

# 시멘틱 웹 환경에서의 개인화 검색

김 제 민<sup>+</sup> · 박 영 택<sup>\*\*</sup>

## 요 약

웹에 분산된 모든 웹 페이지는 구조가 서로 다르다. 시멘틱 웹 환경은 이형적인 구조를 갖는 웹 페이지들의 메타데이터 바탕으로 시멘틱 검색이 가능하다. 그러나 일반적으로 사용자의 요구에 따른 시멘틱 검색은 상황에 따라 엄청난 수의 검색 결과를 내놓는다. 따라서 검색 결과에 대해 각 사용자에 맞는 검색 결과 순위를 적용할 필요가 있다. Culture Finder는 시멘틱 웹 검색 에이전트들이 개인화된 문화 정보를 검색할 수 있도록 도움을 준다. Culture Finder는 웹에 존재하는 각 웹 페이지에 대한 메타 데이터를 작성하고, 시멘틱 검색을 이행하며, 사용자 프로파일을 기반으로 삼아 검색 결과에 대한 순위 점수를 계산한다. Culture Finder에는 개인화된 시멘틱 검색을 효율적으로 실행하기 위해 중요한 5가지 기법이 적용되었다. 사용자의 검색 행위로부터 사용자 프로파일을 생성하기 위한 기계 학습기법, 시멘틱 웹 검색 에이전트를 위한 효율적인 시멘틱 검색 기법, 사용자 질의의 효과적인 파악을 위한 질의 분석 기법, 각 사용자에게 적합한 검색 결과를 제공하기 위한 순위 적용 기술, 메타데이터를 생성하기 위한 상위 온톨로지 표현 방법. 본 논문에서는 Culture Finder의 구조를 통해서 시멘틱 개인화 검색에 대한 기법을 제안한다.

**키워드 :** 시멘틱 웹, 온톨로지, 메타데이터, 개인화 정보 검색

## Personalized Search Service in Semantic Web

Je-Min Kim<sup>+</sup> · Young-Tack Park<sup>\*\*</sup>

## ABSTRACT

The semantic web environment promise semantic search of heterogeneous data from distributed web page. Semantic search would result in an overwhelming number of results for users is increased, therefore elevating the need for appropriate personalized ranking schemes. Culture Finder helps semantic web agents obtain personalized culture information. It extracts meta data for each web page(culture news, culture performance, culture exhibition), perform semantic search and compute result ranking point to base user profile. In order to work efficient, Culture Finder uses five major technique: Machine learning technique for generating user profile from user search behavior and meta data repository, an efficient semantic search system for semantic web agent, query analysis for representing query and query result, personalized ranking method to provide suitable search result to user, upper ontology for generating meta data. In this paper, we also present the structure used in the Culture Finder to support personalized search service.

**Key Words :** Semantic Web, Ontology, Meta-data, Personalized Information Search

### 1. 서 론

현재 시멘틱 웹은 HTML 문서 기반의 웹과 더불어 중요한 웹 환경 패러다임으로 부각되고 있다[1]. 시멘틱 웹은 에이전트, 응용 프로그램, 기업 서버, 웹 커뮤니티들 간의 데이터 공유와 재사용이 가능한 프레임워크를 제공한다[2]. 일반적으로 웹에 분산된 모든 웹 페이지는 구조가 서로 다르다. 시멘틱 웹은 이형적인 구조에 속하는 데이터들을 바탕으로 의미가 통합된 검색을 가능하게 한다. 시멘틱 검색은 시멘

틱 웹 환경에 맞게 작성된 웹 페이지의 메타 데이터를 바탕으로 실행되며, 메타데이터를 이용하여 웹 페이지의 숨겨진 의미를 추론하여 보다 향상된 검색이 가능하다. 개인화가 적용된 시멘틱 검색은 각 웹 페이지의 메타데이터와 사용자의 성향이 기록된 사용자 프로파일을 사용하여 각 사용자에 맞는 검색 결과를 제공한다.

메타 데이터는 데이터에 관한 구조화된 데이터로, 여기서는 웹 페이지의 특징과 의미를 설명해주는 데이터를 의미한다. 시멘틱 웹에서 중추적인 역할을 하는 온톨로지는 각 웹 페이지에 대한 메타데이터를 구축하기 위한 구조를 제공한다. 사용자 프로파일 구축에 있어서 중요한 것은 프로파일을 구성하는 속성 필드의 선택이다. 보통 사용자 프로파일 구축을 위한 속성 필드 선택은 특징 추출(Feature Selection) 단계가

\* 본 논문은 숭실대학교의 지원을 받았습니다.

<sup>+</sup> 준 회 원 : 숭실대학교 대학원 컴퓨터학과 박사 과정

<sup>\*\*</sup> 정 회 원 : 숭실대학교 컴퓨터학과 교수

논문접수 : 2005년 12월 7일, 심사완료 : 2006년 9월 13일

필요하다. 그러나 온톨로지는 사용자 프로파일 구축에 필요한 속성들을 Property형태로써 명확하게 기술하기 때문에, 온톨로지를 기반으로 검색에 필요한 메타데이터를 관리하는 시스템에서는 사용자 프로파일 학습에 필요한 속성 및 속성 값을 정확하게 제공 받을 수 있다.

본 논문에서는 문화 정보에 관련된 다양한 웹 페이지로부터 시멘틱 검색을 효과적으로 실행하는 Culture Finder의 구조를 통해서 시멘틱 개인화 검색에 대한 기법을 제안한다. Culture Finder는 개인화된 시멘틱 검색을 효율적으로 실행하기 위해 중요한 5가지 기법을 적용한다. 사용자의 검색 행위로부터 사용자 프로파일을 생성하기 위한 기계 학습 기법, 시멘틱 웹 검색 에이전트를 위한 효율적인 시멘틱 검색 시스템, 사용자 질의의 효과적인 파악을 위한 질의 분석, 각 사용자에게 적합한 검색 결과를 제공하기 위한 순위 적용 기술, 메타데이터를 생성하기 위한 상위 온톨로지 표현.

본 논문에서 제안하는 시멘틱 개인화 검색 기법은 검색 에이전트가 키워드나 구문으로 표현된 검색 질의를 시멘틱 검색 서버에 전송하고, 시멘틱 검색 서버는 메타 데이터를 기반으로 개인화가 적용된 검색 결과를 검색 에이전트에 전송한다. 효과적인 시멘틱 개인화 검색을 위해서 본 논문에서는 웹 온톨로지 기술 언어인 OWL을 기반으로 효율적으로 메타데이터를 구축하기 위한 방안과 사용자 프로파일 구축 및 검색 결과에 대한 순위 계산 방안을 고려했다.

본 논문의 주 내용은 Culture Finder 구조 및 제안된 기술이다. 따라서 본 논문은 Culture Finder의 구조, 효율적인 시멘틱 개인화 검색을 위한 메타데이터 구축, 시멘틱 검색 기법, 메타데이터 기반의 사용자 프로파일 구축을 위한 방법, 각 사용자의 프로파일을 기반으로 개인에 맞게 검색 결과에 순위를 적용하는 방법에 대해 설명한다.

## 2. 관련 연구

이 장에서는 본 논문에서 제안한 시멘틱 개인화 검색 시스템과 직·간접적으로 연관된 관련 연구에 대해서 설명한다. RSS[3]과 Atom[4]은 다목적으로 사용하기 위해 적은 용량의 확장된 메타데이터 명세 형식이다. RSS·Atom 형식은 RDF 명세 기준을 따르고 있으며, XML 기반의 응용시스템에 적용된다. 이러한 응용시스템은 웹 사이트의 기본적인 내용을 간결하게 RSS·Atom 형식으로 요약하여 온라인으로 등록하고, 등록된 정보들의 검색 및 전달 또는 회수한다. 이 외에 시멘틱 웹과 시멘틱 웹 서비스 기술을 통해서 향상된 정보처리를 위한 새로운 가능성과 목표들이 제공되고 있으며 시멘틱 웹 환경에서 온톨로지 기반의 시멘틱 검색 시스템에 관한 연구들이 진행 중에 있다. Semantic search[5]는 OWL로 작성된 메타데이터 페이지에 존재하는 각각의 객체(Object)들을 검색한다. 따라서 검색 결과는 지식 베이스에 존재하고 있는 객체의 리스트다. Semantic search의 기본 아이디어는 시멘틱 웹 상에 존재하는 다양한 페이지들로부터 연관된 데이터만을 획득하는 검색 방식을 채택함

으로써, 향상된 메타데이터 검색을 실행하는 것이 목적이다. Swoogle[1]은 키워드를 사용하여 메타 데이터가 저장된 지식 베이스를 검색한다. Swoogle의 검색 결과는 메타데이터 페이지에 존재하는 각각의 객체뿐 아니라, 주제(subject) 및 속성(property)까지 포함한다. 이러한 종류의 검색은 일반적인 검색 기법을 시멘틱 웹 상에 적용한 것으로 볼 수 있다. 그 밖에 시멘틱 웹 기술을 개인 정보 검색 시스템으로 적용한 예로서 Haystack[6]과 Myportal[7]이 있다.

이러한 시스템들은 여러 가지 형태로 웹에 존재하는 자원을 이용하여 시멘틱 검색을 하는 응용시스템들을 위해 좋은 해결책이 된다. 그러나 아직까지 시멘틱 웹의 특징과 개인화 정보가 동시에 적용된 시멘틱 개인화 검색 서비스에 관한 관련연구들은 아직 진행 중에 있다. 따라서 본 논문에서 제안한 Culture Finder는 위의 설명된 여러 시스템들의 특징들을 기반으로 시멘틱 개인화 검색 서비스를 위한 여러 가지 방법들을 정의함으로써 설계되었다.

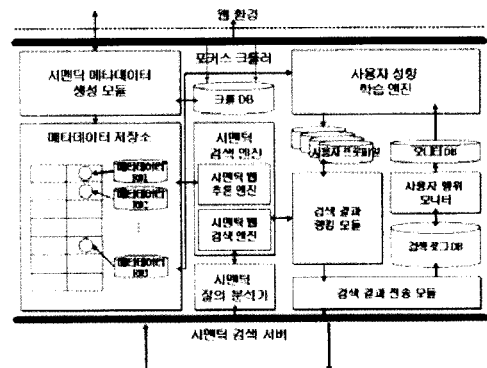
## 3. Culture Finder의 구조

### 3.1 구조

Culture Finder는 시멘틱 웹 검색 에이전트가 문화(문화 뉴스, 공연, 전시)에 관련된 다양한 정보를 검색하는데 도움을 준다. 따라서 시멘틱 웹 검색 에이전트는 문화 정보에 대한 시멘틱 검색을 실행하는 Culture Finder에 필요한 문화 정보를 요청하는 독립적인 클라이언트이며, Culture Finder는 문화 정보에 대한 검색 결과의 정보를 각각의 클라이언트에게 전송하는 서버 형태의 구조를 갖는다. Culture Finder는 각 문화 정보 웹 페이지에 대한 메타데이터를 기반으로 문화 정보 검색을 실행하며, 검색에 사용되는 질의어로는 RDQL (RDF Data Query Language)을 사용한다. (그림 1)은 Culture Finder의 전체적인 구조를 보여준다.

Culture Finder는 크게 일곱 파트로 구성된다.

- 문화 정보 웹 페이지의 메타 데이터를 구축하기 위한 시멘틱 메타데이터 생성 모듈
- 메타 데이터를 효율적으로 관리하기 위한 메타데이터 저장소



(그림 1) Culture Finder의 구조

- Rete 알고리즘을 기반으로 사용자의 질의와 메타데이터간의 정확한 매치를 통하여 사용자가 원하는 정보를 검색하는 시멘틱 검색 엔진
- 사용자가 입력한 질의어에서 검색 의미(속성)를 파악하는 시멘틱 질의 분석기
- 사용자의 검색 행위를 파악하기 위한 사용자 행위 모니터
- 사용자의 성향을 파악하여 사용자 프로파일에 기록하기 위한 사용자 기호 학습 엔진
- 각 개인에 맞는 정보를 우선적으로 제공하기 위해 각 정보의 추천 신뢰도를 계산하여, 이를 기반으로 검색된 결과의 순위를 결정하는 검색 결과 순위 모듈

Culture Finder는 온톨로지를 기반으로 구축된 시멘틱 메타데이터를 바탕으로 검색을 실행한다. 시멘틱 웹에서 중추적인 역할을 하는 온톨로지는 사람이 이해함은 물론이고 기계 역시 인식할 수 있는 형식으로 실 세계 영역들의 개념들을 계층적으로 표현한 것이다. 따라서 기계가 처리할 수 있도록 온톨로지는 실 세계 영역들의 개념을 개체(Entity), 속성(Attribute), 관계(Relationship) 및 개체간의 규칙(Axiom)을 통해 표현한다. 온톨로지는 실 세계 영역들에 존재하는 다양한 개체와 개체 관계를 표현하기 위한 풍부한 표현(Vocabulary)을 제공한다. 따라서 Culture Finder는 영화 분야, 애니메이션 분야, 예술 공연에 대해 미리 구축된 문화 온톨로지와 메타데이터를 사용하여 검색을 실행한다.

### 3.2 문화 관련 웹 페이지를 수집하기 위한 포커스 크롤러

시멘틱 검색에 바탕이 되는 메타 데이터를 작성하기 위해서는 대상이 되는 웹 페이지들을 먼저 수집해야 한다. 스파이더 또는 웹 로봇이라고 불리는 웹 크롤러는 특정 토크에 따라 작성된 웹 페이지를 웹으로부터 자동으로 수집해주는 프로그램이다. 웹 크롤러는 시드 페이지(Seed page)의 주소를 입력 받은 후 해당 시드 페이지의 수집 및 다른 페이지들과 연결 링크를 모두 추출하고, 연결된 모든 페이지를 수집한다[8]. 이러한 과정은 연결 링크가 특정 범위를 넘어서지 않을 때까지 반복된다.

본 논문에서는 메타 데이터 구축 대상이 되는 문화 관련 웹 페이지를 효과적으로 수집하기 위해서, 몇 가지 휴리스틱한 방법(분류 키워드와 시드 페이지 URL 입력)을 적용하여 웹 페이지를 선별해서 찾는 포커스 크롤러 방식을 Culture Finder에 적용하였다. Culture Finder에 적용된 크롤러는 효과적인 웹 페이지 수집을 위해 다음과 같은 작업을 수행한다. 먼저 크롤링 되는 페이지의 URL 정보를 참고하여, 해당 사이트의 도메인을 규정하고 크롤러의 수행 범위를 결정한다. 예를 들어 “무비스트”의 영화 소식만을 한정하여 크롤링 할 경우, “무비스트” 사이트에 속하는 모든 페이지 URL의 앞부분은 “www.movist.com/”으로 시작되기 때문에, 수집할 페이지의 URL이 “www.movist.com/”으로 시작하면 그 페이지는 크롤링 범위에 속하게 된다. 두 번째로, 반복적인 크롤링 현상을 막기 위해서 해당 웹 사이트의 크롤링 로그 정보를 저장하여, 크롤링 수행 시 이를 기반으로

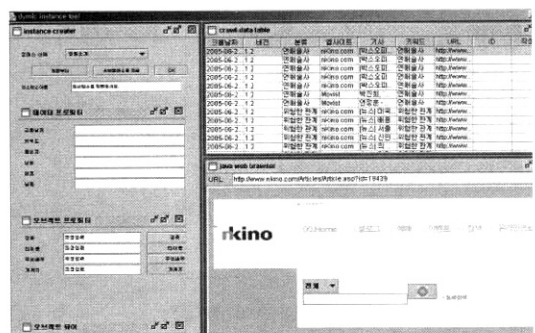
무한 반복 크롤링 문제를 해결한다. 마지막으로 크롤링을 수행 할 때 마다 해당 사이트의 모든 웹 페이지를 수집하는 방식을 탈피하여 수집 대상 페이지의 히스토리 정보를 저장함으로써, 새로 업데이트(Update) 되거나 추가된 페이지만을 수집하여 DB에 추가하는 방식을 사용한다.

### 3.3 메타 데이터 생성

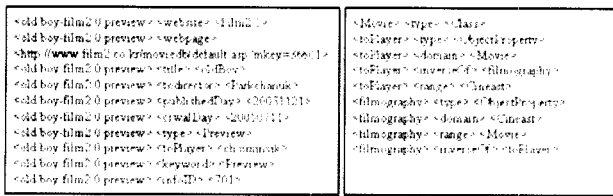
시멘틱 검색 기술은 웹 페이지를 작성한 저자가 제공하는 메타데이터의 수준에 따라서 큰 영향을 받는다. 현재 자동으로 완벽한 메타데이터를 구축하는 연구는 아직 초기 단계에 머물고 있다. 따라서 모든 메타데이터 구축작업은 온톨로지 저작 도구나 메타 데이터 저작도구를 통해 수작업으로 진행되고 있다.

메타데이터 저작 도구(Protege[9], Ontomat[10])는 온톨로지나 메타데이터를 작성하기 위한 일반적인 용어(Class, Property, Instance, and so on)를 간단한 아이콘 형태로 제공하기 때문에, 이러한 도구를 사용하면, 보다 간단하게 온톨로지 및 메타데이터를 만들거나 수정할 수 있다. 그러나 한글로 메타데이터를 구축 할 경우 기존의 저작 도구들은 다음과 같은 문제점으로 인해 많은 시간이 소요된다. 먼저 객체 속성(Object Property)의 연관성을 들 수 있다. 객체 속성은 미리 정의된 다른 객체를 속성 값을 갖는다. 이때 해당하는 객체를 선택하기 위해 해당 객체 아이콘을 찾게 되는데, 객체는 시간이 흐를수록 기하급수적으로 수가 늘어나기 때문에 해당아이콘을 찾는데 많은 어려움을 겪게 된다. 두 번째 문제점으로 메타데이터 구축자가 해당 페이지의 메타 데이터를 구축하기 위해서 직접 URL을 입력해서 해당 페이지를 웹 브라우저 로드하고, 웹 브라우저와 저작 도구를 서로 번갈아 팝업 시키면서 작업을 해야 하는 것인데, 이 역시 많은 시간을 소모한다.

본 연구에서 시멘틱 검색을 위한 메타데이터를 구축하기 위해 몇 가지 사항을 고려한 시멘틱 메타데이터 생성모듈을 Culture Finder에 적용하였다. 적용된 시멘틱 메타데이터 생성모듈은 온톨로지 파일을 읽어 들인 후 온톨로지를 구성하는 클래스의 특징에 맞는 속성들의 동적 변화, 키보드를 통한 객체 프로퍼티 값의 직접 입력, 시멘틱 검색 엔진이 편리하게 검색할 수 있는 데이터 포맷으로 메타데이터를 저장해 주는



(그림 2) 시멘틱 메타 데이터 생성모듈



(그림 3) 트리플로 표현된 메타 데이터

지에 대한 여부를 고려하여 구현되었다. (그림 2)는 Culture Finder에 적용된 시멘틱 메타데이터 생성모듈을 보여준다.

시멘틱 메타데이터 생성모듈의 오른쪽 상단에는 포커스 크롤러가 크롤링한 웹 페이지의 제목과 URL 및 부가정보가 표시되며, 메타데이터 구축자가 각각의 URL을 클릭하게 되면, 오른쪽 하단 브라우저에 해당 페이지가 브라우징 된다. 메타데이터 구축자는 브라우징된 페이지를 참조하면서, 온톨로지를 로드하고, 메타데이터를 구축한 후 메타데이터 저장소에 저장하게 된다.

보통 메타데이터는 OWL과 같은 시멘틱 웹 언어를 사용해서 문서형태로 작성되는데, 문서형태로 작성된 메타데이터는 효율적인 시멘틱 검색을 실행하기에는 부적절한 형식 갖는다. Culture Finder에 적용된 메타데이터 생성모듈은 트리플(RDF triple)형식으로 메타데이터를 생성한다. 트리플은 RDF 구문 기반의 표현 형식으로서 Subject, Predicate, Object로 구성된 세 개의 필드를 사용한다. Predicate 필드는 Subject 필드에 표현된 객체와 Object 필드에 표현된 객체 및 값과의 관계를 표현한다. 이러한 관계는 시멘틱 검색에서 검색 질의어와 검색 결과와의 연결 관계(의미)를 의미한다. 따라서 트리플은 효율적인 시멘틱 검색을 실행하기 위한 적절한 형태라고 할 수 있다. (그림 3)은 트리플로 표현된 메타데이터의 예이다.

3.4 시멘틱 검색 엔진과 질의 분석기

본 논문에서 제안하는 Culture Finder에 적용된 시멘틱 검색 엔진은 다음 두 가지 사항에 중점을 두고 설계되었다. 첫째 사항은 양질의 검색 결과를 내기 위해 메타데이터와 검색 질의어간의 효율적인 매치를 기반으로 하는 검색 방법이고 두 번째 사항은 검색 결과를 요청하기 위한 사용자 또는 검색 에이전트의 검색 질의어의 의미를 파악하는 것이다.

일반적으로 웹 검색 엔진은 키워드와 같이 단순한 질의어를 검색어로 인식하도록 설계된다. 이러한 방식은 각각의 검색 어플리케이션이 질의에 대한 결과를 검색하는데 비효율인 변을 보여준다. 왜냐하면 단순 키워드 매칭은 사용자가 의도하지 않는 검색 결과를 보여주기 때문에 엄청난 양의 정보를 검색한다. 이 경우 수많은 사람이 사용하는 검색 어플리케이션은 검색 엔진의 과 부하와 속도저하현상을 나타내게 된다. 따라서 검색어를 통해 사용자의 검색 의도를 파악하는 것이 중요하다.

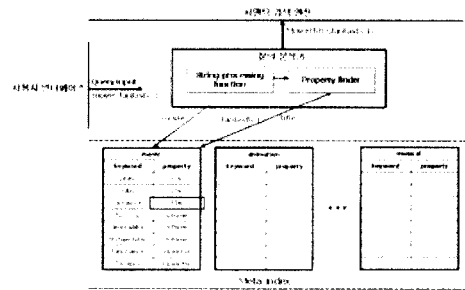
검색 질의어의 의미를 파악하기 위해서, 시멘틱 검색 엔진은 메타데이터 기반의 메타색인(Meta Index)을 사용하는 질의 분석기가 필요하다. 앞 장에서 서술했듯이 트리플로

표현된 메타데이터는 검색 질의어와 검색 결과와의 연결 관계(의미)를 나타내기 때문에, 메타데이터 참조에 의해 구축된 메타색인에는 검색 질의어와 질의어의 의미가 기록된다. 질의 분석기는 두 개의 변수(질의 변수, 의미변수)를 사용하여 메타 색인을 인덱싱 하는 과정을 통해서 검색 질의어의 의미를 파악한다. 즉, 질의 분석기는 질의 변수에 저장되는 질의어를 메타색인에 인덱싱 함으로써 질의어가 갖는 의미의 의미 변수를 통해 알아낼 수 있다. 예를 들어 “movie fantastic4” 라는 질의어가 사용자 또는 검색 에이전트로부터 들어오면 질의 분석기는 movie에 대한 메타색인을 참조함으로써 “movie title”이라는 검색 질의어의 의미를 알아낸다. (그림 4)는 질의 분석기가 검색 질의어의 의미를 파악하는 과정을 보여준다.

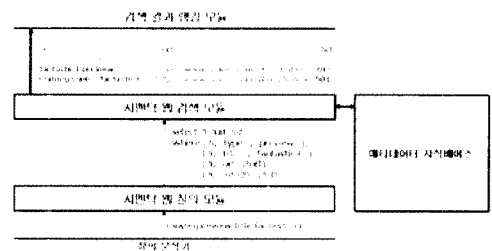
시멘틱 검색 엔진은 시멘틱 웹 관련 기술과 정보 관리 기술을 바탕으로 구현되었다. Culture Finder에 적용된 시멘틱 검색 엔진은 질의 분석기와 함께 메타데이터 지식 베이스(Metadata Knowledge Base), 시멘틱 웹 검색 모듈(Semantic Web Search Module), 시멘틱 웹 질의 모듈(Semantic Web Query Module)로 구성 된다.

메타데이터 지식 베이스는 시멘틱 검색에 필요한 메타데이터의 저장소다. 일정 영역의 온톨로지를 기반으로 구축된 메타데이터는 모든 시멘틱 검색 엔진이 공유하고 재사용할 수 있다.

시멘틱 웹 질의 모듈은 질의 분석기로부터 검색 질의어와 질의어의 의미를 전달 받아서, 시멘틱 웹 검색 모듈이 처리할 수 있는 형태(RDQL)로 질의문을 재구성한다. RDQL[11]은 그래프 패턴으로 표현이 가능한 트리플 패턴으로 구성된다. 각각의 트리플 패턴은 변수와 RDF 형식의 값(URIs and Literals)을 포함한다. 즉, RDF 질의문은 사용자가 요구하는 검색 결과를 바인딩하기 위한 변수들을 선언하고, 트리플 패턴으로 검색 조건을 표현함으로써 시멘틱 웹 검색 모듈이 검색을 실행할 수 있도록 한다.



(그림 4) 검색 질의어 분석 과정



(그림 5) 시멘틱 검색 실행 과정

시맨틱 웹 검색 모듈은 단일화(Unification)을 기반으로 Rete 알고리즘을 사용하여 시맨틱 검색을 실행한다. 단일화는 두 개의 문장이 동일한 표현이 될 수 있도록 변수부호에 항을 대치하는 것을 말한다[12]. 예를 들어서  $(\forall x) W(x)$  문장에서  $W(A)$  라는 문장을 생성하기 위해서는 변수  $x$  에 문제의 대상 영역에 있는 상수부호  $A$  가 할당되어야 한다. 즉,  $W(x)$  라는 문장과  $W(A)$  문장이 동일하게 되기 위해서는  $x$  가  $A$  로 대치되어야 한다. 이러한 단일화 과정에서 여러 개의 항이  $x$  에 대치될 수도 있다. 변수를 대응하는 항으로 대체해서 얻어진 표현을 대치문 이라고 하는데,  $W(A)$  는  $W(x)$  문장에서 얻어진 대치문이 된다. Rete 알고리즘(Forgy, 1982)은 전방향추론 규칙 기반 시스템의 속도를 향상시키기 위한 만들어졌다. Rete 알고리즘은 순환이 없는 방향성 그래프(Acyclic Directed Graph)상에 규칙에 관한 정보를 저장함으로써 속도를 향상시키는 패턴 매치 알고리즘이다. 즉 그래프의 모든 노드 상에서 모든 규칙에 대한 사실(Facts)들을 매치 시키는 것이 아니고, 변화된 사실들에 대해서만 매치 시킨다. 따라서 각 노드에서 변화가 없었던 정적인 데이터는 무시되기 때문에 사실들의 매치 속도는 크게 향상된다. Rete 알고리즘이 적용된 시맨틱 웹 검색 모듈은 질의문의 트리플 패턴들을 하나씩 분리하여 루트 노드를 제외한 노드들에 표현한다. 즉, 루트 노드로부터 최하위 노드까지의 경로를 통해서 질의문의 모든 트리플 패턴들이 표현되며 루트 노드부터 현 위치의 노드까지 각 경로에 저장된 질의 패턴을 만족하는 메타데이터에 대한 정보를 각각의 노드에 저장한다. 이 정보는 패턴내의 변수에 대치되는 모든 값(Values)들을 나타낸다. 이것은 시간 중복성(Temporal Redundancy)을 줄여주는 장점이 있다. 따라서 루트 노드로부터 경로가 진행될수록 보다 적은 메타데이터가 매치된다. (그림 5)는 메타데이터를 기반으로 시맨틱 검색을 실행하는 과정을 보여준다.

### 3.5 사용자 행위 모니터

이 장에서는 시맨틱 개인화 검색을 실행하기 위한 가장 기본적인 데이터를 제공하는 사용자 행위 모니터 모듈에 대해서 설명한다. 사용자 행위 모니터 모듈은 시맨틱 검색 시스템을 사용하는 사용자 개개인의 검색 행위(무엇을 검색하고, 어떤 검색 결과에 관심을 가졌으며, 각 검색 결과가 가지는 특징)에 대한 데이터를 기록한다. 일반적으로 웹 포털 서비스를 이용하는 모든 사용자의 행위는 포털 서비스를 제공하는 웹 서버의 로그에 기록된다. 그러나 로그 정보는 오직 사용자가 관심(마우스 클릭에 의한 선택이나 방문)을 가지는 페이지의 주소에 대한 정보만을 포함하고 있기 때문에, 사용자의 검색 행위를 학습하기에는 충분하지 못하다. 이러한 문제점을 해결하기 위해서 사용자 행위 모니터 모듈은 웹 서버의 로그 정보를 바탕으로 각 사용자의 모니터 테이블을 만든다.

Culture Finder에서는 사용자의 로그정보를 각 사용자의 로그 테이블로 만들어 사용자 로그 데이터베이스에 저장하며, 사용자 로그 테이블은 다음과 같은 속성으로 구성된다.

- 시맨틱 검색 서비스를 사용할 수 있도록 할당된 사용자ID
- 문화 정보의 분류 정보(영화, 애니메이션, 음악)
- 특정 부류의 문화 정보에 대한 세부 분류(영화 리뷰, 인터뷰, 전문가 영화평)
- 검색된 결과(웹 페이지)에 대한 메타데이터 ID
- 검색된 결과(웹 페이지)가 위치하고 있는 장소의 URL
- 검색된 결과에 대한 사용자 관심도(마우스 클릭에 의한 선택이나 방문)
- 순위(추천)가 적용된 결과에 대한 사용자 관심도(피드백 정보)

Culture Finder의 사용자 로그 테이블에는 검색된 결과에 대한 사용자의 관심도가 마우스 클릭의 유무로서 기록된다. 마우스 클릭은 검색된 결과에 대해서 사용자의 관심 여부를 알아내기 위한 기준이다. 사용자 모니터 테이블을 구성하기 위한 필수 조건은 검색 결과에 대한 마우스 클릭 정보와 메타데이터 ID이다. 메타데이터 ID는 검색된 각각의 페이지와 이러한 페이지들의 메타데이터를 연결해주는 매개체다. 사용자 모니터 테이블은 사용자 성향 학습 엔진이 사용자의 정보 검색 성향을 학습하기 위한 상세한 데이터(웹 페이지에 대한 여러 가지 속성 및 속성 값)를 제공한다. 검색된 페이지의 상세한 데이터는 온톨로지를 기반으로 생성된 메타데이터에 대응되기 때문에, 사용자의 정보 검색 성향을 학습하기 위한 상세한 데이터는 각각의 페이지에 대한 메타데이터로부터 메타데이터 ID를 통해서 생성된다. 다음 알고리즘은 Culture Finder에 사용된 사용자 모니터 테이블을 생성하기 위해 정의한 알고리즘이다.

#### <사용자 모니터 테이블 생성 알고리즘>

```
Define learning field based on ontology;
for(fetchloopsizeA=0; fetchloopsizeA<size of user table;
  fetchloopsizeA++)
  User ID fetch in user table(fetch(fetchloopsizeA));
  Make user's monitor table to name user ID;
for(fetchloopsizeB=0; fetchloopsizeB<size of log table;
  fetchloopsizeB++)
  Category, subcategory, and information ID fetch in log
  table(fetch(fetchloopsizeB));
  Detailed data load in meta data
  repository(load(category, subcategory, information ID);
  Record detailed data in learning field;
```

### 3.6 사용자 기호 학습

사용자 모니터 테이블은 사용자가 관심을 갖는 검색 결과들을 사용자의 직접적인 개입 없이 마우스 클릭 유무로서 표현한다. 즉, 마우스 클릭을 통해서 사용자가 검색된 페이지 중 특정페이지를 열어보았다면 사용자가 그 페이지에 관심이 있다고 판단하고, 반면에 열어보지 않은 페이지는 모두 비 관심 페이지로 가정한다. 이러한 방법은 신뢰성이 다소 떨어지지만, 웹 환경에서 사용자의 관심을 가장 쉽게 측정할 수 있다. 웹 콘텐츠 기반 순위 시스템은 검색 결과의 신

뢰성에 따라서 순위를 적용한다. 따라서 시멘틱 개인화 검색을 실행하는 Culture Finder는 각 사용자의 정보 성향에 맞는 검색 결과를 우선적으로 보여주기 위해, 사용자의 성향이 기록되어 있는 사용자 프로파일을 사용한다.

사용자 프로파일을 구축하는 방식에는 지식 기반 프로파일 구축 방법과 행위 기반 프로파일 구축 방법이 있다. 지식 기반 프로파일 구축 방법은 엔지니어가 미리 사용자들의 프로파일 모델들을 구현하고, 설문 조사와 인터뷰를 통해서 실제 서비스를 이용하는 사용자와 가장 가까운 모델을 그 사용자의 프로파일로 선택한다. 행위 기반 프로파일 구축 방법은 보통 기계 학습(Machine Learning)기법을 사용하여 사용자 행위의 패턴을 학습하고, 이러한 행위 패턴을 기반으로 프로파일을 구축한다. 본 논문에서 제안하는 Culture Finder의 사용자 프로파일 생성 모듈은 행위 기반 프로파일 구축 기법을 기반으로 한다. 사용자 모니터 테이블에 저장된 상세 데이터는 검색된 웹 페이지의 여러 특성을 나타내는 속성과 속성 값이 저장된다. 따라서 사용자 프로파일 생성기는 각 사용자의 모니터 테이블과 사용자 성향 학습 알고리즘을 기반으로, 페이지 정보를 구성하는 각 속성마다 사용자의 기호를 나타내는 값들이 기록된 사용자 프로파일을 생성한다.

사용자 프로파일은 특정 주기(하루에 한번 또는 일주일에 한번)마다 새로 생성되는데, 이 때 사용자의 피드백이 큰 변수로 작용한다. 즉, 순위가 적용된 검색 결과 페이지(추천 검색 페이지)에 사용자가 관심을 갖지 않을 경우, 그 페이지를 구성하는 속성 값과 일치하는 프로파일의 속성 값 가중치를 낮추어야 하는 반면, 순위가 적용되지 않은 검색 결과 페이지(비 추천 검색 페이지)에 사용자가 관심을 갖는 경우, 그 페이지를 구성하는 속성 값과 일치하는 프로파일의 속성 값 가중치를 높여 주어야 한다. 사용자 피드백 정보 역시 순위가 적용된 결과에 대한 사용자의 마우스 클릭 유무로 제공 받으며, 모든 피드백 정보는 로그 테이블에 저장된다. 따라서 사용자 프로파일 생성 모듈은 모니터 테이블의 상세 데이터를 기반으로 사용자 관심 속성 값들의 가중치를 계산하고, 사용자 피드백 정보를 바탕으로 관심 속성 값들의 가중치를 자동으로 조절하여 최종적으로 사용자의 성향이 반영된 사용자 프로파일을 생성한다.

Culture Finder에서 사용되는 사용자 프로파일에는 각각의 문화 정보(영화, 애니메이션, 음악 등등)의 세부 분류 정보(영화 리뷰, 인터뷰, 전문가 영화평 등등)에 대한 속성들과 각 사용자의 관심 속성 값의 가중치가 저장됨으로써, 사용자의 문화 정보 성향을 파악할 수 있는 단서를 제공한다. Culture Finder의 사용자 프로파일 생성 모듈이 관심 속성 값의 가중치를 계산할 수 있도록, 다음과 같은 확률 기반의 가중치 계산식(WNP Function)을 정의하였다.

$$\text{Feedback Weight} = [\text{positive weight}, \text{negative weight}]$$

$$\text{WNP ratio} = \text{NP ratio} + ((\text{positive ratio} * \text{positive weight}) + (\text{negative ratio} * \text{negative weight}))$$

$$\text{NP ratio} = \text{Frequency ratio} * \text{positive ratio}$$

$$\text{Positive ratio} = \text{positive Count(attribute value)} / \text{tot\_Count(attribute)}$$

$$\text{Frequency ratio} = \text{positive Count(attribute value)} / \sum \text{positive Count(attribute)}$$

(그림 6) WNP Function

subcategory	Attribute	attribute value	Existence	Viewing
music	K-Pop	Psy's Told	Yes	Yes
music	K-Pop	Brad Pitt	Yes	Yes
music	K-Pop	Brad Pitt	Yes	Yes
music	K-Pop	Movie Song	Yes	Yes
music	K-Pop	Movie Song	Yes	Yes
music	K-Pop	Brad Pitt	Yes	Yes
music	K-Pop	Brad Pitt	Yes	No
music	K-Pop	Brad Pitt	Yes	No

$\text{NP ratio (Player, BradPitt)} = 0.17 * 1 = 0.17$       $\text{NP ratio (Player, MovieSong)} = 0.17 * 1 = 0.17$   
 $\text{NP ratio (Player, BradPitt)} = 0.17 * 0.6 = 0.102$       $\text{Player Weight (MovieSong)} = 0.2$   
 $\text{Player Weight (BradPitt)} = 0.1$       $\text{WNP ratio (Player, MovieSong)} = 0.17$   
 $\text{WNP ratio (Player, BradPitt)} = 0.40$       $\text{WNP ratio (Player, MovieSong)} = 0.53$   
 \* Weight (BradPitt) (BradPitt) (BradPitt, BradPitt)  
 $\text{Feedback Weight} = [0.1, 0.1], \text{Weight Ordering threshold} = 0.15$  (무기함)

(그림 7) 사용자 관심 속성 값의 가중치를 계산한 예

positive\_Count(attribute\_value)는 특정 속성의 특정 속성 값에 대해 사용자가 관심을 갖는 횟수,  $\sum \text{positive\_Count}(\text{attribute\_value})$ 는 특정 속성의 모든 속성 값에 대해 사용자가 관심을 갖는 횟수, tot\_Count(attribute\_value)는 특정 속성의 속성 값들이 나타나는 총 횟수, Feedback Weight = [positive weight, negative weight]는 순위가 적용된 검색 결과에 사용자가 관심을 보일 경우, 이러한 정보를 구성하는 속성 값들을 조절하는 추가적인 가중치를 의미한다. (그림 7)은 WNP function을 사용하여 관심 속성 값의 가중치를 계산한 예이다. 사용자 프로파일은 가중치를 갖는 모든 속성 값들을 반영할 수 없으므로 특정 기준치를 제시하는 Weight\_Ordering threshold를 설정하여, Weight\_Ordering threshold 이상의 가중치를 갖는 속성 값들로 프로파일을 구성한다.

Culture Finder는 검색된 정보의 분류에 따라서 속성 값들의 가중치가 정해지며, 일정 기준(Weight\_Ordering threshold) 이상의 가중치를 갖는 속성 값을 선택하여 사용자 프로파일을 구축하게 된다. 사용자 프로파일은 각 사용자의 ID명과 일치하는 프로파일 테이블들로 구축된다. 본 연구에서 사용자 프로파일을 구축하는 방법은 다음과 같이 정리할 수 있다. 모니터 테이블을 바탕으로 각 속성별로 속성 값들의 가중치를 구하고, 가중치가 높은 순으로 정렬한 후 일정 가중치를 넘는 속성 값들만 프로파일 테이블의 해당 속성 필드에 삽입한다.

### 3.7 검색 결과 순위 적용

일반적으로 사용자는 자신이 원하는 정보만을 검색하기는 어렵지만, 검색된 결과가 자신에게 얼마나 유용한지 인식할 수 있다. 이 때문에 검색결과에 대한 사용자의 관심뿐 아니라 사용자의 정보 성향에 맞게 추천해 준 정보(순위가 적용된 검색 정보)에 대한 관심을 알아내는 것이 가능하며, 검색 결과에 대한 관심 평가를 바탕으로 개인화 검색 시스템들은 사용자가 관심을 가지는 특징들을 반영하는 정보를 검색한다. 검색 결과 순위 모듈은 사용자 프로파일을 참조하여 검색 결과에 추천 점수(Recommend Confidence)를 할당한다. 일정한 기준 이상의 추천 점수(Recommend Confidence)를 갖는 검색 결과는 추천 점수가 큰 순으로 순위가 적용되어 시멘틱 개인화 검색 결과로 제공된다. 즉, 순위가 적용된 정보는 사

용자의 정보 성향이 반영된 시멘틱 개인화 검색 결과이며, 순위가 적용되지 않은 정보는 일반적인 시멘틱 검색 결과다.

추천 점수는 랭커(Ranker) 알고리즘에 의해 계산되며, 다음은 Culture Finder의 검색 결과 순위 모듈이 검색 결과들의 추천 점수를 계산하여 순위를 적용하기 위한 알고리즘을 정의한 것이다.

<검색 결과 순위 적용 알고리즘>

```

for(loopsize=0; loopsize<number of search results;
  loopsize++)
information ID fetch from search results(fetch(loopsize);
for(fetchloopsizeC=0; fetchloopsizeC<number of attribute;
  fetchloopsizeC++)
  A value of attribute fetch in meta data DB(fetch
    (fetchloopsizeC, information ID, attribute);
  Interest values fetch in user profile(fetch(attribute,
    userID);
    if(first interest value == value)
      confidence point = 1st_point(1.0);
    else if(second interest value == value)
      confidence point = 2st_point(0.8);
    else if(third interest value == value)
      confidence point = 3st_point(0.5);
    else confidence point = zero;
    rank_point = rank_point + confidence point;
  if(rank_point >= threshold)
    assign recommend confidence(rank_point) to search
  result;
  else assign recommend confidence(zero) to search
  result;
Sort search result according to recommend confidence;
  
```

#### 4. 검색 결과에 대한 고찰

본 논문에서 제안하는 Culture Finder는 시멘틱 웹 환경을 지원하는 여러 가지 서비스 시스템중의 하나인 시멘틱 검색 시스템에 적용되어 사용자에게 기호에 맞는 정보를 제공할 수 있다. 이와 같은 기능은 수시로 변하는 사용자의 검색 행위와 사용자가 제공받은 정보의 특징을 파악하고, 이러한 사실들을 정량화된 데이터로 정제하여 사용자 프로파일을 학습함으로써 가능하다. 그래서 본 논문이 제안하는 Culture Finder의 적용 가능성을 평가하기 위해 PDA에서 영화 정보를 간편하게 검색 할 수 있도록 구현된 음성 인식 기반의 클라이언트 프로그램을 Culture Finder에 연결하여 영화 정보를 검색하는 과정을 보인다. 또한 검색 결과에 대한 만족도의 증감 수치를 측정하여 이를 바탕으로 Culture Finder에서 제공하는 시멘틱 개인화 검색에 대한 효율성을 실험해 보았다.

본 실험을 위해 사용된 PDA 프로그램은 음성 인식을 기반으로 사용자가 간편하게 영화 정보를 검색하고, 정보가 존재하는 웹 페이지를 보여주는 클라이언트 프로그램으로서 2005년 주식회사 니츠(Nitz)가 개발한 시스템이다. PDA 검

색 시스템은 사용자가 검색 단어를 PDA 마이크를 통해 음성으로 입력하면, 키워드 데이터로 변환한 후 Culture Finder에 전송하고, 검색 결과(웹 페이지 제목, 웹 페이지 URL)를 전달 받는다.

다음 그림들은 사용자가 Culture Finder에 연결된 PDA 프로그램을 사용하여 검색하는 과정들을 보여준다.



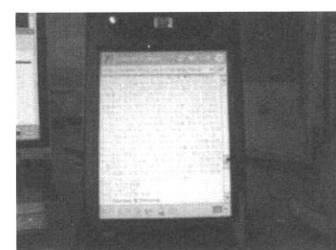
(그림 8) 음성을 통한 검색 키워드 입력



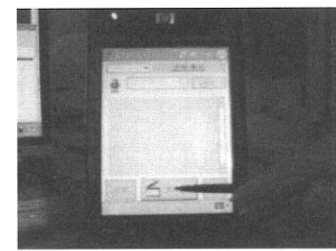
(그림 9) 검색 시작



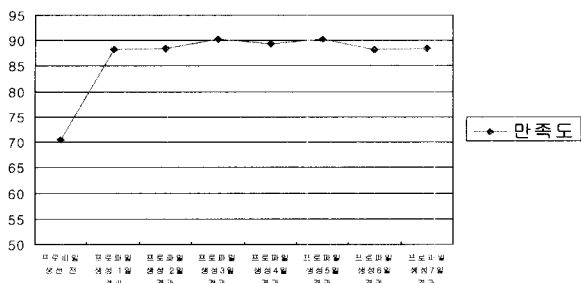
(그림 10) 세부 검색 메뉴 선택



(그림 11) 사용자 기호가 적용된 영화 정보들이 검색



(그림 12) 웹 브라우저를 통한 해당 정보의 상세 내용 확인



다음은 Culture Finder에서 제공하는 시멘틱 개인화 검색에 대한 효율성을 실험해 보았다. 이를 위해 50명의 임의의 사용자를 선정하고, 사용자 프로파일이 만들어지기 일주일 전의 사용 만족도와 사용자 프로파일이 만들어지기 시작한 후의 일주일 동안의 사용만족도를 비교해보았다. 다음 도표는 측정된 사용자 만족도의 변화를 보여준다. 먼저 만족도는 사용자 프로파일이 적용되어 개인화 정보 서비스를 제공한 이후, 15% 이상 상승하였다. 이는 Culture Finder가 제공하는 시멘틱 개인화 검색이 어느 정도 효율성이 있다는 것을 증명한다. 그러나 프로파일이 만들어지기 시작한 후에, 시간이 지나면서, 오히려 만족도가 약간 하락하였다. 이는 사용자 기호를 학습하는 방법이 아직은 완전치 않다는 것을 반영해준다. 본 논문에서 제안된 시멘틱 개인화 검색 시스템은 현재 계속 진행 중인 연구로서 사용자의 정보 성향 학습 기법에 대해서 보다 자세한 연구가 진행될 것이다.

### 5. 결론 및 향후 연구

본 논문에서는, 문화 정보에 대한 시멘틱 개인화 검색을 실행하는 Culture Finder의 구조를 제안하고, 이러한 구조를 통해서 시멘틱 개인화 검색에 적용되는 여러 가지 기술에 대해 설명하였다. Culture Finder를 설계하기 위해서, 본 논문에서는 웹 온톨로지 기술 언어인 OWL을 기반으로 효율적으로 메타데이터를 구축하기 위한 방안, 질의 분석 및 시멘틱 검색 방법과 사용자 프로파일 구축 및 검색 결과에 대한 순위 계산 방안을 고려하였다.

Culture Finder는 시멘틱 웹 검색 에이전트가 문화(문화 뉴스, 공연, 전시)에 관련된 다양한 정보들에 대해서 개인화된 검색을 하는데 도움을 준다. 본 논문에서 제안된 구조는 웹 콘텐츠 및 서비스를 일괄적으로 자동(또는 반 자동)통합하여, 웹 상에 존재하는 자원을 정확하게 사용하는 시멘틱 웹, 웹 서비스 및 다중 에이전트 기술에 적용될 수 있다.

본 논문에서 제안된 시멘틱 개인화 검색 시스템은 현재 계속 진행 중인 연구로서, 향후에는 온톨로지 기반의 사용자의 정보 성향 학습과 사용자 프로파일을 기반의 검색 결과 순위 적용 기법에 대해서 보다 자세한 연구가 진행될 것이다.

### 참고 문헌

[1] Li Ding, Tim Finin, "Swoogle: A Search and Metadata

Engine for the semantic web", CIKM'04, November 8-13, 2004, Washington DC, USA

[2] Semantic web, <http://www.w3.org/2001/sw/>  
 [3] RDF Site Summary (RSS) 1.0, <http://web.resource.org/rss/1.0/>  
 [4] Atom, <http://www.mnot.net/drafts/draft-nottingham-atom-format-02.html>  
 [5] Guha, R. V., McCool, R. and Miller, E. "Semantic search", Proceedings of the twelfth international conference on World Wide Web(WWW2003), ACM Press, 2003  
 [6] D. Quan and D.R.karger. "How to make a semantic web browser", WWW 2004  
 [7] Haibo Yu, Tsunenori Mine, Makoto Amamiya, "An Architecture for Personal Semantic Web Information Retrieval System", WWW 2005, May. 10-14, 2005, Chiba, Japan  
 [8] MSRBot, <http://www.research.microsoft.com/research/sv/msrbot/#webcrawler>  
 [9] Welcome to protege, <http://protege.stanford.edu/index.html>  
 [10] OntoMat Annotizer, <http://annotation.semanticweb.org/ontomat/index.html>  
 [11] RDQL A Query Language for RDF, <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>, January 2004  
 [12] George F Luger, "ARTIFICIAL INTELLIGENCE", Addison-Wesley, pp.67-68, 1994  
 [13] The RETE Algorithm, <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>  
 [14] Stuart. E. Middleton, Nigel. R. Shadbolt, David C de Roure, "Ontological User Profiling in Recommender System", ACM Transaction on Information System, Vol. 22, No. 1, January 2004, pp.54-88.



김 제 민

e-mail : kimjemins@hotmail.com  
 2001년 숭실대학교 컴퓨터학과(학사)  
 2004년 숭실대학교 대학원 컴퓨터학과(석사)  
 2004년~현재 숭실대학교 대학원 컴퓨터학과 박사과정

관심분야: 인공지능, 게임 AI, 시멘틱 웹



박 영 택

e-mail : park@computing.soongsil.ac.kr  
 1978년 서울대학교 전자공학과(학사)  
 1980년 KAIST 전산학(석사)  
 1992년 Univ. of Illinois at Urbana-Champaign(박사)  
 1981년~현재 숭실대학교 컴퓨터학과 교수

관심분야: 인공지능, 에이전트, 전문가 시스템