ISSN(Print): 2287-5905

http://dx.doi.org/10.3745/KTSDE.2012.1.1.019

Mobile Cloud Context-Awareness System based on Jess Inference and Semantic Web RL for Inference Cost Decline

Se-Hoon Jung† · Chun-Bo Sim††

ABSTRACT

The context aware service is the service to provide useful information to the users by recognizing surroundings around people who receive the service via computer based on computing and communication, and by conducting self-decision. But CAS(Context Awareness System) shows the weak point of small-scale context awareness processing capacity due to restricted mobile function under the current mobile environment, memory space, and inference cost increment. In this paper, we propose a mobile cloud context system with using Google App Engine based on PaaS(Platform as a Service) in order to get context service in various mobile devices without any subordination to any specific platform. Inference design method of the proposed system makes use of knowledge-based framework with semantic inference that is presented by SWRL rule and OWL ontology and Jess with rule-based inference engine. As well as, it is intended to shorten the context service reasoning time with mapping the regular reasoning of SWRL to Jess reasoning engine by connecting the values such as Class, Property and Individual which are regular information in the form of SWRL to Jess reasoning engine via JessTab plug-in in order to overcome the demerit of queries reasoning method of SparQL in semantic search which is a previous reasoning method

Keywords : Semantic Web, Jess, OWL, SWRL, Context-Awareness, Mobile Cloud

1. 서 론

상황인식 서비스는 현재 유비쿼터스 시대와 맞물려 사용자의 다양한 모바일기기에서 의료, 교육, 재난, 구호, 쇼핑, 개인 비서등과 같이 매우 광범위한 서비스를 사용자에게 제공하고 있다. 상황인식 서비스는 컴퓨터가 사용자의 생각을 미리 확인할 수 없지만 사용자의 주변 상황을 인식해서 사용자의 의도에 적합한 수많은 데이터를 사용자에게 제공할 수 있는 상황인식 시스템을 지칭하고 있다[1]. 상황인식 시스템은 유비쿼터스 컴퓨팅의 한 분야이다. 유비쿼터스 컴퓨팅의 목적은 1, 2차원 모든 공간 및 환경 그리고 사물들에 컴퓨터가 융합되고, 사물은 융합되어진 컴퓨터를 통하여 네트워크의 단말로 인식되며 현실보다 더 민감한 상황을 파악할 수 있고 더욱 강력한 제어를 수행할 수 있는 공간의 형상화 기술인 AR(Augmented Reality)과 공간형태 변화인식의 고도화를 위한 CA(Context Awareness)컴퓨팅과 같은 방법을 통하여 사용자로 하여금 자의적으로 컴퓨팅 능력을 발휘할 수 있도록 모든 업무를 한 자리에서 해결할 수 있게 해주는 것이다. 최근 이러한 상황인식 서비스는 IT 기술의 개발과 더불어 인간의 질적인 변화를 더욱 더 향상 시킬 것으로 예상되며, 현재 수많은 연구가 국내·외적으로 진행중이다[2-7].

기존 연구의 상황인식 시스템은 단일 형태의 온톨로지와 규칙 추론 기반의 Rule-ML로 상황정보 추론 방식을 대부분 활용하였다[8]. 하지만 이러한 추론 방식은 추론엔진이 시스템에 존재하지 않아서 외부의 상황 추론 엔진으로 지식베이스에 포함된 Fact와 Rule을 모듈을 통해서 보내는 추론 결과를 리턴 받는 형태로 구성되는 단점이 있다. 이는 상황인식 서비스의 상황 추론에 대한 불필요한 응답지연 시간을 포함하게 되는 문제점으로 추론 비용을 증가시키는 근본적인 원인이기도 하다. 또한 상황인식 시스템의 추론 비용 증가에 대한 다른 원인은 기존 상황인식 서비스가 특정 플랫폼에 종속적으로 배치되어 특정 모바일기기 아닌 불특정 다수의 모바일기기에서 사용할 수 없는 불편함이 추론 비용 증가에 대한 단점으로 지적되고 있다. 이러한 단점은 추론 비용 증가와 현재 다양한 플랫폼으로 출시되고 있는 테블릿 PC나 스마트폰에 대한 시스템의 확장성과 이기종간의 데이터 송·수신에 대한 시스템 신뢰성은 현저히 떨어뜨리는 단점이 지적되고 있다[9]. 그리고 현재의 모바일기기에서 연구되고 있는 상황인식 서비스는 제한된 모바일 환경과 메모리 공간 및 데이터 처리 능력으로 인하여 소규모 상황 인식 추론 및 단일 추론 시스템을 연구하고 있는 실정이다[8]. 이에 본 논문에서는 특정 플랫폼에 종속되지 않고 제한되지 않는 메모리와 데이터 처리 능력을 구축하여 대규모 상황 추론 서비스가 가능하며, 다양한 모바일기기에서 상황인식 서비스를 제공받을 수 있는 모바일 클라우드 방식을 제안한다. 현재 클라우드는 다양한 서비스 형태로 개발되고 있지만 본 논문에서는 PaaS(Platform as a Service)기반의 GAE (Google App Engine)을 이용한 모바일 클라우드 상황인식

† 준 회 원 : 순천대학교 멀티미디어공학과 박사과정
†† 정 회 원 : 순천대학교 멀티미디어공학과 부교수
논문접수 : 2012년 6월 7일
수 정 일 : 1차 2012년 8월 10일
심사완료 : 2012년 8월 16일
* Corresponding Author : Chun-Bo Sim(cbsim@sunchon.ac.kr)

시스템을 제안한다. 제안하는 시스템은 특정 플랫폼의 종속성을 배제하여 이기종간의 데이터 송수신에도 상황 데이터 추론 비용을 단축시킬 수 있다. 또한 시스템의 추론 설계 방식은 OWL(Ontology Web Language)의 온톨로지와 SWRL(Semantic Web Rule Language) 규칙으로 표현되는 시멘틱 추론을 이용한 지식베이스 프레임워크와 규칙 기반의 추론 엔진을 제공하는 Jess(Java expert system shell)를 활용하여 설계한다. 그리고 기존 추론 질의 방식인 시멘틱 검색의 SparQL 질의 추론 방식의 단점을 극복하고자 SWRL형태의 Rule 규칙 정보인 Class, Property, Individual 등의 속성값들을 특정 플러그인을 이용하여 Jess 추론 엔진에 연결하도록 설계하였다. 추론 질의 데이터인 SWRL의 규칙 추론은 Jess 추론 엔진에 매핑한다. 지식베이스 프레임워크와 규칙기반의 추론 엔진을 연결할 수 있는 특정 플러그인은 JessTab을 활용하여 상호 보완적인 관계를 유지한다. 이는 기존 연구 방식에 비해 상황인식의 서비스 추론 시간을 단축하는 장점을 제공한다. 아울러, 모바일 클라우드 기반의 상황인식 시스템 구축을 위해 S/W 설계를 객체지향 설계 모델링을 기준으로 설계 결과 산출물들을 제시한다. 이는 S/W 구현 후 발생하는 시스템의 확장성과 재사용성을 극대화하는 장점을 제공하고 있다.

2. 관련 연구

제안하는 모바일 클라우드 상황인식 시스템의 목적은 크게 2가지로 구분한다. 상황인식 추론 비용 감소와 특정 플랫폼에 종속적이지 않은 프로세스 구축이다. 이번 장에서는 기존에 연구된 몇가지의 상황인식 시스템에 대한 선행 연구를 제시한다.

[2]의 연구는 Singapore National University에서는 OSGi를 기반으로 온톨로지를 이용한 상황인식 시스템인 SOCAM (Service-Oriented Context Awareness Middleware)을 설계 및 개발하였다. 해당 연구는 시스템에서 다양한 컨텍스트의 수집과 패턴 분석, 접근을 위한 효과적인 미들웨어를 제공한다. SOCAM는 OWL기반의 2계층 컨텍스트 온톨로지를 이용하여 추론 입력 데이터의 의미기반 컨텍스트 표현과 컨텍스트 추론 및 지식 공유, 컨텍스트 분류와 컨텍스트 상호 의존성에 관하여 연구 문제를 다루고 있다.

[5]의 연구는 Italy의 SMDC Lab of Udine University에서 제안한 시스템이다. 제안된 시스템은 스마트폰에 장착된 센서들을 이용하여 사용자의 원시 리소스를 활용하여 상황을 인식하고 인식된 패턴을 웹을 통하여 패턴 분석 후 추론 데이터 검색을 하게 된다. 그리고 인식된 패턴 분류를 통해 사용자에게 맞는 정보를 추출하여 결과를 사용자에게 보여주는 시스템인 CAB(Context Awareness Browser)를 제안하였다. CAB는 스마트폰 센서로부터 들어온 원시 데이터로부터 규칙 추론기법을 통하여 사용자의 상황정보를 추출하고 베이지안 네트워크 기술을 사용하여 추출된 상황정보로부터 현재 제공 가능한 서비스를 추론한다. CAB는 최적의 상황인식 시스템을 제공하고자 분석된 패턴을 기반으로 검색 엔진으로부터 가능한 서비스를 브라우저에 표현하므로 사용자가 특정 서비스를 요청하지 않아도 된다는 장점이 있다.

[6]의 연구는 숭실대학교에서 연구된 상황 추론 시스템이다. 해당 연구에서는 모바일이라는 제한된 환경에서 상황정보 분석을 위한 원시 데이터의 수집 및 가공을 위하여 일차 술어 논리기반의 추론 엔진을 활용한 정보추천 시스템을 구현하였다. 원시 데이터의 1차적으로 하위 추론을 통해 사용자에게 추천 가능한 정보만을 메모리에 로드하고 전위 추론을 통해 실제 사용자 GUI(Graphic User Interface)에게 보여줄 상황정보 데이터를 추가한다. 또한 제안된 시스템은 상황정보 제공자, 상황정보 소비자, 상황정보 중계자로 구분하고 있고 상황정보 제공자는 모바일 디바이스를 통해 인식되는 원시 리소스를 추론엔진에서 처리가 가능하도록 전처리과정을 거쳐 상황정보 중계자에게 전달하게 된다. 전달된 원시 리소스를 상황정보 중계자에서는 현재 확인된 리소스를 토대로 최적의 서비스를 찾기 위해 추론 엔진을 이용하여 일차 술어 논리 기반의 상황인식 데이터를 추출한다. 상황정보 소비자는 추론된 리소스를 사용자에게 제공할 최적의 서비스로 제공하는 단계이다.

기존 상황인식 시스템은 모두 상황모델 분석 언어로 온톨로지를 이용하였지만 추론 규칙은 상하 추론, 베이지안 네트워크 등 조금은 서로 다르게 구현되었다. 사용자 정보가 중요시 되는 모바일 처리 시스템인 CAB, 모바일 추천 시스템은 보안과 관련된 연구는 전무하였다. 또한 CAB는 획득된 추론 데이터를 웹으로 전송한 후 패턴 비교 및 데이터 분석을 하기 위한 검색엔진이 따로 필요하였고, 데이터를 XML(eXtensible Markup Language) 형태로 변환하고 검색한 후 사용자에게 정보를 제공하는 시간이 데이터의 양에 따라 매우 길어져 추론 비용의 증가를 초래하였다.

3. 제안하는 시스템

이번 장에서는 제안하는 상황인식 시스템의 객체지향 모델링 결과물을 바탕으로 시스템의 설계를 제시한다.

3.1 시스템 개요

Fig. 1은 제안하는 시스템의 전체 흐름도이다. 제안된 시스템의 내부는 클라우드상에 존재하게 되고 상황인식 서비스를 제공을 받기 위해 사용자의 GPS값과 선호도를 상황인식 시스템에 자동 업로드 된다. 클라우드에서는 상황인식 추론을 위해 데이터스토어에 저장된 상황정보 리소스를 바탕으로 Jess 추론 엔진을 활용하여 상황 정보 추론을 처리한다. 추론을 위해 미리 정의된 RDF/OWL 및 RL기반의 시멘틱 검색을 실행하며, 시멘틱 검색은 SWRL의 규칙 추론을 바탕으로 상황정보를 추론하게 된다. 추론된 결과는 다시 데이터스토어에 저장되고 NoSQL의 질의를 통해 모바일 클라우드 기반의 상황인식 시스템을 실행하게 된다.

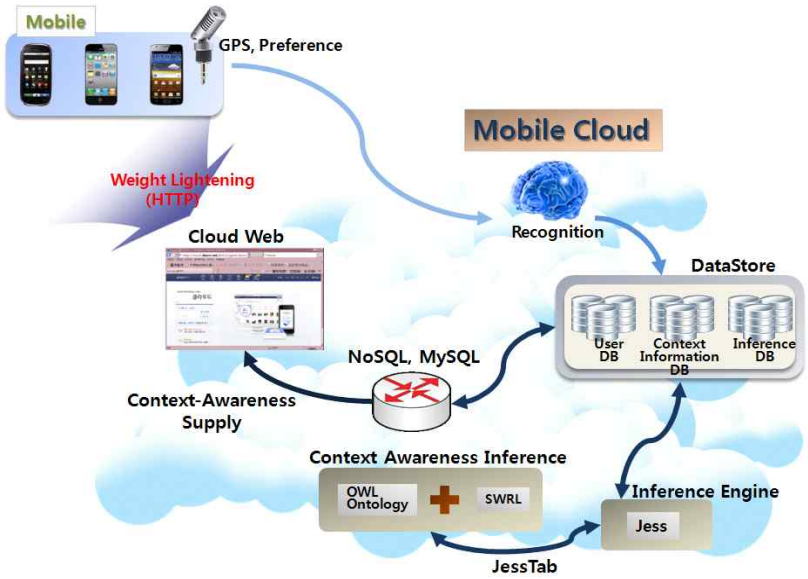

Fig. 1. Overall flow chart of the Proposed context-awareness system

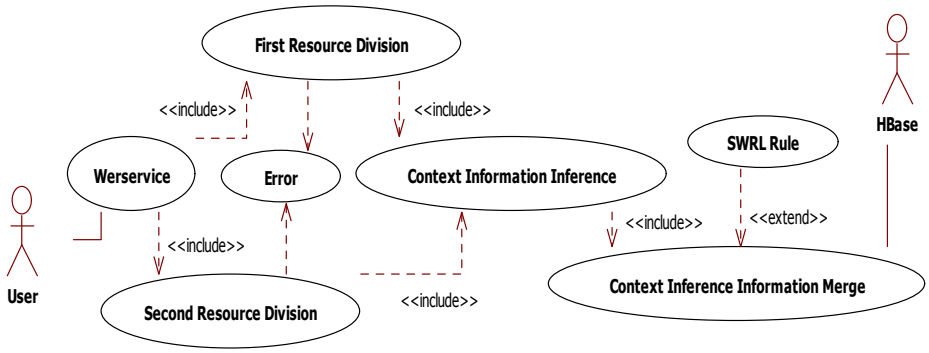

Fig. 2. Usecase diagram of inference data process

3.2 객체지향 설계

Fig. 2는 상황인식 시스템의 상황인식 리소스의 처리에 대한 요구사항을 정의하고 기능에 해당하는 부분에 대해서 유스케이스로 정의한 유스케이스 다이어그램이다. 사용자의 디바이스의 GPS 및 메모와 같은 사용자 정보를 바탕으로 상황정보 리소스의 분배 과정을 거친다. 리소스 분배 과정은 사용자의 정보를 추론 과정을 보내기 전에 불필요한 정보를 제거하고 필요한 부분만 추론 과정으로 보내는 과정인 전처리 과정을 말한다. 분배 과정 후 정보 추론을 위한 SWRL규칙과의 비교를 통해 사용자에게 제공될 정보를 추론하게 되고 각각 추론된 정보를 마지막으로 통합하게 된다. 통합된 상황추론 정보는 가상 서버에 존재하는 빅테이블 데이터스토어의 데이터베이스에 저장하게 된다. 만약 상황 정보 리소스를 처리하지 못한다면 추론 과정은 종료하게 된다.

Fig. 3은 본 논문에서 제안한 상황인식 정보 처리 과정을 오퍼레이션 중심의 알고리즘 형태로 표현한 액티비티 다이어그램이다. 유스케이스에서 구분된 각각의 객체에서 *Context Information Send, First Context Resource*

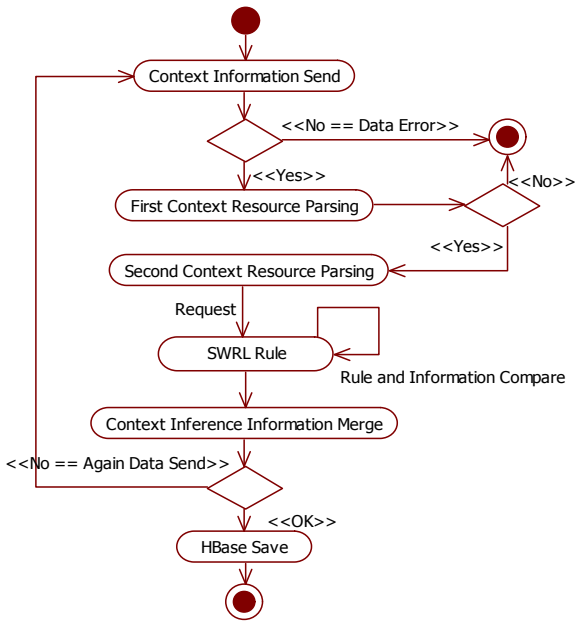

Fig. 3. Activity diagram of inference data process

*Parsing, Context Inference Information Merge*의 오퍼레이션에서의 시스템 흐름을 제어하는 선택 분기문을 삽입하여 시스템의 제어를 추가한다. *SWRLRule* 클래스의 오퍼레이션을 호출하게 되고 정의된 상황정보 규칙을 비교처리하는 *InferenceDataCompare* 오퍼레이션을 통해 데이터를 처리하고 빅테이블 데이터스토어에 저장되게 된다.

Fig. 4는 Fig. 2 유스케이스 다이어그램의 유스케이스를 정적으로 표현한 클래스 다이어그램이다. 모바일 클라우드 기반으로 설계된 상황인식 시스템은 클라이언트측엔 실행파일이 존재하지 않는다.

이에 본 설계에서는 클라이언트측의 프로그램 실행은 진행되지 않으며 단순히 상황정보 접속과 추론정보 제공만을 받기 때문에 디바이스 설계는 *intro* 클래스와 *Context Awareness* 클래스로만 설계하였다. 상황정보 전송을 위한 설계 클래스 전개는 *Webservice* 클래스를 통해 이루어진다. *Webservice* 클래스의 *request()* 메서드를 통해 사용자의 GPS와 같은 상황정보 데이터를 연결한 후 획득한 데이터를 파싱하기 위해 *DataConvert* 클래스로 보낸다. 1, 2차 상황정보 데이터 파싱을 마치고 *DataConvert* 클래스로부터 획득한 상황정보 리소스는 *SWRLRule* 클래스의 *getSWRLProcess()* 메서드를 통해 상황인식 정보 추론을 위한 단계로 진행하게 된다.

Fig. 5는 제안하는 시스템의 상황 처리부분에 관하여 실제 구축된 컴포넌트의 배치를 나타낸 컴포넌트 다이어그램이다. *CWebservice.java*의 컴포넌트와 인터페이스를 통해 상황정보의 획득과 전처리 및 추론을 위한 *CResource Inference .class* 및 *CRequestSWRLRule.class* 인터페이스를 활용하게 된다. 상황정보 추론을 통해 얻어진 데이터는 *CContextInforma tionInference.java*를 통해 가공된다. 또한 *CContextInference InformationMerge.class*의 인터페이스를 통해 가공된 상황정보 추론 데이터의 통합과 *HBase Funtion.java*를 통해 추론 데이터를 데이터베이스에 저장된다.

3.3 상황 데이터 추론 설계

1) 의미기반 OWL 설계

온톨로지는 공유하는 개념화의 형식적이고 명확한 명세를 의미하며, “합의된 지식”을 표현해야 하는 것이다[10]. 합의된 지식이라는 의미는 몇몇 개인이 임의로 정한 것이 아니라 관련된 모든 구성원의 동의에 의해 수용되는 개념과 개념사이의 관계를 표현한 지식을 말한다. 그리고 OWL 온톨로지는 XML, RDF(Resource Description Framework) 보다 더 많은 의미 표현 수단을 제공하므로, 웹 상에서 컴퓨터가 해석할 수 있는 콘텐츠를 작성하는데 있어 이들 언어보다 뛰어나다.

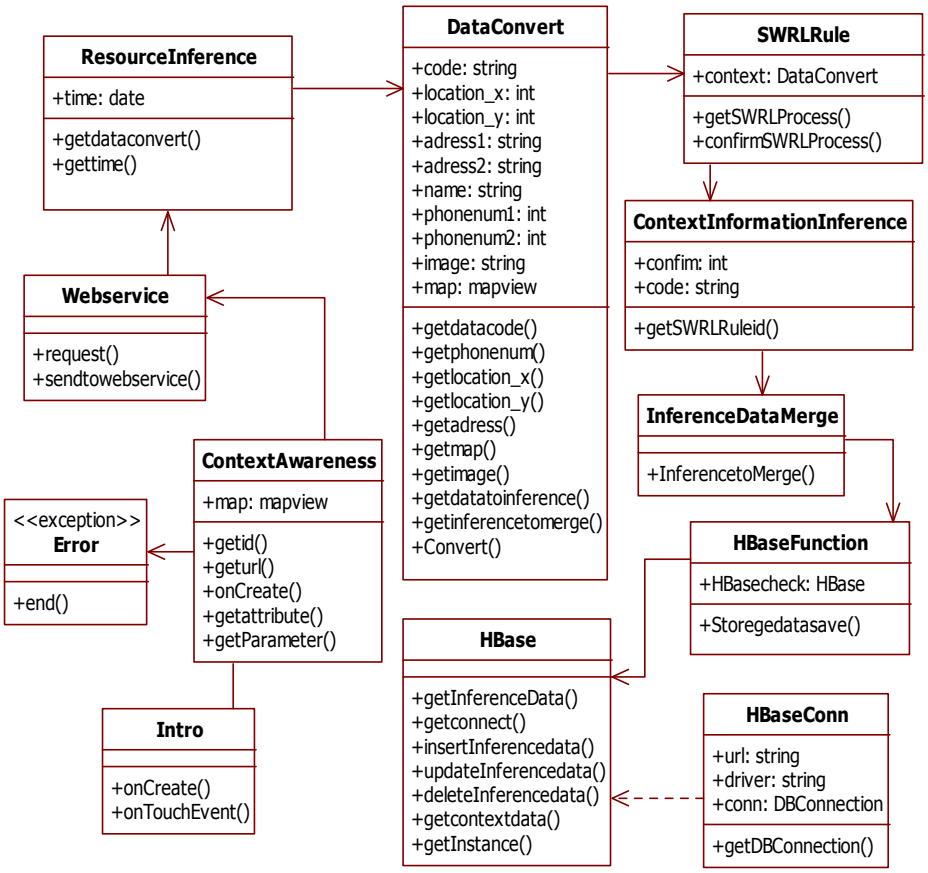

Fig. 4. Class diagram of inference data process

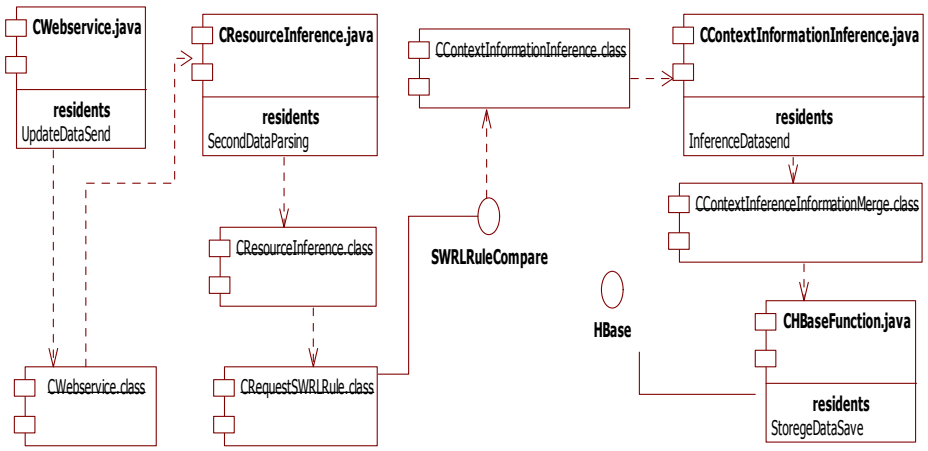

Fig. 5. Component diagram of inference data process

Fig. 6은 제안하는 시스템 추론의 OWL의 *Phone* 클래스 구조를 설계한 것이다. 클래스간의 계층 구조를 통해 클래스의 배열을 결정하였다. 하지만 클래스간의 계층 구조로는 클래스의 내부 구조를 작성하기엔 무리가 있다. 그래서 본 단계에서는 클래스 슬롯과 구조를 결정한다. 이를 위해 클래스간의 관계도와 상속관계 모델 및 도메인과의 관계 목록 등을 정의한다. 클래스간의 관계도는 클래스 슬롯의 상속 개념을 확인할 수 있다. Fig. 6은 *Phone* 클래스 중심의 다중 클래스간의 클래스 관계도이다. 하위 클래스는 상위 클래스가 포함하고 있는 상위 클래스의 속성 모두를 상속받게 된다. *Phone* 클래스를 상위 클래스로 가지는 모든 하위 클래스들은 *Phone* 클래스의 슬롯인 폰번호를 모두 상속 받게 된다. 이러한 이유 때문에 슬롯은 속성을 가질 수 있는 가

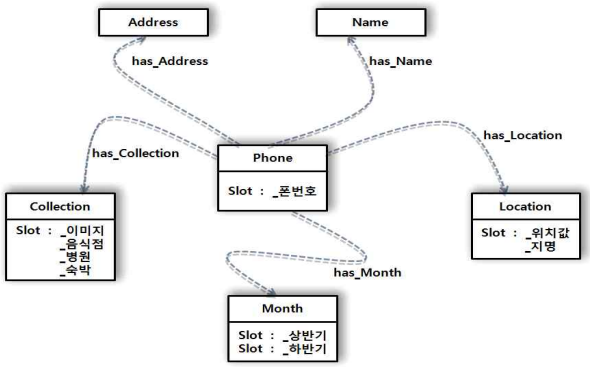

Fig. 6. OWL class diagram of Phone class

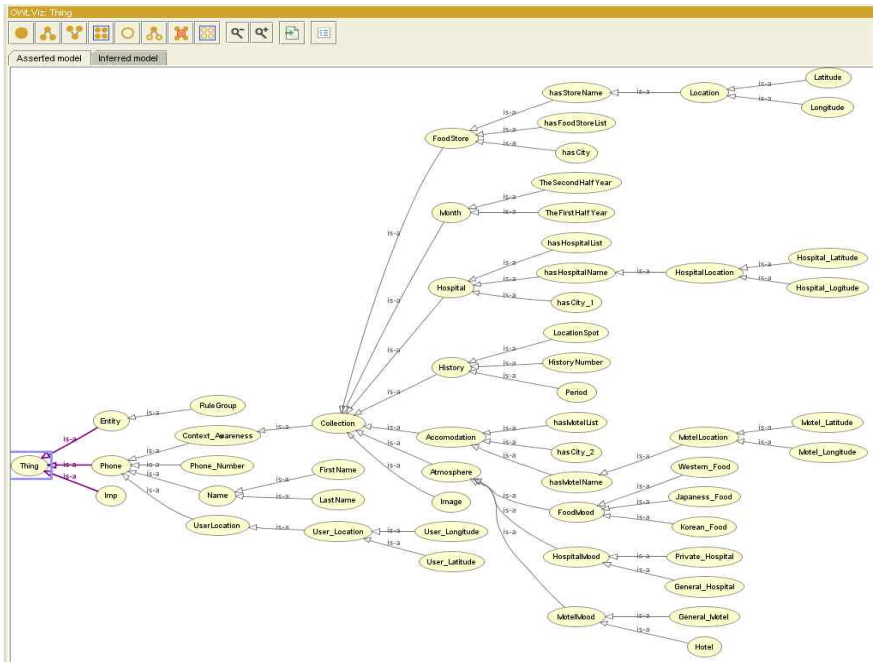

Fig. 7. Design of context-awareness class using Protege

장 일반적인 클래스에 설계하였다. 즉, *Collection, Month, Location* 클래스는 상위 클래스인 *Phone* 클래스의 하위 클래스로 슬롯을 상속 받는다.

Fig. 7은 OWL을 정의하기 위한 지식 및 규칙 베이스 기반의 저작도구인 Protege의 툴을 이용하여 온톨로지 클래스를 설계한 화면이다. Fig. 7을 통해 인스턴스간의 관계와 인스턴스간의 처리 구조 등을 처리한다.

antecedent :(전제조건)

Phone(?x1) ^ (?x1 UserLocation ?s1,?s2,?s3) ^ (?s1 User_Location?s2,s3) ^ (?s2 User_Longitude ?x2) ^ (?s3 User_Latitude ?x3) ^ (?y1 Context_Awareness ?x1) ^ (?y2 Collection) ^ (?y4 Atmosphere ?p1) ^ (?p2 FoodMood p3) ^ (?p4 Westen_Food ?p4) ^ (?1f FoodStore ?m1) ^ (?m1 hasCity) ^ (?1f hasFoodStoreList?db) ^ (?db hasStoreName ?i) ^ (?i Location ?z1) ^ (?z1 Longitude ?cs1) ^ (?z1 Latitude ?cs2)

⇒ consequent :(조건 결과)

(?x1 Phone ?lf) ^ (?lf :FoodStore ?sf1) ^ (?sf1 :Locaton ?sm1) ^ (?sm1 :Latitude ?df1)^(?sm1 :Longitude ?df2)

Fig. 8. SWRL rule inference for context-awareness inference

```
<ruleml:_rlab ruleml:href="FoodStore"/>
<ruleml:_body rdf:parseType="Collection">
 <swrl:ClassAtom>
  <swrlx:individualPropertyAtom
                              swrlx:property="hasCity">
   <ruleml:var>x1</ruleml:var>
   <ruleml:var>x2</ruleml:var>
  </swrlx:individualPropertyAtom>
  <swrlx:individualPropertyAtom
                          swrlx:property="hasFoodStoreList">
   <ruleml:var>x2</ruleml:var>
   <ruleml:var>x3</ruleml:var>
  </swrlx:individualPropertyAtom>
</ruleml:_body>
</swrl:ClassAtom>
<ruleml:_head rdf:parseType="Collection">
 <swrl:ClassAtom>
  <swrlx:individualPropertyAtom
                           swrlx:property="hasFoodStoreName">
   <ruleml:var>x1</ruleml:var>
   <ruleml:var>x3</ruleml:var>
  </swrlx:individualPropertyAtom>
 </ruleml:_head>
</swrl:ClassAtom>
```

Fig. 9. SWRL rule expression for restaurant recommendation inference

2) 규칙기반의 SWRL 설계

SWRL은 OWL의 하부 언어인 OWL DL 및 RuleML의 하부 언어인 Unary&Binary Datalog RuleML을 결합한 통합 규칙 표현 언어이다[11]. SWRL의 규칙은 머리 부분에 해당하는 conseq uent(결론)와 몸체 부분에 해당하는 antecedent(조건) 사이에 포함된 의미적 형태를 추론하는 것이다. Fig. 8은 제안하는 시스템의 규칙 추론의 일부분이며 사용자의 위도, 경도, 고도의 데이터의 값을 전제조건으로 설정하고 미리 설정된 사용자의 선호 음식을 설정하여 최종적인 결과 규칙을 추론엔진에 저장하게 된다.

Fig. 9는 SWRL 구문을 RuleML 규칙에 관한 속성을 정의한 것이다. 결과적으로 음식점 인식은 두 가지 형태로 선언된 클래스에 만족하는 새로운 클래스의 추론을 생성하는 구조를 설명하고 있다. *<swrl:AtomList>*로 선언된 *FoodStore*은 *City, Food StoreList* 클래스를 통해 *hasFoodStoreName*이라는 새로운 클래스를 도출하고 도출된 클래스와 *Location*을 통해 *hasFood StoreList*이라는 새로운 클래스 속성을 도출한다.

Fig. 10은 상황 추론 규칙에 관한 SWRL의 최종적인 XML 형태의 규칙 설계 화면이다.

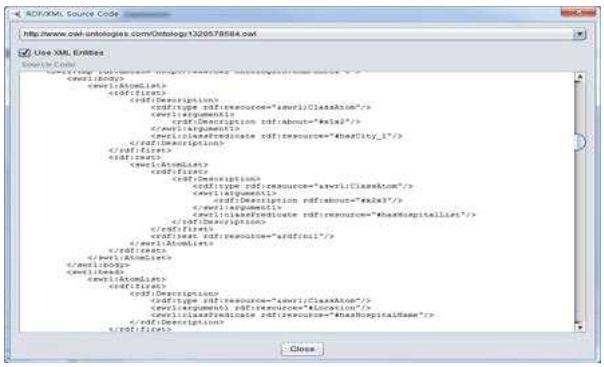

Fig. 10. Rule design screen of finally SWRL(XML)

본 논문에서 제안하는 파일의 형태는 RDF/XML 형태의 소스 코드를 표현하였다. 이를 통해 최종적인 규칙의 관한 SWRL과 RDF형태의 속성구조를 추론 파서 형식에 맞도록 Class, Property, Individual 형식의 구문 규칙을 제공할 수 있다. 이와 같은 규칙 설계는 Jess 추론 엔진에서 규칙 추론 호출에 대한 빠른 응답을 제공할 수 있다. 기존 연구 방식인 Subject, Object, Predicate 속성에 대해 1차 추론과 속성값의 매핑을 통한 2차 추론을 통해 제공되는 단점을 규칙 기반의 구문 형식의 연속적인 매핑으로 상황정보를 처리할 수 있는 장점을 제공한다.

Fig. 11은 시스템의 추론 저작 도구인 Protege를 활용하여 SWRL 규칙과 매핑한 것이다. 이를 통해 각 규칙에 정의 표현식을 비교하여 상황인식 규칙의 결과값을 최종적으로 추론 엔진에 반환하게 된다.

위와 같은 SWRL은 Horn 논리와 비슷한 구문으로 규칙을 사용자가 표현하기 쉬운 상위 레벨의 구문으로 작성할 수 있도록 하였고 모든 규칙은 OWL 개념들인 *Class, Property, Individual*의 용어로 표현한다. 이러한 형태를 통하여 설계된 규칙 추론은 상황인식 분석을 위한 온톨로지

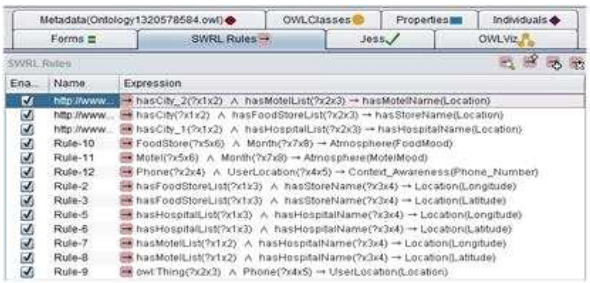

Fig. 11. Protege and SWRL rule connection(Protege)

추론 설계 중 사용자에게 제공될 사용자 추론 서비스에만 활용한다. 다른 부분인 위치정보 분석은 Java을 이용한 규칙 기반의 추론 엔진인 Jess를 활용한다.

3) Jess(Java Expert System Shell) 설계

Jess 추론엔진은 실행엔진, 정보 베이스, 규칙 베이스로 구성되며, 실행엔진에는 규칙 베이스에 있는 정해진 규칙을 통해 정보 베이스에 있는 사실정보들을 비교분석한 결과로 작동한다. 규칙들은 새로운 정보를 생성하고 정보 베이스에 저장하거나 Java 기능을 실행한다. Fig. 12는 Jess의 기본 구조를 도식화 하였다. 구성은 다음과 같다.

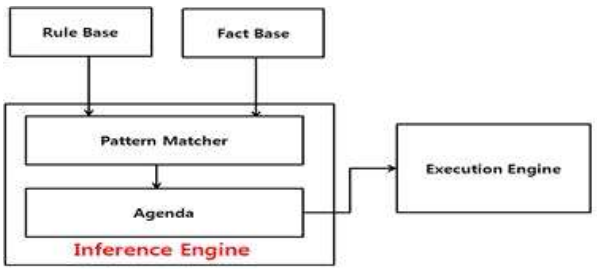

Fig. 12. Process flow chart of Jess inference system

Rule Base는 시스템에 저장한 상황인식에 대한 모든 규칙을 저장하고 있다. 예를 들어 서비스를 제공하기 위해 사용자의 정보로부터 획득한 리소스의 추론을 위한 쿼리와 기능에 관한 규칙의 저장소로 설명할 수 있다. Fact Base는 Jess 시스템에서 작업을 수행해야할 상황정보 제공을 위한 상황정보 리소스의 데이터 보관소이다. 이는 사용자의 상황정보라고도 할 수 있다. Patter Matcher는 Rule Base의 규칙 정보와 Fact Base의 상황 정보의 매칭을 통해 알맞은 정보를 추출하는 단계이다. Agenda는 추론된 정보간의 혼동 문제를 해결하기 위한 논리적 수행 작업에 대한 규칙을 저장하는 단계이며, 이는 똑같은 위치정보 추론 상황에 대한 서비스 제공시 중복되는 부분에 대한 우선 결정권을 정할 수 있는 단계이다. 마지막으로 Execution Engine은 추론 정보를 사용자의 요구사항에 맞도록 최적화 및 상황정보 결과값의 정형화를 추진하는 단계이다. 추론 규칙에 요소가 될 상황인식 정보는 기본 베이스 원소로 이루어지며, 기본 베이스 원소는 Rule Base와 Fact Base의 형태로 존재하게 된다. Jess의 Rule Base는 Java의 Jess 툴을 이용해 생성되는 지식 기반 인스턴스를 JessTab을 이용하여 SWRL에서 파서된 규칙과 패턴 매칭을 통해 매핑하게 된다. Rule Base에 매핑된 인스턴스들은 규칙들의 집합으로 명명된다.

```
(deftemplate FoodStore
"Context-Awareness Recomendation"
(declare (slot Latitude(INTEGER))
         (slot Longitude(INTEGER))
         (slot hasCity(STRING))
         (slot DetailAddress(STRING))
         (slot Store(default Food()))
         (slot hasStoreName(STRING))))

(defrule find-UserLocation
;; User of Locatoion(Latitude, Longitude) is
                                    Maegokdong in
  suncheon(FoodStoreName).
  hasRecomLatitude(of User)(is ? a1)
  hasRecomLongitude(of User)(is ? a1)
  hasCity(of User)(is ? b1)
  hasRecomDetailAddress(of User)(is ? c1)
  hasStoreName(of User)(is ? d1)
)
```

Fig. 13. Jess rule define for location trace using location value

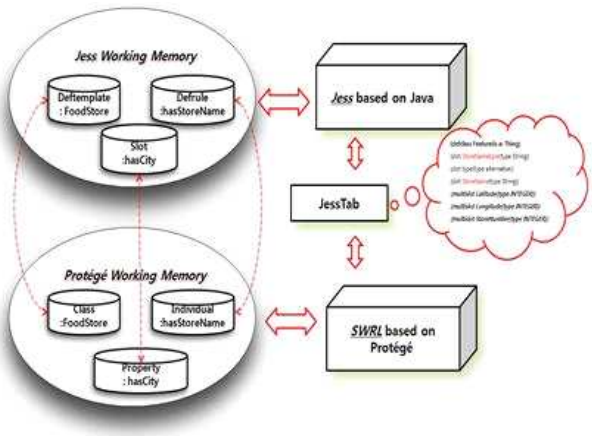

Fig. 14. Proposed system JessTab mapping map diagram

Fig. 13은 상황인식 서비스의 위치값을 통한 현재 사용자 지역의 정보 추론을 위한 *deftemplate, defrule*의 규칙 생성에 관한 코드이다. 상황정보를 추론하기 위해 매핑된 규칙 생성 코드는 Java의 Jess를 기반으로 작성된다.

그러나 Jess 규칙으로 정의된 클래스 정보는 위치정보의 분석을 위해 활용되므로 사용자 추천 서비스를 위한 상황정보 추론은 Protege로 정의된 SWRL과의 연결을 설정해야하며, 본 논문에서는 JessTab 플러그인을 통해 이 부분을 해결하였다. 즉, Jess를 통해 정의된 인스턴스의 규칙 정보를 Protege로의 SWRL과의 연결을 표현하기 위해 JessTab의 클래스 연결을 통해 해결할 수 있다.

Fig. 14는 이러한 연결 기능을 가지는 JessTab의 구조 및 역할과 제안하는 시스템의 추론 기능의 연결된 부분을 표현하고 있다. JessTab은 Jess와 SWRL사이의 구조 맵핑을 처리하였으며, SWRL의 *Class*와 *Jess*의 *deftemplate*는 *FoodStore*로 서로 매핑된다. 또한 SWRL의 *Individual*과 Jess의 *defrule*는 *hasStoreName*으로, SWRL의 *Property*와 Jess의 *slot*은 *hasCity*로 서로 매핑 처리된다.

4) 모바일 클라우드 설계

본 논문에서 제안하는 시스템은 클라우드 서비스 전달 방식의 일부인 PaaS(Platform as a Service)방식을 이용하여 사용자의 모바일기기에 별도의 프로그램 설치 과정 없이 상황인식 애플리케이션을 네트워크로 이용하여 서비스를 제공받을 수 있도록 설계한다. 이를 위해 본 논문에서는 모바일 클라우드의 설계 및 구현을 구글에서 제공하는 Google App Engine(GAE)을 활용한 상황인식 클라우드를 구축하였고, 다양한 사용자와 다양한 플랫폼에서의 접속을 통한 상황인식 서비스 처리를 위해 클라이언트기반의 애플리케이션을 설치하는 방식이 아닌 웹 애플리케이션 서버에 관련 시스템을 설치하여 데이터를 처리할 수 있도록 하였다.

Fig. 15는 제안하는 시스템의 모바일 클라우드 가상서버의 아키텍처 구성도이다. 본 논문에 활용되는 GAE의 아키텍처 구조는 여러 애플리케이션과의 리소스 공유는 가능하지만 빅테이블 데이터스토어나 보안과 관련된 애플리케이션을 다른 애플리케이션과 격리되는 특징으로 구성되어 있다. 이러한 특징을 이용하기 위해 사용자는 GAE의 상황인식 서비스 호출을 위해 모바일기기의 웹서비스인 HTTP를 활용하게 된다.

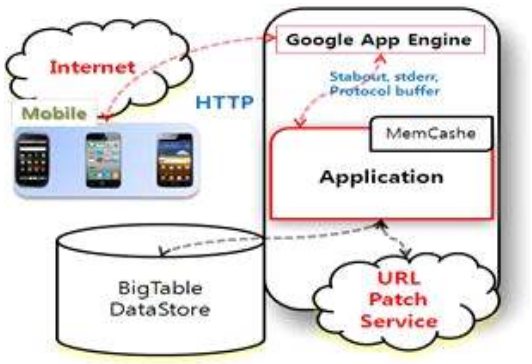

Fig. 15. Mobile cloud(GAE) architecture block diagram

이를 기반으로 상황인식에 관한 상황정보 리소스와 추론 제공 데이터의 저장소인 데이터베이스는 빅테이블 데이터스토어에 저장되며 NoSQL을 사용해 데이터스토어에 질의를 한다. 또한 시스템 처리 중 특정 URL의 내용을 확인하기 위해서는 애플리케이션이 직접 통신 포트를 확인하는 URL Patch Service를 이용하게 된다. 또한 구글에서 제공하는 GAE의 데이터베이스는 관계형 데이터베이스와 비관계형 데이터베이스를 모두 활용할 수 있지만 본 논문에서는 데이터 클라우드 방식을 유지하고 현재 구글에서 발표한 관계형 데이터베이스 활용이 MySQL로 한정되어 있어 본 시스템의 서비스 구현은 NoSQL의 DBMS를 이용한다.

Fig. 16은 본 논문에서 활용된 NoSQL 중 빅테이블 데이터스토어인 데이터베이스에 키값을 구분으로 로그인 데이터를 저장하는 코드이다.

사용자 로그인시 로그인 ID를 키값으로 지정하고 나머지 데이터를 속성값으로 구분하여 User 맵을 생성 및 업로드하게 된다. 상황정보 데이터의 인터페이스 제공 방법은 put구문을 이용하면 해당 데이터를 처리한다.

```
from google.appengine.ext import
                                dbclassUser(db.Model):
  id = db.StringProperty()
  phonenumber = db.IntegerProperty()
  password = db.StringProperty()
  address1 = db.StringProperty()
  jumin_num1= db.IntegerProperty()
  jumin_num2 = db.IntegerProperty()
  email = db.StringProperty()

User(key_Id='sehun', phonenumber=010111)sehun.put()
User(key_Id='kyung', phonenumber=010114789).put()

User(key_Id='minjoo',
 phonenumber=01011111114).put()
User(key_Id='hyungjin', phonenumber=01011111115,
 email=iam1710@nate.com).put()
User(key_Id='sodam', phonenumber=01011111116).put()
User(key_Id='kilhee', phonenumber=01011111117).put()
User(key_Id='chnbo', phonenumber=01011111118).put()
```

Fig. 16. Login data call code of NoSQL

Table 1. Implementation environments

분 류	세부 사항
H/W 시스템	Intel Core2 Quad Q9400 2.66Ghz RAM : 8Gb
운영체재(OS)	Windows 7(64bit)
설계 Tool	StarUML 5.0, ER-Win 4.0, Microsoft Visio 2007, Protege 3.4.2
클라우드 구현 Tool	Google App Engine SDK 1.5.5
서비스 구현 Tool	Eclipse 3.7(INDIGO), Google Android 2.3(Gingerbread)
상황인식 추론 Plug-in	JessTab, SWRLTab
구현 언어	Java, OWL, SWRL
DBMS	NoSQL(BigTable)

4. 구현 결과 및 성능평가

4.1 시스템 구현 환경

제안하는 시스템의 구현환경 Table 1과 같으며, Intel Core2 Quad Q9400 2.66Ghz를 기반으로 클라우드 설계는 Google App Engine SDK 1.5.5를 활용하였다. 서비스 구현 툴은 Eclipse 3.7(INDIGO), Google Android 2.3(Gingerbread) 이며, 상황인식 추론 플러그인은 JessTab, SWRLTab이다. 또한 DBMS는 NoSQL (BigTable)을 사용하였다.

4.2 제안하는 시스템의 모바일 GUI 결과

본 논문은 시스템의 추론 프로세스 확장을 추론 비용 감소가 주요 목적이다. 본 시스템의 구현 확인을 위해 여기서는 하나의 시나리오 작성 후 조건을 설정하여 시스템을 테스트하였다. 시나리오와 환경 조건은 다음과 같다.

- 사용자는 원하는 경로 지점에서 모바일기기를 통해 사진을 촬영한다.
- 촬영한 사진은 사용자의 위치정보 값과 사진을 클라우드 서버에 저장하게 된다.
- 사용자는 자주 이용하는 지역을 설정할 수 있으며, 원하는 음식점의 종류를 사전에 입력할 수 있다.
- 사용자 추천 서비스는 병원, 음식점, 숙박시설 등을 사용자의 현재 위치에 해당하는 주변의 시설들을 리스트 형태로 사용자에게 제공될 수 있어야 한다.
- 사용자의 경로 이동 시 GPS 센서를 이용하여 과거 모바일기기를 통해 촬영했던 이미지를 볼 수 있도록 제공해야 한다.
- 일정 및 메모목록에 저장된 날짜와 시간을 기반으로 특정 날짜에 방문할 음식점 테마 및 음식점의 위치를 사용자에게 제공해야 한다.

Fig. 17~20은 시나리오를 기반으로 제안된 시스템의 GUI를 구현한 결과이다.

Fig. 17은 사용자의 모바일 기기를 통해 사용자의 현재 이동하는 위치 값의 변화를 주소 형식으로 10초에 한 번씩 업데이트하며, 최종 인식된 위치의 위도와 경도값을 표시하는 화면이다. 해당 위치 값은 클라우드 서버에 자동 전송된다. Fig. 18은 클라우드 서버에 접속하기 위해 사용자 인증 부분이다. 이 단계에서는 기존 연구에서 클라이언트측의 단순 인증과정과 사용자 인증이 제공되지 않았던 단계를 클라우드 형태의 인증과정을 통해 프로세스의 접근성 및 사용자 정보 권한을 강화하였다. Fig. 19는 미리 정의된 사용자가 선호한 음식점(분식점)을 기반으로 SWRL로 정의된 규칙과 Jess 추론 엔진을 통해 인식된 음식점 카테고리에서 추론의 우선순위로 결정된 사용자 선호 음식점을 기준으로 음식점을 추론한다. 추론된 음식점의 주소는 가상 서버에 저장된 빅테이블 데이터스토어에 NoSQL의 질의를 통해 구글 맵과 함께 사용자에게 제공된다. Fig. 20은 5개의 추론 리스트에 해당 음식점을 누르면 해당 음식점의 주소와 구글맵의 위치가 함께 제공된다.

Fig. 17. Recognized context information location data sending

Fig. 18. Context-awareness system service(Cloud Login GUI)

Fig. 19. Offer of context-awareness system service inference data(List)

Fig. 20. Offer of context-awareness system service inference data(Map)

4.3 성능평가 결과

제안하는 시스템의 성능평가는 크게 2가지 형태로 제시한다. 첫 번째로는 제안하는 시스템과 기존에 연구된 시스템과의 수치적인 평가 방법인 정량적인 평가이다. 두 번째로는 2장 관련 연구에서 제시된 기존 연구와의 객관적인 비교평가표를 통해 제안하는 시스템의 우수성을 입증한다.

1) 정량적인 시스템 성능평가

제안하는 시스템의 정량적인 평가 방법은 크게 3가지로 구분하여 제시한다. 첫 번째 방법은 제안된 모바일 클라우드상의 GPS 데이터 로드를 통한 데이터 추론 상황정보 예측 비용에 들어가는 Working Memory 데이터 적재 비용을 측정하는 평가를 수행한다. Fig. 21은 SWRL을 이용한 상황 추론 리소스인 GPS 데이터를 모바일 클라우드의 Jess 추론 엔진의 Working Memory에 적재되는 비용을 측정한 결과이다. [6]의 연구에서 측정한 전위 추론 방식은 모바일 기기 내부에서 Working Memory에 대한 적재 비용을 제시하고 있다. 또한 데이터 적재 횟수는 2000~5000까지만 제시되어 있다.

그러나 본 논문에서는 500~5000까지 비교하여 각 구간별 데이터의 데이터 추론을 위한 Working Memory 적재 비용 제안된 모바일 클라우드 시스템과 모바일 기기 내부의 데이터적재 비용을 측정 및 비교하였다. 비교 평가 결과

2000~3000개의 GPS 측정값에 따른 추론 리소스의 메모리 적재 비용은 [6]의 연구보다 우수하였지만, GPS 데이터가 누적된 3800개 이상의 데이터를 적재하여 추론하기 위한 비용에서는 [6]의 연구보다 다소 높은 적재 비용에 대한 결과를 초래하였다. 이는 GPS 데이터를 가상 서버로 전송하기 위한 통신상의 트래픽양의 임계값 설정으로 인해 모바일기기 내부적으로 추론과정을 해결하는 [6]의 연구와 데이터 적재 비용의 차이를 보였다.

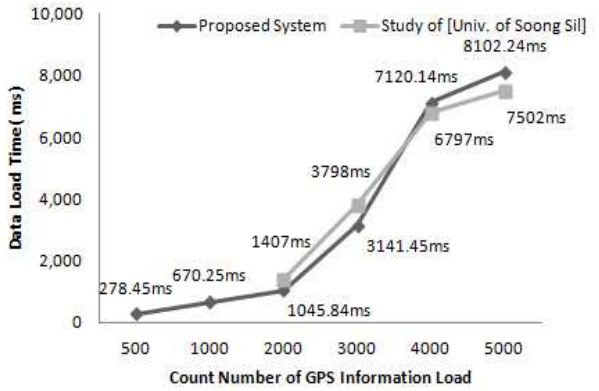

Fig. 21. Working Memory data load cost measurement

두 번째 방법은 본 논문에서 제안한 추론 방식과 추론 언어에 따른 상황인식 데이터 추론 비용에 대한 비교평가이다. Fig. 22는 추론 방식과 추론 언어에 따른 상황인식 추론 데이터 반환 비용의 측정 결과를 나타내고 있다.

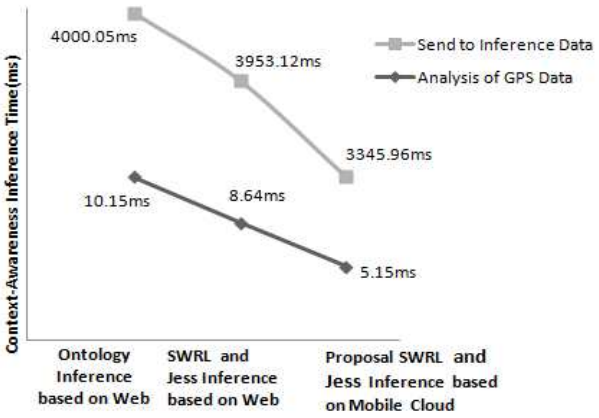

Fig. 22. Measurement result of context-awareness inference data cost according to inference method and inference language

비용 측정을 위한 조건은 GPS 데이터의 분석 비용과 최종적으로 추론된 데이터의 클라이언트 전송 비용을 각각 10개의 GPS 데이터에 대한 추론 비용을 20회 측정하여 평균 반환 비용을 제시하였다. 제안된 시스템의 상황추론 언어는 OWL과 SWRL을 이용하였고, 추론 방식은 Jess 추론 엔진과 서로 다른 상황추론 언어와 추론 엔진을 연결시키기 위한 JessTab을 이용하였다. 비교 평가 항목은 기존 연구에서 진행된 웹기반의 상황인식 시스템으로 추론 방식과 추론 언어의 변경을 통한 비교 항목을 설정하고 제안된 모바일 클라우드 처리 방식의 추론 방식과 상황 추론 언어기반의 시스템과 추론 비용을 평가하였다. 성능평가 결과 기존에 제안된 두 가지 형태의 웹 기반 상황인식 방식은 상황 리소스 분석과 분석된 리소스의 추론 엔진 질의 방식에 대한 연결 과정 프로세스가 길어서 질의에 대한 추론 엔진의 비용이 상대적으로 높게 측정되었다. 그리고 질의 방식에 대한 처리 프로세스가 소규모 데이터로 한정되어 있는 방식으로 인해 멀티 환경에서의 데이터 추론 비용은 매우 높다. 하지만 제안된 모바일 클라우드기반의 상황인식 시스템은 모바일 클라우드를 활용하여 대규모 데이터를 JessTab을 이용한 플러그인을 통해 추론 엔진에 보다 정확하고 빠른 질의 응답이 가능하고 모바일 클라우드방식의 가상 서버를 통한 프로세스 구조는 추론 데이터의 반환 비용을 감소시키는 결과를 보였다.

정량적인 평가를 위한 세 번째 방법은 제안된 시스템의 정확성 및 적합성을 측정한다. Fig. 23은 제안하는 시스템의 적합성을 검증하기 위해 추론 서비스의 완전성에 해당하는 재현율*(Recall ratio, 추천된 음식점중 조건에 맞는 음식점의 수/조건에 해당하는 음식점의 총 갯수)*과 추론 서비스의 일치성에 해당하는 정확률*(Precision ratio, 추천된 음식점중 조건에 맞는 음식점의 수/서비스를 통해 추천된 음식점의 수)*을 측정한 결과이다. 결과를 도출하기 위한 측정 조건은 특정 장소로 이동하면서 평소 선호하는 음식점(조건은 분식집과 패밀리 레스토랑으로 한정함) 분류의 음식점 추천과 장소를 2곳으로 선정하여 사용자가 현재 위치(전남 순천시 매곡동 순천대학교 정문)에서의 주변 음식점을 평가하도록 하였다. 측정 횟수는 같은 장소에서 300번을 측정하였다. 정확률에 비해 재현율이 상대적으로 낮게 측정된 이유는 사용자 모바일기기에 추천할 수 있는 음식점의 리스트수를 5곳으로 제한하였기 때문에 전체 비율이 낮게 측정되었다. 상대적으로 정확률이 높은 이유는 모바일기기에 제공된 5곳의 리스트와 실제 사용자가 선호하는 음식점의 비율을 계산하여 전체적으로 수치가 높게 측정되었다. 300회를 측정한 시스템 서

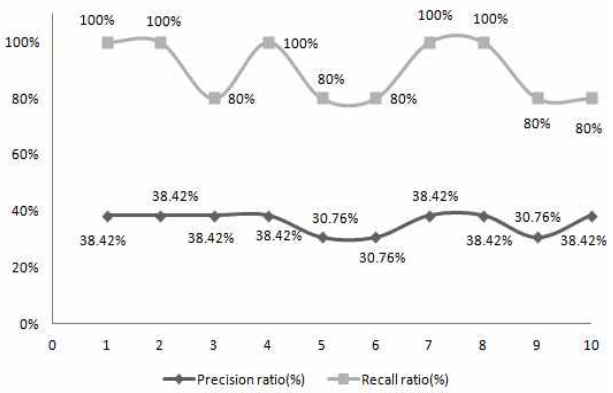

Fig. 23. Proposed system recall ratio and precision ratio measurement result

비스의 평균 재현율은 30.76%였으며, 평균 정확률은 80%로 측정되었다. 이는 사용자가 모바일 기기를 통해 어느 지역에서 본 논문에서 제안하는 상황인식 시스템의 서비스를 활용하면 서비스에서 제공되는 5곳 중 4곳을 주변 음식점으로 안내해주는 비교적 높은 서비스 성능결과를 도출하였다.

2) 정성적인 시스템 성능평가

Table 2는 제안하는 시스템과 기존 연구된 상황인식 시스템과의 정성적인 평가 지표이다. 기존 연구에서는 대부분 서비스 처리 방식을 서버 형태의 상황인식 처리를 제공하고 클라이언트측을 PC 및 PC와 연동되는 이기종간의 클라이언트로 제한되거나 모바일 클라이언트에서 추론 서비스를 제한하였다. 또한 대다수의 기존 연구에서는 상황인식의 데이터 및 추론 정보에 대한 보안성을 제공되지 않는 문제점을 발생시켰다. 상황 추론 언어는 대부분 온톨로지 기반으로 작성되었으며, [2]의 연구에서는 RDF Triple 추론 방식, [5]의 연구에서는 베이지안 네트워크, [6]의 연구에서는 Jess와 같이 기존 선행 연구에서는 추론 방식 및 엔진에 관한 부분은 다르게 나타났다. 이와 같은 기존 선행 연구와의 비교를 통해 제안하는 시스템에서는 기존 선행 연구에서 제안되지 않았던 서비스 설계 방식을 UML을 이용하여 객체지향 설계를 통해 소프트웨어 및 시스템의 재사용성과 확장성을 극대화하였다. 또한 추론 방식은 OWL/RDF 기반으로 시멘틱 검색 방식의 SWRL 규칙 결합을 통해 상황인식을 추론하였다. 또한 추론 속도 향상을 제시하고자 시멘틱 검색의 SWRL과 추론 엔진인 Jess를 JessTab으로 연동하고 모든 데이터 추론을 가상 서버인 클라우드상에서 시스템을 처리하였다.

Table 2. Qualitative evaluation with other study

	[2]의 연구	[5]의 연구	[6]의 연구	제안하는 시스템
서비스 설계	N/A	N/A	N/A	객체지향 모델링
상황추론 언어	OWL/RDF	OWL	OWL	SWRL/OWL
추론 방식	RDF Triple	베이지안 네트워크	Jess/상하 추론	Jess/Jess Tab
시스템 구조	통합서버	에이전트	모바일	모바일 클라우드
보안성	N/A	N/A	N/A	클라우드 인증
서비스 처리 방식	서버처리	서버처리	클라이언트 처리	가상 서버처리

5. 결론 및 향후연구

본 논문에서는 모바일 클라우드 환경에서 Jess 및 SWRL를 이용한 상황인식 시스템을 제안하여 사용자의 GPS값과 미리 정의된 사용자의 선호도 등을 바탕으로 사용자의 현재 위치에서 최적의 상황인식 서비스 구현하였다. 첫째, RDF/OWL의 온톨로지에 대한 시멘틱 검색인 SWRL의 DL+RuleML 추론 방식을 이용하여 보다 빠른 추론 속도를 보였고 JessTab 플러그인을 활용하여 Jess 추론 엔진과 Protege의 지식베이스에 저장된 SWRL의 Class, Property, Individual의 값을 매핑하여 OWL 문법 형식을 연동함으로써 Rule과 Fact의 표현을 그대로 사용할 수 있다는 장점을 보였다. 또한 외부 추론 엔진과의 연결로 인한 추론 속도의 감소 등을 성능평가를 통해 확인하였다. 둘째, 상황인식 시스템은 특정 플랫폼에 종속되지 않도록 하기 위하여 클라우드 기반의 가상화 서버 형태인 PaaS(Google App Engine)를 활용하여 프로세스를 구현함으로써 다양한 플랫폼 형태에서 상황인식 시스템을 가능하도록 하였다. 마지막으로, 제안된 시스템 구현을 위하여 객체지향 관점으로 설계 형태로 접근하여 각 기능들을 모듈화시켜 소프트웨어의 재사용성 및 유지보수성을 극대화하였다. 또한 각 공정들에 해당하는 결과 산출물들을 코드별로 분류함으로써 프로세스의 확장성을 극대화하였다.

향후 연구로는 본 논문에서 활용한 특정 벤더의 제공 방식인 PaaS방식의 모바일 클라우드 형태에서 벗어나 순수 개발자 입장에서의 SaaS(Software as a Service)방식을 이용하여 가상화 서버를 다양한 형태의 상황인식 시스템으로 구현할 예정이다.

참 고 문 헌

[1] Hye-Kyoung Jeon, Yang-Jae Park, Jung-Hyun Lee, “Design of Infant Danger Notification System using Context-Information on the Home Network Environment”, In Proceedings of the Fall Conference of Korean Institute of Information Technology, pp.324-329, 2007.

[2] Tao Gu, Hung Keng Pung, Da Qing Zhang, “A service oriented middleware for building context-aware service”, Journal of Network and Computer Application, Vol.28, No.1, pp.1-18, 2005.

[3] Tao Gu, Hung Keng Pung, Da Qing Zhang, “A Middleware for Building Context-Aware Mobile Services”, Proceeding of IEEE Vechicular Technology Conference, pp.2656-2660, 2004.

[4] Tao Gu, Xiao Hang Wang, Hung Keng Pung, Da Qing Zhang, “An Ontology-based Context Model in Intelligent Environment”, Proceeding of Communication Networks and Distributed Systems Modeling and Simulation Conference, pp.270-275, 2004.

[5] P. Coppola, V.D. Mea, L. D. Gaspero, D. Menegon, D. Mischis, S. Mizzaro, I. Scagnetto, L. Vassena, “The Context-Aware Browser”, In Proc. of the IEEE Intelligent Systems, Vol.25, No.1, pp.38-47, 2010.

[6] Gil Heo, “The Context-aware Recomendation System with

Rule-based Inference Engine in the Mobile Environment", Soongsil University, A Master's Thesis, 2011.

[7] Jong-In Lee, Dong-Gyu Yeo, Byeong-Man Kim, "Detection of Music Mood for Context-aware Music Recommendation", The Transactions of the Korea Information Processing Society : Part B, Vol.17B, No.4, pp.263-274, 2010.

[8] Joong-Kyung Ryu, Kyung-Yong Chung, Jong-Hun Kim, Kee-Wook Rim, Jung-Hyun Lee, "Recommendation using Service Ontology based Context Awareness Modeling", Journal of Korea Contents Association, Vol.11, No.2, pp.22-30, 2011.

[9] Hae-Yoon Park, Hae-Young Yoo, "Portability Evaluation Technique of Application for Heterogeneous Mobile Platform", The Transactions of the Korea Information Processing Society : Part D, Vol.18, No.2, pp.123-132, 2011.

[10] Gruber, T. R. "What ia an Ontology?", http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, 1993.

[11] SWRL (A Semantic Web Rule language Combining OWL and RuleML), W3C Member Submission, http://www.w3.org/ Submission/SWRL, 2004.

정 세 훈

e-mail : iam1710@hanmail.net

2010년 순천대학교 멀티미디어공학과(학사)

2012년 순천대학교 멀티미디어공학과 (공학석사)

2012년~현 재 순천대학교 멀티미디어 공학과 박사과정

관심분야 : 상황인식 시스템, 모바일 클라우드, 시멘틱 웹, 분산 데이터베이스

심 춘 보

e-mail : cbsim@sunchon.ac.kr

1996년 전북대학교 컴퓨터공학과(학사)

1998년 전북대학교 컴퓨터공학과(공학석사)

2003년 전북대학교 컴퓨터공학과(공학박사)

2004년~2005년 부산카톨릭대학교 컴퓨터 정보공학부 전임강사

2005년~현 재 순천대학교 멀티미디어공학과 부교수

관심분야 : 데이터베이스 색인구조, 멀티미디어 정보검색, 유비쿼터스 및 클라우드 컴퓨팅

추론 비용 감소를 위한 Jess 추론과 시멘틱 웹 RL기반의 모바일 클라우드 상황인식 시스템

정 세 훈[†] · 심 춘 보[††]

요 약

상황인식 서비스라는 개념은 컴퓨팅과 통신을 기반으로 서비스를 제공 받는자의 주변 상황을 컴퓨터가 인식하고 스스로 판단하여 사용자에게 유용한 정보를 제공하는 서비스이다. 그러나 모바일 환경에서 제한된 모바일 기능과 메모리 공간 및 추론 비용 증가로 인해 소규모의 상황인식 처리 능력을 가지는 단점과 추론 엔진의 부분 개발로 인한 상황 정보 추론 방식의 제한적인 형태로 나타나고 있다. 이에 본 논문에서는 특정 플랫폼에 종속되지 않고 다양한 모바일기기에서 상황인식 서비스를 제공받을 수 있도록 PaaS기반의 GAE을 이용한 모바일 클라우드 상황인식 시스템을 제안한다. 제안하는 시스템의 추론 설계 방식은 OWL의 온톨로지와 SWRL 규칙으로 표현되는 시멘틱 추론을 이용한 지식베이스 프레임워크와 규칙 기반의 추론 엔진을 제공하는 Jess를 활용하여 설계한다. 아울러 기존 추론 질의 방식인 시멘틱 검색의 SparQL 질의 추론 방식의 단점을 극복하고자 SWRL형태의 Rule 규칙 정보인 Class, Property, Individual등의 속성값들을 특정 플러그인을 이용하여 Jess 추론 엔진에 연결하도록 설계한다.

키워드 : 시멘틱 웹, Jess, OWL, SWRL, 상황인식, 모바일 클라우드