

# 웹 에이전트를 이용한 웹기반 교수-학습 시스템의 설계 및 개발

김갑수 · 이건민

서울교육대학교 컴퓨터 교육과

## 요약

최근에, 컴퓨터를 이용한 학습은 CAI 환경에서 WBI 학습 환경으로 옮겨가고 있다. WBI 학습에 필요한 대부분의 웹 문서는 검색 엔진을 이용하여 얻는다. 교수자는 검색한 웹 문서의 사용가능성을 평가한 후 학습 자료로 사용한다. 그러나 이런 교수-학습 과정은 다음과 같은 문제점이 있다. 첫째, 교수자에 의해 선정된 웹 문서를 반복해서 검색한다. 둘째, 웹 문서를 학습자에게 제시하기 위한 별도의 교수 설계 과정이 필요하다. 셋째, 학습자의 평가 결과와 웹 문서 사이의 관련성 분석이 매우 어렵다. 본 연구에서는 WBI 학습에 필요한 웹 문서를 검색하고 학습자에게 학습 과정을 안내하는 WAILS(Web Agent Instruction Learning System)를 제안한다. WAILS는 웹 에이전트를 이용하여 WBI 학습에 필요한 웹 문서를 수집한다. 교수자는 수집한 웹 문서를 평가하고 교수-학습 생성기를 통해 학습자에게 제시한다. 교수자는 WAILS를 이용하여 웹 문서의 검색과 교수-학습 설계를 동시에 하게 된다. 이는 WBI 학습을 더욱 촉진시키게 된다.

## An Implementation and Design Web-Based Instruction-Learning System Using Web Agent

Kim, Kap Su · Lee, Keon Min

Seoul National University of Education, Dept. of Computer Education

## ABSTRACT

Recently, the current trend for computer based learning is moving from CAI environment to WBI environment. Most web documents for WBI learning are collected by aid of search engine. Instructors use those documents as learning materials after they evaluate availability of retrieved web documents. But, this method has the following problems. First, we search repeatedly the web documents selected by instructor. Second, there is a need for another course of instruction design in order to suggest the web documents for learner. Third, it is very difficult to analyze for relevance between the web documents and test results. In this work, we suggest WAILS(Web Agent Instruction Learning System) that retrieves web documents for WBI learning and guides learning course for learners. WAILS collects web documents for WBI learning by aid of web agent. Then, instructors can evaluate them and suggest to learners by using instruction-learning generating machine. Instructors retrieve web documents and the instruction-learning design at the same time. This can facilitate WBI learning.

## 1. 서론

정보화 사회가 도래함에 따라 사회는 더욱 복잡해지고 다양해져 가고 있다. 이에 교육현장에는 획일적인 대량 교육에서 탈피하여 창조력과 개성을 키우는 교육, 개인을 존중하는 교육을 실시하려는 노력이 일고 있고, 그 일환으로 개개 학생의 능력과 배경에 알맞은 개별화 학습 지도가 이루어지고 있다[1]. 개별화 학습을 위해 시청각 매체활용 자습 교수법, 개별 처방 교수법, 개별지도 교수법, 학습자 요구부응 교수법 등의 다양한 방법이 제시되었다. 컴퓨터를 이용한 개별화 학습도 그 중의 하나로서 학습자에게 지속적인 흥미, 즉각적인 피드백, 학습 진전의 측정 등을 제공한다[2]. 특히, 근래에는 과거 CAI 위주의 컴퓨터 학습 환경을 벗어나 인터넷을 이용한 컴퓨터 학습 환경을 바탕으로 교수-학습이 이루어지고 있다. 인터넷 학습 환경을 구축하려는 노력은 웹을 교육적으로 활용하려는 움직임으로 나타나 새로운 교수 모형인 WBI 학습이 등장하였다. WBI 학습은 정보의 다양성, 역동성, 상호작용성, 즉시성 등을 학습자에게 제공하는 장점이 있다. 이를 위해 WBI 학습을 원활하게 할 수 있도록 도와주는 웹 기반 교수-학습 시스템을 사용한다. 그러나 현재의 웹 기반 교수-학습 시스템은 WBI 학습을 위한 웹 문서의 선정과 교수-학습 설계에 다음과 같은 문제점을 안고 있다.

첫째, 교수자에 의해 선정된 웹 문서는 효용 가치가 매우 높음에도 이를 체계적으로 관리하는 도구가 존재하지 않아 같은 문서를 반복해서 검색한다. 교수자가 선정한 웹 문서는 학습자에게 가장 적절한 것이다. 그러나 현재 WBI 학습에서는 이를 단지 일시적인 학습 자료로만 활용할 뿐 다른 교수-학습 상황에서 공유할 수 있는 기회를 제공하지 못한다.

둘째, 교수자가 학습 목표에 적합한 웹 문서를 검색한 후 학습자에게 제시하기 위해서는 별도의 교수 설계 과정이 필요하다. 현재, 교수자는 검색한 웹 문서를 학습자에게 제공하기 위해 별도의 교수 설계를 해야 한다. 이는 교수자로 하여금 별도의 HTML 문서를 작성하도록 요구한다. 이런 과정은 교수-설계 시간의 낭비를 가져오고 HTML 문서에 대한 교수자의 부담을 가져온다. 그러므로 검색한 웹 문서를 즉시 활용 가능한 자료로 구성하는 도구가 요구된다.

셋째, 학습자의 학습 결과와 웹 문서와의 관련성 분석이 매우 어렵다. WBI 학습에서 어떤 웹 문서를 학습 자료로 이용하였을 때 학습 목표 도달에 용이한가는 웹 문서의 선택에 큰 영향을 미친다. 여기에 웹 문서를 학습 자료로 이용하여 학습한 결과를 지속적으로 누적 평가하여 선택한 웹 문서에 반영하는 도구가 필요하다. 이에 본 연구는 WBI 학습에 필요한 웹 문서를 검색하여 학습자에게 학습 과정을 안내하는 WAILS(Web Agent Instruction Learning System)의 개발을 목적으로 한다. WAILS는 웹 에이전트 프로세스, 인터페이스, 교수-학습 프로세스로 구성된다. 웹 에이전트 프로세스는 웹 상에 존재하는 문서에 대한 정보를 수집하여 교수자에게 제공한다. 교수-학습 프로세스는 교수자가 웹 문서를 선택하고 평가하는 작업을 처리한다. 인터페이스는 교수자의 정보·학습 내용을 요구하고 교수-학습에 필요한 자료를 제시한다.

본 연구에서 WAILS에 필요한 웹 문서는 초등학교 5학년 1학기 교육 과정 관련 문서로 한정하며 메타 검색을 수행하여 수집한다. 또한 WAILS의 검색 방법은 연산자를 사용하지 않는 단순 키워드 검색으로 한정한다. 그리고 웹 문서에 대한 평가는 교육 전문가가 실시한다.

## 2. 이론적 배경

에이전트에 대해 한 마디로 정의하기에는 너무나 많은 논의가 있어 왔다. 이는 자칫 에이전트의 정의를 모호하게 하는 측면이 있다. 그러나 지금까지의 연구 논문에 나타난 에이전트의 특성에 대해 자세하게 알아봄으로써 에이전트의 정의에 접근하고자 한다. 이 장에서는 에이전트에 대해 다양한 시각을 통해 정의를 하고 에이전트가 갖추어야 할 속성에 대해 기술한다.

### 2.1 에이전트의 정의

에이전트에 대한 정의는 연구 논문마다 다양하게 나타난다. 예를 들어 테드 셀커(Ted Selker)는 사용자 자신이 해야만 하는 작업을 처리하는 방법을 알고 있는 소프트웨어로, 러키-톤손(Lecky-Thomson)은 작업을 성공적으로 완수하기 위해 적절한 방법으로 환경에서 정보를 수집하여 주어진 작업을 수행하

는 소프트웨어로 에이전트를 정의한다[3]. 이외에도 에이전트에 대한 연구자들의 정의는 제각각 다르게 나타난다. 본 논문에서는 에이전트가 가지는 일반적인 특성을 통해 에이전트가 무엇이며 다른 프로그램과 어떻게 구별되는지 알아보는 접근 방식을 통해 정의를 내리고자 한다. 지금까지 연구·개발되었거나 개발중인 에이전트의 특성은 다음과 같다.

- 특정 목적에 대하여 사용자를 대신하여 작업을 수행하는 자율적인 프로세스
- 독자적으로 존재하지 않고 어떤 환경의 일부이거나 그 안에서 동작하는 시스템
- 지식베이스와 추론 기능을 가지며 사용자, 자원 또는 다른 에이전트와의 정보교환과 통신을 통해 문제 해결을 도모하는 프로그램
- 스스로 환경의 변화를 인지하고 그에 대응하는 행동을 취하며, 경험을 바탕으로 학습하는 기능을 가지는 시스템
- 수동적으로 주어진 작업만을 수행하는 것이 아니고, 자신의 목적을 가지고 그 목적 달성을 추구하는 능동적 자세를 지닌 프로그램

위의 내용은 에이전트의 일반적인 특성이다. 이들을 고려해 볼 때 에이전트는 특정한 목적을 가지고 사용자를 대신하여 작업을 수행하면서, 작업 환경의 변화를 인지하여 그에 대응하는 행위를 취하는 자율적인 프로세스라고 할 수 있다.

## 2.2 에이전트의 속성

에이전트의 속성은 에이전트가 갖추어야 할 필수적인 구성 요소다. 에이전트의 속성에 따라 에이전트의 종류를 분류한다. 에이전트의 속성은 에이전트의 정의를 더욱 명백하게 해주는 역할을 한다. 일반적으로 언급되는 에이전트의 속성으로는 자율성(Autonomy), 지능(Intelligence), 이동성(Mobility), 사회성(Social Ability), 반응성(Reactivity)이 있다.

자율성은 사용자와 지속적인 상호작용이나 감독 없이 작업을 수행하거나 인간의 간섭을 전혀 받지 않고 독자적으로 행동하는 능력을 말한다[4]. 이는 단순히 사용자의 입력을 처리하는 수준을 넘어 일정한 목표를 갖고 환경의 변화를 감지하여 변화에 대처하는 능력을 갖추고 있다는 말이다. 자율성을 통해 에이전트 시스템은 다른 일반 프로그램과 구별되며

사용자나 다른 에이전트의 직접적인 지시나 간섭 없이도 스스로 판단하여 행동하는 능력을 갖추게 된다.

지능은 어느 정도 독립성이나 자율성을 가지고 사용자나 다른 프로그램의 행위를 근거로 하여 일련의 작업을 수행함에 있어 지식이나 사용자의 목표를 활용하는 능력이다[5]. 지능을 갖춘 에이전트는 지속적으로 에이전트의 작업 환경의 동적인 변화를 감지하며, 환경의 상태에 영향을 주고 추론을 통해 문제를 해결한다. 이런 능력을 갖춘 에이전트를 지능형 에이전트라 하며 인공지능과 맥락을 같이한다. 지능형 에이전트 시스템은 지식 베이스와 추론 능력을 갖추고 사용자의 의도를 파악하여 계획을 세우고 학습을 통하여 새로운 지식을 스스로 터득하는 능력을 반드시 갖추게 된다.

이동성은 사용자에게 의해 지정된 특정 작업을 수행하기 위해 에이전트 자신의 통제 하에 네트워크상의 노드에서 노드로 이주할 수 있는 능력이다[6]. 이동성에 중점을 두고 설계된 에이전트를 이동 에이전트라 한다. 이동 에이전트는 언제 어느 때 네트워크상의 노드를 이주할 지를 선택하며 자신의 실행을 통제하고 지속적으로 네트워크를 이동한다. 이동성을 통해 에이전트는 사용자가 요구한 작업을 현재의 호스트에서 수행하지 않고 실제 그 작업을 처리하는 호스트로 이동시켜 수행할 수 있는 능력을 갖추게 된다.

사회성은 전형적인 프로그램과 프로그램 수준의 통신을 벗어나 인간과 비슷한 에이전트 통신 언어를 사용하여 사용자 또는 다른 에이전트 시스템들과 통신하는 능력이다[7]. 에이전트의 통신 언어로 KQML(Knowledge Query and Manipulation Language)을 일반적으로 많이 사용한다. KQML은 에이전트간의 상호접속과 정보교환을 위해 만들어진 언어이며 프로토콜의 집합체이다[8]. KQML을 통해 에이전트는 사회성을 지니게 되고 다양한 종류의 에이전트를 사용하여 다중 에이전트 시스템을 구현한다.

반응성은 에이전트가 환경(여기서 환경은 물리적 환경, GUI를 통한 사용자, 다른 에이전트, 인터넷 또는 이 모든 것들을 조합한 것이 될 수 있음)을 감지하고 환경의 변화에 시기 적절하게 대처하는 능력이다[9]. 반응성을 갖춘 에이전트는 자신의 환경 변화나 다른 에이전트의 메시지에 반응을 하고 수행해

야 할 작업에 대해 능동적인 계획을 세운다.

### 2.3 에이전트 개발 사례

에이전트 개발을 위해 사용된 프로그래밍 언어로는 C, C++, 텔레스크립트(Telescript), Tcl, 자바(Java) 등이 있다. 본 논문의 WAILS 시스템은 이중 자바를 사용하여 개발한다. 자바는 네트워크 프로그래밍을 위해 개발된 언어로 현재 상당수의 에이전트들이 자바를 이용하여 개발되고 있다. 자바를 이용하여 개발된 대표적인 시스템으로는 AWB(Aglets Work-bench)[10], Odyssey[11], Concordia[12], Voyager[13], Ara[14], JAT Lite[15]가 있다. <표1>에 이들 시스템의 특징을 보인다.

AWB는 IBM사의 Tokyo Research Lab(TRL)에서 개발한 것으로 네트워크 기반의 이동 에이전트 응용 프로그램을 작성하는데 시각적 환경을 제공한다. 실행 환경은 Tahiti라 불리는 GUI 기반의 자바 어플리케이션으로 에이전트의 관리를 쉽게 하였으며 프레임워크인 JAAPI(Java Aglets API)를 사용하여 보다 쉽게 개발 할 수 있다[16].

Odyssey는 General Magic사에서 개발한 에이전트 시스템으로 텔레스크립트로 작성되었다가 자바 기반으로 이식되었다. Odyssey는 자바 클래스 라이브러리안에서 개발자가 에이전트 응용프로그램 개발을 가능하게 해 주며 Sun의 RMI(Remote Method Invocation), Microsoft사의 DCOM을 지원하는 API를 제공한다[17].

<표1> 자바를 이용하여 개발된 에이전트 시스템

시스템	주요 특징
AWB	ATP(Agent Transfer Protocol)을 이용하여 에이전트 이동가능
Odyssey	자바 클래스 형태 안에서 텔레스크립트(Telescript)의 개념을 효과적으로 구현
Concordia	컴포넌트를 사용하여 분산 어플리케이션을 개발할 수 있는 완벽한 환경 제공
Voyager	객체 메시지 교환의 광범위한 기능 제공
Ara	이형 네트워크에서 이동 에이전트가 이식성을 갖고 안전하게 실행되는 플랫폼 제공
JAT Lite	인터넷에 대한 접속/접속 해제, 메시지 전송과 수신, 다른 에이전트와의 정보 교환 가능

Concordia는 Mishubishi사가 개발한 것으로 이동 에이전트 어플리케이션의 개발과 관리를 위한 프레임워크를 제공한다. Concordia 시스템은 표준 자바 VM(Virtual Machine), 서버, 에이전트들로 구성되어 있다. 자바 VM은 대부분의 네트워크 플랫폼에 유용한 표준 환경을 제공하며 Concordia 서버는 에이전트의 생명 주기를 관리한다.

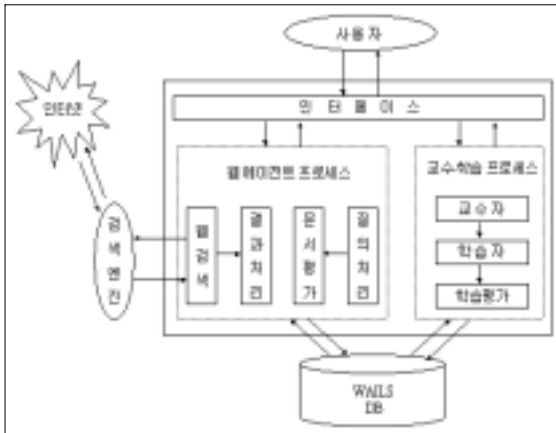
Voyager는 Object-Space사가 자바로 구현된 에이전트 확장 분산 컴퓨팅을 위해 개발한 플랫폼으로 자바 기반 ORB(Object Request Broker)의 속성을 조합하여 에이전트 시스템을 구현한다.

Ara는 독일의 Kaiserslautern 대학의 분산 시스템 그룹이 개발한 것으로 무선 접속이나 간헐적인 접속이 필요한 컴퓨터 같은 저접속/고볼륨(Weak Connection/High-volume) 시스템 또는 대역폭/접속과 데이터의 비율이 중요한 대형 분산 데이터시스템에 대한 응용프로그램 작성을 목적으로 한다.

JAT Lite는 Standford 대학에서 인터넷에서 통신을 하는 소프트웨어 에이전트를 프로그래머들이 빠르게 개발할 수 있도록 하기 위해 자바를 사용하여 개발된 프로그램 패키지다. JAT Lite는 자바 템플릿(Template)과 독특한 자바 에이전트 하부구조를 제공하여 쉽게 에이전트 시스템을 구성할 수 있으며 점차적으로 특정 레이어(Layer)로 옮겨가는 계층구조를 갖는다.

### 3. WAILS 시스템 설계

WBI 학습에 이용되는 웹 문서는 대부분 교수자에 의해 선별됨으로써 묵시적인 평가가 이루어진다. 이런 평가 결과를 지속적으로 누적한다면 WBI 문서에 대한 보다 정확한 평가와 함께 학습 목표에 적합한 웹 문서를 제공하게 된다. 그러므로 교수자의 평가를 누적하기 위해 WAILS는 웹 문서의 수집과 검색 및 평가 그리고 교수-학습 설계 과정이 동일한 공간에서 이루어지도록 구성될 필요가 있다. 이를 위해 WAILS는 웹 에이전트 프로세스(Web Agent Process), 교수-학습 프로세스(Instruction-Learning Process), 인터페이스(Interface)로 구성한다. WAILS의 전체적인 시스템 구조는 <그림 1>과 같다.



<그림>] WAILS 시스템 구조

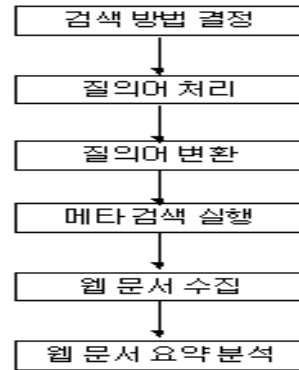
웹 에이전트 프로세스는 교수-학습을 위한 웹 문서를 수집하여 WAILS DB에 저장하고 교수자의 검색 요청을 처리하기 위해 구성한다. 교수-학습 프로세스는 교수-학습에 필요한 웹 문서를 일정한 형태로 제공하고 학습자의 학습 결과를 저장하는데 필요하다. 학습자와 WAILS의 상호작용을 원활히 하고 사용자로부터 발생하는 이벤트를 받아 적절한 메시지 처리를 위해 인터페이스가 요구된다.

### 3.1 웹 에이전트 프로세스

웹 에이전트 프로세스는 웹 문서를 수집하고 교수자의 검색 요청을 처리하는 역할을 한다. 그러므로 웹 에이전트 프로세스는 웹 검색을 원활하게 하고 교수자의 검색 요청의 종류를 판별하여 처리하도록 설계한다. 웹 문서 검색 요청을 처리하기 위한 웹 검색 모듈, 웹 문서 평가를 위한 문서 평가 모듈, 교수자의 질의 처리를 위한 질의 처리 모듈, 각각의 작업 결과 처리를 위한 결과 처리 모듈로 구성된다. 각각의 모듈이 수행되는 과정은 <그림2>와 같다.



<그림2> 웹 에이전트 프로세스의 처리 과정도



<그림3> 웹 검색 모듈의 검색 과정도

시스템 관리자의 요청에 의해 웹 검색 모듈은 메타 검색을 수행하여 웹 문서를 수집한다. 수집된 문서 정보는 WAILS DB에 저장되며 질의 처리 모듈에 의해 교수자의 검색을 수행하는 자료로 이용된다. 질의 처리 모듈은 교수자의 검색 요청을 처리하여 적절한 결과를 제시한다. 평가 모듈은 사용자의 평가와 학습 결과에 근거하여 웹 문서에 대한 평가를 한다. 평가가 끝난 웹 문서는 결과처리 모듈을 통해 WAILS DB에 저장된다.

웹 검색 모듈은 질의를 입력받아 다른 검색 엔진을 이용하여 메타 검색을 수행하거나 색인 파일을 이용하여 주기적으로 검색을 수행한다. 웹 검색 모듈의 전체적인 처리 과정은 <그림3>과 같다.

문서 평가 모듈은 교수자에 의해 평가 된 웹 문서에 평가 가중치를 부여한다. 웹 문서에 대한 평가는 교수자가 교수-설계 과정에서 선택한 문서에 대해서는 적합을 선택하지 않은 문서에 대해서는 부적합으로 구분하여 이루어진다. 적합한 문서에 대해서는 다음의 평가식을 이용하여 가중치를 부여한다.

$$S_w = \sum (B_w + \frac{A_s}{S_s} + \frac{1}{A_s})$$

평가식에서  $S_w$ 는 현재 가중치,  $B_w$ 는 이전 가중치,  $A_s$ 는 검색된 전체 사이트 수,  $S_s$ 는 교수자에 의해 선택된 사이트 수를 나타낸다. 검색된 전체 사이트 중에서 선택된 사이트의 수가 작을수록 그 사이트는 다른 사이트에 비해 학습 주제와의 관련도가 높은 것으로 평가한다. 반면에 전체사이트에서 선택

된 사이트의 수가 많을수록 가중치는 감소하게 된다.

질의 처리 모듈은 WAILS 시스템을 이용하는 교수자의 질의를 처리한다. 질의 처리 모듈은 키워드 검색과 디렉토리 검색을 지원한다. 키워드 검색을 하는 경우 질의 처리 모듈은 입력받은 키워드와 일치하는 키워드 ID를 WAILS DB에서 검색한다. 키워드 ID 검색에 성공하면 키워드 ID와 일치하는 URL을 최종적으로 WAILS DB로부터 가져온다. 검색 결과는 교수자에게 학년, 교과, 단원에 관계없이 종합적으로 보여주게 된다. 디렉토리 검색을 수행하는 경우 학년, 교과, 단원의 순서로 코드를 생성하여 WAILS DB에서 관련 URL을 가져온다. 디렉토리 검색 결과는 학년, 학기, 교과, 단원별로 일목 요연하게 교수자에게 제시된다.

웹 문서 메타 검색 결과는 결과 처리 모듈을 통해 WAILS DB에 저장된다. 웹 문서를 WAILS DB에 저장하기 전에 결과 처리 모듈은 웹 문서에 대한 최초의 우선 순위 가중치를 결정한다. 결과 처리 모듈은 문서를 TFIDF(Term Frequency Times Inverse Document Frequency) 방법을 사용하여 가중치를 갖는 URL을 추출한다. TFIDF는 문서에 자주 나타나는 용어에 가중치를 부여하는 정보 검색 가중치 메커니즘으로  $W = Hc \cdot Tf \cdot idfk$ 로 표현된다. 이는 본래 웹 문서에서 중요 키워드를 산출하는 방법이나 본 논문에서는 여러 검색 엔진에 걸쳐 공통적으로 나타나는 URL에 대한 가중치를 부여하기 위한 기법으로 사용한다. TFIDF 방법에서 Hc는 상수를 나타내며 Tf는 현재 문서에서 자주 나타나는 URL 횟수를 나타낸다. idfk는 수집된 문서 전체에서 URL이 나타나는 빈도 수를 말한다. URL이 결정되면 가중치 우선 순위별로 WAILS DB에 저장한다.

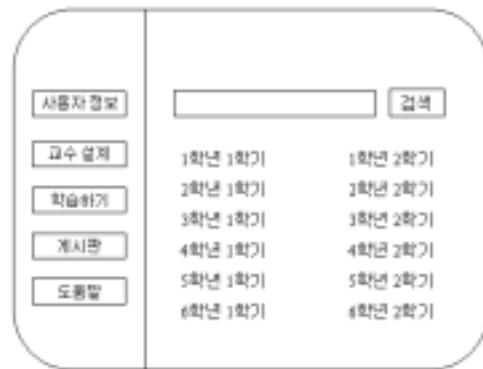
<표2> 학습 내용 코드

구분	학년						학기		교과									
	1	2	3	4	5	6	1	2	국 어	수 학	사 회	자 연	음 악	미 술	체 육	실 과	도 덕	영 어
코드	1	2	3	4	5	6	1	2	01	02	03	04	05	06	07	08	09	10

### 3.2 인터페이스

인터페이스는 사용자의 상호작용과 교수자에 의해 발생하는 이벤트를 처리한다. 상호작용으로는 교수자 정보의 입력, 교수자 질의 생성, 교수-학습 설계가

있다. 교수자의 데이터는 교수자 이름, ID, 비밀번호, 전자우편 주소로 구성되며 사용자 정보 메뉴에 의해 처리된다. 교수-학습 설계는 학년, 교과, 키워드의 데이터를 갖는다. 인터페이스 모듈을 통해 입력된 학습 내용은 코드로 바뀌게 된다. 학습 내용이 코드화되는 과정은 <표2>에 의해 이루어지며 학습 내용 코드는 문자로 이루어져 있다. 이러한 기능에 대한 인터페이스는 사용자정보, 교수설계, 학습하기, 게시판, 도움말의 메뉴를 갖는 화면을 설계한다. 각의 메뉴를 통해 교수자와 학습자는 원활한 웹 기반 교수-학습을 한다. 인터페이스의 화면 설계는 <그림4>와 같다.



<그림4> 인터페이스 화면 설계

### 3.3 교수-학습 생성기

교수-학습 생성기는 교수자의 교수 계획에 의해 학습자가 학습을 하게 한다. 교수-학습 생성기는 교수자와 학습자 모드로 구성된다. 교육 전문가인 교수자는 학습자에게 전달하고자 하는 학습 내용과 관련이 있는 웹 문서를 웹 에이전트로부터 제공받는다. 학습할 웹 문서는 초등학교 교육과정에 적합한 것으로 한정한다. 교수자는 웹 문서를 바탕으로 학습자에게 학습을 할 수 있는 환경을 구성한다. 학습 단위, 학습 주제, 학습 목표를 기술한 후 학습자가 학습할 순서를 정한다. 학습자가 학습 순서에 따라 학습을 완료하면 학습 결과를 평가하기 위한 평가 문제를 작성한다. 평가 문제는 교수자가 출제한다. 교수자가 새롭게 작성한 학습 평가 문제는 WAILS DB에 저장되며 다른 교수자가 사용한다.

### 3.4 WAILS DB 설계

본 데이터베이스는 관계형 데이터 모델로서 키워드, URL, 학습단위, 교수자 테이블로 구성한다. 각 테이블간의 E-R 다이어그램은 <그림5>와 같다.



<그림5> WAILS 데이터베이스 E-R 다이어그램  
E-R 다이어그램에서 생성한 주요 테이블은 다음과 같이 구성한다. 키워드 테이블은 웹 에이전트의 메타검색과 교수자의 검색을 위한 키워드 정보를 저장하며 <표3>과 같이 테이블을 구성한다.

<표3> 키워드 테이블

속성	컬럼명	Key	자료형
키워드ID	keyID	Primary Key	문자(8)
이름	keyNAME		문자열(20)
검색 체크	keyCHECK		숫자(1)

URL 테이블은 웹 에이전트의 검색 결과인 URL 목록을 저장하며 <표4>와 같이 테이블을 구성한다.

<표4> URL 테이블

속성	컬럼명	Key	자료형
URL ID	urlID	Primary Key	숫자(8)
URL 주소	urlADD		문자열(20)
URL 설명	urlDESC		문자열(300)
키워드 ID	keyID		문자(8)
가중치	urlVALUE		숫자(3)
선택횟수	urlSELECT		숫자(8)
단원코드	chapCODE		문자(6)

학습 단위 테이블은 키워드와 교과목 단위과의 관계를 저장하며 <표5>와 같이 테이블을 구성한다.

<표5> 학습 단위 테이블

속성	컬럼명	Key	자료형
과목	SUBJECT		문자(2)
학년	GRADE		문자(1)
학기	TERM		문자(1)
키워드 ID	keyID		숫자(8)
단원코드	chapCODE		문자(6)

사용자 테이블은 사용자의 정보를 저장하며 <표

6>과 같이 테이블을 구성한다.

<표6> 사용자 테이블

속성	컬럼명	Key	자료형
이름	usrName		문자열(10)
사용자 ID	usrID	Primary Key	문자열(8)
비밀번호	usrPASSWD		문자열(8)
선택 URL	usrURL		문자열(10)

#### 4. WAILS 시스템 구현

WAILS 시스템은 WINDOW NT 4.0 서버를 운영체제로 하고 80M 메모리를 갖는 Pentium MMX 233을 사용하여 구현하였다. WAILS의 웹 에이전트 프로세스는 JDK 1.1.8을 사용하여 개발하였다. 인터페이스와 교수-학습 프로세스는 사용자와의 상호작용에 중점을 두고 IIS(Internet Information Server)4.0의 ASP(Active Server Page)를 이용하였다.

##### 4.1 웹 에이전트의 검색

웹 에이전트는 초기 상태에서 교수자의 검색 요청을 처리할 수 없다. 웹 에이전트가 교수자의 검색을 처리하려면 최초의 WAILS DB를 생성할 필요가 있다. 최초의 WAILS DB는 웹 검색 모듈이 색인 파일의 키워드를 이용하여 메타 검색을 함으로써 생성된다. 최초의 WAILS DB가 생성된 후 교수자의 검색 요청을 키워드 검색과 디렉토리 검색으로 구분하여 **처리한다. 디렉토리** 검색은 WAILS DB의 웹 문서 항목별 분류를 참고하여 결과를 제시한다. WAILS에서 키워드 검색은 초등학교 교과서의 단원 및 학습 주제의 단어로 수행된다. 키워드 선정은 교육 전문가에 의해 이루어진다. 그러므로 WAILS의 키워드는 일반 검색엔진의 키워드 추출방법을 적용하지 않는다. 교수-학습에 필수적인 키워드만 선정함으로써 키워드와의 관련성이 높은 웹 문서를 수집한다. 이렇게 선정된 키워드는 메타 검색을 위해 각각의 검색 엔진에 맞는 길의 형태로 변환된다. 특히, 교육용 문서의 효용성을 높이기 위해 검색엔진에게 교육 항목에 대한 메타 검색을 의뢰한다. 교육 분야 메타 검색 결과 형성된 웹 문서를 분석하여 교육용 웹 문서로 선정될 가능성이 높은 웹 문서의 URL을 분석하여 추출한다. 추출한 URL은 WAILS DB에 저장된다. 웹 에이전트의 웹 문서 인출 알고리즘은 <그림6>과 같

다.

```

public class Crawler extends Thread {
    public Crawler() {
        indexer.start(); //웹 문서 분석 시작
    }
    public boolean URLAllowed(URL url){ }
    public void addURL(URL url_to_queue) { } //검색할 URL을 큐에 저장
    public void run() {
        if (!q.hasMoreElements()) return; //검색할 URL이 큐에 없으면 종료.
        while (true) {
            url_status = new URLStatus()
            //URL.Status Class를 생성하여 웹 문서 내용 인출
            url_status.readContent();
            if (url_status.loaded()) {
                indexer.queueURL(url_status);
            } else if (url_status.moved()) {
                addURL(url_status.actual_url);
            }
        }
        Monitor m = engprefs.getMonitor();
        if (m != null) m.crawlerDone(this);
        indexer.stopWhenDone(exit_when_done);
    }

    public static void main(File file, EnginePrefs prefs, boolean exit) {
        Crawler cr = new Crawler(); // 웹문서 수집 시작
        DataInputStream in = new DataInputStream(new FileInputStream(file));
        line = in.readLine(); // 검색 엔진의 URL 읽어오기
        try {
            while (line != null) {
                cr.addURL(new URL(line)); // 검색 엔진의 URL 저장
                line = in.readLine();
            }
            cr.start();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

<그림6> 웹 검색 모듈의 웹 문서 인출 알고리즘

<그림6>의 알고리즘을 설명하면 다음과 같다. 먼저, 메타 검색을 할 검색엔진의 URL 주소를 인출한다. 다음은 교육용 웹 문서 관련 질의어를 입력받고, 메타 검색을 위한 질의어를 검색엔진별로 생성하고, 질의어를 검색엔진에게 전송하여, 메타 검색 결과 웹 문서를 읽어온다. 그후에 웹 문서에서 해당 문서의 URL을 추출하여 추출된 URL을 WAILS DB에 저장한다.

웹 에이전트의 전체 작업 시간은  $T_w = T_f + T_d$ 의 식으로 측정한다.  $T_f$ 는 검색엔진 URL을 인출하여 큐에 저장하는 시간을 나타내며  $T_d$ 는 HTTP 요청에 의해 웹 문서를 읽어오는 시간을 나타낸다.  $T_f$ 는 메타 검색을 수행할 엔진의 수를  $N$ 이라 할 때  $N$ 번 수행하게 된다. 그리고  $T_d$ 는  $T_f$ 가  $N$ 번 수행된 상태에서 더 이상 검색을 수행할 URL이 없을 때까지 루프를 최대  $N$ 번 수행한다. 즉,  $T_d$ 는 최대  $N^2$ 번을 수행한다. 그러므로  $T_w$ 는 최대  $(N + N^2)$ 번을 수행한다. 이를  $O$ 의 표기법으로 나타내면  $O(N^2)$ 의 값

을 얻으므로 알고리즘의 효율성을 입증하여 준다.



<그림7> WAILS 웹 에이전트의 실행 화면

그리고 WAILS 시스템의 웹 에이전트는 시스템 관리자가 구동시키며 웹 검색에 필요한 키워드를 시스템 관리자가 입력하거나 색인 파일로부터 인출하여 메타검색을 한다. WAILS의 웹 에이전트가 키워드 입력을 받아 메타 검색을 수행하는 화면은 <그림7>과 같다. WAILS의 웹 에이전트는 현재 검색 중인 URL 주소와 처리 중인 URL 주소를 보여줌으로써 시각적인 웹 에이전트의 관리가 가능하다.

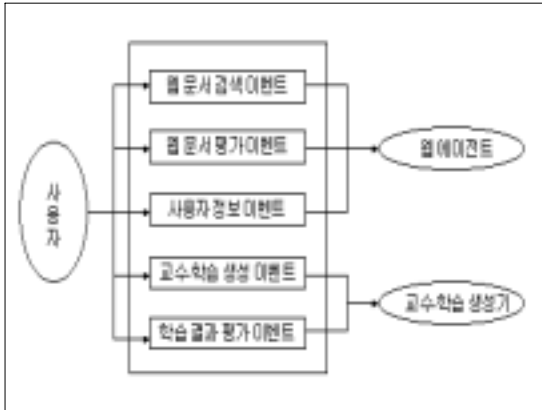
#### 4.2 인터페이스의 이벤트

교수자가 WAILS를 이용하여 최초의 교수-학습 설계를 할 경우 인터페이스를 통해 등록을 해야 한다. 교수자가 등록에 필요한 데이터를 입력하면 즉시 WAILS의 교수-학습 설계 도구를 이용하게 된다. 이미 등록을 한 교수자는 ID와 비밀번호를 입력할 것을 요청 받는다. 인터페이스는 교수자 확인 작업을 거친 후 교수-학습 설계 과정을 제시한다. 이런 일련의 과정은 인터페이스의 이벤트 처리를 통해 이루어진다. 인터페이스가 처리하는 이벤트는 웹 문서 검색 이벤트, 웹 문서 평가 이벤트, 사용자 정보 이벤트, 웹 문서 선택 이벤트, 교수-학습 생성 이벤트, 학습 결과 평가 이벤트로 이루어진다. <그림8>에 인터페이스가 이벤트를 처리하는 과정을 보인다.

교수자가 검색 버튼을 클릭하면 웹 문서 검색 이벤트가 발생하고 이는 웹 에이전트로 전달된다. 검색

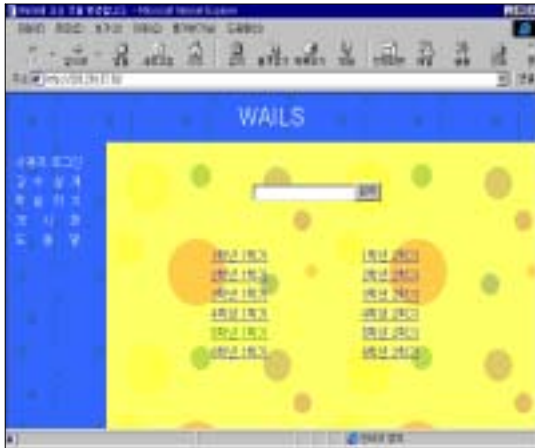


된 문서가 존재하면 교수자는 교수 설계를 하고 교수-학습 생성 이벤트가 발생한다. 교수-학습 생성 이벤트 결과에 따라 학습자는 학습을 하고 학습결과 평가 이벤트를 발생시킨다. 평가 결과에 따라 WAILS DB를 갱신한다.



<그림8> 인터페이스의 이벤트 처리 과정도

WAILS의 인터페이스 화면은 <그림9>와 같다. 교수자는 검색창에 키워드를 입력하거나 검색창 아래의 디렉토리에서 학년, 학기별로 원하는 웹 문서를 검색한다.



<그림9> WAILS의 인터페이스 화면

### 4.3 교수-학습 설계 화면

교수-학습 생성기는 교수자에게 교수 설계 환경을 제공한다. 교수자는 웹 에이전트에 의한 검색 결과가

제시되면 필요한 웹 문서의 URL을 선정한다. 검색 결과가 제시되는 화면은 <그림10>과 같다. 검색 결과와 필요한 웹 문서를 체크한 후 교수 설계 메뉴를 클릭 하여 교수 설계를 한다. 교수 설계는 초등학교의 일반적인 교수-학습 과정에 따라 이루어진다. 교수자는 단원, 학습목표, 학습내용을 기술해야 하고 학습에 필요한 웹 문서의 URL은 교수-학습 생성기에 의해 자동 생성된다. <그림11>에 교수-학습 설계 화면을 보인다.



<그림10> 검색 결과 화면



<그림11> 교수-학습 설계 화면

### 5. 결론 및 향후 연구과제

정보의 홍수를 이루는 인터넷에서 교수자가 검색한 웹 문서는 효용가치가 높음에도 같은 문서를 반복해서 찾는 경향이 있다. 또한 교수자가 웹 문서를 선정한 후 별도의 교수-학습 설계 과정을 거쳐야만 한다. 이 문제를 해결하기 위해 본 논문은 웹 에이전트를 이용한 개별 학습시스템인 WAILS를 개발하였다. WAILS는 WBI 학습에 필요한 자료를 웹 에이전트를 이용하여 제공해 주는 시스템으로 웹 에이전트, 인터페이스, 교수-학습 생성기로 구성되어 있다. 웹 에이전트는 시스템 관리자의 요청에 의해 메타검색을 하고 검색 결과를 WAILS DB에 저장한다. 인터페이스는 교수자로부터 정보를 입력받아 이벤트 처리를 거쳐 각각의 프로세스로 보낸다. 교수-학습 생성기는 교수자가 교수 계획을 세워 학습자가 학습을 하도록 한다.

WAILS는 인터넷의 정보를 학습 내용에 적합하게 제시하여 WBI 학습이 이루어지도록 한다. WAILS를 이용함으로써 교수는 검색엔진을 이용한 웹 문서 검색과 교수 설계를 동시에 한다. 교수가 선정한 웹 문서는 시간이 지남에 따라 학습 내용에 더욱 적합한 형태로 재평가를 받는다. 이는 웹을 활용한 교수-학습을 더욱 촉진시키는 역할을 하게 된다.

본 논문에서는 웹 에이전트를 이용한 WAILS를 개발하였다. 향후 연구과제는 교수-학습 생성기에 지능에이전트를 도입하여 학습자의 학습 결점을 찾아내고 이를 자동으로 교정해 주는 것이다.

### 참고문헌

- [1] 이진민 · 김갑수, “이동 에이전트를 이용한 개별 학습 시스템”, 한국정보교육학회 학술발표논문집 제4권 1호, 1998.
- [2] 이진민 · 김갑수, “교육용 웹 문서 수집 에이전트의 이동 알고리즘”, 한국정보교육학회 학술발표논문집 제4권 2호, 1998.
- [3] BjÖrn Hermans, “Intelligent Software Agents on the Internet”, <http://www.he-rmans.org/ag-ents>, 1999.
- [4] Michael Bleyer, “Multi-Agent Systems for Information Retrieval on the World Wide Web”, <http://www.uniulm.de>, 1999.
- [5] “The IBM Agent”, <http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm>, 1999.
- [6] Benjanin Falchuk and Ahmed Karmouch, “A Mobile Agent Prototype For Autonomous Multimedia Information Access, Interaction, and Retrieval”, <http://deneb.genie.uottawa.ca>, 1999.
- [7] Jeffrey Bradshaw, “Software Agents”, AAA Press/The MIT Press, 1998.
- [8] Tim Finin and Richard Fritzon, “KQML as an Agent Communication Language”, <http://cs.umbc.edu>, 1998.
- [9] Michael Wooldridge, “Intelligent Agent: Theory and Practice”, <http://www.elec.qmw.ac.uk/dai/people/mikew/pubs/ker95/ker95-html.html>, 2000.
- [10] Aglets Software Development Kit (ASDK), <http://www.trl.ibm.co.jp/aglets>, 1998.
- [11] “Odyssey Information”, <http://www.generalmagic.com/technology/technology.html>, 1998.
- [12] Introducing Concordia, <http://www.meitica.com/HSL/Projects/Concordia>, 1998.
- [13] ObjectSpace Voyager Technical Overview, <http://www.objectspace.com/developers/voyager/white/>, 1998.
- [14] “The Ara Platform for Mobile Agents”, [http://www.uni-kl.de/AG-Nehmer/Projekte/Ara/index\\_e.html](http://www.uni-kl.de/AG-Nehmer/Projekte/Ara/index_e.html), 1999.
- [15] “JAT Lite Introductory FAQ”, <http://cdr.stanford.edu/ProcessLink/papers/JATL.html>, 1999.
- [16] Danny B. Lange and Daniel T. Chang, “IBM Aglets Workbench Programming Mobile Agent in Java”, <http://www.agent.org/>, 1998.
- [17] Introduction to the Odyssey API, General Magic Inc, <http://www.genmagic.com/agents>, 1997.