

개념정보 추출을 위한 이동에이전트 시스템

공용해* · 최인석**

1. 서 론

인터넷의 급속한 발달로, 밀려드는 많은 정보들 속에서 우리가 필요로 하는 정보를 찾는 것은 중요한 일이다. 인터넷상에는 다양한 정보 자원들이 존재하고 있으며 최근에는 XML(eXtensible Markup Language)에 대한 관심이 날로 증가하고 있다. XML은 SGML (Standard Generalized Markup Language)에 기반을 둔 단순하고, 매우 유연성 있는 텍스트형식이다[1]. XML은 정보의 구조를 자유롭게 정의할 수 있기 때문에 문서의 표준화가 어렵다. 따라서 인터넷상에 존재하는 많은 비정형 XML 문서들은 동일한 정보에 대하여 서로 다른 구조와 속성을 지니고 있으므로 응용프로그램의 접근이 극히 제한적일 수 있으며, 정보의 표현이 상이하여 동일 정보의 검색에 어려움이 발생할 수 있다[2].

많은 사이트가 제공하는 비정형 XML 문서들이 동일한 정보에 대해서 동일한 구조와 약속된 개념 표현을 따를 경우 정보 검색은 매우 쉽고 정확해질 수 있지만, 현실적으로는 그렇지 못하다. 현재 W3C 등에서 XML 문서의 구조를 정의하는 DTD(Document Type Definition)의 표준화 작업

을 진행하고 있으나 표준화된 DTD가 보편화되기까지는 많은 시간과 노력이 필요하며, 수많은 정보에 대해 모든 DTD를 표준화하기는 사실상 어려운 일이다[3].

만약 특정 정보가 적절히 개념화되어 표현된다면, 이러한 개념 수준의 정보를 이용하여 비정형 XML 문서에 포괄적으로 적용할 수 있는 DTD를 생성함으로써 구조적 제약을 극복할 수 있고, 이러한 포괄적 DTD를 이용하여 비정형 XML 문서들에 대한 접근이 가능하게 된다. 또한 개념 수준의 정보는 사용자의 요구사항을 개념적 질의로 변환함으로써, 문서내의 포함된 개념수준의 정보를 추출해 낼 수 있게 한다[4].

그러나, XML을 기반으로 하는 정보 자원에서 필요한 정보를 얻기 위하여 모든 문서들을 호스트로 전송 받아 정보 추출작업을 실행한다면 호스트의 실행 시간과 네트워크의 부하가 증가할 것이다. 만일 정보문서가 존재하는 원격지에서 필요한 정보만을 전송하는 이동 에이전트를 활용한다면 호스트의 실행시간과 네트워크의 부하를 감소시킬 수 있을 것이다[5].

본 연구에서는 다양한 비정형 XML문서에서 개념수준의 정보 검색이 가능한 이동 에이전트 시스템을 개발하였다. 개념정보추출 이동 에이전트 시스템의 개발 과정은 다음과 같다. 먼저 원격지 사이트에 이동하기 위한 에이전트를 개발하고,

본 연구는 한국소프트웨어진흥원의 ITRC 사업에의해 수행된 것임.

*순천향대학교 정보기술공학부

**홍성기능대학 전자계산기과

원격지 사이트의 다양한 XML 문서에서 개념 정보를 추론하기 위하여 특정 정보에 대한 일상의 표현을 개념화한다[6]. 이러한 개념 수준의 정보를 이용하여 정보의 구조에 독립적으로 접근할 수 있는 포괄적 DTD를 자동으로 생성하는 DTD 생성기를 개발하고, 또한 사용자의 질의와 정보의 개념을 추론하여 표현이 다른 XML문서에서 개념수준의 정보의 검색이 가능하게 하는 추론 알고리즘을 연구한다. 그림 1은 개념정보 검색을 위한 이동에이전트 시스템의 구성도이다.

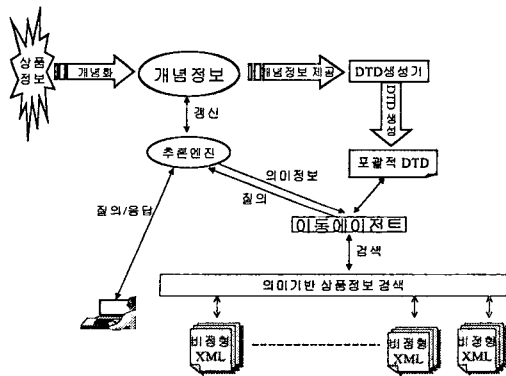


그림 1. 개념정보 검색을 위한 이동에이전트 시스템의 구성도

본 논문의 구성은 다음과 같다. 2장에서 이동에이전트를 살펴보고, 3장에서 정보의 개념화와 포괄적 DTD의 생성 과정을 보여준다. 4장에서는 개념의 추론 방법을 기술하고, 5장에서 개념정보 추출을 위한 이동에이전트를 구현한다. 6장에서 결론 및 기대효과를 알아본다.

2. 이동에이전트

이동 에이전트란 사용자를 대신하여 이기종 분산 환경의 네트워크에서 자율적으로 이동 및 반응하는 소프트웨어를 지칭하며 자율성, 지능성, 이동

성과 대행성, 비동기성 등의 특징을 가지고 있다.

자율성은 에이전트가 어떤 환경에 속하여 환경의 일부분으로써 주위 환경을 인지하고 자신의 목표에 따라 스스로 작동하여 결과를 갖는 시스템이다[7]. 대행성은 에이전트가 사람을 대신하여 어떤 일을 수행한다는 의미를 갖고 있다. 지능성은 에이전트가 지능을 가지고 활동한다는 의미이다. 에이전트의 지능성은 필수 조건이 아니며 오히려 임무수행 성능을 높이는데 수반되는 부대효과라 할 수 있다. 에이전트의 이동성은 코드와 데이터로 표현된 에이전트가 물리적으로 통신망을 타고 한 호스트에서 다른 호스트 컴퓨터로 이동하는 능력을 말한다. 비동기성은 파견된 이동에이전트가 원격지에서 수행되므로 호스트에 대해서 독립적으로 실행되는 특성이다[8].

이러한 특성들을 가진 이동에이전트가 실행하기 위해서는 생성, 복제, 파견, 회수, 소멸, 활성화, 비활성화등의 기본 연산이 필요하다. 생성은 컨텍스트라 불리는 서버의 주소와 이름으로 구성된 작업실에서 발생한다. 컨텍스트에서 자바의 이동에이전트가 생성되면서 식별자가 주어지고 초기화된 후 곧바로 실행하게 한다. 복제는 같은 컨텍스트 안에서 원래의 에이전트와 똑같은 에이전트를 복사한다. 그러나 이 복사된 에이전트는 자신의 식별자를 가지며 별도로 실행 시켜야 한다. 파견은 현재의 컨텍스트에서 해당 에이전트를 제거하고, 이동될 컨텍스트안에 해당 에이전트를 추가하여 실행하게 한다. 회수란 특정 에이전트가 현재의 컨텍스트에서 제거되고 해당 에이전트를 요청한 컨텍스트안에 추가시키며 파견에 비해 수동적이다. 비 활성화는 에이전트의 실행을 임시로 중지한 후 상태를 디스크에 저장한다. 활성화는 이러한 에이전트를 같은 컨텍스트안에서 다시 실행하게 하다. 소멸은 현재 실행중인 에이전트를

중지시키고 그 에이전트가 있는 컨텍스트로부터 에이전트를 제거한다.

이와 같이 이동 에이전트는 실행 프로그램이 실제로 데이터가 존재하고 있는 장소로 이동하여 사람을 대신하여 목적을 수행한다. 이동에이전트는 임무를 수행 완료 후에는 일반적으로 소멸하거나 홈 사이트로 돌아오게 되므로 효율적인 네트워크 활용이 가능하다[9]. 그림 2는 이동에이전트의 실행에 필요한 기본 연산과 실행 과정을 보여준다 [10].

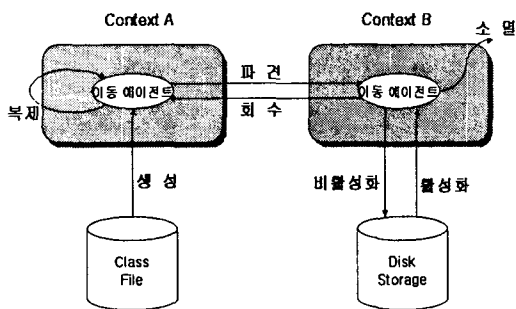


그림 2. 이동 에이전트의 실행 과정

3. 개념 수준의 정보 추출

개념 수준의 정보는 일상의 정보 표현을 개념화하여 사용자와 응용 시스템들 사이에 전달될 수 있는 정보 영역의 공유 개념과 공통된 지식을 제공할 수 있게 하며, 웹 응용프로그램들 사이의 지식을 공유하고 일관된 처리를 가능하게 한다. 이러한 개념수준의 정보는 지식의 공유와 재사용을 촉진하기 위하여 인공 지능 분야에서 사용되고 있다[11].

개념수준의 정보를 추출하기 위해서는 비정형 XML문서의 구조에 독립적으로 접근이 가능해야 하며, 이를 위하여 개념정보를 내포한 포괄적 DTD의 생성이 필요하다. 접근된 비정형 XML문

서에서 개념 수준의 정보를 추출하기 위해서는 정보의 개념화가 필요하며, 정보를 개념화하기 위해서는 계층적 구조화와 속성화가 필요하다[12].

3.1 개념의 계층적 구조화

계층적 구조화는 정보를 표현하기 위한 일상적인 개념의 포함관계를 계층적 구조로 표현한다. 그림 3은 음악 CD를 표현하는 계층적 구조의 예이다. 계층적 구조의 하위개념은 상위개념의 모든 속성을 상속받는다.

개념 'Object'는 하위 개념으로 'Person', 'Company', 'Record', 'Song'을 가지며 하위개념은 상위개념 'Object'의 모든 속성을 상속받는다. 또한 개념 'Singer'는 모든 상위개념 'Object'의 속성을 상속받은 'Person'의 속성을 모두 상속받는다. 결과적으로 개념 'Record'와 'Person'의 모든 속성을 상속받는 결과가 된다. 정의된 개념적 계층구조는 DTD생성기에 의해서 적절한 포괄적 DTD를 표현하는데 사용된다.

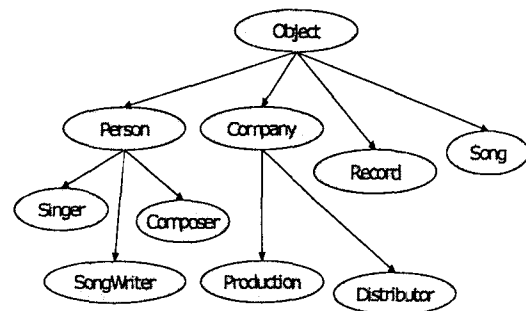


그림 3. 상품정보의 개념적 계층구조의 예

3.2 개념의 속성화

개념의 속성화는 계층적 구조로 표현된 각 개념들이 가질 수 있는 속성과 상호관계를 표현한다. 그림 4의 예는 정의된 개념들 사이의 속성과

상호 관계를 정의하였다. 개념 'Record'는 'trackList'를 비롯한 여러 개의 에트리뷰트(attribute)와 그에 대한 값으로 정의되어 있다. 'trackList'는 'Record'의 에트리뷰트로서 개념 'Song'을 값으로 갖는다. 개념 'Record'의 에트리뷰트 'production'과 개념 'Production'의 에트리뷰트 'product'는 상호 관계를 규정하여 같은 정보로 해석된다. 정의된 속성화는 DTD에서 표현되는 에트리뷰트들의 속성 값과 추론에 필요한 규칙을 정의할 수 있게 한다.

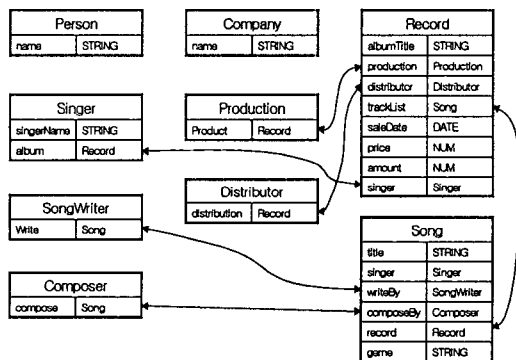


그림 4. 정의된 개념의 속성과 상호관계

3.3 개념화된 정보로부터 포괄적 DTD 생성

인터넷에 넓게 분포되어 정보를 표현하는 비정형 XML문서는 구조가 매우 다양하여 응용프로그램의 접근이 극히 제한적일 수 있다. 그러나 특정 정보를 개념화한 개념수준의 정보로부터 유추된 DTD는 특정 정보를 표현하는 XML 문서들의 구조에 독립적으로 접근한다.

DTD생성기는 개념 수준의 정보를 토대로 만들어졌다. 따라서 DTD 생성기는 개념수준의 정보에서 표현된 개념들이 적절히 포함된 포괄적 DTD를 생성할 수 있다. 이를 위해서는 개념수준의 정보에서 표현되는 개념들간의 구조를 DTD의 ENTITY로 표현하는 과정이 필요하며, 개념

들간의 상속을 적용할 수 있게 한다. DTD 생성기는 개념의 계층적 구조를 DTD의 엘리먼트 타입으로 매핑하며, 개념의 에트리뷰트는 부 엘리먼트와 엘리먼트들의 XML 속성으로 정의한다. 만약 서로 다른 개념의 에트리뷰트가 엘리먼트와 관계가 있다면 다른 개념과 관계를 엘리먼트로 표현하고 그렇지 않을 경우에는 PCADATA로 나타낸다. 그림 5는 DTD의 엔티티로 표현된 개념구조이다.

```
<!ENTITY % Object "Object | Song | Record | Company |Distributor | Production | Person | Composer |SongWriter | Singer">
```

그림 5. DTD ENTITY로 표현된 개념구조

개념의 계층적 구조의 표현이 완료된 후 완료된 개념을 상위개념으로 갖는 클래스들을 순회하면서 엔티티와 하위개념을 찾아 정의한다. 정의된 개념의 속성과 상호관계는 상위 개념의 속성을 상속받아 DTD의 엘리먼트로 표현한다. 그림 6에서는 DTD ELEMENT로 표현된 개념 속성의 일부를 보여준다.

```
<! Singer (#PCDATA | name | singerName | album)*>
```

그림 6. DTD ELEMENT로 표현된 개념의 속성

엔티티(ENTITY)에 실제 할당할 수 있는 값의 속성을 정의하는 ATTLIST는 상위 개념의 속성들을 그대로 상속받게 된다. 그림 7은 상위개념으로부터 속성을 상속받은 ATTLIST를 보여준다.

```
<!ATTLIST Singer
    name CDATA #IMPLIED
    singerName CDATA #IMPLIED
    album CDATA #IMPLIED>
```

그림 7. 상위개념의 속성을 상속받은 ATTLIST

다음은 정보의 속성 타입이 STRING이나 NUM 인 것은 #PCDATA로 표현하고, 다른 개념 참조 하는 것은 #PCDATA 이후에 참조하는 개념 목록 을 참조하여 DTD 엘리먼트를 생성한다. 그림 8 은 생성된 DTD 엘리먼트의 일부이다.

```
<!ELEMENT name (#PCDATA)>
<!ELEMENT singerName (#PCDATA)>
```

그림 8. 생성된 DTD ELEMENT

그림 9는 DTD생성기에 의해서 자동으로 생성 된 포괄적 DTD의 일부이다.

```
<!-- entities for realizing the is-a hierarchy -->
<!ENTITY % Object "Object | Song | Record |
Company | Distributor | Production | Person |
Composer | SongWriter | Singer">
:
:
<!-- element declarations for ontology concepts -->
<!ELEMENT Object (#PCDATA)* >
<!ELEMENT Person (#PCDATA | name)* >
<!ELEMENT Singer (#PCDATA | name |
singerName | album)* >
:
:
<!-- ATTLIST declarations for ontology attributes -->
<!ATTLIST Object >
<!ATTLIST Person
name CDATA #IMPLIED >
<!ATTLIST Singer
name CDATA #IMPLIED
singerName CDATA #IMPLIED
album CDATA #IMPLIED >
:
:
<!-- element declarations for ontology attributes -->
<!ELEMENT name (#PCDATA) >
<!ELEMENT singerName (#PCDATA) >
<!ELEMENT album (#PCDATA | %Record;)* >
:
:
```

그림 9. DTD생성기에 의한 포괄적 DTD

DTD 생성기에 의해 생성된 포괄적 DTD는 개 념수준의 정보를 반영하였으므로 XML 문서에 구조적으로나 개념적으로 유효하다. 따라서 포괄 적 DTD는 다양한 XML 문서에 개념적으로 접근 이 가능하며 동일 정보를 표현하는 다른 구조의 XML 문서에 모두 적용될 수 있다.

```
<?xml version="1.0"?>
<!DOCTYPE Record SYSTEM "MusicCD.dtd">
<Record singer="Bee Gees">
<albumTitle>The Very Best Of</albumTitle>
<distributor>Universal Music</distributor>
<saleDate>1990</saleDate>
<trackList>
<Song title="Don't Forget To Remember">
<writeBy>B.Gibb</writeBy>
<composeBy>M.Gibb</composeBy>
</Song>
</trackList>
</Record>
```

그림 10. "MusicCD.dtd"를 적용한 상품문서 예1

```
<?xml version="1.0"?>
<!DOCTYPE Singer SYSTEM "MusicCD.dtd">
<Singer>
<singerName>Bee Gees</singerName>
<album>
<Record>
<albumTitle>The Very Best Of</albumTitle>
<distributor>Universal Music</distributor>
<saleDate>1990</saleDate>
<trackList>
<Song>
<title>Don't Forget To Remember</title>
<writeBy>B.Gibb</writeBy>
<composeBy>M.Gibb</composeBy>
</Song>
</Record>
</album>
</Singer>
```

그림 11. "MusicCD.dtd"를 적용한 상품문서 예2

그림 10과 그림 11은 DTD생성기에 의해서 생성된 "MusicCD.dtd"를 서로 다른 XML문서에 적용시키는 예를 보여준다. 두 예문은 동일한 정보를 표현하고 있으나 문서의 구조와 정의된 엘리먼트가 서로 다름을 알 수 있다. 그러나 통합 DTD는 두 문서에 아무런 제약 없이 적용이 가능함을 보여주고 있다.

4. 개념의 추론

인터넷에 존재하는 비정형 XML문서를 대상으로 응용프로그램을 이용하여 동일한 개념의 정보를 검색하기란 어려운 일이다. 그러나 개념 수준의 정보에서 정의된 에트리뷰트의 관계를 이용하여 정의한 규칙에 의하여 새로운 정보를 추론할 수 있다[13].

예를 들면 " '김갑동'이라는 사람이 '밀레니엄'이라는 타이틀로 음악 CD롬 타이틀을 발표했다."라고 가정을 한다. 이 가정에서 음악 CD롬 타이틀을 발표한 사람 또는 '김갑동'이가 발표한 음악 CD롬 타이틀을 찾는 것은 어려운 일이 아니다. 그러나, 위의 정의에서 '김갑동'이가 '가수'라는 직접적인 정보가 없기 때문에 '가수'로서의 '김갑동'이는 찾지 못한다. 우리는 일상적인 개념으로 음악 CD롬 타이틀을 발표했다는 사실에서, '김갑동'이라는 사람이 '가수'라고 추측할 수 있다. 즉, "음악 CD롬 타이틀을 발표한사람은 당연히 가수다"라는 일상적인 개념이 정의되어있고, 이를 이용한다면 '김갑동'이는 가수로서 충분히 검색될 수 있으며, 새로운 개념으로 정의된다[14].

이처럼 새로운 개념을 정의하고 추론하여 정보를 만들어내는 것은 추론 엔진이라는 클래스에서 담당하며, 추론엔진을 통하여 표현의 이질적인 문제를 극복할 수 있다.

5. 개념정보 추출 이동에이전트 시스템의 구현

본 연구는 인터넷상에 넓게 분포되어있는 많은 비정형 XML문서에서 개념 수준의 정보를 원활히 검색하기 위한 이동에이전트를 개발하였다.

개념정보 추출 이동에이전트를 위하여 개발한 DTD 생성기는 특정 개념에 대한 모든 XML 문서에 적용할 수 있는 포괄적 DTD를 생성하여 구조적인 제약을 극복하였으며, 개념 수준의 정보로부터 생성된 추론엔진은 문서에 내포된 개념을 추론하여 새로운 개념을 정의하고 개념 수준의 정보 검색을 지원한다.

개념 수준의 정보를 검색하기 위하여 원격지에 존재하는 모든 비정형 XML문서를 호스트로 가져와서 필요한 작업을 수행한다면 호스트의 실행 시간과 네트워크 부하가 가중되므로 이동에이전트를 이용하였다. 그림 12는 개념정보추출 이동에이전트 시스템을 도식화하였다.

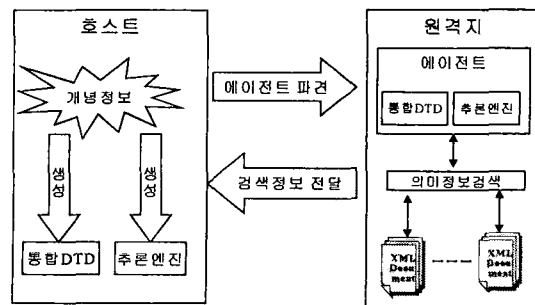


그림 12. 개념정보 추출 이동 에이전트 시스템

이동에이전트는 포괄적 DTD와 추론엔진을 원격지 사이트에 이동시킨다. 원격지 사이트에 도착한 이동에이전트는 포괄적 DTD를 이용하여 비정형 XML문서에 구조적 제약을 극복하고 접근하고, 추론 엔진을 통한 개념정보를 검색한다. 이

동에이전트는 검색된 개념 수준의 정보를 원래의 호스트로 전송한다. 개념정보 추출을 위한 이동에이전트는 모든 정보 추출 활동이 완료되면 호스트로 귀환 또는 소멸한다.

6. 결론

최근 인터넷에는 XML 문서를 기반으로 하는 많은 정보 자원들이 제공되고 있다. 다양한 XML 기반의 정보 자원들을 대상으로 개념 수준의 정보를 검색하기 위한 이동에이전트를 개발하였다.

이동에이전트는 이기종 분산 환경의 네트워크에서 자율적으로 이동 및 반응하면서 사용자의 요구에 의하여 원격 사이트에서 필요한 개념 수준의 정보만 추출하여 전송하므로 최소한의 호스트의 실행시간만 필요하며, 네트워크의 부하를 줄일 수 있었다.

그러나, 정보 추출의 대상이 되는 비정형 XML 문서는 구조와 정보의 표현이 매우 다양하여 응용 프로그램의 접근에 제약을 받을 수 있으며 동일 정보로 취급받지 못할 수 있다. 본 연구에서는 동일 정보를 추출하기 위하여 일상의 정보를 개념화하였으며, 포괄적 DTD를 자동으로 생성해주는 DTD생성기와 개념수준의 정보를 추론할 수 있는 추론엔진을 연구하였다. DTD 생성기는 개념 수준의 정보를 내포한 포괄적 DTD를 생성하여 비정형 XML문서에 접근할 때 발생하는 구조적인 제약을 극복할 수 있었으며, 추론 엔진은 이미 정의된 개념수준의 정보로부터 새로운 개념 수준의 정보를 추론하고 정의하여 보다 다양하고 신뢰성 있는 정보를 검색할 수 있었다.

그러나, 특정 정보로 제한된 개념화는 많은 정보를 표현하지 못했다. 따라서, 향후 보다 포괄적인 정보의 개념화에 대한 연구가 필요하다.

참고 문헌

- [1] Richard Anderson외, Professional XML, Wrox Press, 2000.
- [2] Kristin Stock, David Pullar, "Identifying Semantically Similar Elements in Heterogeneous Spatial Databases Using Predicate Logic Expressions", Lecture Notes in Computer Science, No. 1580, pp. 231-252, 1999.
- [3] W3C, Extensible Markup Language(XML) 1.0 Specification, <http://www.w3.org/TR/REC-xml>, 1998.
- [4] <http://aicore.chungbuk.ac.kr/~dobest/document/Agents/AmEC/amec.html>.
- [5] 석황희, 김인철 "이동 에이전트의 개념과 응용", 한국멀티미디어학회, 제3권, 제2호, pp. 29-39, 1999.
- [6] M. Erdmann and R. Studer, "How to Structure and Access XML Document with Ontologies," Data & Knowledge Engineering, Vol. 36, No.3, pp. 317-335, 2001.
- [7] S. Frabklyn and A. Graesser, "Is it an Agent, or just a Program": A Taxonomy for Autonomous Agents," Proc. of the Third Int'l Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.
- [8] 박지은, 김상욱, "이동 에이전트를 지원하는 프로그래밍 언어", 정보처리학회지, Vol. 4, No. 5, 1997.
- [9] 김평중, 윤석환, "이동 컴퓨팅을 위한 이동 에이전트 시스템", 정보처리학회지, Vol. 4, No. 5, pp. 67-75, 1997.
- [10] 권은정, 김백선, 용환승, "자바 기반의 이동 에이전트를 이용한 분산 XML 문서 검색 시스템 설계 및 구현", 한국멀티미디어학회 추계학술발표논문집, pp. 740-745, 1999.
- [11] A. Gomez-Perez and R. Benjamins "Applications of Ontologies and Problem Solving Methods," AI Magazine, Vol. 20, No. 1, pp. 199-222, 1999.
- [12] Sokratis Karkalas, Nigel Martin, "Automatic Semantic Object Discovery and Mapping from

Non-normalised Relation Databases Systems, ADVIS2000, pp. 92-107, 2000.

[13] <http://www.aifb.uni-karlsruhe.de/WBS/broker>.

[14] Patrick Henry Winston, Artificial Intelligence, Addison Wesley, U.S.A, 1993.



공 용 해

- 1982년 연세대학교 전자공학과 학사
- 1986년 Polytechnic University 전산학 석사
- 1991년 Polytechnic University 전산학 박사
- 1982년 한진중공업 연구원
- 1983년 삼성전자 연구원
- 1991년 ~ 현재 순천향대학교 정보기술공학부 교수
- 관심분야: 지능에이전트, 신경회로망, 컴퓨터비전 등



최 인 석

- 1988년 서울산업대학교 전산학과 학사
- 1996년 서울산업대학교 전산학 석사
- 2000년 순천향대학교 전산학 박사수료
- 1992년 한국산업인력공단
- 1995년 ~ 현재 홍성기능대학 전자계산기과 부교수
- 관심분야: 이동에이전트, XML, 인터넷 응용