세로중성 1, vV2 : 세로중성 2, vV3 : 세로중성 3
 C21 : 중성 1, C22 : 중성 2, C23 : 중성 3

한글 음절을 생성 시에는 중성, 중성에 따라 형식이 달라진다. 이는 가로 중성인지, 세로 중성인지에 따라 초성 글꼴 모양이 달라지며, 중성 입력시 또한 글자 모양이 형식에 따라 다르게 표현된다. 예를 들어, 그림4에서

- ‘가’ = ‘ㄱ’ + ‘ㅏ’
 - 2형식 초성 ‘ㄱ’ + 세로중성 ‘ㅏ’



- ‘고’ = ‘ㄱ’ + ‘ㅓ’
 - 4형식 초성 ‘ㄱ’ + 가로중성 ‘ㅓ’



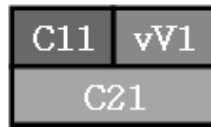
그림 4 음절 구성(2형식과 4형식)의 ‘ㄱ’ 글자모양 차이

‘가’의 ‘ㄱ’과 ‘고’의 ‘ㄱ’은 같은 글자라 하더라도 모양이 다르다. (‘가’는 2형식이고, ‘고’는 4형식을 보여주고 있

다.) 이는 음절 구성형식에 따라 폰트 모양이 다르게 표현된다.

하지만, 현재 옛한글 입력기에서 조합형의 옛한글을 표현하는 데에 있어서 음절 구성 형식과 형에 맞춰지지 않고 단지 한 음절 구성 형식에 맞춰져 출력되고 있는 실정이다. 예를 들면,

- ① 초성 복자모 ‘ㄲ’ + 중성 자소 ‘ㅏ’ + 중성 ‘ㄴ’ -> 𑖏
- 3형식 초성 + 세로중성 + 중성



- ② 초성 복자모 ‘ㄲ’ + 중성 자소 ‘ㅓ’ + 중성 ‘ㄴ’ -> 𑖏
- 5형식 초성 + 가로중성 + 중성



그림 5 음절 구성(3형식과 5형식)의 글자모양 문제점

그림5를 보면 ①은 세로중성에 맞춰 글자 모양이 갖춰져 있지만, ②는 가로중성에 대한 초성 ‘ㄲ’의 모양이 가로중성에 맞지 않다. 이는 음절 구성 형식에 맞춰진 폰트가 없기 때문이다. 단지 한 형식에 맞춰져 있기 때문에 이러한 문제점이 발생한다. 즉, 폰트 자체가 이러한 음절 구성 형식과 형에 맞춰서 제작되어 있지 않기 때문이다. 본 연구의 자소형 최적 폰트 1,651자는 이러한 한글 음절 9가지 형식과 3가지 형에 맞춰서 폰트가 제작되었다. 그 예로 그림6을 보면, 초성 ‘ㄱ’의 음절 구성 형식의 9가지 글자 모양을 보여주고 있다. 또한, 초성 ‘ㄱ’의 2형식의 3가지 형에 따른 글자 모양이다. 형에 따라 날자소만으로 복자모로 조합할 수 있다.

하지만, 기존의 MS워드나 한글에서 조합형 옛한글 약 160만자를 표현한다고 하지만, 단지 한 형식에 맞춰진 글꼴을 표현하는 게 전부다. 이유는 폰트가 한 형식에 맞춰서 제작되어 있기 때문이다. 또한 완성형 옛한글 약 5천자를 제공하고 있지만, 본 연구에서 제시한 약 399억 음절에 비하면 턱없이 부족하다. 즉, 현재 제작된 폰트들이 이러한 문제점을 해결하기 위해서 음절 구성 형식에 맞춰 폰트를 제작하려면, 폰트 글자 모양이 기존에 제시한 자소의 모양이 몇 십 배의 제작이 필요할 것이다. 하지만 본 연구의 자소형 최적 폰트는 단지 날자소 1,651자만으로 이 문제점을 해결할 수 있다.

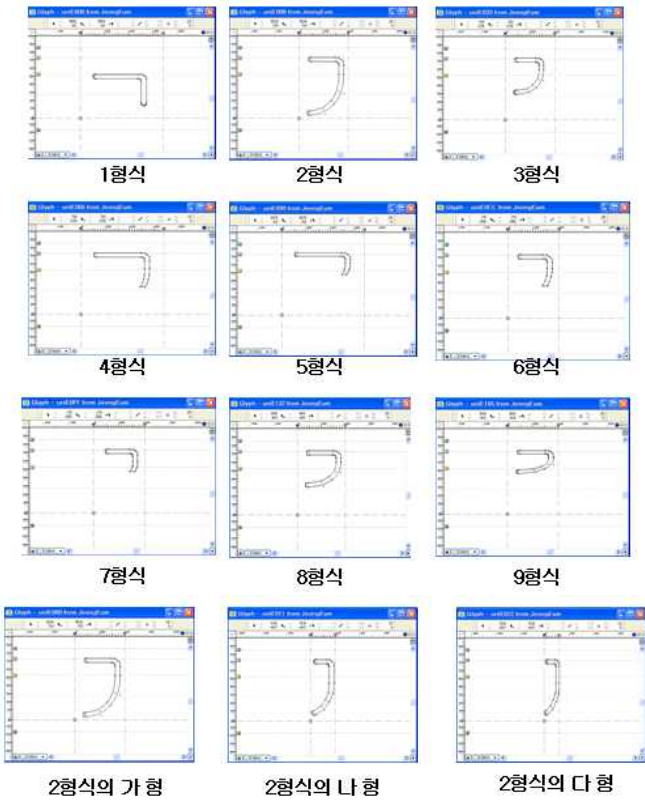


그림 6 초성 ‘ㅜ’의 음절 구성 9가지 형식과 3가지 형

6. 결론

본 연구에서는 현재 컴퓨터에서 한글 및 옛한글 그리고 훈민정음을 표현하는 방안과 음절의 표현범위에 관한 연구들을 살펴보았다. 특히 날자소를 기반으로 한 훈민정음 399억 음절 표현 방법 연구에서 최적의 날자소 폰트의 수 1651개로 그 표현이 가능함을 살펴보았다. 그리고 현행 운영체제는 텍스트 코드와 폰트 코드가 일치하는 것을 지원하기 때문에 폰트코드를 이용해서 모음의 글자꼴에 따라서 다른 글자꼴을 가지고 399억 음절 조합폰트 생성의 가능성을 기존의 방법과 함께 비교분석해 보았다.

결론적으로 1651개 날자소 폰트로 훈민정음 창제원리에 따른 약 399억 음절폰트를 조합으로 생성하는 방법이 기존의 다른 방법에 비하여 가장 근본적인 해결책이며 글자꼴의 미려함이나 조합 가능한 음절의 범위로 보아서 최적한 폰트 설계 방법으로 판단된다. 기존의 한글 및 옛한글 표현법은 훈민정음 창제원리로 볼 때 50만 또는 160만 음절을 표현할 수 있는 임시방편적 방법에 불과하며 근본적인 해결책이 되지 못한다. 한글은 훈민정음 창제원리에 따른 구현을 통하여 날자소 코드인 정음형 코드로 문서를 표현하고 음절글자꼴은 화면이나 프린터에 표시할 때 표현하면 된다. 따라서 폰트파일에 접근

할 때 기본적으로 음절조합 자동틀을 포함시켜서 한 음절 단위로 문자열을 입력하면 약 399억 음절의 조합음절글꼴이 반환되도록 시스템 차원에서 지원되어야 한다. 또한 본 연구에서 제시한 1651개의 자소형 최적 폰트로써 399억 음절자를 조합해 냄으로써 기존의 구현들보다 훨씬 효율적이라 사료된다. 따라서 현행한글정보교환용 코드체계는 임시방편적이며 근본적 해결책이 되지 못하므로 훈민정음 창제원리에 따른 자소 중심으로 코드를 제정하고, 음절자는 그 보조수단으로 삼으며, 폰트는 날자소로 설계하고 폰트파일에 접근할 때 음절조합 모듈을 거처도록 조합음절폰트를 지원해야 한다고 본다.

추후 연구로는 자소형 최적 폰트를 제작하는 데에 좀 더 줄일 수 있는 방안을 연구하고 기존 코드와의 호환성에 대한 방안을 연구할 계획이다.

참고문헌

[1] 변정용, “국제 문자 부호계에 포함된 한글 부호계의 개선 방안”, 98 한국정보과학회 가을 학술발표 논문집 Vol. 25 No.2

[2] 변정용, “훈민정음 원리의 공학화에 기반한 한글 부호계의 발전방향”, 1994 정보과학회지 제12권 8호

[3] 변정용, “한글 부호계 제정에서 반복되는 시행착오의 고리를 끊으려면”, 98 정보처리 제5권 5호

[4] 김용목, 날개셋 한글 입력기, <http://moogi.new21.org/>

[5] 하늘 입력기, <http://cvs.osxdev.org/view/Hanulim/?root=cvs>

[6] 제주 사이버 삼다관 옛한글, <http://www.jejusamda.com/>

[7] 변정용, “훈민정음 원리에 따르는 우리글 코드 제정 방안”, 96 코리언 컴퓨터처리 학술대회 논문집

[8] Byun JY, "A Representation of Hangeul Syllables defined in Hunminjeongeum", MMM2007 LNCS4351, SpringerVerlag, 2007

[9] 변정용, 김경옥 “훈민정음 원리에 기반한 자소형 최적 폰트 개발”, 한국정보처리학회 2007년 춘계학술대회, VOL. 14 NO. 01 pp. 0690 ~ 0693, 2007

[10] Kim KW, Byun JY "An Implementation of Optimized Hangeul Font Set for Multilingual IM Engine", ALPIT2007, VOL. 6 NO. pp. 200~205, 2007

[11] 안대혁, 박영배, “유니코드 환경에서의 올바른 한글 정규화를 위한 수정 방안”, 정보과학회지: 소프트웨어 및 응용 제34권 제2호(2007.2.), p169-177

[12] The Unicode Consortium, "The Unicode Standard 4.0", Addison-Wesley, 2003

[13] 강신향, 훈민정음연구,성균관대학출판부,1991