

# Clipping 기반의 무선 인터넷 사이트 구축용 변환 서버 구현

조 승 호<sup>†</sup> · 차 정 훈<sup>††</sup>

## 요 약

현재 유선 인터넷 콘텐츠와 비교할 때 무선 인터넷 콘텐츠가 크게 부족하여 유선 콘텐츠를 무선 인터넷 콘텐츠로 변환시키고자 하는 요구가 높다. 본 연구의 변환 서버는 사용자 단말기로부터 요청을 자동적으로 인식하여 대상 URL 서버상의 원본 문서를 검색하여 이들을 중간 언어인 metaXML 문서로 생성하고 이를 사용자 단말기에 적합한 무선 마크업 언어 문서로 변환 출판한다. 또한, 본 변환 서버는 이미지 변환을 위하여 이미지 변환기와 기존 웹 페이지들의 재구성 저작 도구인 Clipper와 연동하여 동작한다. 대상 URL의 정적/동적 웹 페이지를 처리하는 변환 서버의 처리 능력에 대하여 성능 실험을 수행하였으며, 동적 문서에 대한 변환 실험 결과에 따르면, 변환 서버내 스레드 풀의 스레드 개수를 5개로 유지할 때 변환 서버가 가장 나은 처리량을 나타내었다.

## Implementation of A Clipping-based Conversion Server for Building Wireless Internet Sites

Seung-Ho Cho<sup>†</sup> · Jeong-Hoon Cha<sup>††</sup>

## ABSTRACT

Because a quantity of wireless internet contents is much less than wired internet contents, it exist high necessities that wired internet contents should be converted into wireless internet contents. The conversion server implemented in this paper, automatically recognizes the type of user agents when they request, retrieves source documents on the web site specified by an URL, generates metaXML documents as an intermediate form, and converts them wireless markup documents appropriate for user agents. The conversion server interoperates with the image converter for image conversion and the clipper which is an authoring tool for clipping existing wired internet documents. We performed experiments about capability of the conversion server transcoding static/dynamic web pages specified by an URL. According to performance results on dynamic web pages, the conversion server showed better throughput when a thread pool in the server maintains 5 threads compared with 1 and 10 threads.

키워드 : 마크업 언어 변환(Markup Language Conversion), XML, XSLT, 이미지 변환(Image Conversion), Clipping, 처리량 (Throughput)

### 1. 서 론

지난 2~3여년간 무선통신 대역폭의 개선, 카메라 장착 휴대폰의 보급, 멀티미디어 데이터 활용 증가 등으로 무선 인터넷 인프라를 효과적으로 활용할 수 있는 환경이 조성되었다. 이러한 무선 통신 환경은 이동 중인 사용자들이 장소, 시간, 매체 등에 구애됨이 없이 생활에 편리한 다양한 정보 서비스의 공급을 요구하고 있다. 유선 인터넷 콘텐츠와 비교해 볼 때 무선 인터넷 콘텐츠들은 공급자 수뿐만 아니라 콘텐츠의 양에 있어서도 매우 부족한 수준이다. 현

재 이동통신사들의 무선망에 대한 개방이 임박하여 유선 포털들이나 기업들은 무선망 개방시 기존의 유선 인터넷 사이트들에 대한 무선 인터넷 서비스를 준비하고 있어 유선 콘텐츠를 무선 콘텐츠로 변환하고자 하는 요구가 높다.

무선 인터넷 사이트를 구축하기 위한 방법으로는 직접 무선 마크업 언어를 활용하여 콘텐츠를 구축하는 방법과 기존에 있는 유선 인터넷 콘텐츠를 무선용 인터넷 사이트로 변환하는 방법이 있다. 직접 콘텐츠를 구축하는 경우, 각 단말기에 최적화된 무선 콘텐츠를 서비스할 수 있는 장점이 있으나 제작 및 유지 보수 시간 및 비용 등이 많이 소요되는 단점이 있다[17, 19, 22]. 반면에, 기존 콘텐츠를 변환하여 재사용하는 경우에는 각기 다른 단말기들에 일관

<sup>†</sup> 정 회 원 : 강남대학교 컴퓨터미디어공학부 교수

<sup>††</sup> 정 회 원 : (주)엠플넷소프트 모바일인터넷 연구소 선임연구원  
논문접수 : 2003년 7월 23일, 심사완료 : 2004년 2월 23일

된 출판이 가능할 뿐만 아니라 무선 인터넷 서비스 구축에 소요되는 비용을 대폭 절감할 수 있고, 무선 인터넷 사이트 개발 기간을 단축시킬 수 있는 장점이 있다[1, 3, 4, 5, 7, 8].

유무선 마크업 언어(Markup Language) 변환시 사용되는 모델로는 마크업 언어간 직접 변환모델과 중간 언어로서 XML을 활용하여 변환하는 모델로 나눌 수 있다[15]. 직접 변환 모델은 원본 콘텐츠의 문서들을 목적 마크업 언어로 표현된 문서로 직접 변환하는 것으로 새로운 마크업 문서 변환시, 추가될 변환 모듈의 개수가 기존의 변환 모듈 수만큼 개발되어야 하는 부담이 있다. 즉, 원본 마크업 언어 문서이든 목적 마크업 언어 문서이든 새로운 마크업 언어가 변환 서버에 추가될 때마다 기존의 변환 모듈 수만큼 새로운 모듈이 추가로 개발되어야 하는 것이다.

반면 중간 언어(intermediate language)를 사용하는 변환 모델은 미리 정의된 중간 언어를 사용하여 원본 마크업 언어와 목적 마크업 언어간에 독립성을 제공한다. 미리 정의된 중간 언어를 매개로하여 원본 마크업 언어를 중간 언어로 변환하는 모듈, 중간 언어를 목적 마크업 언어로 변환하는 모듈들로 구성된다. 이 모델에서 새로이 마크업 언어를 변환하고자 하는 경우에는, 원본 마크업 언어에 대해 중간 마크업 언어로 변환하는 모듈, 그리고 중간 언어를 목적 마크업 언어로 변환하는 모듈, 단지 두 개의 변환 모듈만 요구되므로 확장이 용이하다. 현재 무선 마크업 언어에 대한 다양한 표준들이 사용되고 있고 새로운 무선 마크업 언어들이 제안되고 있는 상황에서 시스템의 확장성을 중요시하여 본 연구에서는 후자의 중간 언어를 사용하는 모델을 채택하였다.

본 연구는 기존의 관련 연구인 PersonalClipper[2, 7]가 낮은 무선 대역폭을 효율적으로 활용하기 위한 목적으로 clipping을 추구하고 있는데 비하여, 본 연구에서는 사용자가 유선 콘텐츠로부터 무선 콘텐츠를 추출/정의하기 위하여 clipping을 적용하고 있다는 점이 주요한 차이점이다. 또한, 사용자는 본 연구의 저작도구인 clipper를 활용하여 정확하고 간편하게 변환 규칙을 정의할 수 있으며, 본 연구의 결과인 변환 서버가 이와 같이 정의된 변환 규칙에 의거 기존의 웹 페이지로부터 사용자 단말기에 적합한 무선용 페이지를 자동적으로 변환 생성한다.

2장에서는 본 연구와 연관된 연구들에 대하여 기술하고, 3장에서는 본 연구의 변환 서버 개요, 특징, 시스템 구성, 처리 절차, 이미지 변환기의 특징, 이미지 변환기와 연동 절차 등에 대하여 설명하고, 4장에서는 변환 서버와 상호 연동하여 기존의 웹 페이지들을 재구성하는 저작 도구인 Clipper의 특징, 구현 사례 등에 대하여 기술하고, 5장에서는 처리량과 평균 응답시간을 성능 척도로 변환 서버의 성능 실험 및 결과를 분석하고, 6장에서 결론을 맺는다.

## 2. 관련 연구

무선통신망의 대역폭이나 단말기의 화면 크기 등을 고려하여 기존의 유선 사이트를 무선 인터넷으로 서비스하기 위하여 주로 세가지 접근 방식이 존재한다.

### ① 다수의 웹 사이트들을 재구축하는 방식

무선 단말기들을 위한 콘텐츠들을 별도로 구축하는 것으로 예를 들면, New York Times의 PDA 사이트[22]나 Amazon의 폰용 사이트[17] 등이 있다. 또한 이를 구축하기 위한 저작도구로는 AnyBuilder[18] 등이 있다.

### ② 웹 페이지들에 대해 서로 다른 뷰(view)를 제공하는 방식

주가 조회, 교통, 기상 정보 등의 특정 정보들을 무선 통신망을 통해 서비스하기 위한 wrapper[14]를 제공하는 접근 방식으로 byair.com 또는 oraclemobile.com 등의 사이트에서 제공한다.

### ③ 웹 페이지들을 필터링 또는 재구성하는 변환 방식

단말기의 화면 크기나 대역폭 등을 고려하여 변환 서버나 프록시 등이 콘텐츠를 변환한다. 본 연구를 포함하여 Hori의 연구, HAN, WebAlchemist, PersonalClipper 등이 이 방식에 속한다[2, 3, 5, 7-12, 24].

콘텐츠 제공자의 관점에서 보면, 다수의 웹 사이트 버전을 관리해야 하는 첫 번째 방식은 인력 및 비용이 많이 소요되는 단점이 있고, 두 번째 방식은 각 웹 사이트에 대응되는 wrapper들이 각 사이트가 변경될 경우 이에 따라 변경되어야 하는 부담이 크다. 세 번째 방식은 프록시나 변환 서버 등이 콘텐츠 변환을 수행하고 사용자는 다만 자신의 단말기에 적합한 콘텐츠 내용들을 전송받는 것으로서 현재 주로 이 방식이 채택되고 있는 추세이다.

Hori 등의 연구[9, 13]는 기계가 해석할 수 있는 형태의 외부 주석(external annotation)을 기반으로 방대한 웹 페이지로부터 무선 단말기용 콘텐츠를 추출하고자 한다. 사람의 개입을 최소화하기 위하여 웹 페이지 중에서 어느 요소들을 추출하여 사용자 단말기로 내보낼 것인지에 대한 판단은 주석을 활용한다. 주석 파일은 페이지 요소들간 상대적 우선 순위를 부여하기 위해 RDF(Resource Description Framework) 데이터 모델[25]을 준수한다. 이 시스템은 페이지 분리 모듈과 사용자 단말기용 페이지 생성 모듈로 구성되었고, 사용자가 보다 용이하게 주석을 부가할 수 있는 저작 도구를 제공한다.

HAN(Hierarchical Atomic Navigation)[8] 연구는 주로 PDA용 콘텐츠 변환을 위하여 원래의 웹 페이지들을 구역(zone)으로 분할하는 개념에 착안하고 있다. 이들 구역은

이미지, 텍스트 단락, 헤더, 하이퍼링크, 폼, 자바 애플릿 등의 요소들로 구성된다. 사용자들은 임의의 구역을 선택하여 볼 수 있으나 원래 크기대로 보기 위해서는 하나의 구역만이 표시된다. 이러한 구역을 통해 축약된 페이지 미리보기 형태인 항해 페이지(navigator page)을 통해 웹 페이지 요소들을 식별한다. 원래 페이지상의 요소들을 읽을 수 있으면 축소판으로 표시하고, 읽을 수 없을 수준이면 아이콘으로 표시된다. 이러한 표시 분류는 트리 탐색 및 가중치 부여를 통해 수행된다.

WebAlchemist[10] 연구는 가능한 한 웹 페이지의 전체 구성을 유지하면서 웹 페이지의 구문 분석으로부터 부분적으로 의미를 추출하여 변환시 이를 활용하고자 한다. 이 연구에서는 기존에 연구된 긴 웹 페이지를 일련의 작은 크기의 부페이지(subpage)들로 구획 변환, 이미지 축소 또는 생략 변환, 첫 문장의 생략 변환 등을 확장하였다. 이러한 확장된 변환을 구조 인지 변환(structure-aware transcoding)이라 하였는데, 이 변환에서는 패턴 매칭에 의한 반복된 화면 구조를 발견하여 이를 스트링들의 배열로 매핑하거나, 표나 셀 등을 선택적으로 생략하는 변환을 수행한다.

PersonalClipper[2, 7]의 연구에서는 웹 사이트에 대한 사용자의 개인화된 클리핑을 쉽게 생성하고 유지하도록 허용한다. 유선 인터넷 검색시 사용자는 자신에게 부적합한 콘텐츠일지라도 통상 많은 양의 데이터를 전송받게 된다. 그러나, 무선 환경에서는 낮은 대역폭과 작은 화면 크기로 인하여 유선 인터넷처럼 대량의 데이터 전송은 큰 부담이 된다. 이러한 문제에 대하여 PersonalClipper는 사용자의 검색 행동들을 기록하여 필요한 정보만이 검색 가능하도록 클리핑 명세를 정의 및 저장한다. 정보 필요시 사용자는 클리핑 명세만을 실행시킴으로써 사용자의 클릭 횟수와 데이터 전송량을 줄인다.

본 연구는 접근 방식 측면에서 PersonalClipper와 가까우나, PersonalClipper가 낮은 무선 대역폭을 효율적으로 활용하기 위한 목적으로 clipping을 추구하고 있는데 비해, 본 연구에서는 사용자가 유선 콘텐츠로부터 무선 콘텐츠를 추출/정의하기 위하여 clipping을 적용하고 있다는 점이 주요한 차이이다. 본 연구의 clipper에 의해 정확하고 간편하게 정의된 변환 규칙에 의거 기존의 웹 페이지로부터 사용자 단말기에 적합한 무선용 페이지가 자동적으로 변환 생성되게 하는 것이 본 연구의 특징이다.

### 3. 시스템 구성

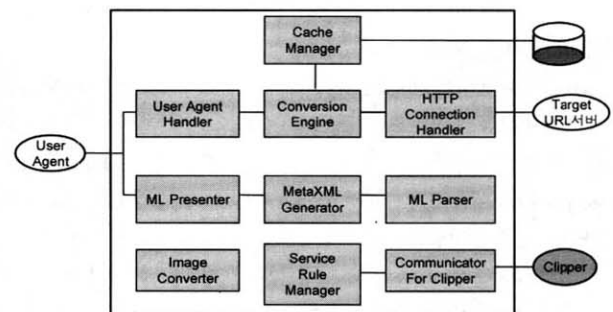
본 절에서는 변환 서버의 전체 시스템 구성을 기술한 후, 변환 서버를 구성하는 주요 모듈, 처리 절차, 이미지 변환기 및 구현 사례에 대하여 기술한다.

#### 3.1 시스템 개요

일반적으로 유선 인터넷 문서들을 무선 인터넷 문서로 변환하는 방식은 크게 완전 자동 변환(Fully Automatic Conversion) 방식과 사용자 설정 변환(User-Configurable Conversion) 방식으로 분류된다. 전자의 경우는 변환 프로그램이 유선 콘텐츠 전체 페이지에 대하여 미리 정해진 규칙대로 변환 작업을 수행하는데 비하여, 후자의 경우에는 유선 인터넷 사이트로부터 무선 단말기 사용자에게 서비스하기 원하는 콘텐츠만을 추출하여 선별적으로 출판하는 방식이다. 본 서버는 두 가지 방식을 모두 지원한다.

변환 서버 서버의 주요 특징들을 살펴보면,

- ① 단말기가 요청한 URL(Unified Resource Locator)을 통해 원본 문서를 검색한다.
  - ② URL의 원본 문서들을 파싱하여 중간 언어인 metaXML로 생성한다.
  - ③ metaXML 문서를 사용자 단말기에 적합한 무선 마크업 언어 문서로 변환한다.
  - ④ HTML 문서들을 WAP 1.x, ME, XHTML, WML 2.0로 변환한다[27, 31].
  - ⑤ 무선 마크업 언어들인 HDML, WML, mHTML, cHTML, XHTML간 상호 변환이 가능하다.
  - ⑥ 무선통신망의 낮은 대역폭과 무선 단말기들의 작은 화면 크기를 고려하여 콘텐츠들을 자동 분할한다.
  - ⑦ 사용자 단말기로부터 접속을 자동 인식한다.
- 이러한 변환 서버의 구조는 (그림 1)과 같다.



(그림 1) 변환 서버의 구조

다음 절에서 기술될 이미지 변환 모듈을 제외한 각 모듈이 수행하는 기능들을 살펴보면,

##### 3.1.1 User Agent Handler

사용자 단말기로부터 URL 요청을 받아 들어 요청 큐(request queue)에 저장함으로써 서버의 변환 작업이 개시된다. 이 모듈은 사용자 단말기로부터 요청을 받아들이기 위해 웹 서버로서의 일부 기능을 갖추어 HTTP 프로토콜을 처리한다.

### 3.1.2 Conversion Engine

전체 변환 과정을 제어하고 조정하는 모듈로서 요청 큐로부터 사용자 단말기의 URL 요청을 가져와서 Request 객체를 생성하여 처리를 시작한다. 이 모듈은 스레드(thread) 단위로 실행되며 변환 서버 서버 구동시 스레드 풀에 생성된 스레드들이 대기한다. 요청 객체를 통해 획득한 사용자 단말기 브라우저, 대상 URL, 사용자 입력 파라미터 등에 대한 정보들을 검사하고 보관한다. 그리고, 대상 URL 서버로부터의 응답을 처리하기 위해 Response 객체를 생성하여 응답 큐(response queue)에 저장한다.

### 3.1.3 HTTP Connection Handler

이 모듈은 대상 URL, 사용자 파라미터 등 사용자 요청 정보들을 넘겨 받아서 대상 URL에 접근하여 원본 문서를 읽어 입력 스트림으로 반환한다. 실제 대상 URL에 대한 HTTP 연결을 추상화하였다. 대상 문서가 HTML인 경우, 웹 브라우저처럼 user agent 정보로 검색 요청을 하고 이에 대한 HTML 응답을 받는다. HTTP GET/POST 방식을 지원하며, HTTP Cookie를 지원 및 관리한다.

### 3.1.4 ML Parser

대상 URL로부터 전송받은 문서를 파싱하여 내부에서 처리할 수 있는 XML DOM[6] 문서를 반환한다. HTML 문서인 경우에는 W3C의 Jtidy[26]를 사용하여 XHTML 문서로 변환하고, WML 문서인 경우에는 Apache Xerces [30]의 DOM Parser를 사용하여 DOM 문서로 변환한다. 또한 원본 페이지의 상대 주소를 절대 주소로 치환하는 역할도 수행한다.

### 3.1.5 MetaXML Generator

ML 파서로부터 DOM 트리를 넘겨 받아서 metaXML로 작성된 문서를 파일 또는 스트림 형식으로 생성한다. 이 모듈은 DOM 트리에 대해 XSLT(eXtensible Stylesheet Language Transform)을 적용하여 중간 언어인 metaXML 형식으로 출력을 생성한다. 각 변환을 위한 XSL 파일은 미리 템플레이트(template)로 작성되어 있다. XSLT의 적용을 위하여 Apache Xalan를 사용하며, 다중스레드 환경에 최적화될 수 있도록 템플레이트 객체를 사용하였다[30, 32].

### 3.1.6 ML Presenter

이 모듈은 metaXML 문서를 파일 또는 스트림 형태로 입력받아서 스트림 형식으로 사용자 단말기에 적합한 마크업 언어 문서로 변환한다. 이 모듈에는 각 마크업 언어마다 독립된 서브모듈(submodule)들인 WMLPresenter, MHTML Presenter, PDAPresenter, CHTML Presenter 등이 구현되어 있다. 이 서브모듈들은 개별 마크업 언어의 특성에 맞게 미리 작성된 XSL 파일을 적용하여 사용자 단말기용 최종 출력을 생성한다. 템플레이트 객체가 사용되며, 사용자

단말기들의 화면 크기에 따라 일정한 크기로 출력 화면을 분할해 주는 분할(fragmentation) 기능 등이 구현되어 있다. 예를 들어, WML의 경우에는 입력 HTML를 다수의 카드(card)로 분할하고 카드들을 링크로 연결한다.

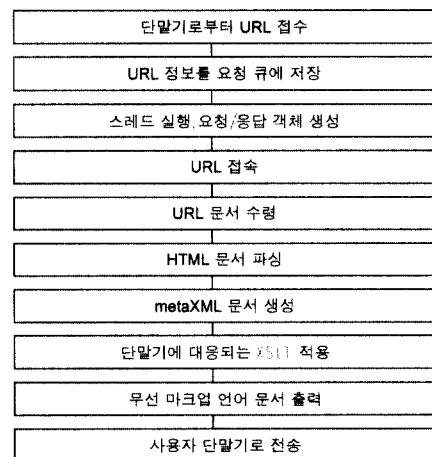
### 3.1.7 Cache Manager/Service Rule Manager

이전 페이지 주소 및 태그 위치와 관련된 정보인 SessionInfo, WML의 <prev> 태그 동작을 위해 이전 페이지의 URL과 카드 id를 가진 PrevInfo, 사용자 쿠키 정보를 가진 Mcookie, Clipper에서 생성된 변환 규칙 WIDL(Web Interface Definition Language)[29] 등은 무선 페이지 항해(navigation)을 위해 유지 관리되어야 할 정보들이다. 본 변환 서버가 운영 중에 생성된 각종 정보들은 메모리나 파일에 저장된다. WIDL을 제외한 나머지 정보들은 캐쉬 클래스를 활용하여 관리하고 WIDL은 명시적으로 삭제하기 전까지 정보가 유지되어야 하므로 Service Rule Manager 모듈에 의해 관리된다.

### 3.1.8 Communicator for Clipper

4장에서 설명될 Clipper는 무선 서비스를 위한 정보들을 유선 사이트로부터 추출/정의하는 저작 도구이다. 사이트 관리자는 이러한 Clipper를 활용하여 사용자 단말기에 전송할 내용들을 규정한 변환 규칙(conversion rule)을 WIDL 형식으로 작성하여 저장소에 저장한다. 이 모듈은 이러한 Clipper와 통신하는 변환 서버측 대응 모듈로서 Clipper와 접속을 통해 사용자 인증후 무선용 사이트 구축을 위한 변환 규칙 저장, 전송, 또는 열기 등을 수행한다. Clipper로부터 접속 요구를 받은 때, 변환 서버 서버는 Clipper가 전송한 ID/Password에 대해 인증을 수행하고 적합한 사용자라면 인증번호를 전송한다. 적합한 사용자 ID에 대해 발급된 인증번호는 별도로 관리된다.

이러한 변환 서버가 수행하는 변환 절차는 (그림 2)와 같다.



(그림 2) 변환 서버의 변환 절차

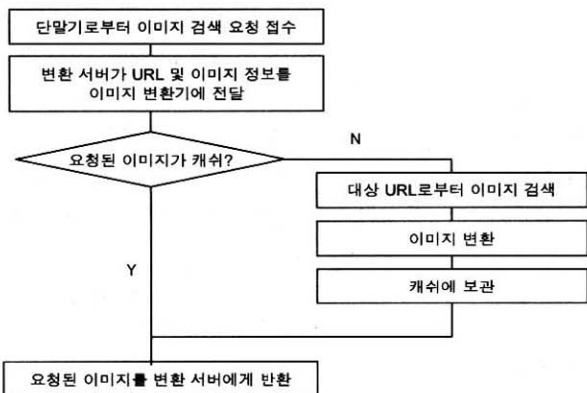
### 3.2 이미지 변환

무선 인터넷 분야에서 마크업 언어에 대한 표준화는 비교적 많은 진전이 있었던 반면에, 이미지 형식에 대한 표준화 작업은 미미하다. WAP Forum에서 지원하는 WBMP [27] 포맷이 있으나 이는 흑백의 단색 이미지만 표현할 수 있어 표현에 한계가 있으며 ME 브라우저를 사용하는 무선 인터넷 단말기에서는 지원되지 않는 단점을 갖고 있다. 현재 국내의 이동통신사에서는 WBMP, NBMP, GIF, PNG, SIS(Simple Image Service)[21] 등의 다양한 형식들이 사용되고 있다.

이러한 상황에서 이미지를 서비스하는 포털같은 사이트들은 같은 내용의 그림을 BMP, WBMP, NBMP, SIS 등의 다수의 포맷으로 미리 작성해 놓고 단말기에 따라 적합한 이미지 형식을 제공하고 있다. 이러한 방식으로는 정적인 이미지 지원은 가능하나 주가 그래프나 날씨 정보와 같이 실시간으로 생성되는 이미지 정보들을 처리하기에는 역부족일 뿐만 아니라, 새로운 이미지 포맷이 추가될 경우에 신속한 서비스 제공이 어려운 난관이 있다. 본 이미지 변환기는 실시간 이미지 변환을 지원하여 정적 이미지뿐만 아니라 동적 이미지에 대해서도 처리 가능하여 새로운 이미지 형식이나 단말기에 대한 지원이 용이하다.

본 이미지 변환기는 다음과 같은 특징을 가지고 있다.

- ① JPG, GIF, BMP, PNG, TIF 등의 유선 인터넷 이미지 형식들을 무선 인터넷용 이미지 형식인 BMP, WBMP, NBMP로 실시간 변환한다.
- ② 단색의 BMP, WBMP 및 NBMP 등의 무선 인터넷용 이미지를 GIF, JPG, PNG 등의 유선 인터넷 이미지로 실시간 변환한다.
- ③ 이미지의 색상수 및 품질 선택 기능을 통해 다양한 대역폭의 단말기 지원한다.
- ④ 이미지 영역 선택을 통한 이미지 분할 변환 기능 지원한다.(이미지 맵 또는 만화 등)
- ⑤ Animated GIF의 장면별 변환 지원한다.



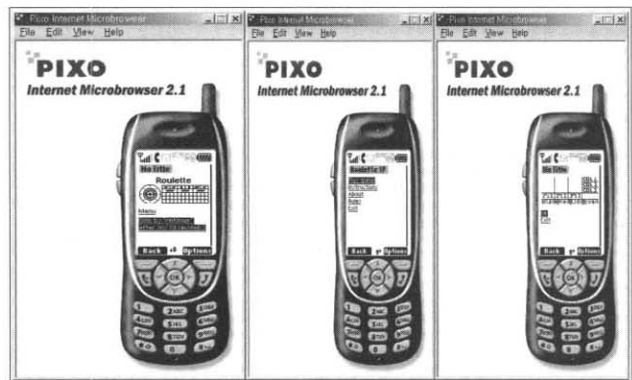
(그림 3) 이미지 변환기와 변환 서버의 연동 절차

- ⑥ Animated NBMP 형식의 이미지 생성 지원한다.
- ⑦ 사용자 단말기 프로파일을 사용하여 새로운 단말기의 추가가 용이하다.

이미지 변환을 위하여 이미지 변환기가 변환 서버와 연동하는 절차를 보면 (그림 3)과 같다.

### 3.3 구현 예

다음은 본 서버가 WML로 표현된 WAP 게임[23, 28]을 cHTML[20]로 표현된 일본의 i-Mode용 게임으로 변환하는 예를 보여준다. 실제 서비스되고 있는 WAP 게임에 대하여 게임 서버에 접속하여 변환된 게임 출력을 i-Mode폰 에뮬레이터를 사용하여 진행하는 단계들을 보여준다. 다음의 i-Mode 폰 에뮬레이터가 본 변환 서버를 통하여 게임 서버에 접속하면 게임 진행시마다 바뀌게 되는 게임 장면 이미지들이 이미지 변환기에 의해 변환되고 WML 문서들은 마크업 언어 변환에 의해 cHTML과 GIF 이미지로 구성된 문서로 변환 출판된다.



(그림 4) WAP 게임의 cHTML 게임 변환 사례

## 4. Clipper

본 절에서는 변환 서버와 상호 연동하여, 기존의 웹 페이지들을 무선 단말기에 적합하도록 재구성하는 저작 도구인 Clipper에 대하여 기술한다.

### 4.1 특 징

변환 서버와 연동하는 클라이언트측 도구인 Clipper는 기존의 웹 페이지들을 무선 인터넷 환경에 적합하도록 재구성해 준다. 이러한 Clipper는 편리한 사용자 인터페이스를 통해 사용자 정의 편집 기능을 제공하여, 기존의 웹 페이지에 없었던 부분을 모바일 페이지에 포함시킬 수 있는 기능도 제공한다. 이러한 Clipper는 윈도우즈 API를 사용하여 구현되어 Win32 계열의 운영 체제에 탑재된다.

본 Clipper의 특징으로는 다음과 같다.

- ① Clipper를 사용하기 위하여 복잡한 설정을 하거나 별도의 스크립트를 작성할 필요가 없다.
- ② 웹 브라우저와 유사한 사용자 인터페이스를 이용하여 편리하게 검색한다.
- ③ Drag & Drop 방식으로 웹 문서로부터 원하는 콘텐츠만을 추출한다.
- ④ 무선 인터넷용 페이지를 재구성할 수 있는 편집 기능이 제공된다.
- ⑤ 추출 및 편집 결과는 변환 규칙 WIDL로 작성되어 변환 서버에 저장된다.
- ⑥ Clipper와 변환 서버사이에는 소켓 통신을 한다.

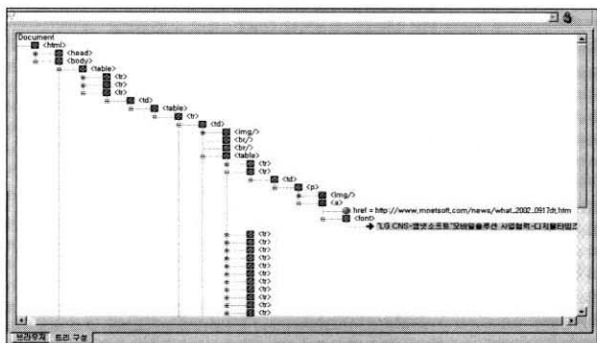
이러한 Clipper가 편리한 편집 방식을 제공하므로 대량의 HTML 문서로 구성된 웹 사이트라 할지라도 무선 서비스를 위한 사이트로 변환하는 시간을 단축할 수 있는 장점이 있다.

#### 4.2 구 현

(그림 5)는 Clipper가 기존의 웹 페이지로부터 추출 작업을 수행하는 두 가지 모드를 보여준다. 그림에서 보는 것처럼 Clipper는 맨 위의 메뉴바/툴바, 왼쪽 상단의 노드편집/미리보기 창, 왼쪽 하단의 환경 설정/노드 검색창, 오른쪽의 브라우저/트리 창 등으로 구성된다.



(a) 웹 문서 clipping 모드



(b) 트리 구조 clipping 모드  
(그림 5) Clipping 모드 화면

#### 4.2.1 브라우저/트리 창

크게 3부분으로 동작하는 브라우저/트리창은 웹 브라우저처럼 동작하는 브라우저창, 클리핑을 위해 웹 브라우저의 HTML 소스를 XHTML로 변환시킨 클리핑 창, 이러한 XHTML 문서를 DOM 형태의 트리 구조로 변경시킨 트리창으로 이루어진다. 무선용 페이지 작성시 트리창을 사용하여 클리핑을 수행하게 되면, 손쉽게 마크업 요소들을 Drag & Drop 방식으로 편집할 수 있다.

#### 4.2.2 노드편집/미리보기 창

노드 편집창은 Clipper에서 브라우저창과 함께 클리핑을 진행하는 중요한 창 중의 하나이다. (그림 5)(a)에서 보는 것처럼 브라우저 창에서 자기가 원하는 내용들을 블록 지정 후, 노드 편집 창으로 이동시키면 간단하게 무선용 페이지를 제작할 수 있다. 노드 편집창은 Drag & Drop 기능 외에도 왼쪽의 8개 버튼을 이용해서 사용자 정의 편집 기능을 이용할 수 있다.

사용자 정의 편집 기능은 실제로는 클리핑시 원본 웹 페이지에는 포함되지 않는 데이터들을 직접 입력시키고자 하는 경우 편리한 기능이다. 이 기능에는 텍스트 입력, 이미지 입력, 폼 입력(한 줄 글상자 입력 버튼, 폼 숨김 필드 입력 버튼, 누름 단추 입력 버튼), 하이퍼링크 입력, 줄 바꾸기 입력 등이 있다. 이 버튼들을 이용하여 데이터를 입력시키고, 입력된 데이터를 편집하고자 할 때에는 Edit 버튼을 사용하고, 삭제시에는 Delete 버튼을 사용한다.

노드 편집창에서 각 노드를 이동 배치시키고자 할 경우에는 상, 하 화살표 버튼을 사용하면 쉽게 문서내에서 이동된다. 미리 보기(Preview) 창은 사용자가 클리핑한 내용을 휴대폰이나 PDA를 사용하지 않고 미리 확인해 보고 싶은 경우 미리보기 탭을 클릭하면, 간단히 닫닫기 상에 보여질 화면을 미리 볼 수 있다.

#### 4.2.3 환경 설정/노드 검색창

환경 설정 창에서는 사용자가 무선용 페이지를 전송받을 무선 단말기 환경을 휴대폰용 또는 PDA용으로 설정할 수 있다. 이 창은 클리핑된 새 모바일 페이지의 페이지명을 입력하는 항목과 작성된 페이지 특성에 따라 일반 페이지, 게시판, 사용자별 페이지로 나누어 구별하는 항목으로 구성된다. 노드 검색창은 노드 편집창과 함께 사용되는 것으로서 클리핑하고자 하는 웹 페이지의 특정 부분을 검색하여 찾을 수 있다. 이 기능은 문서가 크고 복잡할 때 유용하다.

#### 4.2.4 메뉴바/툴바

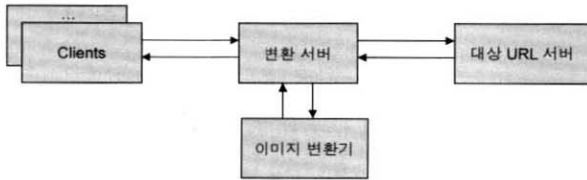
메뉴바에는 파일, 연결, 주소 입력창, 웹 브라우저 기능 등으로 구성된다.

- 파일 : 저장하기, 불러오기, 백업파일에서 불러오기, 종료하기 등의 4개 부메뉴로 구성되어 있다.
- 저장하기(☐) : 클리핑 작업 결과를 저장하면 변환 규칙 WIDL 파일로 저장되고 변환 서버로 전송된다. 변환 서버에 저장된 후에는, 휴대폰이나 PDA 등으로 URL 접근시 클리핑된 페이지를 볼 수 있다. 변환 서버와 연결되어 있지 않은 경우에는 백업 파일로 저장한다.
- 불러오기(☐) : 변환 서버에 저장된 변환 결과 파일들을 불러와서 재편집할 수 있게 한다.
- 백업파일에서 불러오기 : Clipper가 설치된 시스템에 저장된 백업 파일을 불러와서 재편집할 수 있게 한다.
- 연결 : 변환 서버와 연결을 설정하며 변환 서버 IP, 접속 포트 번호, 현재 Clipper를 사용하는 사용자 ID/Password 등을 입력할 수 있다.

5. 실험 환경 및 결과

본 절에서는 변환 서버의 성능을 평가하기 위한 척도로서 처리량 및 응답시간을 기준으로 사용하였다[16].

5.1 실험 환경



(그림 6) 실험 구성

일반적인 사용자들의 웹 문서 접근 양식과 변환 서버의 구성을 고려하여 (그림 6)과 같이 실험 환경을 구성하였으며, 이 실험 환경은 3대의 Client용 PC, 변환 서버용 1대, 대상 URL을 적재한 웹 서버 1대로 구성하였다. 이들의 상세한 사양은 <표 1>과 같다.

<표 1> 실험 장비 현황

항 목	대 수	CPU(MHz)	RAM(MB)	운영체제
클라이언트용 PC	3	PIII 650/750/866	256	W2K Professional
서버	1	PIII 1.1GHz	1GB	Redhat Linux 7.1
대상 URL용 웹 서버	1	PIII 600	256	W2K Professional/ Apache 1.3

클라이언트 모듈에서 변환 서버에 변환 요청 방법은 사용자의 일반적인 웹 문서 접근 양식을 고려하였다. 즉, 클라이언트가 하나의 대상 URL에 대한 변환 요청을 변환 서

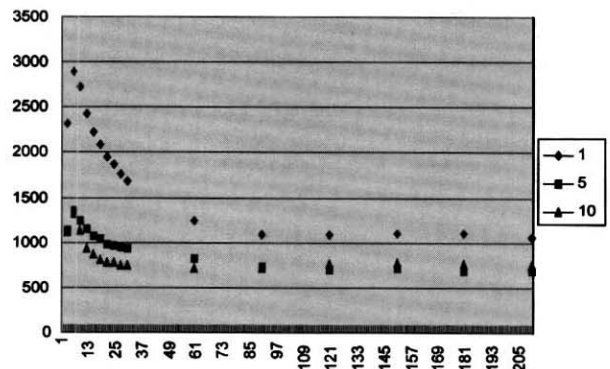
버로 전송하고 그에 대한 변환 결과를 받은 후, 다시 새로운 변환 요청을 발생시키는 방식으로 실험을 수행하였다. 본 실험에서는 변환 서버내 변환 스레드들의 스레드 개수를 어느 정도 범위에서 유지하는 것이 적절한지 분석하고자 하였으며, 또한, 변환 서버의 처리 능력 외에도 안정성 검증에도 비중을 두어 사용자의 생각 시간(think time)을 0으로 설정하여 변환 서버의 스트레스 테스트도 병행하여 실험하였다[16].

변환 서버에서 동작하는 변환 엔진 스레드 풀의 스레드 개수는 1, 5, 10개 순서로 증가시켰으며 각 경우에 대해 Client 모듈이 발생시키는 요청 개수를 1, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 60, 90, 120, 150, 180, 210개로 증가시켜가면서 측정하였다. 클라이언트 요청 개수를 변경하여 실험을 진행할 때 최초 30초 이내 응답 시간은 전이 상태 현상으로 간주하고 이후 1분간의 응답에 대해서 응답 시간을 수집하였다.

5.2 실험 결과

5.2.1 정적인 웹 문서

대상 URL의 최초 접근시 대상 URL 사이트로부터 원본 웹 문서들을 변환 서버로 가져온다. 이후부터는 변환 결과가 변환 서버의 캐쉬에 보관되어 있으므로 대상 URL로부터 원본을 가져올 필요없이 변환 서버에서만 변환 작업이 이루어진다. (그림 7)에서 x축은 클라이언트 모듈에서 생성시킨 사용자 요청 개수를 나타내며, y축은 변환 서버의 처리량(requests/second)을 나타낸다. 각 그래프는 변환 엔진에서 수행되는 스레드 개수를 1, 5, 10개로 설정하였고 클라이언트 모듈에서 발생시키는 요청 수를 증가시키면서 처리량의 변화를 관찰하였다.



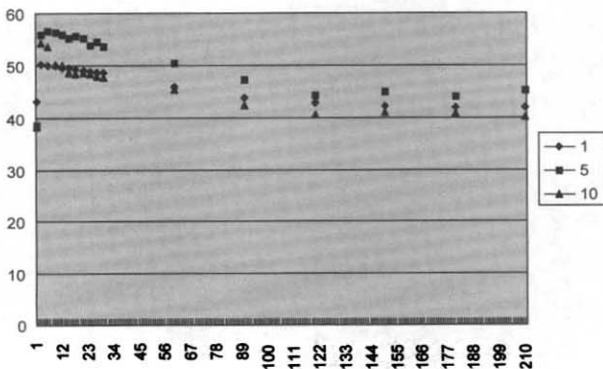
(그림 7) 처리량 : 정적 웹 문서의 경우

이 실험에서는 정적 URL 문서를 대상으로 하였으므로 한번 변환된 후 변환 결과가 변환 서버내 캐쉬에 보관되어 있어 사용자 요청에 대한 처리가 변환 서버 내부에서만 이루어지는 특성이 있다. 이 경우 변환 스레드가 수행하는 작

업은 단지 변환 서버내 캐쉬로부터 URL에 대응하는 변환 결과를 찾아 사용자 단말기로 전송하는 것뿐이다. 따라서, 변환 서버가 첫 번 검색시 URL 서버에 접근하는 경우를 제외하고는 네트워크를 통해 URL 서버에 접근하지 않게 되므로, URL 서버에 대한 접근 지연시간이 소요되지 않게 된다. (그림 7)을 보면, 스레드 개수가 1인 경우의 처리량이 5, 10인 경우보다 높게 나타났다. 이것은 스레드 개수가 1인 경우가 스레드풀의 오버헤드가 없었기 때문에 보다 나은 처리량을 나타낸 것으로 분석된다. 이 그래프에서 최대치들을 보면, 스레드 개수가 1, 5, 10일 때, 각각 최대 2897.83, 1353.86, 1323.68 요청/초의 처리량을 나타내었다.

5.2.2 동적인 웹 문서

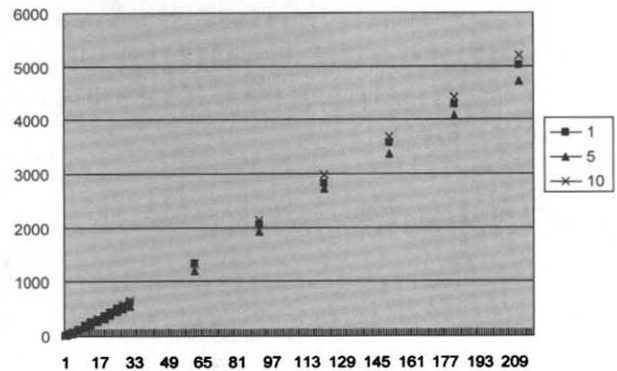
이 실험은 정적인 웹 문서와 달리, 대상 URL의 내용들이 사용자 요청시마다 계속 바뀌는 경우이다. 3.3절에서 소개한 WAP 게임을 i-Mode용 게임으로 변환하는 경우가 대표적인 예라고 할 수 있다. 이러한 사례에서는 변환 서버가 내보낸 변환 결과에 대해 사용자의 반응이 또 다른 게임 장면으로 이동하는 것이 일반적이다. 따라서, 변환 서버상의 캐쉬 효과는 없고 변환 서버는 사용자 요청시마다 대상 URL에 접근하여 요청된 화면을 검색한 후 변환 작업을 수행하여야 한다. 이러한 경우는 변환 서버에 사용자 요청 개수가 증가하면 할수록 서버의 부하가 계속 증가하는 결과를 초래한다.



(그림 8) 처리량 : 동적 웹 문서의 경우

(그림 8)은 변환 서버의 변환 엔진 스레드 풀이 보유한 스레드 개수를 1, 5, 10개로 설정하고 사용자 요청 개수를 1에서부터 210개까지 증가시켜 가면서 변환 서버의 처리량을 관찰한 결과를 보여준다. 여기서 Y축의 단위는 초당 처리된 요청 수(requests/sec)를 나타낸다. 변환 서버내 스레드 풀의 스레드 개수가 5개인 경우가 가장 우수한 성능을 나타내었고, 10개의 스레드를 스레드풀에 유지한 경우가 오히려 스레드 개수를 1개만을 유지한 경우보다 낮은 성능을

나타내었다. 이것은 사용자 요청에 대해 스레드 풀내 스레드를 배정하고 반환하는 오버헤드가 하나의 WAP 게임 화면의 이미지 변환 시간보다 더 오랜 처리 시간이 소요된 것으로 분석된다. 또한, 스레드 수가 5인 경우를 보면, 전체적으로 사용자 요청 개수를 30개까지 증가시킨 구간에서는 변환 서버의 처리 성능이 잘 발휘하고 있는 것으로 나타났으나, 그 이후부터는 계속 감소하다가 사용자 요청 개수를 120개이상 발생시킨 구간이후부터는 처리량이 정체된 모습을 나타내었다.



(그림 9) 평균 응답시간 : 동적 웹 문서의 경우

(그림 9)는 (그림 8)과 동일한 조건하에서 사용자 요청들의 평균 응답시간을 측정된 결과를 보여준다. X축은 사용자 요청 개수를 나타내며 Y축은 평균응답시간을 ms 단위로 표시하였다. 이 그래프는 사용자 요청 개수가 증가함에 따라 평균 응답시간이 거의 선형적으로 증가하고 있는 결과를 보여준다. 대략적으로 사용자 요청 개수가 180개 이하 구간에서는 평균 응답시간이 약 4초이하를 기록하였다.

6. 결 론

유선 콘텐츠와 비교해 볼 때 무선 콘텐츠들은 공급자뿐만 아니라 콘텐츠의 양에 있어서도 매우 부족한 수준이다. 현재 무선망에 대한 개방이 시행되면 유선 포털들이나 유선 사이트를 보유한 기관들이 자신들의 독자적인 무선 인터넷 서비스를 실시할 것으로 예상되기 때문에 유선 콘텐츠들의 무선 콘텐츠로의 변환 요구는 매우 높다.

본 연구의 변환 서버는 사용자의 단말기로부터 접속을 자동으로 인식하여 단말기가 요청한 URL에 대한 정보를 통해 대상 URL상의 원본 문서를 가져온 후, 이들을 파싱하여 중간 언어인 metaXML로 생성하고 이 문서를 다시 사용자 단말기에 적합한 마크업 언어 문서로 변환한다. 이미지 변환기는 대상 URL로부터 가져온 이미지를 사용자



단말기에 적합한 형식으로 실시간 변환하여 전송하며, 변환된 이미지는 다음의 재요청에 대비하여 캐쉬에 보관한다. JPG, GIF, BMP, PNG, TIF 등의 이미지 형식들을 무선 인터넷용 이미지 형식들인 BMP, WBMP, NBMP으로 변환을 수행한다. Clipper는 편리한 사용자 인터페이스의 제공을 통해 기존의 웹 페이지들에 대해 무선 인터넷 환경에 적합하도록 재구성해 준다. 대상 URL의 웹 페이지가 정적인 경우와 동적인 경우에 대하여 성능 실험을 수행하여 변환 서버의 안정성을 검증하였을 뿐만 아니라 스타드 플에 따른 처리량 및 평균 응답시간을 관찰하였다.

향후 연구 과제들로는 기존 웹 사이트들에 문법적으로 부정확한 문서들이 존재하므로 이에 대한 에러 처리 능력을 향상시킬 필요가 있으며, JavaScript 등으로 작성된 웹 페이지에 대한 사용자 요청을 처리할 수 있는 기술에 대한 연구 등이 진행되어야 한다. Clipper에 대해서는 보다 다양한 편집 기능 제공, 다수의 관리자가 존재하는 마스터 컨트롤 제공자나 교육용 응용시 다수의 사용자 식별자 지원 등의 기능 개선이 요구된다. 성능 면에서는 변환 서버의 스타드 플 관리에 대한 성능 향상 방안이 연구되어야 할 것이다.

## 참 고 문 헌

- [1] C. Anderson, P. Domingos, and D. Weld, "Personalizing Web sites for Mobile Users," Proc. of the 10th Int. WWW Conference, Hong Kong, pp.565-575, May, 2001.
- [2] V. Anupam, J. Freire, B. Kumar and D. Lieuwen, "Automating Web Navigation with the WebVCR," Proc. of the 9th Int. WWW Conference, Amsterdam, Netherlands, pp.503-517, May, 2000.
- [3] T. Bickmore, A. Girgensohn and J. Sullivan, "Web page filtering and re-authoring for mobile users," The Computer Journal, Vol.42, No.6, pp.534-546, 1999.
- [4] T. Bickmore and W. Schilit, "Digester : device-independent access to the World Wide Web," Computer Networks and ISDN, Vol.29, No.8, pp.1075-1082, 1997.
- [5] O. Buyukkokten et al., "Power Browser : Efficient Web Browsing for PDAs," Proc. Conf. Human Factors in Computing Systems(CHI 00), pp.430-437, ACM Press, NY, 2000.
- [6] N. Chase, *XML and Java from Scratch*, Que Publishing co., 2001.
- [7] J. Freire and B. Kumar, "Web Services and Information Delivery for Diverse Environments," Proc. of the VLDB Workshop on Technologies for E-Services, 2000.
- [8] F. Gonzalez-Castano, A. Rifon, and E. Costa-Montenegro, "A New Transcoding Techniques for PDA Browsers, based on Content Hierarchy" Proc. of Fourth International Symposium on HCI with Mobile Devices, Italy, 2002.
- [9] H. Hori, et al., "Annotation-based Web Content Transcoding," Proc. of the 9th International WWW Conference, Netherlands, 2000.
- [10] Y. Hwang, C. Jung, J. Hong, and S. Chung, "WebAlchemist : A Web Transcoding System for Mobile Web Access in Handheld Devices," Proc. of Mobile Computing Data Management, Denver, Aug., 2001.
- [12] B. Knutsson, H. Lu, J. Mogul and B. Hopkins, "Architecture and Performance of Server-Directed Transcoding," ACM Trans. on Internet Technology, Vol.3 No.4, pp.392-424, Nov., 2003.
- [13] K. Nagao, Y. Shirai and K. Squire, "Semantic Annotation and Transcoding : Making Web Content More Accessible," IEEE Multimedia, Vol.34, No.2, pp.69-81, 2001.
- [14] A. Sahuget and F. Azavant, "Building light-weight Wrappers for Legacy Web Data Sources using W4F," Proc. of VLDB, pp.738-741, 1999.
- [15] B. N. Schilit et al., "Web Interaction using Very Small Internet Devices," IEEE Computer Vol. 35, No.10, pp.37-45, IEEE Computer Society, 2001.
- [16] IBM, "웹 기반 시스템 하에서의 성능 이론 및 웹사이트 진단 사례", Technical Report, 2002.
- [17] Amazon Anywhere, <http://www.amazon.com/anywhere>.
- [18] AnyBiulder, <http://www.anybil.com/>.
- [19] Avantgo, <http://www.avantgo.com>.
- [20] cHTML, <http://www.nttdocomo.com/source/tag/index.html>.
- [21] NeoMtel, <http://www.neomtel.co.kr/>.
- [22] New York Times, <http://channel.nytimes.com/partners/palmpilot/index.html>.
- [23] Openwave, <http://www.openwave.com/>.
- [24] ProxiWeb, <http://www.proxinet.com/>.
- [25] Resource Description Framework(RDF), <http://www.w3.org/TR/REC-rdf-syntax/>.
- [26] Tidy, <http://www.w3.org/People/Raggett/tidy/>.
- [27] WAP forum, <http://www.wapforum.org/>.
- [28] WAP 게임, <http://www.wirelessgames.com>.
- [29] WIDL, <http://www.w3.org/TR/NOTE-widl>.
- [30] Xerces & Xalan, <http://xml.apache.org/>.
- [31] XHTML, <http://www.w3.org/TR/xhtml2/>.
- [32] XSL, <http://www.w3.org/Style/XSL/>.



### 조 승 호

e-mail : shcho@kangnam.ac.kr  
1985년 서울대학교 전자계산기공학과(학사)  
1989년 서울대학교 전산과학과(석사)  
1993년 서울대학교 전산과학과(박사)  
1985년~1987년 삼성전자 컴퓨터부문  
연구원

1997년~1999년 Maryland 대학교 및 Johns Hopkins 대학  
연구교수

1993년~현재 강남대학교 컴퓨터미디어공학부 부교수  
관심분야 : 병렬 I/O, 실행시간 시스템, 모바일 컴퓨팅



### 차 정 훈

e-mail : meaculpa@mnetsoft.com  
1998년 공주대학교 전기공학과 졸업  
2001년 공주대학교 대학원 전기전자  
정보공학(공학석사)  
2000년~현재 (주)엠넷소프트 모바일  
인터넷 연구소 선임연구원

관심분야 : 모바일 컴퓨팅, 멀티미디어 변환