

무선 환경에서의 웹 콘텐츠 서비스를 위한 HDML 전용 저작도구의 설계 및 구현

박 영 충[†] · 박 병 하[†] · 김 건 희[†] · 신 동 일^{††} · 신 동 규^{††}

요 약

본 논문에서는 HDML(Handheld Device Markup Language)로 구성된 문서를 보다 손쉽게 효과적으로 생성하고, 관리될 수 있는 HDML 전용 저작도구를 설계 구현하였다. HDML은 무선 환경에서 소형 휴대용 장비를 이용해서 웹에 대한 접근을 가능하게 하는 마크업 언어이다. 본 저작도구의 개발과정에서, 휴대용 장비의 기본적인 제약과 HDML 데크/카드의 표현과 키 카드의 행동의 정의, 카드 사이의 하이퍼링크(Hyper-Link)의 관리의 같은 HDML 문서만이 가지는 특성으로 인해서 발생하는 기술적인 난제들을 극복하였다.

Design and Implementation of HDML Authoring Tool for Web Contents Services on Wireless Environments

Young-Choong Park[†] · Byoung-Ha Park[†] · Gun-Hee Kim[†] · Dong-Il Shin^{††} · Dong-Kyoo Shin^{††}

ABSTRACT

In this paper, we present a design and implementation of HDML(Handheld Device Markup Language) authoring tool that can create and manage HDML documents easily and conveniently. HDML is a markup language which enable access to the Web using small mobile equipment through mobile networks. The basic limitation of mobile equipments and technical difficulties such as representation of HDML deck/card, definition of activation in each card and management of hyper-link between cards are solved, in the development process of the HDML authoring tool.

1. 서 론

인터넷 기술의 발달로 인해서 사용자는 언제 어디에서나 개인용 컴퓨터와 네트워크 연결이 있는 곳이면 자신이 원하는 정보를 손쉽게 취득할 수 있게 되었다. 현재 웹에서 정보를 얻을 수 있는 장비로 쓰이는 개인용 컴퓨터들은 통신만을 위해서 이용하기에는 기능이 복잡하고 부피가 크며 무거우므로, 소형 휴대용 장비를 이

용해서 다양한 정보를 취득하고자 하는 요구가 점차 증가되고 있다. 특히 휴대용 전화기나 PDA(Personal Digital Assistant)와 같은 무선 터미널(Wireless Terminals)을 이용한 웹 정보에 대한 접근이 절실히 요구되고 있다 [17]. 현재 이동통신망에 가입하여 휴대용 전화기나 PDA 같은 휴대용 무선장비를 이용하는 사용자의 수는 국내에서만 2000만 명을 넘었으며 앞으로 개인용 컴퓨터 사용자보다 더 많은 수의 사용자가 이를 이용하게 될 것으로 보인다. 이러한 휴대용 무선장비를 통한 웹 콘텐츠 이용지의 확대는 데이터통신의 급격한 증가를 가져올 것이고, 엄청난 국내외적인 수요를 불러일으킬 전망

[†] 준 회원 : 세종대학교 대학원 전산학과
^{††} 종신회원 : 세종대학교 컴퓨터공학과 교수
논문접수 : 1999년 12월 29일, 심사완료 : 2000년 2월 16일

이다. 휴대용 무선장비를 위한 웹 콘텐츠(contents) 제공 서비스는 현행 휴대전화 서비스뿐 아니라 차세대서비스로 일컬어지는 TMT2000에서와 같은 멀티미디어를 중심으로 하는 통신서비스에서도 필수적인 기능이 될 것이다[19].

기존의 웹 기반 응용 업무를 지원하기 위한 언어인 HTML(Hyper Text Markup Language)로서는 제한된 제원[15]을 가진 휴대용 무선장비에서 많은 정보를 효율적으로 나타낼 수 없다. 이에 대한 대안으로 휴대용 무선장비에서의 효과적인 웹 접근을 위한 언어인 HDML(Handheld Device Markup Language)[1, 16]이 Phone-Com(사립 초기에는 Unwired Planet이라는 회사 명칭을 이용했음)에 의하여 개발되었고, 1997년 5월 W3C(World Wide Web Consortium)에 휴대용 무선장비의 웹(Web) 접근에 관한 표준안으로 제출되었다. 현재 휴대용 무선장비에서의 웹 접근을 위한 표준제정을 위하여 Ericsson, Motorola, Nokia 및 Phone.Com 등이 WAP(Wireless Application Protocol) Forum을 형성하여 HDML을 기반으로 하는 WML(Wireless Markup Language)을 포함하는 무선 응용 프로토콜(WAP)의 표준제정을 추진하고 있다[2, 3]. WML은 HDML의 기본구조인 데크와 카드(Deck/Card) 단위의 작동개념을 채택하여 데크와 카드들간의 하이퍼링크(hyper link)등 HDML의 작동원리를 그대로 승계하고 있으며 현재 WML 형식 1.1 버전이 발표되었다. 휴대용 무선장비를 이용한 웹 콘텐츠 제공 서비스를 위한 각종 응용프로그램의 제작이 전세계적으로 이루어 졌으며, 그 대부분은 HDML을 이용한 것이고 최근에는 WML을 이용한 응용프로그램의 제작이 이루어지고 있다.

본 논문에서는 HDML로 구성된 문서를 보다 손쉽게 효과적으로 생성하고, 관리할 수 있는 HDML 전용 저작도구의 설계와 구현에 대하여 기술한다. 본 논문에서는 휴대용 장비의 기본적인 제약과 HTML의 페이지에 해당하는 HDML의 데크 안에 존재할 수 있는 수많은 카드의 표현과 각 카드의 행동의 정의 그리고 카드 사이의 하이퍼링크(Hyper-Link)의 관리와 같은 HDML 문서만이 가지는 특성으로 인해서 발생하는 기술적인 난제들을 사용자 입장에서 보다 손쉽게 작성하고, 관리할 수 있는 HDML 전용 저작도구를 구현하였다.

논문의 구성은 다음과 같다. 2장에서는 WAP과 HDML 작동환경에 관해 기술한다. 3장에서는 HDML 전용 저작도구의 설계 및 구현에 대해 기술하고, 마지막으로 4

장에서 결론 및 향후과제를 제시한다.

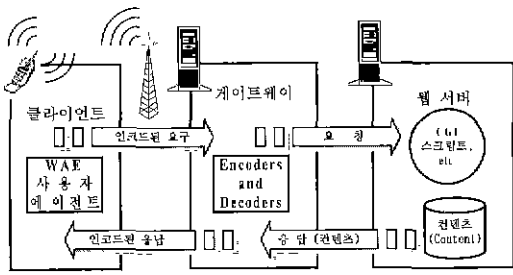
2. 관련연구

현재까지 이 분야의 연구는 통신사업자 및 통신사업자의 업계 표준화 기구인 WAP Forum에 의해 주로 이루어졌다. Phone.Com이 마크업 언어 HDML과 무선망과 유선망을 연결하는 프로토콜인 HDTP(Handheld Device Transport Protocol)를 W3C에 제안함으로써 연구 개발이 확산되기 시작되었으며, 이를 기초로 하여 WAP과 WML이 WAP Forum에 의해 무선 인터넷 표준으로 개발되고 있다. HDML 작동환경과 WAP은 유사한 개념이므로 WAP 환경을 소개하고 HDML 작동환경에 대하여 기술한다.

2.1 Wireless Application Protocol

WAP(Wireless Application Protocol)[3]는 네트워크 기술과 무선 데이터 기술 및 인터넷의 빠른 발전으로 생겨난 프로토콜이다. WAP의 전반적인 구조는 웹을 기반으로 하여 만들어 졌기 때문에 많은 부분에서 유사한 면을 볼 수 있다. 웹 구조는 매우 유연하고 강력한 프로그래밍 모델을 제공한다. 즉, 콘텐츠는 표준 데이터 포맷인 HTML로 표현되며 웹 브라우저로 작업을 처리한다. 클라이언트에서 사용자가 원하는 데이터 객체를 웹 서버에 요청하면, 웹서버는 표준 포맷을 사용하여 코드화된 데이터를 가지고 클라이언트에 응답하게 된다. WAP 프로그래밍 모델은 인터넷 웹 프로그래밍 모델과 매우 비슷하며 (그림 1)과 같이 표현될 수 있다. 이 모델은 WAP 응용 개발자들에게 몇 가지 편리함을 제공한다. 즉, 사용자에게 익숙한 프로그래밍 모델이라든지, 안정된 구조, 그리고 현재 사용되고 있는 도구인 웹 서버나 XML(eXtensible Markup Language) 저작도구 등과의 호환능력 등을 들 수 있다.

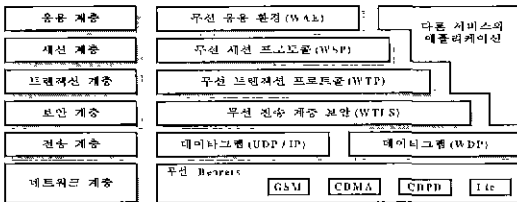
WAP의 콘텐츠와 응용프로그램은 우리에게 익숙한 웹 콘텐츠 형식(WWW Contents Format)을 기반으로 해서 만들어진다. WAP 콘텐츠는 웹 통신 프로토콜을 기반으로 하여 만들어진 통신 프로토콜인 WTP(Wireless Transaction Protocol)를 사용하여 전송된다[8]. 무선 터미널에서는 사용자 인터페이스인 마이크로 브라우저를 사용하여 전체 응용프로그램을 제어하는데, 이것은 웹에서 이용되는 표준 웹 브라우저와 비슷한 기능을 가진



(그림 1) WAP 프로그래밍 모델

다. WAP 컨텐츠 유형(contents type)과 프로토콜은 통신시장과 모든 휴대형 무선 장비를 위해 최적화 되었으며, WAP은 무선 도메인과 웹 서버간의 원활한 연결을 위해 프록시(proxy) 기술을 사용한다. WAP 프록시는 전형적으로 다음의 기능을 포함하고 있다

- 프로토콜 게이트웨이(gateway) : 이것은 WAP 프로토콜 스택(stack)인 WAE (Wireless Application Environment)[4]와 WSP(Wireless Session Protocol)[7]와 WTP(Wireless Transaction Protocol)[8] 및 WTLS (Wireless Transport Layer Security)[9], 그리고 WDP (Wireless Datagram Protocol)[10]를 웹 프로토콜 스택, 즉 HTTP와 TCP/IP로 전환하는 기능을 갖는다. WAP 구조는 이동 통신 장치를 위한 응용프로그램 개발을 위해 (그림 2)에서 보는 바와 같이 확장 가능한 프로토콜 환경과 전체 네트워크의 계층화된 구조를 제공한다

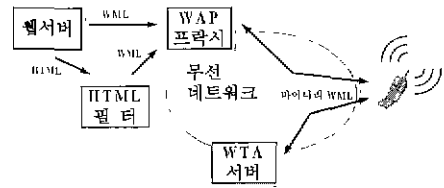


(그림 2) WAP 스택(Stack)

- 컨텐츠 부호화기와 복호화기 : 컨텐츠 부호화기는 네트워크 상에서의 데이터의 크기를 줄이기 위해 WAP 내용을 간결하게 부호화된 포맷으로 해석한다

WAP 프록시는 표준 웹 서버 상에서의 컨텐츠와 응용프로그램이 추가 되도록 즉, 우선권을 가지도록 하며,

CGI(Common Gateway Interface) 스크립팅과 같은 안정된 웹 기술의 사용을 가능하도록 한다. WAP 표준에서는 하나의 웹 서버와 WAP 프록시 그리고 WAP 클라이언트를 정의하고 있으므로 WAP 구조는 다른 구성 요소들을 더 쉽게 지원 할 수 있으며, WAP 프록시 기능[5]을 포함한 웹 서버의 구현을 가능하게 해준다



(그림 3) WAP 네트워크의 동작

일반적인 WAP 네트워크의 동작은 (그림 3)과 같다 WAP 클라이언트는 무선 네트워크 망에서 2개의 서버와 통신을 한다. WAP 프록시는 WAP 요청을 웹 요구로 바꾸고, 그것에 의하여 WAP 클라이언트가 웹 서버에 자신의 요청을 보낼 수 있도록 해준다 또한 WAP 프록시는 클라이언트가 이해 할 수 있는 축소된 바이너리 포맷으로 웹서버로부터 응답을 코드화 한다. 만일 웹 브라우저가 WML로 만들어진 WAP 컨텐츠를 제공한다면, WAP 프록시는 그것을 웹 브라우저로부터 직접 가져오지만 만일 웹 브라우저가 HTML로 만들어진 웹 컨텐츠를 제공한다면, 특정 필터인 HTML 필터가 웹 컨텐츠를 WAP컨텐츠로 바꾸기 위해 사용된다. WTA[5, 6]는 전화 서비스와 이를 위한 프로그래밍 인터페이스를 제공한다.

WAP컨텐츠는 WML 및 WMLScript로 작성된다. WML은 태그에 기초한 문서언어이며, XML에 기반을 둔 일반적인 마크업 언어이다[2, 12]. WML은 HTML과 HDML이 가진 능력을 공유한다 WML은 XML 문서 형식으로서 나타내어지며, 전화나 다른 무선이동터미널과 같은 제한된 능력을 가진 장치 상에서 프리젠테이션과 사용자의 상호작용을 나타내는데 최적화 되어있다. WML은 HDML로부터 적용된 몇 가지 요소를 바꾸고 HTML과 비슷한 요소들 가진 새로운 일부 요소를 추가하였다. WML에서도 HDML과 마찬가지로 카드(card)와 데크(deck)라는 개념을 사용한다. 이것은 응용프로그램이 다중 키드로 만들어진 문서(일반적으로 데크라고 말한다)를 나타낼 수 있게 설계되어 있다. 논리적으로 시

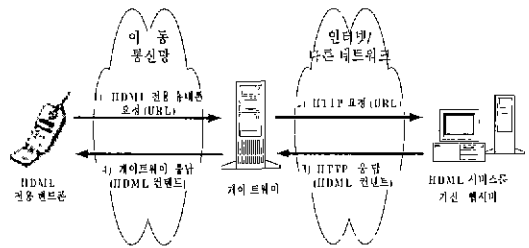
용자는 WML 키드를 통해서 웹을 탐색할 수 있다. 카드의 내용을 다시 봄으로써, 요구된 정보로 들어갈 수도 있고, 선택을 만들 수도 있으며 다른 카드로 움직일 수도 있다. 카드 안에 있는 명령어는 특별한 상호작용에 의해 필요에 따라 웹 서버 상에서 서비스를 실행할 수 있다. WML 데크는 웹 서버 상에서 정적 파일로 저장될 수 있으며, 또한 동적으로 웹 서버 상에서 실행되는 내용 생성기에 의해 생성될 수도 있다. 데크 안에 있는 각각의 카드는 특별한 사용자 상호 작용을 위한 기능을 가지고 있다. WMLScript는 표준 브라우저와 행동 능력을 가진 WML의 표현 능력을 강화시켜 배우기 쉽게 고안된 스크립트 언어이다[11] WMLScript는 진보된 사용자 인터페이스를 지원하며, 클라이언트에 기능을 추가하고, 이며한 장치와 그것의 주변기에 귀속하기에 편리한 기법 등을 제공한다. 또한 웹서비스의 왕복 여정에 따른 시간지연을 감소시킨다. WMLScript는 WML 데크에 프로시저 구문을 추가하기 위한 표준 수단을 제공한다.

2.2 HDML 작동환경

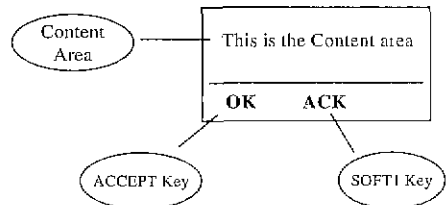
HDML은 휴대용 장비로부터의 효과적인 웹의 접근을 위하여 미국의 Phome.Com에 의하여 개발된 언어로서 1997년 5월 W3C(World Wide Web Consortium)에 무선장비의 웹 접근에 관한 표준안으로 제출되었다[13]. HDML 태그(tag)에 기반을 둔 언어이며, 기존의 HTML과는 달리 제한된 제원을 가진 시스템 상에서의 데이터의 표현과 사용자와의 상호작용을 나타낸다. HDML을 근간으로 현재 W3C에서 WML의 표준화를 진행중에 있으며, 현재 전세계적으로 다수의 무선 데이터 서비스 응용 프로그램이 HDML로 구성되어 있다. HDML을 지원하는 HDML 플랫폼 상에서 HDML 플랫폼의 구성요소로는 HDML 전용 휴대폰, HDML 문서, 휴대형 장비와 웹을 연결해주는 게이트웨이(gateway) 및 HDML을 위한 웹서비스를 들 수 있다. 각 구성요소들의 관계는 (그림 4)에 나타나 있다. 앞 절에서 설명한 WAP 환경과 유사하다.

HDML 전용 휴대폰(HDML phone)은 브라우저(browser)를 실행하는 데이터 교환이 가능한 휴대용 장비이다. HDML 전용 휴대폰에서 사용하는 마이크로 브라우저는 휴대용 무선장비가 HDML을 해석하거나 게이트웨이(gateway)와 통신할 수 있도록 해주는 소프트 모듈이며, 진보된 이동 폰 서비스(Mobile Phone Service)를 위

해 이동 패킷 데이터 기술(Mobile Packet Data Technology)[18]을 이용하고, 웹 브라우저(web browser)와 매우 비슷한 구조를 가진다. 사용자는 (그림 5)에서 보는 바와 같이 HDML 전용 휴대폰의 키패드를 이용하여 데이터를 교환할 수 있으며 사용자가 다른 위치로 움직일 때, HDML 전용 휴대폰은 다른 카드를 적재하며 디스플레이와 입력 사양(input option)을 바꾸게 된다



(그림 4) HDML 플랫폼 구성요소

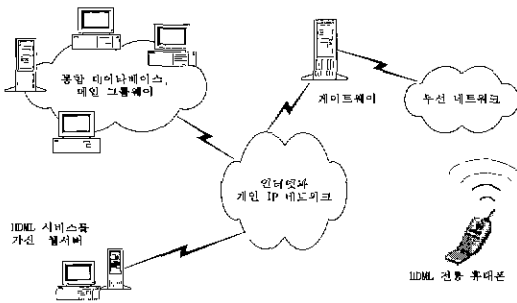


(그림 5) HDML 전용 브라우저의 구조

게이트웨이는 HDML 플랫폼의 핵심부이며, HDML 전용 휴대폰 사용자들의 데이터베이스를 관리한다. 게이트웨이는 HDML 전용 휴대폰의 요청을 웹 서버에 전달해주며 웹 서버로부터의 응답을 HDML 전용 휴대폰에 전달해 준다. HDML 휴대폰이 게이트웨이와 함께 동작할 수 있도록 하기 위해서는 게이트웨이의 데이터에 등록해야만 한다. 사용자가 HDML 전용 휴대폰의 키(key)에 전달된 URL(Uniform Resource Locator) 요청을 눌렀을 때, HDML 휴대폰은 HDTP(Handheld Device Transport Protocol)를 사용하여 게이트웨이에 그 요청을 보내게 된다. HDTP는 휴대형 장비를 이용한 무선망에서의 통신을 위해 최적화된 개방형 프로토콜로서, 클라이언트와 서버 사이의 원격작용을 위한 세션 계층의 프로토콜이다. (그림 6)은 게이트웨이를 이용한 데이터의 전달방법을 보여준다.

웹 서비스는 HDML과 HTTP(HyperText Transfer

Protocol) 요청에 대한 응답 내용을 제공하는 웹서버 응용 프로그램을 말한다. 웹서비스의 예로서는 다음과 같은 것들이 있다.



(그림 6) 게이트웨이를 이용한 데이터 전달방법

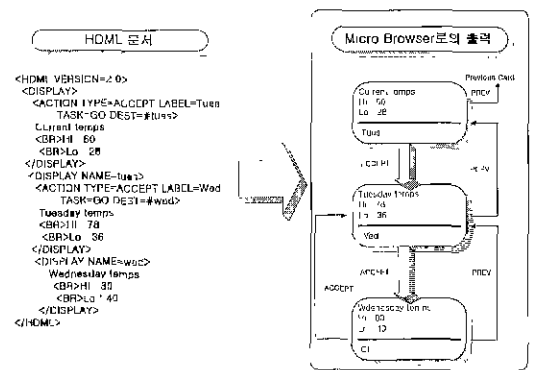
- 보안 추적 시스템(Securities tracking systems)
- 주문 추적 시스템(Order tracking systems)
- 기상 및 도로정보 보고 (Weather and road condition reports)

웹서버는 CGI(Common Gateway Interface)가 HTML을 생성하는 것과 같은 방법으로 유동적으로 HDM을 생성한다. 예를 들면, 데이터베이스나 다른 외부 데이터 제공자에게 질문함으로써 웹 상에 원래 존재하는 HTML 내용으로부터 HDM을 생성하는 웹서버를 쉽게 개발할 수 있다. 이 단계에서 다음과 같은 풀(PULL) 거래(transaction)와 푸시(PUSH) 거래를 원하는 웹서버에서 실행할 수 있다.

- 풀(PULL) 거래 : HDM 전용 휴대폰의 요청에 응답 서비스를 제공
- 푸시(PUSH) 거래 : HDM 전용 휴대폰에 비동기적으로 데이터를 보내는 서비스 제공

HDM은 태그에 기반을 두며 제한된 재원을 가진 장치 상에서의 표현과 사용자 상호작용을 보이는데 최적화 되어 있는 언어이다. HDM은 카드(card)와 데크(deck)라는 개념을 사용한다. 이것은 응용프로그램이 다중 카드로 만들어진 문서(일반적으로 이를 데크라고 하며 이는 HTML의 페이지에 해당됨)를 나타내도록 하는 설계를 포함한다. 웹서버가 HDM 전용 휴대폰에 전달할 수 있는 HDM의 가장 작은 단위는 데크이다. 데크는 하나 이상의 카드를 포함하여 HDM 전용 휴대폰이

데크를 받아들일 때, 데크에 있는 첫 번째 카드를 실행한다. HDM 전용 휴대폰은 카드가 제공하는 정보를 보여주며 사용자가 카드에 응답하도록 해준다. 카드의 종류에 의존하기 때문에 사용자는 텍스트를 넣거나 옵션을 선택하거나 기능 키를 누름으로써 응답할 수 있다. 따라서 HDM의 기본단위는 카드이며, 이것은 전화와 사용자 사이의 일관된 상호 행동을 나타낸다. HDM이 제공하는 카드의 종류로는 디스플레이 카드(display card), 입력 카드(entry card), 선택 카드(choice card) 그리고 노디스플레이 카드(no-display card)가 있는데 이 네 가지 종류의 카드를 이용하여 여러 개의 카드를 하나의 데크 안에 구성함으로써 사용자가 원하는 형태의 서비스 구조를 만들 수 있다. 각각의 카드들은 다음과 같은 기능을 한다. 먼저 디스플레이 카드는 정보를 단순히 사용자에게 보여주며, 입력 카드는 사용자에게 메시지를 보여주고 사용자가 데이터를 입력할 수 있도록 하기 위해서 사용된다. 선택 카드는 메뉴 리스트를 보여 주어 사용자가 원하는 메뉴를 선택할 수 있도록 지원하며, 노디스플레이 카드는 실제로는 디스플레이카드와 똑같이 실행은 되지만 사용자에게 보이지는 않는다. 노디스플레이 카드는 변수를 초기화한다든지, 또는 사용자 인터페이스를 가지지 않는 하부 활동(activity)을 실행하는 것과 같은 특별한 기능을 수행하는데 사용된다.



(그림 7) HDM 문서와 브라우징

HDM 프로그램의 예를 (그림 7)에서 살펴보자. 이 예제에서는 1개의 데크 안에 3개의 카드를 사용하여 날씨 정보를 나타내고 있다. 휴대폰이 데크를 맨 처음 실행할 때, 휴대폰은 데크 안에 있는 처음의 카드인 현재의 기온이 담긴 디스플레이 카드를 사용자에게 보여준

다 휴대폰이 계속해서 종속되는 하부 카드를 사용자에게 보여주는 순서는 사용자의 요구 사항이 표현된 텍스트 속의 HDML 문법에 의해 나타나는 링크의 진행과정에 의존한다. 여기서 정의된 텍스트는 그림에서 보이는 대로 화살표의 지시에 따른 링크의 진행과정을 보여준다. 사용자가 첫 번째 카드에 있는 ACCEPT 키를 누르면, 휴대폰은 두 번째 카드를 사용자에게 보여준다. 이러한 행동은 첫 번째 카드에 있는 다음과 같은 <ACTION> 문장에 의존하여 행동하게 된다. 이 <ACTION> 문장은 HDML 전용 휴대폰에 장착된 브라우저의 행동을 규정한다. 다음은 액션(ACTION) 문장의 예이다

```
<ACTION TYPE=ACCEPT LABEL=Tues
TASK=GO DEST=#tues>
```

이 문장은 휴대폰에 있는 ACCEPT 키의 라벨을 "OK"에서 "Tues"로 바꾸도록 지시하며, 사용자가 이 키를 눌렀을 때, 텍스트 안에 있는 두 번째 카드의 이름인 "Tues"로 이동하게 된다. 마찬가지로 두 번째 카드에 있는 <ACTION> 문장은 휴대폰에 있는 ACCEPT 키의 라벨을 "OK"에서 "Wed"로 바꾸도록 지시하며, 사용자가 ACCEPT 키를 눌렀을 때, 카드의 이름이 "Wed"인 카드로 이동하게 된다. ACCEPT 키의 초기 행동은 이전 카드로 되돌아가는 것이다. 예를 들어, 위의 예제에서 텍스트 안에 있는 세 번째 카드는 ACCEPT 키가 어떻게 행동할 지를 결정하는 <ACTION> 문장을 가지고 있지 않기 때문에, 이 세 번째 카드에서 사용자가 ACCEPT 키를 누른다면, 휴대폰은 이전의 카드로 되돌아가게 된다.

3. HDML 전용 저작도구의 설계 및 구현

HTML 응용업무가 몇 개의 페이지로 구성되는 것과 달리, 간단한 HDML 응용업무의 구성에도 다수의 텍스트와 텍스트내의 많은 수의 카드(Card)들이 이용되기 때문에 각각의 카드에 대한 콘텐츠를 제작, 수정, 관리, 표현하는데 많은 어려움이 따른다. HDML은 기존의 웹에서 사용되고 있는 HTML과 같이 태그에 기반을 두고 있다는 점에서는 유사한 특징을 가지지만, 태그의 행동이나 정의 그리고 링크에 있어서 매우 다른 구조를 가지고 있다. 따라서 HDML 카드의 생성과 링크 측면에서의 기술적인 문제들이 발생한다.

HDML 카드의 생성 측면으로 볼 때, HDML의 4가지

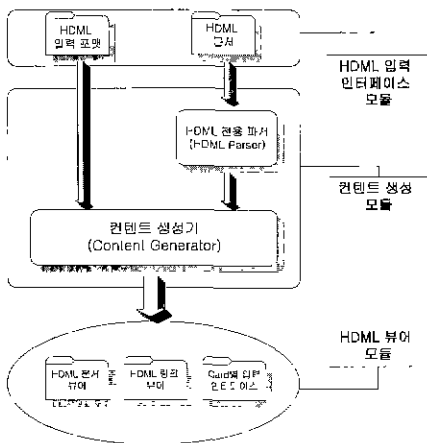
카드, 즉, 디스플레이 카드, 입력카드, 선택카드, 노디스플레이카드는 행동 방식이나 사용되는 형태가 명확하게 구분되어 있으며, 상호간에 종속(nestod)되지 않고 독립적으로 동작하는 특성을 가지고 있다. HDML에 관련된 전용 저작도구는 현재까지 발표되지 않았지만, 기존의 도구인 Allare사의 Cold Fusion3.1[20]에서는 HTML 저작 뿐만아니라 간단한 HDML의 조작성 가능하게 되어 있다. Cold Fusion3.1을 이용하면, HDML 문서를 저작할 때, 에디터 상에서 사용자가 소스를 직접 고쳐야 한다. 즉 에디터를 이용하여 HDML 텍스트를 구성하면 텍스트에 해당되는 태그들이 나타나고, 태그를 선택하면 선택된 태그가 에디터 창에 소스 형태로 입력되므로 소스를 직접 고쳐야한다. 카드 빌 동작의 형태가 명확히 구분되며, 텍스트를 기반으로 한다는 요인 때문에 태그 기반의 저작방식도 설득력은 있다. 그러나 저작도구가 필요한 근본적인 이유는 사용자로 하여금 해당 언어의 태그에 대한 지식이 없이도 간편하게 자신이 원하는 작업을 할 수 있도록 도와주며, 비주얼 환경을 만들어 줌으로써 사용자가 용이하게 작업할 수 있는 환경을 만드는 데 있다. Cold Fusion3.1과 같은 기존의 저작도구는 사용자가 HDML 태그 및 문법에 대해 완전히 숙지하고 있는 상태에서만 작업이 가능하며, GUI(Graphic User Interface) 환경을 지원하지 않기 때문에 사용자가 HDML의 모든 페이지를 직접 타이핑해야 한다.

하이퍼링크 측면에서 볼 때, HDML은 HTML보다 매우 복잡하다. HTML에서는 각각의 페이지를 링크하지만, HDML에서는 텍스트 뿐만 아니라 텍스트 안에 있는 각각의 카드까지 링크가 가능하기 때문이다. HDML에서 카드는 그 자체만으로 HTML의 작은 페이지와 같은 기능을 한다. 텍스트는 서로 관련된 카드들을 모아서 전송하기 위한 전송 수단이며, 관련 카드들을 구별하고, 식별하기 위한 단위라고 할 수 있다. 이런 HDML의 특성을 효과적으로 표현하기 위해서는 링크에 대한 관리가 필수적이라고 할 수 있다. Cold Fusion3.1에서는 이런 링크에 대한 관리의 측면을 고려하지 않았다.

이러한 난점을 효과적으로 해결하기 위하여 다음의 사항을 고려하여 저작도구의 설계를 하였다. WYSIWYG 기능을 지원하며 카드 상호간의 비종속적, 독립적 측면을 보다 효과적으로 나타내기 위해 4개의 내부프레임을 이용하여 각각의 카드를 표현하였으며, 내부 프레임 안에는 휴대폰의 화면에 표시될 텍스트만 입력하게 하였다. 태그나 속성은 일부는 자동 생성하고 일부는 마우

스의 클릭만으로 만들 수 있게끔 하여 WYSIWYG을 통한 효과적인 GUI 환경을 제공하도록 하였다. 한편 링크에 대한 관리에는 트리 구조를 이용하였다. 윈도우 95/98의 디렉토리 구조처럼, 링크되는 데크나 카드를 하부로 계속 중속시켜나가는 방식을 취하였다.

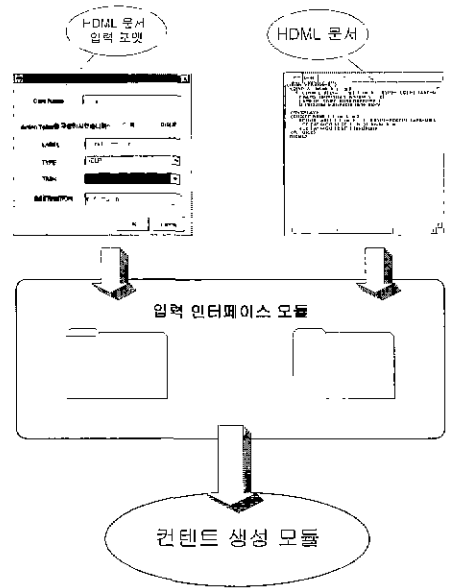
위의 설계를 지원 하기 위하여, HDML 전용 저작도구는 크게 3개의 논리적 모듈로 구성되었다. 사용자로 하여금 각각의 HDML 태그의 속성(attribute)을 규정하도록 지원하는 데이터의 입력 인터페이스 모듈(Data Input Interface Module)과 데이터의 입력 인터페이스 모듈에서 정의된 내용을 이용하여 HDML 소스(source)를 생성하는 콘텐츠 생성 모듈(Content Generating Module), 그리고 생성된 콘텐츠의 분석 및 에디팅 뷰를 제공해주는 HDML 뷰어(HDML Viewer)가 저작도구의 구성 모듈이다. 특히 HDML 뷰어는 사용자에게 보다 쉽고 편리하게 HDML 문서를 생성하고 수정하며, 관리하고 표현할 수 있는 비주얼한 환경을 제공하도록 구현되었다. (그림 8)은 HDML 전용 저작도구 시스템의 내부 구조 및 처리 절차를 보여준다. HDML입력 인터페이스 모듈은 데이터를 생성하거나 기존의 데이터를 불러오게 되며, 이는 콘텐츠 생성 모듈에 보내진다. 콘텐츠 생성 모듈은 HDML 뷰어 모듈안의 각각의 뷰어들의 형태에 맞도록 데이터를 가공한다.



(그림 8) 시스템 구성도

3.1 입력 인터페이스 모듈(Data Input Interface Module)
HDML 문서의 입력 인터페이스 모듈은 HDML 문서

의 각각의 카드에 대한 입력 대화 상자를 통해 카드 종류 및 카드에 대한 엘리먼트의 속성 그리고 속성 값을 지정하며, 이미 작성된 HDML 문서를 파서에 전달해주는 역할을 한다 (그림 9)에서 보는 바와 같이 2개의 요소로 구성되어 있다. 하나는 새로운 HDML 문서의 작성시 요구되는 입력 포맷(format)이며, 다른 하나는 이미 작성된 HDML 문서를 수정하고자 할 때, 이미 만들어진 HDML 문서를 가져오는 기능이다. 문서작성시 선택된 카드에 따라 입력 포맷이 다르게 나타나며, 이 입력 포맷에 데이터를 입력함으로써 해당 카드에 대한 태그(Tag)와 관련 속성(Attribute)을 정의하며, 액션(action) 테이블에 값을 입력함으로써 카드의 행동 방법을 정의할 수 있다 액션(action)태그는 카드들 사이의 이동위치와 이동방법을 정의하며, HDML 뷰어 모듈의 링크 뷰어를 구성하는데 매우 중요한 역할을 한다. 데이터 입력 인터페이스 모듈을 통하여 입력된 데이터들은 구조화된 데이터로 필터링(Filtering)된다.



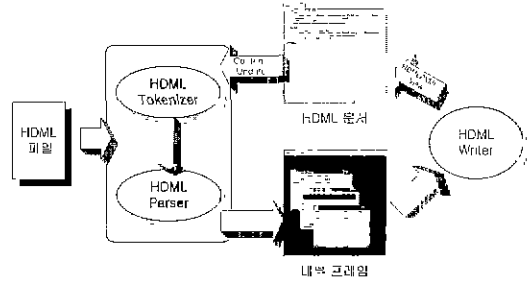
(그림 9) 입력 인터페이스 모듈

3.2 콘텐츠 생성 모듈(Content Generating Module)

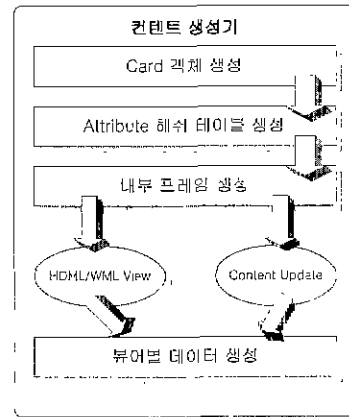
콘텐츠 생성 모듈은 HDML 입력 인터페이스 모듈을 통해 받아들여진 데이터를 분리, 저장 함으로써 HDML 뷰어 모듈의 각각의 뷰어에 알맞는 형태로 데이터를 가공하는 역할을 한다. 이 모듈은 HDML 저작도구의 예

심 기능을 하는 모듈이며, (그림 10)에서 보는 바와 같이 2개의 요소로 구성된다. HDML 파서(Parser)는 HDML 문서를 파싱하며, 콘텐츠생성기는 입력된 데이터를 카드별로 객체화 하며 객체화된 데이터를 이용하여 각각의 카드에 대한 프레임을 생성한다.

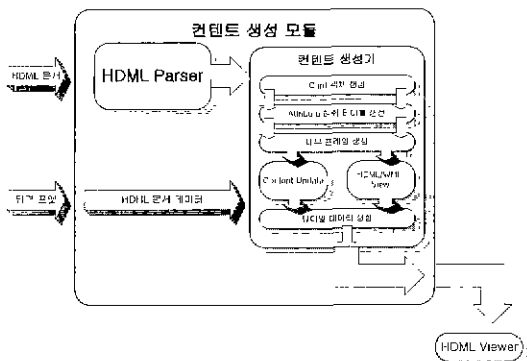
콘텐츠 생성기에서 뷰어 모듈에 적합한 형태로 데이터를 가공하기 위해서는 입력으로 전달되는 데이터를 카드별, 태그별, 속성별로 분리, 저장하는 작업이 필요하다. 이러한 일련의 작업을 HDML 파서가 담당하게 된다. 사용자가 입력 포맷을 통해 새로운 데이터를 생성했다면, 파서를 거치지 않고 바로 콘텐츠 생성기로 보내진다. 입력 포맷을 통해서 데이터를 생성할 경우는 콘텐츠 생성기에 입력될 수 있는 형태로 데이터가 저장되기 때문이다. (그림 11)에서 보는 바와 같이 파서 모듈은 HDMLTokenizer와 HDMLParser의 서브 모듈로 나누어지며, 이 2개의 모듈이 순차적으로 직동함으로써 HDML 문서를 파싱한다. HDML 문서가 입력될 때, HDMLTokenizer는 입력된 데이터를 토큰화하여 HDMLParser에게 넘겨주고, HDMLParser는 해당 카드에 속하는 엘리먼트나 속성 그리고 속성 값을 카드의 타입별로 그룹화 시킨후 카드 타입 식별자를 지정한다. 콘텐츠 생성기는 카드의 타입 식별자를 통해서 어떤 카드 객체를 만들것인지를 결정하게 된다.



(그림 11) HDML 문서의 파싱



(그림 12) 콘텐츠 생성기



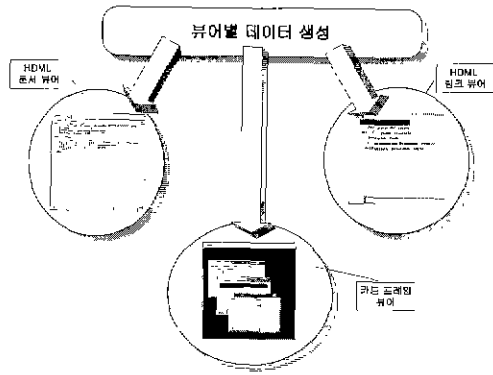
(그림 10) HDML 콘텐츠 생성 모듈

콘텐츠 생성기는 (그림 12)에서 보는 바와 같이, 카드 객체를 생성하고, 속성들을 저장하기 위한 해쉬 테이블을 생성하며, 이런 데이터들을 가지고 내부 프레임을 생성한다. 입력된 데이터를 통해서 HDML의 4가지 카드인 디스플레이 카드, 입력 카드, 선택 카드 그리고 노-

디스플레이 카드에 대한 4개의 객체를 생성하며, 이렇게 객체가 생성될 때, 각각의 카드에 대한 속성을 저장할 수 있는 속성(attribute) 해쉬 테이블이 생성되어 데이터가 저장되게 된다. 이렇게 생성된 객체를 통해서 각각의 카드에 대한 내부 프레임을 생성하여 카드별 텍스트를 입력할 수 있는 포맷을 가진 카드별 프레임을 디자인 템에 생성하게 된다. 생성된 카드 프레임의 입력 인터페이스를 통해 사용자는 각각의 카드에 들어갈 텍스트를 입력할 수 있다. 이렇게 입력된 텍스트와 이미 생성된 카드 객체에 저장되어 있는 데이터를 이용하여 HDML 문서를 생성하고, 사용자에게 HDML 문서가 어떻게 연결되는지를 보여주기 위해 필요한 데이터를 추출하여 링크 데이터를 생성하며, 상위 계층에서 만들어진 카드 프레임들을 저장한다. 이러한 일련의 과정을 뷰어별 데이터 생성 계층에서 처리한다. 이렇게 처리된 데이터는 HDML 뷰어 모듈을 통해 각각의 뷰어에 전달됨으로써 사용자에게 보여지게 된다.

3.3 HDML 뷰어(Viewer)

HDML 뷰어는 마이크로브라우저를 통해 조회 가능한 HDML 문서를 보여주는 HDML 문서 뷰어와 HDML 문서에 대한 효율적인 관리와 직관적인 인지를 제공하기 위해 카드간의 연결을 트리 형태의 링크로 보여주는 HDML 링크 뷰, 그리고 카드별 편집을 통해 마이크로브라우저에 표시될 화면을 편리하게 편집할 수 있는 카드 프레임 뷰를 제공한다. (그림 13)은 HDML 뷰어 모듈의 구성을 나타낸다. HDML 뷰어 모듈을 이렇게 3가지 형태의 뷰어를 통해 사용자로 하여금 보다 쉽고, 편리하게 HDML 문서를 생성하고, 관리 할 수 있는 구조를 제공한다. 각각의 뷰어들은 서로 매우 의존적이다. 뷰어에서의 데이터의 변화는 일련의 처리 절차를 거쳐 다른 뷰어에서 즉시 업데이트된다.



(그림 13) HDML 뷰어의 구성

3.3.1 HDML 문서 뷰어(HDML Document Viewer)

HDML 문서 뷰어는 HDML 문서를 사용자에게 보여주는 역할을 담당하며, 여기서 보여지는 HDML 문서는 마이크로 브라우저의 소스로서 브라우저를 통해 HDML 문서의 내용이 사용자에게 보여지게 된다. HDML 문서 뷰어를 통해 보여지는 HDML 문서는 HDMLWriter를 통해 생성된다. HDML 콘텐츠 생성모듈의 콘텐츠 생성기에서 만들어진 카드 객체의 데이터와 카드 프레임 뷰어를 통해 입력된 텍스트는 HDMLWriter를 통해서 HDML 문서화 되어 HDML 문서 뷰어에 전달되고, HDML 문서 뷰어는 이렇게 생성된 HDML 문서를 사용자에게 보여지게 된다.

3.3.2 HDML 링크 뷰어(HDML Link Viewer)

HDML 링크 뷰어는 HDML 문서를 구조화하여 트

리 구조로 제공함으로써 데크 안의 카드들을 효율적으로 관리하도록 해준다. 링크는 같은 데크 안의 카드로 이어질 수 있고 다른 데크의 카드로 이어질 수 있다. 링크 관리가 비효율적이면 직관적이고 이해하기 쉬운 HDML 문서의 네비게이션(navigation)을 유지하기 어렵다. 본 HDML 저작도구에서는 키드의 링크를 관리하는 기능을 모듈로써 설계, 구현하였다. 이 모듈은 트리 구조로 카드의 링크를 관리하며 다음과 같이 작동한다. 먼저 루트 노드(Root Node)는 저장되어 있던 HDML 파일의 테크이거나 새로 만든 파일의 데크가 된다. 이 루트(Root)의 자식 노드(Node)는 이 데크에 포함되어 있는 각 카드로 구성된다. 각 키드를 나타내는 노드는 만약 카드의 <action> 엘리먼트의 DEST 속성의 값이 있다면 "<Destination = value>"으로 표시된다. 이는 이 카드의 다음 링크는 value 값을 갖는 같은 데크 안의 카드나 다른 데크의 카드거나 URL이 나타내는 리소스(resource)로 연결된다는 것을 나타낸다. <action> 엘리먼트 안에 DEST 속성 값이 없다면 "<Destination = NONE>"으로 표시된다. HDML 카드 중 선택 카드는 사용자가 여러 항목 중 하나를 선택하도록 하는 기능을 가진 카드이다. 사용자가 항목 중 하나를 선택했다는 것은 그 항목을 나타내 주는 카드로 이동했다는 것인데, 이때에도 해당 카드로의 링크가 이어진다. 이 선택 카드의 링크 관리 또한 저작도구 링크 관리에서 중요하다. 선택 카드의 링크 관리는 다음과 같다. 선택 카드에 해당하는 노드에는 선택 카드가 가지고 있는 항목이 "<<ITEM (항목) Destination = dest attribute의 value 값>>"로 나타나게 된다. 만약 <ce> 엘리먼트에 dest 속성 값이 없다면 "<<ITEM (항목) Destination = dest attribute의 value 값>>"로 나타난다. 각 카드를 나타내는 노드를 선택하면 디자인 탭에 나타나 있는 카드 프레임 중 해당 카드에 해당하는 카드가 활성화(activated)된다. 또 각 카드 노드에 있는 "<Destination = value 값>"을 선택하면 해당 카드의 링크에 해당하는 카드 프레임이 활성화된다. 또 선택 카드의 항목 중 하나를 선택하게 되면 그 항목을 가지고 있는 선택 카드 프레임이 활성화되고 그 항목도 선택된다. 만약 카드의 링크를 변경하고 싶다면 속성변경 모듈에서 링크를 수정해 주면 된다.

3.3.3 카드 프레임 뷰어(Card Frame Viewer)

카드 프레임 뷰어는 사용자에게 각각의 카드를 보여

주며, 카드별 입력 포맷을 제공함으로써 보다 쉽게 콘텐츠를 생성할 수 있게 해준다. HDML 문서는 어떤 카드를 생성했느냐에 따라 마이크로 브라우저에 표현되는 방식이 달라지게 된다. 이런 점들 때문에 테크로 구분되는 웹페이지 안에 내부 프레임은 각각의 카드로 생성하게 하고, 이 프레임에 카드별 입력 포맷을 줌으로써 사용자로 하여금 보다 편리하게 HDML 문서의 카드들을 생성하고, 원하는 콘텐츠를 입력 포맷을 통해서 자유롭게 생성하도록 구현하였다. 디스플레이 카드, 입력 카드, 노-디스플레이 카드는 입력 포맷으로 텍스트를 가지며, 선택 카드는 리스트 포맷과 대화 상자로 이루어진다. 선택 카드는 나머지 3가지의 카드들과 다른 구조를 가진다. 선택 카드는 브라우저에서 콘텐츠를 선택할 수 있는 항목을 가지며, 원하는 항목을 선택하면 해당 항목에 대한 콘텐츠가 있는 곳으로 제어가 옮겨가게 된다. 이렇게 다른 카드와 약간 다른 구조를 가졌기 때문에, 카드 프레임 뷰어에서는 카드 프레임을 리스트 박스로 구성하여 항목을 Add와 Delete를 이용하여 추가, 삭제하도록 구성하였으며 항목 추가 시에 해당 항목의 행동 방법을 대화 상자를 통해 입력함으로써 선택 카드 전체 뿐만 아니라 각각의 항목별 카드의 이동을 제어할 수 있도록 구성되었다.

34 부가 기능 모듈

DB연동, 팩스 송신등의 부가 기능 제품을 원활히 하기 위해 다음의 모듈들을 구현하였다.

(1) 웹 서버를 이용한 CGI 스크립팅

웹 서버를 이용한 CGI 스크립팅은 현재 다음 5가지의 스크립트를 가지고 있으며, 스크립트 언어인 perl로서 작성되었다. 본 저작도구에서는 팩스송신, 경보송신, 페치, 현재시간출력, 리스트 관리자 등의 스크립트를 가지고 있으며, 각 사이트의 특성에 맞추어 변경하여 저장할 수 있다.

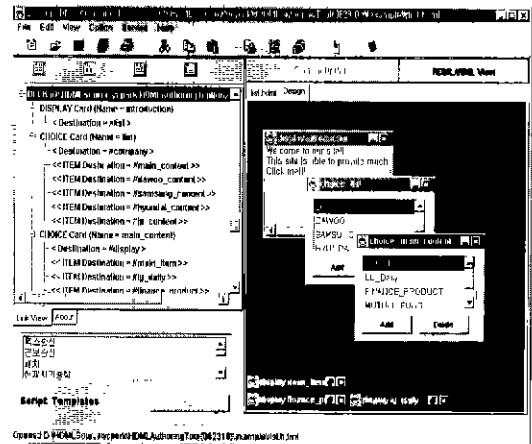
(2) 폰 시뮬레이터(Phone Simulator)를 이용한 브라우저

폰 시뮬레이터는 Phone.Com사에서 제공하는 UPSDK2.0.2를 이용하여 정의된 문서에 맞게 구현하였다.

35 구현환경

본 저작도구는 Windows 95/98/NT 환경 하에서 개발

되었다. 프로그램의 이식성 및 이동성의 효과적인 제공을 위하여 자바 언어로 작성되었으며, JDK1.1.6과 swing1.0.3을 이용하였으며, Phone.Com사에서 제공하는 시뮬레이터(simulator)인 UPSDK2.0.2를 통해 생성된 HDML 문서의 동작을 미리보기 형태로 볼 수 있다. 개발된 시스템은 이동통신망과 휴대폰을 이용하여 인터넷 상에서의 웹콘텐츠 서비스를 사용자에게 제공할 수 있는 언어인 HDML을 위한 전용 저작 애플리케이션 프로그램이다. (그림 14)는 HDML 전용 저작도구를 이용하여 문서를 편집하고, 브라우징 할 수 있는 애플리케이션 프로그램의 사용자 인터페이스이다.



(그림 14) HDML 전용 저작도구 사용자 인터페이스

4. 결론 및 향후과제

현재 전세계적인 애플리케이션의 제작은 HDML을 통하여 이루어지고 있으며, 동시에 WML로 작성된 애플리케이션도 작성되어 사이트가 구축되는 상황이다. HDML이 1997년 발표되어 진세계적으로 애플리케이션 개발에 이용되는 개방형 마크업 언어임을 감안하면, 현재 표준으로 정의되고 있는 WML 애플리케이션의 확산에는 시간이 걸린 것이다. 무선 인터넷 서비스를 위한 HDML 및 WAP/WML 응용도구의 제작이 전 세계적으로 급속도로 이루어지고 있는 상황에서, HDML에 관련된 기술의 습득과 지원도구의 개발, 그리고 WAP 및 WML에 관한 연구는 미래의 정보통신 기술의 중심에 위치하고 있다고 할 수 있다.

본 논문의 설계구현으로써 무선인터넷 환경을 지원하

는 마크업 언어인 HDML 및 관련 환경에 대한 기초기술을 확보할 수 있었다. 본 논문에서 설계, 구현한 HDML 전용 저작도구의 기능을 요약하면 다음과 같다

- WYSIWYG(What You See Is What You Get)방식의 HDML 페이지 편집 기능
- 링크 관리 기능 : 데크와 카드의 위치 및 연결 관계를 다단계 트리 계층 구조로 나타냄
- 브라우징 기능 . WYSIWYG 편집기상에서 구성을 보거나, 브라우저를 가동하여 직접 페이지가 게이트웨이를 통하여 접속된 상태를 시연함
- CGI 연동 및 자동 생성 기능

본 논문의 후속 연구로써 WML 저작도구를 설계, 구현 중에 있으며, 특히 WMLScript의 기능과 효율적인 브라우저 기능에 대하여 연구하고 있다. 계속적인 연구 개발을 다음분야에 대하여 진행할 예정이다.

- 이동통신망과 인터넷의 연동 시에 필수적인 WAP 게이트웨이에 관한 연구 및 개발
- 이동통신망을 위한 응용 프로토콜에 관한 연구
- 효과적인 웹 정보의 접근을 위한 휴대용 장비의 연구 및 개발

참 고 문 헌

[1] Handheld Device Markup Language Specification, <http://www.w3.org/TR/NOTE-Submission-HDML-spec.html>.

[2] [WML] "Wireless Markup Language," WAP Forum, April 30, 1998. URL <http://www.wapforum.org/>.

[3] [WAP] "Wireless Application Protocol Architecture Specification," WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>.

[4] [WAE] "Wireless Application Environment Specification," WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>.

[5] [WTA] "Wireless Telephony Application Specification," WAP Forum, April 30, 1998 URL <http://www.wapforum.org/>

[6] [WTAI] "Wireless Telephony Application Interface," WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>.

[7] [WSP] "Wireless Session Protocol," WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>.

[8] [WTP] "Wireless Transaction Protocol Specification," WAP Forum, April 30, 1998 URL : <http://www.wapforum.org/>.

[9] [WTLS] "Wireless Transport Layer Security Protocol," WAP Forum, April 30, 1998 URL : <http://www.wapforum.org/>

[10] [WDP] "Wireless Datagram Protocol," WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>

[11] [WMLScript] "Wireless Markup Language Script," WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>

[12] Laurent, S., XML : A Primer, MIS Press, 1998

[13] Getting Started with the UP.SDK, <http://updev.phone.com/dev/ts/>

[14] HDTP Specification, <http://www.phone.com/>, <http://updev.phone.com/>.

[15] Tatsuo Nakajima, Akihiro Hokimoto. "Adaptive Continuous Media Applications in Mobile Computing Environments," Japan Advanced Institute of Science and Technology 1-1 Asahidai, Tatsunokuchi, Ishikawa, pp.923-12, JAPAN, <http://mmmc.jaist.ac.jp:8000/>.

[16] Peter M. Corcoran, Ferenc Papai, Arpad Zoldi, "Portable User Interfaces for Remote Access to Embedded Home Systems and Home Networks," University College, Galway, Proceedings of the IEEE 1998 International Conference on Consumer electronics, pp.174-175, 1998.

[17] Letueri P, Srivastava MB. "Advances in wireless terminals." University of California at Los Angeles, IEEE Personal Communications, Vol.6, No.1, pp.6-19, 1999

[18] Salkintzis AK, "Packet data over cellular networks : The CDPD approach," IEEE Communications Magazine, Vol.37 No.6, pp.152-159, 1999.

[19] Ken Buchanan, Rodger Fudge, David McFarlane, Tim Philips, Akio Sasaki, and Sabah Towaj, "IMT-2000 : Service Provider's Perspective." IEEE Personal Communications, Vol 4 No.4, pp.8-13, 1997

[20] [ColdFusion3.1] <http://www.allaire.com>.



박 영 충

email : ycpark@ce.scjong.ac.kr

1999년 세종대학교 전산과학과
(이학사)

1999년~현재 세종대학교 전산과
학과 석사과정 재학중

관심분야 무선 응용 프로토콜(Wireless Application Protocol),
무선 데이터 서비스, 인터넷 응용



신 동 일

email : dshin@sejong.ac.kr

1988년 연세대학교 전산과학과
(이학사)

1993년 M.S. in Computer Science,
Washington State University

1997년 Ph.D in Computer Science, University of North
Texas

1997년~1998년 시스템공학연구소 선임연구원

1998년~현재 세종대학교 컴퓨터공학과 진임강사

관심분야 : 이동통신, WAP, 멀티미디어 데이터베이스,
CSCW, 지능형 에이전트



박 병 하

email : bhpark@cc.sejong.ac.kr

1999년 세종대학교 전산과학과
(이학사)

1999년~현재 세종대학교 전산과
학과 석사과정 재학중

관심분야 무선 응용 프로토콜(Wireless Application Pro-
tocol), 무선 데이터 서비스, 분산객체



신 동 규

email : shindk@sejong.ac.kr

1986년 서울대학교 계산통계학과
(이학사)

1992년 Illinois Institute of Tech-
nology 전산학과(공학석사)

1997년 Texas A&M University 전산학과
(공학박사)

1986년~1991년 한국국방연구원 연구원

1997년~1998년 현대전자 멀티미디어연구소 책임연구원

1998년~현재 세종대학교 컴퓨터공학과 조교수

관심분야 : 웹기반 멀티미디어, WAP, 멀티미디어 DB,
영상압축, 웨이브릿 부호화



김 건 희

email : ghlam@dmmlab.sejong.edu

1997년 한림대학교 컴퓨터공학과
(공학사)

1998년~현재 세종대학교 전산과
학과 석사과정 재학중

관심분야 : 내용기반 멀티미디어 검색(Content-Based
Retrieval), 무선 응용 프로토콜(Wireless Ap-
plication Protocol), SMIL(Synchronized Multi-
media Integration Language), XML