

시맨틱 웹 환경에서 웹 서비스를 위한 RDF Triple 처리기법[☆]

RDF Triple Processing Methodology for Web Service in Semantic Web Environment

정 관 호* 김 판 구** 김 권 천***
Jeung, Kwan-Ho Kim, Pan-Koo Kim, Kweon-Cheon

요 약

온톨로지를 사용하여 웹 서비스 검색 기능을 향상시키려는 연구가 진행되고 있다. 그 중 DAML, DAML+OIL를 이용하여 UDDI를 검색하는 방안이 제안되었다. 하지만 이 방식은 UDDI 검색 시 검색에 적합한 연산자들을 사용해야 하는 불편함과 그에 따른 온톨로지를 정의해 주어야 하는 번거로움이 존재한다. 이러한 문제점을 해결하기 위해 본 논문에서는 시맨틱 웹 환경에서 정보검색 및 의미적인 추천을 위한 N-Triple 처리, 핵심 Triple 추출을 위한 필터링, Triple 통합, Triple간 의미적 연결, Triple 탐색 기법을 제시하고 더 나아가 제시된 기법으로 시스템을 설계하고 구현한다. 끝으로 가상의 시맨틱 웹 환경에서 제안된 시스템을 실험하여 본 논문의 유용성을 입증한다.

Abstract

Researches on enhancing the searching function of the web service using the ontology concept have been studying. One of them suggests a searching method for UDDI using DAML and DAML+OIL. However this approach has inconveniences to use operations proper to the circumstance and to define respective ontologies according to them. To solve these problems, we introduce an effective method of dealing with N-Triple, filtering core Triples, merging Triples, semantic connection between Triples and searching Triples for searching information and recommending the results in semantic web environment. Furthermore, we implement this proposed method in a system to test it. Finally, we test the system in the virtual semantic web environment for our research analysis.

☞ Keyword : RDF, RDFS, 시맨틱 웹(Semantic web), 웹 서비스(Web service)

1. 서 론

현재의 웹은 다양한 기술들이 융합되어 복합적인 서비스를 제공하고 있다. 관련정보를 연결하여 작업의 효율성을 높이고자 했던 초기의 단

순 메커니즘은 다양한 사용자의 요구조건이 더해지면서 급속도로 발전해 현재는 IT 기술의 핵심 코어로 급상승하였다. 웹이 폭발적으로 진화할 수 있었던 이유는 웹이 가지고 있는 편의성과 확장성 그리고 응용성에서 비롯된다. 웹은 정보 표현의 단순성을 이용하여 다양한 정보를 생성하게 되고 생산된 정보는 다른 정보와 융합되어 재사용되었다. 이런 과정에서 웹은 정적인 정보제공 기능에서 정보를 동적으로 주고받을 수 있는 상호 정보제공체제로 변화하였다. 웹 기술의 발달에 따라 웹은 플랫폼 독립적으로 시스템간의 연계, 통합 및 자원 공유를 가능하게 하는

* 준 회 원 : 한양대학 정보통신학과 연수연구원
kwanho@chosun.ac.kr

** 정 회 원 : 조선대학 컴퓨터공학부 교수
pkkim@chosun.ac.kr

*** 정 회 원 : 조선대학 의학과 교수
mdkckim8@hanmail.net

[2005/09/30 투고 - 2005/10/14 심사 - 2006/02/13 심사완료]

☆ 본 논문은 2004년 조선대학교 연구 보조비 지원에 의하여 연구되었음.

웹 서비스(Web Services) 기술로 진화하고 있다. 웹 서비스는 XML을 기반으로 한 공개 표준을 사용하여 기존의 분산 컴퓨팅 환경의 문제점이었던 이기종 시스템 간 상호 운용성 문제를 해결하였다[3]. 웹 서비스에서 서비스 공급자는 WSDL을 이용하여 서비스 내역과 이용 방식을 공시하고 자신의 서비스 존재를 UDDI와 같은 서비스 정보 저장소를 통해 광고하게 된다. 서비스 요구자는 UDDI와 같은 서비스 정보 저장소를 통해 필요한 웹 서비스 제공자를 검색하게 되며, 서비스 제공자의 WSDL을 분석함으로써 서비스를 실질적으로 호출, 이용이 가능하다[4]. 웹 서비스의 문제점은 UDDI를 통해 단순한 키워드 기반의 메타데이터 정보를 검색하고 WSDL을 통한 호출 메커니즘만 정의하고 있으며 구문적(syntactic) 상호 운용성을 보장할 뿐 의미적인 부분은 보장하지 못한다. 이런 문제점을 해결하기 위해 온톨로지 개념을 도입하여 의미적인 검색을 시도하려는 연구가 진행되고 있다. 서비스 정보 저장소의 정보를 기존의 XML 형식이 아닌 온톨로지 언어로 구성하고 서비스 제공자가 제공하는 서비스 정의 또한 온톨로지 구성하여 서비스 사용자가 의미기반 정보검색이 가능하도록 시도하고 있다. 온톨로지 개념을 기반으로 만들어진 언어에는 Topic Map, DAML, OIL, DAML+OIL, RuleML, RDF, OWL 등이 있으며 이중 웹 서비스 검색 기능을 향상하기 위해 DAML 프로젝트 그룹에서 제안한 DAML, DAML+OIL를 이용한 연구들이 활발하게 진행되고 있다. 하지만 DAML+OIL은 웹 서비스 검색 시 상황에 적합한 각 연산자들을 사용해야 하는 불편함과 그에 따른 각각의 온톨로지를 정의해 주어야 하는 번거로움이 존재한다. 이러한 문제점을 해결하기 위해 본 논문에서는 RDF N-Triple를 사용하여 기업의 정보를 구성하고 구성된 정보를 해석할 수 있는 처리기법을 보이고자 한다. 더 나아가 검색된 정보를 활용하여 사용자가 최종적으로 사용가능한 정보

를 추천할 수 있는 시스템을 구축하여 제안된 기법을 검증하고자 한다.

본 논문의 구성은 2장에서 시맨틱 웹과 온톨로지 언어인 RDF 그리고 웹 서비스에 대해 살펴보고 3장에서는 RDF로 웹 서비스를 수행할 수 있도록 정보를 구성하고 처리하는 기법을 제안한다. 이 기법을 활용해 4장에서는 가상의 시맨틱 웹 환경 속에서 시스템을 구축하여 성능을 평가하였고 끝으로 연구결과 요약과 향후 연구에 대해 서술하였다.

2. 관련 연구

본 장에서는 차세대 웹인 시맨틱 웹과 온톨로지 언어인 RDF에 대해 알아보고 현재 웹 서비스의 기술적 현황 및 문제점에 대해 알아본다.

2.1 시맨틱 웹

현재의 웹이 정보의 양적 확장에 초점을 두었다면 시맨틱 웹은 온톨로지와 의미적 해석을 기반으로 정보의 질적 성장에 초점을 두고 있다. 시맨틱 웹에 대한 표준화 및 기술연구는 W3C를 중심으로 진행이 되고 있다. 현재 시맨틱 웹에 관한 연구 주제는 크게 언어(RDF, OWL), 기반구조, 온톨로지로 나뉘지며 각 세부분류별 하부 모임을 조직하여 활발히 연구하고 있다. 또한 복합 응용으로서의 시맨틱 웹과 웹 서비스를 결합시킨 시맨틱 웹 서비스(Semantic Web Services)에 관한 응용 통합 연구들도 활발히 진행되고 있다. 시맨틱 웹의 계층구조를 7단계로 구성된다[12,13]. 가장 기본이 되는 층은 자원 표기를 위한 URI와 다국어 지원을 위한 Unicode다. 그 위에 structure와 syntax를 표현하기 위한 data layer 층으로 XML, Namespace, XML Schema, RDF 등으로 구성되며, 그 위에 schema layer로서 Vocabulary를 정의하고, 간단

한 semantics를 표현하는 Lightweight Ontology로서의 RDF Schema가 있다. 그리고 semantic layer로 Formal Semantic과 Reasoning을 지원하는 Ontology로서의 OWL가 있다. 이러한 주요 layer에 기초한 상위 layer로 Rules, Logic framework, Proof, Trust Layer들로 구분된다. 현재 정보를 표현할 수 있는 표현 언어(RDF, OWL)가 만들어졌다 하더라도 그 언어로 구성된 정보를 기계와 인간이 해석하고 처리하지는 못하고 있다. 이를 위해서는 표현 언어 처리기술 및 상위단계(Rule, Logic, Proof, Trust)의 연구가 더 필요하다.

2.2 RDF(Resource Description Framework)

시맨틱 웹을 위한 대표적인 온톨로지 언어는 RDF와 OWL이 있다. RDF는 자원에 대한 메타 데이터를 기술하는 프레임워크라고 말할 수 있다. RDF는 리소스(Resource), 특성(Property), 서술문(Statement)의 개념으로 구성된다. 웹 페이지나 웹 사이트 등의 모든 사물(thing)은 리소스로 표현되고, 각 리소스의 특성이나 다른 리소스와의 관계 등을 특성으로 나타낸다[14,15]. 어떤 리소스의 한 특성에 대한 값을 나타내는 것이 서술문이다. 이를 RDF Triple이라고 말하며 일반적으로 Subject, Predicate, Object라고 말한다. 리소스의 구성을 표현하기 위해 XML과 RDF를 혼합하여 사용할 수 있는데 이를 RDF/XML Serialization이라고 한다[7,9]. 자원이 가지고 있는 도메인(Domain) 및 범위(Range) 그리고 도메인에 속한 개념들 간의 관계를 정의하기 위해서 RDF 스키마(Schema)가 있다. RDF 스키마에 정의된 관계에 의해 의미적 자원연결이 가능하고 연결된 자원은 사람과 기계가 이해 가능한 메타 데이터 형태로 표현된다. RDF/XML 구문이 자원의 구성과 표현에 가깝다면 RDF 스키마는 자원의 관계정의, 즉 온톨로지 언어의 특성에 가깝다.

현재 W3C의 활발한 연구 활동에 의해 시맨틱 웹을 위한 정보표현 방법들이 구체화되고 있다. 또한 표현 언어를 해석하고 추론하기 위한 연구들이 활발하게 진행되고 있지만 아직 구체적인 해석방법을 제시하지는 못하고 있다. 시맨틱 웹 환경에서 다양한 정보를 공유하고 서비스하기 위해서는 RDF로 구성된 정보를 해석할 수 있는 기술이 절실하게 요구되고 있다.

2.3 웹 서비스

웹 서비스는 인터넷상에서 단일 기업 또는 다수 기업간에 XML을 기반으로 상호 연동시키는 표준화된 소프트웨어 기술 및 이러한 기술을 활용한 비즈니스 서비스를 말한다. 웹 서비스는 기존의 응용프로그램처럼 완벽한 상호작용 사양을 정의하는 것이 아니라 서로 주고받는 데이터 표준에 대한 정의를 규정함으로써 시스템간 유연한 연계를 정의하고 있다. 거래업체간의 실질적인 시스템, 실질적인 프로그램언어간의 커뮤니케이션 차이를 극복해 주는 느슨한 분산 환경을 구현(Loosely Coupled Distributed Computing)할 수 있다는 장점이 있다. 웹 서비스는 다양한 하드웨어와 소프트웨어의 문제를 해결하기 위해 SOAP, UDDI, WSDL과 같은 개방형 표준 기술을 기반으로 서비스 제공자(Service Provider), 서비스 저장소(Service Registry) 및 요청자(Service Requester)를 유기적으로 결합하여 구현된다. UDDI는 웹 서비스의 디렉터리 서비스를 담당하게 되는데 업체가 자사의 웹 서비스를 온라인 디렉터리에 등록·광고하거나 외부에서 웹 서비스를 검색하는데 사용된다. WSDL은 웹 서비스의 서비스를 정의하는 언어로서 프로그램이나 인터페이스 정의를 기술할 때 사용되며 SOAP은 분산된 정보를 교환하는 통신 프로토콜로서 인터넷을 통해 다양한 웹 서비스 사용자가 정보를 교환·전송할 수 있는 기능을 담당한다.

2.4 DAML+OIL

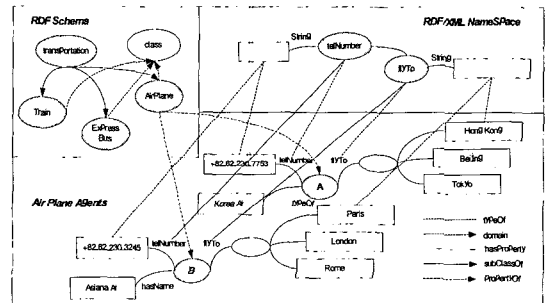
DAML+OIL은 사용자가 웹 서비스를 자동적으로 위치, 선택, 합성, 감독할 수 있게 하는 적절한 온톨로지를 제공한다. 이에 따라 웹 에이전트는 자동적으로 서비스 기술을 판독하고 이러한 서비스들을 자동적으로 접근할 수 있다. DAML+OIL에서 서비스에 대한 온톨로지를 구조화하기 위해 프로파일(Service Profile), 서비스 모델(Service Model), 서비스 그라운드링(Service Grounding)등 3가지 핵심적인 지식 유형이 필요하다[17]. 이러한 언어의 특성을 이용하여 논문 [1]에서 입력, 출력, 제약 연산자를 사용하여 매칭 알고리즘을 제안하였다. 논문 [2]에서는 DAML+OIL를 사용하여 WSDL 형식에 맞게 웹 서비스를 기술하고 있다. 상위 연구의 문제점은 UDDI 검색할 시 Service Template에 정의된 입력, 출력, 제약 연산자를 사용하여 검색을 하는 단점이 있으며 더욱이 각 기능에 맞는 독립된 온톨로지를 구축해야 하는 불편이 따른다. 이런 문제점을 해결하기 위해 본 논문에서는 RDF를 사용해서 서비스 정보를 구축하였으며 정보들 간의 의미적 연결을 시도하고 정보 검색 시 해당 Predicate과 RDF Triple Node 검색을 통해 기존 연구에서 시도되었던 복잡성을 제거하고자 한다. 더 나아가 사용자가 실질적으로 사용 가능한 서비스 정보를 추출하기 위해 검색된 결과를 사용자 정보와 매칭시켜 추천하는 기법을 제시한다.

3. RDF를 이용한 서비스 정보 구성 및 처리 기법

3.1 서비스 정보 표현

UDDI는 웹 서비스 정보를 관리하기 위해 White pages, Yellow pages, Green pages를 사용한다. White pages는 비즈니스의 이름, 성명,

연락처, 식별자 정보 등을 가지며, Yellow pages는 표준 분류법에 따른 산업적인 분류 정보를 가진다. Green pages는 웹 서비스 이용에 관련된 기술적 정보를 나타낸다. 온톨로지 개념으로 Yellow pages는 정보의 분류를 나타내고 있기 때문에 상위 온톨로지라 말할 수 있다. 이는 RDF에서 RDF 스키마에 해당된다. RDF 스키마는 특정 도메인의 상위 개념들 간의 관계 및 분류를 정의할 수 있다. White pages와 Green pages는 상위 개념에 따른 하부의 특정 정보를 담고 있기 때문에 RDF에서는 RDF/XML로 표현할 수 있다. 본 논문에서는 현재 UDDI에서 RDF를 사용하여 정보를 구축할 수 없기에 정보 저장소라는 가상의 공간을 만들어 기업 정보를 구축하였다.



〈그림 1〉 항공권 판매 기업의 정보구성

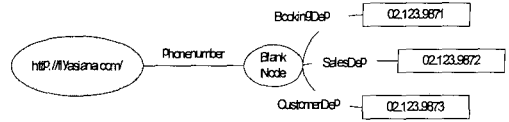
〈표 1〉 RDF/XML로 표현된 기업정보

```
<?xml version='1.0' encoding='UTF-8'?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ns1="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:ns2="http://www.informationreg/xmlns/plane#">
  <rdf:Description rdf:about="http://www.koreanair.co.kr/">
    <ns1:TITLE>Korean Air</ns1:TITLE>
    <ns1:TEL rdf:nodeID="bNode1"/>
    <ns2:FLYTO rdf:nodeID="bNode2"/>
  </rdf:Description>
  <ns1:work rdf:nodeID="bNode1">
    <rdf:value>+82-62-230-7754</rdf:value>
  </ns1:work>
  <rdf:Description rdf:nodeID="bNode2">
    <rdf:value>Hong kong</rdf:value>
    <rdf:value>Beijing</rdf:value>
    <rdf:value>Tokyo</rdf:value>
  </rdf:Description>
</rdf:RDF>
```

그림 1은 가상의 정보 저장소에 항공권 서비스 기업 정보를 RDF의 Triple 개념으로 도식화한 것이다. Yellow pages에 해당되는 기업 분류 및 관계를 RDF schema에 정의하였다. 최상위 개념으로 Transportation을 두고 그 하위에 Air Plane, Express Bus, Train을 서브 클래스로 선언, Air Plane의 Property로 각 항공 에이전트를 두었다. RDF/XML Namespace에서는 각 항공 에이전트에서 사용할 속성들에 대해 정의하였고 이를 이용하여 항공 에이전트 정보를 구성하였다. 그림 1과 같이 구성된 정보는 정보 저장소에 다른 관계들과 연결되어 웹 서비스 에이전트가 정보를 찾고자 할 때 구성 관계를 이용하여 정보를 검색하게 된다. 표 1은 그림 1의 개념을 RDF로 코딩한 기업정보의 일부를 나타내고 있다.

3.2 RDF N-Triple 처리

정보 저장소 내부에 저장된 정보는 그림 2와 같이 RDF N-Triple로 구성된다. 사용자로부터 입력된 값을 바탕으로 정보저장소 내의 기업정보를 검색하고 해석하기 위해서는 우선 RDF N-Triple을 연관된 노드(Node)로 구성하는 전처리작업이 필요하다. 각각의 노드(Subject, Predicate, Object)들은 완전한 Triple를 구성하고 있는 경우와 중간에 연결노드(Blank Node)를 걸쳐 다른 Triple과 연결된 결합형 Triple이 존재한다. 즉 정보를 표현하다보면 1:1 관계뿐만 아니라 1:n의 관계 또한 요구된다. 1:n, 즉 하나의 노드가 다수의 속성 값을 가지고 있는 경우며 그림 2처럼 중간에 연결노드를 두어 그 값들을 연결시켜준다. 이 경우, 연결노드의 Predicate 값에 따라 Object 값이 변화하게 된다. 그림 2의 RDF Graph를 Triple 단위로 세분화하면 표 2같다. 첫 번째 노드의 Object가 두 번째 이하 노드의 Subject로 선언되어 첫 번째 노드의 Subject와 최종 연결된 모습을 보이고 있다.

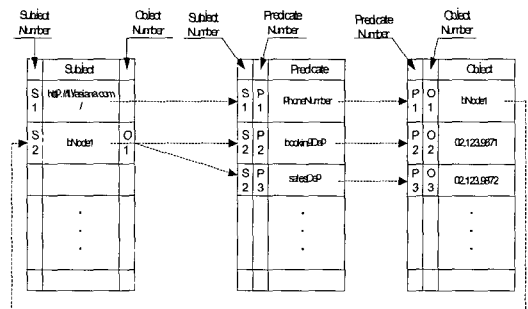


〈그림 2〉 연결노드에 다수의 Object 연결

〈표 2〉 Graph의 Triple 구성

Subject	Predicate	Object
http://www.flyasiana.com/	phoneNumbers	bNode1
bNode1	BookingDep	02-123-9871
bNode1	SalesDep	02-123-9872
bNode1	CustomerDep	02-123-9873

본 논문에서는 N-Triple을 처리하기 위해 그림 3과 같이 3개의 기억 공간을 설정하여 Subject, Predicate, Object 노드를 저장하고 각 노드별로 고유번호를 부여하여 서로 연결하였다. 1:n의 노드관계는 그림 3의 'S2'가 보여주는 것처럼 상위 노드의 고유번호를 공유하여 각각 하부의 서브 노드임을 표기하였다. 기억 장소에 저장된 노드들은 Triple 검색의 모든 과정에서 불필요한 RDF 문서 접근시간을 절약하고 연결된 Triple 탐색을 보다 효율적으로 하기 위해 설계되었다.



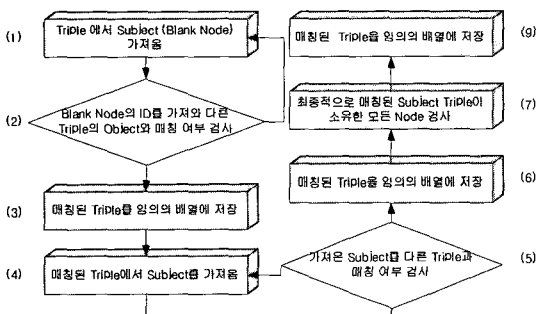
〈그림 3〉 Triple 구성을 위한 메모리 구성

3.3 핵심 Triple 추출을 위한 트리밍 방법

구성된 N-Triple에는 핵심정보를 가지고 있는

노드와 이와 무관한 노드들이 존재하게 된다. 보다 빠르고 정확한 검색을 위해 핵심이 되는 노드 Triple을 제외한 나머지 Triple들을 트리밍(trimming)하였다. 트리밍 과정에서의 문제점은 핵심 노드가 연결노드를 가지고 있는 경우이다. 핵심노드가 연결노드를 가지고 있다면 핵심노드에 따른 하부의 모든 노드들이 가지고 있는 정보를 보존해야 하기 때문에 하부노드를 찾는 작업이 필요하다. 핵심 Triple과 연결된 연결노드, 연결노드와 연결된 각 노드들을 적절하게 연결하기 위해 아래와 같은 해결 방법을 제시한다.

그림 4의 ①에서 ③까지는 연결노드와 동일한 ID를 가지고 있는 다른 Triple의 Object를 찾기 위해 수행되며 ④에서 ⑥은 동일한 Object를 가지고 있는 다른 Subject를 찾기 위하여 수행된다. ⑦과 ⑧은 처음 키워드 매칭된 Triple과 동일한 연결노드를 가지고 있는 모든 Triple을 찾기 위해 수행된다. 즉 어떤 한 개의 Triple이 있을 때 Object로 "bNode"이라는 연결노드를 소유하고 "bNode"에 연결된 다수의 Triple이 존재할 경우가 있으므로 ⑦과 ⑧절차를 수행하여 연결된 다수의 Triple을 가져오게 된다.

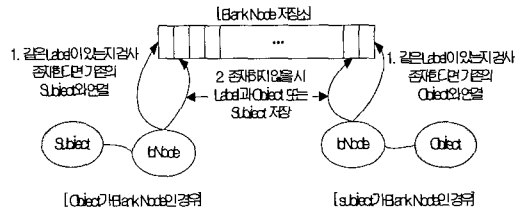


〈그림 4〉 필터링 시 연결노드 처리기법

3.4 RDF/XML 문서 통합 시 연결노드 처리

RDF Triple 통합 시 Subject 및 Object가 연결노드로인 경우 연결노드와 연결된 노드들을

찾아서 연결해 주어야 하는 문제가 발생한다. 즉 n' 가 정보를 가지고 있는 노드일 경우 n' 는 $n = \{n_1, n_2, n_3, \dots, n_n\}$ 의 원소이며 집합 n 은 다시 $S = \{a, b, \dots, n\}$ 중의 한 원소인 경우, 상위의 각 노드들(a, b, \dots, n)과 최상위 노드 S 를 모두 찾아서 연결해 주어야 관련정보의 손실이 발생하지 않는다. 이를 해결하기 위해 다음과 같은 방법을 제시한다. 만약 통합하고자 하는 두 문서에 동일한 이름을 가진 연결노드 Label이 존재할 경우 통합 과정에서 하나의 동일 연결노드 이름으로 묶어 주어야 한다. 이를 위해 우선 연결노드를 위한 임의의 배열을 선언하고 Object가 연결노드일 때 배열에서 같은 Label를 가진 값을 찾는다. 만약 같은 값이 존재하지 않을 경우, Label과 함께 Object를 배열에 저장해 둔다. 동일 Label을 가진 Subject 연결노드 발견 시, 저장된 Object와 Label을 이용하여 각 노드를 연결한다. Subject가 연결노드인 경우도 위와 같은 방법으로 연결한다.



〈그림 5〉 문서통합 시 연결노드 처리

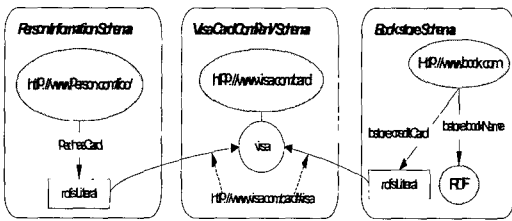
〈표 3〉 연결노드 처리

```

foreach($res ->triples as $value) {
    $subject = $value->getSubject();
    $id = $subject->getId();
    $object = new BlankNode($id);
    $temp = $baseModel->find($object);
    $newModel->add($temp);
    foreach($temp->triples as $value2) {
        $subject2 = $value2->getSubject();
        $label = $subject2->getLabel();
        $subject3 = new Resource($label);
        $temp2 = $temp->find($subject3);
        $newModel->addModel($temp2);
    }
    $temp = $baseModel->find($object);
    $newModel->add($temp);
}
    
```

3.5 자원들 간의 의미적 연결

RDF Triple로 구성된 자원들을 의미적으로 연결하여 정보 검색 시 연결된 Triple 간의 관계를 파악하고 적절한 판단을 내릴 수 있어야 한다. 본 논문에서는 아래와 같은 방법으로 각각의 Triple 간의 연결을 시도하였다.



〈그림 6〉 개인정보와 기업정보 연결

예를 들어, 어느 개인이 소지하고 있는 카드가 'visa'인 경우 표 4와 같이 RDF문으로 표현한다. 'rdf:type'속성에 'rdf:resource'로 'http://www.visa.com/card#visa'라는 속성값을 설정한다.

〈표 4〉 'visa' 카드를 소지한 개인정보

```
<rdf:Description rdf:about="
http://www.person.com/foo/card">
  <rdf:type rdf:resource="http://www.visa.com/card#visa"/>
</rdf:Description>
```

이는 속성값으로 주어진 URI가 type 값이 되는 것을 의미한다. 즉 그림 6의 RDF Graph처럼 개인은 자신이 소지한 카드에 대해 기술하지 않고 'rdf:resource'를 사용하여 카드사가 구축해 놓은 RDF Schema와 연결해 카드의 내역을 상세하게 표현할 수 있다. 만약 'visa' 카드를 소지한 개인이 어떤 웹 서비스를 하는 곳에서 물건을 사고자 했을 경우, 그 물건이 'visa' 카드 로만 결제가 가능하다고 한다면 개인이 소지한 카드사의 정보와 웹 서비스를 하는 카드사의 정보 일치 여부를 파악해 그 물건의 구입 여부를 파악할 수 있게 된다. 표 5는 개인정보 매칭처

리를 위한 알고리즘의 일부를 보여주고 있다.

3.6 RDF N-Tuple 탐색

RDF Triple 탐색은 F-Logic[8]을 사용하여

〈표 5〉 개인정보 매칭 처리

```
function personProfileMatch($nodes) {
  $pNodes = new memNodes();
  $pNodes->load(person information document);
  $subjectLabel = "";
  $matchObjectLabel = array();
  $it = $nodes->getNodesIndex();
  while ($it->hasNext()) {
    $statement = $it->next();
    $predicateLabel =
    $statement->getLabelObject();
    $res = $pNodes->find(new
      Resource($predicateLabel));
    if($res[0] != NULL) {
      foreach($res->triples as $value) {
        $matchObjectLabel[] =
        $value->getObject();
      }
    }
  }
  return $matchObjectLabel;
}
```

Triple들을 탐색하였다. RDF Triple은 객체 지향적 성격을 가지고 있지만 실질적으로 속성(Predicate) 중심 구조로 되어있다. Subject에 해당하는 자원은 Predicate에 의해 Object 자원과 연결된다. 그러므로 Predicate이 어떻게 설정되었느냐에 따라 Subject, Object 관계가 달라진다. Triple 구성요소를 로직으로 표현하면 표 6와 같다.

〈표 6〉 Triple 상황별 Logic 표현

단일 Triple	Subject[Predicate→Object]
다수의 Object	Subject[P ₁ →O ₁ , P ₂ →O ₂ , P ₃ →O ₃ , ...]
연결노드	Subject1[P ₁ →Subject2[P ₂ →O ₂]]

연결노드인 경우, 첫 번째 Object가 두 번째 노드의 Subject로 표현된다. 표 2의 Triple 중 'BookingDep'의 전화번호 '02-123-9871'을 가지고 있는 Subject 노드를 찾고자 할 경우, 다음

과 같은 F-Logic으로 표현할 수 있다.

FORALL X <-

X[phoneNumbers->bNode1[BookingDep->02-123-9871]]

‘X’는 Subject에 대응되고 ‘phoneNumbers’는 ‘bNode1’을 연결하는 Predicate가 된다. ‘bNode1’은 첫 번째 노드의 Object이자 두 번째 노드의 Subject로 사용되었다. ‘BookingDep’은 Object ‘02-123-9871’을 연결하는 두 번째 Predicate로 선언되었다. 최종결과는 첫 번째 Triple의 Subject인 ‘http://www.flyasiana.com/’ 이 된다.

지금까지 RDF 온톨로지 언어를 사용하여 기업의 정보를 구축하고 구축된 정보를 처리할 수 있는 RDF Triple 처리방법, 핵심 Triple 추출기법, 연결노드 처리기법, 자원의 의미적 연결기법, Triple 탐색기법을 제시하였다. 4장에서는 제시된 방법론을 기반으로 시맨틱 웹 환경에서 웹 서비스를 수행할 수 있는 시스템을 제작하여 성능을 평가하였다.

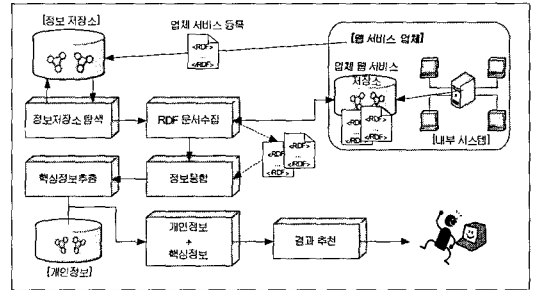
4. 성능평가

4.1 실험 환경

웹 관련 기술 연구자들 사이에 시맨틱 웹은 뜨거운 이슈로 자리 잡고 있다. 많은 분야에서 다양한 기술들을 시맨틱 웹에 적용하는 중이지만 시맨틱 웹이 현실화되기 위해서는 아직 많은 기술적인 부분이 해소되어야 한다. 그래서 본 논문은 가상의 시맨틱 웹 환경을 구성하여 제안된 기법들을 검증하였다. 시스템은 Linux 커널 2.4 와 SunOS 5.8 기반으로 구현되었다. 웹 서버는 아파치 서버 2.0을 사용하였고 사용 언어는 PHP, Java를 이용하였다. 그리고 온톨로지 언어

로 RDF, RDFS를 사용하였다. 구성된 시스템 사용자가 “인천에서 파리로 가는 비행기 티켓을 구매할 수 있는 웹 서비스를 찾아주세요”라는 질의를 바탕으로 테스트 되었다. 그림 7은 전체적인 시스템의 구성을 나타내고 있다.

웹 서비스를 원하는 기업은 기업의 서비스정



(그림 7) RDF Triple 기반 개인화 추천 시스템

보를 정보저장소에 등록하고 실질적인 웹 서비스가 이루어 질 수 있도록 자사의 네트워크에 제공될 서비스를 구축한다. 정보저장소 탐색 모듈은 정보저장소에 구축된 기업의 서비스정보를 검색하여 정보통합모듈에게 검색된 결과를 넘겨주는 역할을 수행한다. 문서수집 모듈은 해당 기업에서 제공하는 웹 서비스에 접속하여 검색된 결과와 관련된 RDF문서를 수집하고 정보통합 모듈에서 수집된 문서를 통합하게 된다. 정보검색 모듈(개인정보+핵심정보)은 핵심정보추출 모듈에서 추출된 정보와 개인의 성향 정보를 기반으로 정확한 결과를 도출해 내고 그 결과를 사용자에게 최종 추천하게 된다.

4.2 Prototype 시뮬레이션

테스트를 위해 10개의 항공권 판매 기업에 ‘AirPlane’의 하부 클래스에 설정하였다. 기업 정보 중 ‘Paris’로 가는 항공권을 판매하는 기업으로 ‘http://www.koreanair.co.kr’와 ‘http://www.flyasiana.com’으로 설정하였다. 정보 저장소

질의는 기업 정보 스키마에 있는 ‘AirPlan’ 클래스의 Subclass 정보를 가져와 각 하부에 존재하는 기업 정보 RDF Triple과 찾고자 하는 정보의 핵심 단어와 매칭시키는 방법으로 검색하였다. 즉, 테스트를 위해 가정된 질의는 ‘인천’, ‘파리’, ‘항공권’, ‘판매’에 해당되는 각 Triple을 매칭하는 방법으로 검색되었다.

```

Predicate: http://mars.chosun.ac.kr/~kiki/xmlns/pland#FLY
Object: bNode6
Statement number: 19
Subject: bNode5
Predicate: http://www.w3.org/1999/02/22-rdf-syntax-ns#type
Object: http://www.w3.org/2001/vcard-dt/3.0#work
Statement number: 20
Subject: bNode5
Predicate: http://www.w3.org/1999/02/22-rdf-syntax-ns#value
Object: +82 62 230 7756
Statement number: 21
Subject: bNode6
Predicate: http://www.w3.org/1999/02/22-rdf-syntax-ns#value
Object: tokyo
Statement number: 22
Subject: bNode6
Predicate: http://www.w3.org/1999/02/22-rdf-syntax-ns#value
Object: osaka
Statement number: 23
Subject: bNode6
Predicate: http://www.w3.org/1999/02/22-rdf-syntax-ns#value
Object: okinawa

Array
Result : http://www.koreanair.co.kr/
Result : http://flyasiana.com/
    
```

〈그림 8〉 정보저장소 탐색 모듈 처리 화면

그림 8의 상단부분은 정보 저장소 탐색 모듈이 RDF로 된 기업 정보 Triple을 해석하는 화면이고, 하단은 파리로 가는 항공권을 판매하고 있는 두 웹 서비스 URL를 결과 값으로 보여준다. 정보 검색모듈은 정보 통합모듈에 의해 통합된 RDF문서와 개인 정보 RDF 문서를 기반으로 정확한 결과를 도출하는 역할을 수행한다. “인천에서 파리로 가는 비행기 티켓을 구매할 수 있는 웹 서비스를 찾아주세요”라는 질의에는 구입하고자 하는 의사가 내포되어 있으므로, 검색 결과 중에서 사용자가 이용 가능한 웹 서비스 업체를 추천해 주어야한다. 이를 위해서, 사용자 정보와 통합된 문서의 핵심 Triple을 매칭하는 작업을 수행하게 된다. 그림 9은 개인 정보를 매칭시키지 않은 결과다. 개인 정보에는 개인이 소지한 카드가 ‘visa’이므로 두 번째 검색 결과는 사용자가 원하는 정보이기는 하지만 사용자가 소지한 카드로는 구매할 수 없다는 것을

알 수 있다.

```

Result : http://flyasiana.com/fal205
Result : fal205
Result : incheon
Result : paris
Result : 17:00
Result : 21:00
Result : 500000
Result : http://www.visa.com/visa
Result : http://www.koreanair.co.kr/ka1308
Result : ka1308
Result : incheon
Result : paris
Result : 14:00
Result : 22:00
Result : 500000
Result : http://www.master.com/master
    
```

〈그림 9〉 개인정보를 적용하지 않은 결과화면

그림 10은 개인이 가지고 있는 카드 정보와 웹 서비스 업체의 결제 카드 정보를 카드사 스키마와 연결하여 검색을 수행한 화면이다. 최종 결과 값으로 항공권 판매 웹 서비스를 수행하고 있는 URI, 비행기 번호, 출발 공항이름, 도착 공항이름, 출발 시간, 도착 시간, 항공료, 결제 가능한 카드사 이름이 출력되었다. 그림 9에서와 달리 ‘Paris’로 가는 항공권을 취급하는 업체 한 곳이 추천되었고 그 업체는 개인이 소지한 카드로 결제할 수 있는 항공권 취급 업체라는 것을 알 수 있다.

```

http://mars.chosun.ac.kr/~kiki/xmlns/schedule#CARD
http://www.visa.com/visa
Array
Result : http://flyasiana.com/fal205
Result : fal205
Result : incheon
Result : paris
Result : 17:00
Result : 21:00
Result : 500000
Result : http://www.visa.com/visa
    
```

〈그림 10〉 최종 추천된 결과 화면

4.3 Triple 증가에 따른 성능평가

기존의 연구 [1]과 [2]에서는 DAML+OIL의 특성을 이용하여 UDDI 검색기법을 제안하였다. 본 논문의 서두에서 지적했듯이 기존 연구는 UDDI 검색 시 Service Template에 정의된 입력, 출력, 제약 연산자를 사용해야 한다. 이는 사용자가 UDDI를 검색 시 정의된 연산자를 사용하여 검색을 수행해야 하기 때문에 검색에 제약이 따른다. 이런 문제점을 해결하기 위해 본

연구에서는 RDF로 정보를 구성하고 검색하는 방식을 제안하였다. 기존의 연구와 본 연구가 서로 다른 온톨로지 언어를 사용하고 있고 다른 방식으로 검색을 처리하고 있으므로 비교평가 어렵지만 다음과 같은 서로의 장단점이 존재한다.

기존의 연구에서는 웹 서비스 검색에 대한 기술을 제안하고 있다. 즉 기술적 문제제시만 존재할 뿐 실질적인 실험데이터가 존재하지 않는다. 두 연구 모두 기술을 제안하는 단계이고 폭넓

었으며 측정횟수는 동일질의에 대해 5회를 실시하였으며 결과 값으로 평균시간을 사용하였다. 초기 테스트를 위해 10곳의 기업 정보와 검색을 수행하는 검색자 개인 정보를 구축하였다. 이 모든 정보를 구성하는 Triple 수는 158개였다. 기업의 수와 검색 수행자 수는 변화시키지 않고 각 정보를 구성하는 정보의 양을 증가시켜 Triple을 증가하였다. 수행 시간은 그림 11에서 나타난 결과처럼 처음 Triple 개수가 158일 때 0.243초, 220일 때 0.311초, 500일 때 0.402초였다. Triple의 개수가 500을 기점으로 그 전까지 보였던 증가 비율 보다 다소 급격한 변화를 보였다.

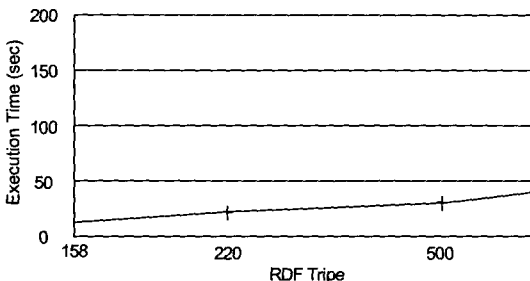
그림 12은 정보 저장소의 기업 정보 수를 증가하여 테스트한 결과다. 각 기업 정보를 구성하는 Triple의 개수를 10으로 제한하고 기업의 수를 증가시켜 정보 저장소 탐색 시간을 측정하였다. 그림 12의 결과에서 나타나듯이 60곳의 기업 정보를 구성한 이후 수행 시간이 급격하게 증가하였다. 두 결과는 정보를 가지고 있는 Triple이 다른 하위 노드와 연결되어 정보의 양이 증가하게 되면 검색 시 Triple을 해석하는데 많은 메모리 공간과 시간이 소요됨을 보여주고 있다. 또한 단순 Triple로 구성된 개인 정보 보다는 다수의 연결노드로 구성된 기업의 정보를 처리하는데 더 많은 시간이 걸림을 보여주고 있다. 연결노드의 증가에 따른 Triple 해석능력 향상은 차후 집중적으로 개선되어야 할 부분이다.

<표 7> 제안방식과 기존방식과의 비교

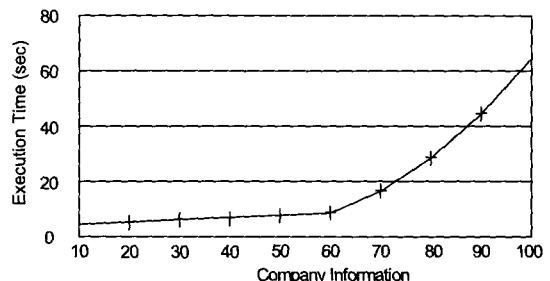
구분	사용언어	검색방식	개인정보기반 추천
기존연구	DAML+OIL	연산자 사용 핵심단어 매칭	없음
제안방식	RDF	N-Triple 해석에 따른 핵심단어 매칭	가능

은 검증 데이터가 존재하지 않기 때문에 정확도를 측정하기엔 다소 무리가 있다. 하여 본 논문에서는 Triple을 해석하고 처리하는데 걸리는 시간을 측정하였다.

기업 정보와 개인 정보를 구성하는 모든 Triple의 개수를 증가시켜 시스템 성능 변화를 측정하였고 또한 정보 저장소의 기업 수를 증가하여 그에 따른 정보 저장소 검색 시간을 측정하였다. 시간측정방법은 각 모듈들의 실행코드 시작과 끝 부분에 시간 계산함수를 두어 측정되



<그림 11> RDF Triple 증가에 따른 수행속도



<그림 12> 기업정보 증가에 따른 수행시간

5. 결론 및 향후 과제

급속도로 발전하는 웹의 진화에 따라 현재 웹에서 사용되는 방법들은 시맨틱 웹 환경에 적합한 방법으로 재구성되거나 새로운 개념에 맞게 재창조되고 있다. 기존의 XML 기반의 웹 서비스는 시맨틱 웹 환경에 맞게 온톨로지 기반으로 변화를 모색하고 있다. 그 중 DAML, DAML+OIL를 기반으로 한 연구가 진행되고 있지만 웹 서비스 검색 시 상황에 적합한 각 연산자들을 사용해야 하는 불편함과 그에 따른 각각의 온톨로지를 정의해 주어야 하는 번거로움이 존재한다. 이러한 문제점을 해결하기 위해 본 논문에서는 RDF를 사용하여 다음과 같은 방식을 제안했다.

- N-Triple 처리기법 : N-Triple로 연결된 각 노드를 탐색하기 위한 기억공간구성 및 접근 기법을 제시
- Triple 필터링 시 발생하는 연결노드 처리기법 : 다수의 연결로드로 구성된 핵심 Triple 추출 시 발생하는 각 노드들의 처리기법을 제시
- RDF Triple 통합 시 연결노드 처리기법 : RDF Triple 통합 시 Subject 및 Object가 연결노드로 되어있는 경우 연결노드와 연결된 상위 노드의 Object 및 Subject를 효율적으로 연결하기위한 처리기법을 제시
- Triple 간의 의미적 연결 및 탐색기법 : Triple의 의미적 연결방법을 제시하고 이를 탐색할 수 있는 방법을 제시
- 개인정보 기반 사용가능 결과 추천 : 개인의 정보를 기반으로 검색된 내용 중 사용자가 사용 가능한 정보를 추천하는 방법 제시

제안된 방법을 바탕으로 웹 서비스를 수행하는 시스템을 구축하여 제안된 기법의 유효성을 검증하였다. 현재 RDF, OWL, DAML+OIL 등

다양한 온톨로지 언어가 개발되고 있지만 온톨로지 언어를 처리할 수 있는 정형화된 기법은 제시되지 못하고 있다. 본 논문에서 제시한 Triple 처리기법 또한 아직 정형화 되지 않았다. 하여 차후 연구를 통해 많은 개선을 하고자 하며 현재 중점으로 개선하고 있는 부분은 다음과 같다. 각 노드들을 연결하는 관계정의 단어(vocabulary)에 따라 상의한 결과가 나온다. 즉, 동일한 어떤 것(thing)을 서로 다른 단어를 사용하여 정의할 경우 검색 시 결과 범주에 포함되지 않거나 찾지 못하는 경우를 말한다. 이 문제는 비단 본 연구만의 문제가 아닌 온톨로지 검색 전반에 걸친 문제로 대두되고 있다. 또한 핵심정보를 추출하기 위한 Triple 트리밍 단계에서 세부정보 소실이 과도하게 발생하는 경향이 있다. 찾고자 하는 핵심정보는 있지만 그 정보에 따른 부수적 세부정보가 트리밍과정에서 많은 부분이 소실되었다. 현재 상위의 문제점을 해결하기 위해 노드의 트리밍 시 트리밍 되는 노드가 가지고 있는 predicate을 이용하여 하부 연결노드검색을 시도하고 있다. 이 단계를 통해 향상된 세부정보를 추출할 수 있었지만 각 노드별 검색에 따른 과도한 하드웨어 자원 소비와 핵심정보와 무관한 정보까지 추출되는 문제점이 발생하였다. 이런 문제점은 차후 연구를 통해 집중적으로 개선되어야 할 부분이며 그에 따른 현재 연구가 진행되고 있다.

참고 문헌

- [1] Paolucci, M. et al., "Semantic Matching of Web Services Capabilities," Proceedings of the 1st International Semantic Web Conference (ISWC 2002), Lecture Notes in Computer Science, Vol. 2342, pp. 333-347, 2002.
- [2] Sivashanmugan, K. et al., "Adding Sem-

- antics to Web Services Standards”, Proceedings of the International Conference on Web Services, pp.395-401, 2003.
- [3] D. Booth, “Web Services Description Language (WSDL),” <http://www.w3.org/TR/2005/WD-wsdl20-primer-20050803/>, 2005
- [4] L. Clement and T. Rogers, “Using WSDL in a UDDI Registry,” <http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-wsdl-v2.htm>, 2004
- [5] D. Martin, M. Burstein, O. Lassila, M. Paolucci, T. Payne, and S. McIlraith, “Describing Web Services using DAML-S and WSDL,” <http://www.daml.org/services/daml-s/0.7/daml-s-wsdl.html>, 2002.
- [6] P. Hayes, “RDF Semantics”, <http://www.w3.org/TR/rdf-mt/>, W3C Recommendation, 2004.
- [7] F. Manola and E. Miller, “RDF Primer,” <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, 2004.
- [8] OntoBroker®, “How to Write F-Logic Programs,” A Tutorial for the Language F-Logic, 2004.
- [9] G. Klyne and J. J. Carroll, “Resource Description Framework (RDF): Concepts and Abstract Syntax,” <http://www.w3.org/TR/rdf-concepts/>, W3C Working Draft, 2004.
- [10] D. Beckett and B. McBride, “RDF/XML Syntax Specification,” <http://www.w3.org/W3C Recommendation>, 2004.
- [11] D. L. McGuinness and F. V. Harmelen, “OWL Web Ontology Language Overview,” <http://www.w3.org/TR/owl-features/>, W3C Recommendation, 2004.
- [12] T. Verner-Lee, J. Hendler, and O. Lassila. “The Semantic web,” Scientific American, pp.35-43, 2001.
- [13] S. A. McIlraith, T. C. Son, H. Zeng. “Semantic Web Service,” IEEE Intelligent System, Vol.17, No.1, pp.46-53, 2001.
- [14] 신호필, “지식기반(Knowledge Base)으로서의 온톨로지(Ontology)와 시맨틱 웹(Semantic Web),” 한국정보처리학회지 제 11권 제2호, pp.64-72, 2004.
- [15] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, K. Tolle. The RDFSutie: Managing Voluminous RDF Description Bases, Technical Report, ICS-FORTH 2003.
- [16] J. Hendler, T. Berners-Lee and E. Miller “Integrating Applications on the Semantic Web,” Journal of the Institute of Electrical Engineers of Japan, Vol. 122, No.10, pp. 676-680, 2002.
- [17] The DAML Service Coalition, “DAML-S: Semantic Markup for Web Services,” <http://www.daml.org/services/owl-s/1.0/owl-s.html>, 2003

● 저 자 소개 ●



정 관 호 (Jeung Kwan Ho)

2002년 조선대학교 전자계산학과 졸업(학사)
2005년 조선대학교 대학원 전자계산학과 졸업(석사)
2005년 ~ 현재 한양대학 정보통신학과 연수연구원
관심분야 : 웹 서비스(Web Service), 온톨로지(Ontology), 시맨틱 웹(Semantic Web)
E-mail : kwanho@chosun.ac.kr



김 판 국 (Kim Pan Koo)

1988년 조선대학교 컴퓨터공학과 졸업(학사)
1990년 서울대학교 대학원 컴퓨터공학과 졸업(석사)
1994년 서울대학교 대학원 컴퓨터공학과 졸업(박사)
1995년 ~ 현재 조선대학 컴퓨터공학부 교수
관심분야 : 시맨틱 웹, 온톨로지, 컴퓨터 비전, Motion & Video Processing, 컴퓨터 보안
E-mail : pkkim@chosun.ac.kr



김 권 천 (Kim Kweon Cheon)

1986년 조선대학교 의과대학 졸업(학사)
1990년 조선대학교 대학원 의학과 졸업(석사)
1999년 전남대학교 대학원 의학과 졸업(박사)
1994년 ~ 현재 조선대학 의학과 교수
관심분야 : 의료정보학, 일반의과학.
E-mail : mdckkim8@hanmail.net

