

지능형 e-비즈니스를 위한 플랫폼에 관한 연구: DAML-S의 규칙기반 프레임워크로의 확장 및 통합방안

양진혁*, 민재홍**, 이윤수*, 정인정*
*고려대학교 전산학과, **한국전자통신연구원
*e-mail:grjinh, arzhna, chung@korea.ac.kr
**e-mail: jhmin@etri.re.kr

A Study on the Platform for the Intelligent e-Business: A Method on Extension and Integration of DAML-S into Rule-based Framework

Jin-Hyuk Yang*, Jae-Hong Min**, Yun-Su Lee*, In-Jeong Chung
*Dept. of Computer Science, Korea University
**ETRI

요 약

본 논문에서 우리는 시맨틱 웹을 기반으로 하는 웹 서비스 발견, 실행, 구성 및 모니터링을 자동화하기 위한 목적을 가지는 DAML-S를 규칙기반 프레임워크로의 확장 및 통합방안에 관한 연구결과를 제시한다. 확장 및 통합된 형태의 DAML-S는 온톨로지 내부에 DamlRuleML로 인코딩된 형태의 규칙들을 포함한다. 내포된 규칙들로 보완된 DAML-S 서비스 온톨로지들은 서비스 발견, 실행, 구성 및 모니터링에서 확장 이전의 DAML-S가 가지는 한계점들을 극복하고 나아가 보다 지능적인 서비스의 구현을 가능하게 한다. 본 논문에서 제안된 우리의 접근법은 온톨로지를 마크업하기 위한 노력과 규칙을 표현하기 위한 노력이 자연스러운 방법으로 통합될 수 있는 근간을 마련할 뿐만 아니라 규칙들을 이용하여 온톨로지들을 보완하고, 규칙들에서 사용되는 용어들을 온톨로지들에서 정의된 용어들과 속성들로 표현할 수 있다는 장점을 가진다. 이와 같은 공헌이외에 우리는 실제 시나리오를 통하여 언급된 사항들이 가능하게 되는 방법들을 구체적으로 보인다.

1. 서론

웹 서비스는 웹에서 접근 가능한 프로그램 또는 장치로서 간단하게 정의될 수 있으며, 웹 서비스를 위한 표준들은 메시지를 위한 SOAP(Simple Object Access Protocol)[1], 서비스 기술을 위한 WSDL(Web Service Description Language)[2] 및 서비스 발표 및 검색을 위한 UDDI(Universal Description, Discovery and Integration)[3]가 존재한다. 게다가 서비스 구성을 위한 표준들로서는 WSCI(Web Service Choreography Interface)[4], BPEL4WS(Business Process Execution Language for Web Services)[5], DAML-S(DAML ontology for web Services)[6]와 같은 마크업 언어들이 존재한다.

웹 서비스가 특히 e-비즈니스에서 중요한 역할을 수행하는 이유는 기존 정적인 웹에서 단순한 EDI 기반의 문서 교환에 머물고 있는 e-비즈니스를 웹 서비스라는 개념을 이용하여 상호 필요한 원격 프로그램을 호출함으로써 기업의 프로세스를 수행하는 동적인 개념으로 변화시키고 있기 때문이다[7]. 상기와 같은 표준들을 이용하여 e-비즈니스 참가자들이 서비스를 웹상에서 기술하고 발견하고 나아가 구성, 실행 및 통신할 수 있다.

우리는 본 논문에서 DAML-S¹⁾를 웹 서비스를 위한 마크업 언어로 채택한다. 그 이유는 DAML-S는 시맨틱 웹과 온톨로지 분야에서 기원된 추론이 가능한 마크업 언어일 뿐만 아니라 기계에 의한 의미처리가 가능한 표현력이 풍부한 언어이기 때문이다. 그러나 DAML-S가 웹 서비스

를 표현하기 위한 매우 훌륭한 수단임에도 불구하고 DAML-S 버전 0.9[6]에서는 2장에서 자세하게 언급되고 있듯이 알려진 많은 한계점들을 가진다.

우리는 본 논문에서 지능형 웹 서비스를 수행하기에 지금껏 알려진 여럿의 마크업들에 비해 상대적으로 훌륭한 마크업 언어인 DAML-S의 한계점들을 극복하고 온톨로지와 규칙 표현의 통합을 위한 방안으로서 DAML-S의 규칙기반 프레임워크로의 확장 방안을 제시한다. 이를 위하여 현재 RuleML 버전 0.8[8]의 확장된 형태인 SCLP RuleML[9]을 DAML+OIL로 나타낸 DamlRuleML[10]을 DAML-S 서비스 온톨로지에 포함시킬 수 있는 이론적인 근거와 실제 예를 제공한다. 이러한 접근법은 온톨로지를 규칙들을 이용하여 보완한다는 측면에서 뿐만 아니라 규칙들에서 사용되는 용어들을 온톨로지들에서 사용되는 용어들로 사용한다는 측면에서 장점들을 가진다. 게다가 우리의 접근법은 규칙들을 내포한 확장된 형태의 DAML-S 온톨로지를 이용할 경우 서비스를 위한 DAML-S 온톨로지들만을 이용해서는 불가능한 것들이 가능하게 되는 장점 또한 가진다. 우리는 본 논문에서 구체적인 시나리오를 통해서 이와 같은 사항들이 가능하게 되는 절차와 방법을 자세하게 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 DAML-S의 특징들 및 한계점들과 RuleML을 살펴본다. 3장에서는 DAML-S의 문제점들이 규칙기반 프레임워크로 확장된 DAML-S를 통해서 개선되는 방법을 시나리오와 함께 보인다. 마지막으로 4장에서는 우리가 취하고 있는 접근법의 특징들 및 향후과제에 대해서 언급한다.

2. DAML-S 및 RuleML

본 장에서는 DAML-S 자체보다는 DAML-S의 알려진

1) OWL-S가 2003년 11월에 발표되었으나 연구수행당시에는 이용가능하지 않았다.

문제점들 및 한계점들과 RuleML에 대하여 언급한다.

2.1 DAML-S의 한계점들

DAML-S는 웹 서비스의 발견, 실행, 구성 및 모니터링을 자동화하기 위한 목적으로 시맨틱 웹기반의 온톨로지 언어 DAML+OIL로 작성된 서비스 온톨로지로서 서비스 프로파일, 서비스 모델 및 서비스 그라운드링으로 구성된다. DAML-S의 알려진 한계점들은 다음들을 포함한다.

- L1: 개념적인 모델이 부정확하다. 즉, DAML-S 모델은 WSDL 모델로 나타낼 때 DAML-S를 구성하고 있는 프로파일 온톨로지와 프로세스 온톨로지 사이에 분명한 연계가 존재하지 않는다[11].
- L2: DAML-S 개념들은 매개변수형의 다형성 (parametric polymorphism)과 재사용성과 같은 소프트웨어 공학적인 개념들을 지원하지 못한다[11].
- L3: DAML-S는 WSDL로의 매핑은 많은 부분에 있어 DAML-S의 표현력의 제약사항 때문에 제한되어진다 [11].
- L4: 사용하기 힘들다. 즉, DAML-S는 버전 0.9가 발표되어 아직 표준으로 정착되어 있지 못하기 때문에 도구의 지원이 제한되어 있을 뿐만 아니라 알려진 보기들이 2개만이 존재하며 DAML, WSDL 및 SOAP과 같은 부가적인 지식이 요구된다[11].
- L5: DAML-S 스펙은 IP뿐만 아니라 OE도 반드시 기술되도록 변경해야만 한다[12].
- L6: DAML-S는 작업들을 구성하기 위해서는 부가적인 논리가 요구된다[12].
- L7: DAML-S는 여러 개의 프로파일들, 프로세스들 그리고 그라운드링들을 명시하지 않고서 같은 서비스에 대한 여러 개의 인터페이스들을 명시할 수 있어야 한다 [13].
- L8: DAML-S는 기존의 서비스들로 새로운 서비스를 기술할 수 있어야 한다[13].
- L9: DAML-S에서 제공하는 현재의 IOPE들과 타입정보가 서비스 발견과 구성을 위한 충분한 속성들이 아니다 [13].
- L10: DAML-S는 IOPE 정보를 자세하게 기술하기 위한 선택스를 제공하고 있지 않다[14].
- L11: DAML-S 서비스 프로파일의 기능적인 속성들 중 하나로 데이터를 포함해야만 한다[15].
- L12: DAML-S는 예측된 반환 시간과 값들 그리고 타입아웃을 보장하는 것과 같은 프리미티브들을 기술할 수 있어야 한다[15].
- L13: DAML-S에서는 순서(sequence)를 표현함에 있어 조기 종료를 위한 조건들과 메커니즘들을 명시해야만 한다[15].
- L14: DAML-S는 에이전트가 진정으로 반사적(reactive)으로 되기 위해서는 이벤트들에 대응할 수 있는 메커니즘과 환경을 모니터링하기 위한 메커니즘이 요구된다 [15].
- L15: DAML-S의 시맨틱스: [16]과 [17] 각각에서 DAML-S의 연산적인 시맨틱스(operational semantics)와 액시엄적인 시맨틱스(axiomatic semantics)를 제공하고 있지만 현재까지 DAML-S를 위한 정형 시맨틱스가 없다[18].
- L16: 프로세스 컨트롤 온톨로지(process control ontology): DAML-S는 웹 서비스 실행 모니터링의 자동화를 위해서 요구되는 프로세스 컨트롤 온톨로지가 현재 제공되지 않았다[18].
- L17: 대화 프로토콜(conversation protocol): DAML-S는 서비스를 수행함에 있어 발행하는 정확한 통신 패턴들(서비스 거래자들 사이에서 교환되는 메시지들의 패턴들)을 명시하는 방법이 현재 버전의 DAML-S에서 충분히 분명한 방법으로 명시되지 않고 있다[18].

2.2 (SCLP) RuleML and DamiRuleML

RuleML은 규칙 지식표현을 위한 XML 선택스로서 주요 상용 규칙 시스템들 사이에서 상호 호환이 가능하다. 현재 상용 규칙 시스템들은 SQL(RDB), 프롤로그, 생성

규칙들 및 ECA(Event Condition Action)으로 분류할 수 있다. 이 4개의 상용 제품들의 공통점은 선언적인 로직 프로그램이라는 것이다. 사실 RuleML의 핵심 표현은 혼 로직 프로그램이다. 그러나 RuleML은 NAF(Negation As Failure)와 같은 부정(negation)문제와 우선권이 정해진 충돌 핸들링 및 질의들과 액션들을 위한 질차적인 프리미티브들이 없다는 표현상의 제한점들을 가진다. 게다가 RuleML에서는 이항(binary) 관계들만을 포함한다는 제약도 가진다.

상기와 같은 RuleML의 제한점을 개선하기 위한 노력의 일환으로 SCLP(Situated Courteous Logic Program)[9] RuleML이 등장하였다. SCLP는 부정문제에 대한 해결책을 가지고, 우선권을 가지고 규칙 충돌들을 핸들링 할 수 있으며 질의와 액션을 위한 프리미티브들을 가진 로직 프로그램으로서 전통적인 논리 추론보다 수행능력이 우수하다.

SCLP RuleML 0.8을 위한 DTD를 위해서는 [9], SCLP RuleML을 DAML+OIL로 표현한 DamiRuleML을 위해서는 [10]을 참조하고, SCLP RuleML 버전 0.8 표현에 대한 DAML+OIL로의 완전한 변환은 [10] 및 [19]를 참조한다.

3. DAML-S의 규칙기반 프레임워크로의 확장

3.1 설계 원칙들 및 전략들

우리는 DAML-S를 규칙기반 프레임워크로 확장하는 방안을 강구함에 있어 다음과 같은 문제점들에 직면하였다.

- R1: 규칙들이 왜 필요한가?
- R2: 규칙의 표현은 어떻게 할 것인가?
- R3: 규칙들을 DAML-S 내부에 포함시킬 것인가? 별도로 저장할 것인가?
- R4: 규칙들을 저장하고 질의하는 것은 어떻게 할 것인가?
- R5: 표현된 규칙과 DAML-S 스펙의 호환성
- R6: 규칙 충돌(rule conflict)이 발생했을 경우에는 어떻게 할 것인가?

3.1.1 규칙의 필요성

온톨로지와 규칙을 통합하여 함께 표현함으로써 얻을 수 있는 장점들로서는 [20]에서도 언급되듯이 온톨로지를 규칙을 이용하여 보완할 수 있다는 점과 규칙에서 사용되는 용어들을 온톨로지에서도 사용되는 용어와 관계들을 이용하여 표현할 수 있다는 것이다. 따라서 우리는 시맨틱 웹의 발전 방향을 준수하면서 상기와 같은 장점을 취함과 동시에 DAML-S 스펙이 자연스럽게 규칙을 포함할 수 있는 규칙프레임워크를 제안한다. 이와 같은 접근법은 팀 버너스리에 의해서 제안된 시맨틱 웹 스택[21]과 일치한다.

3.1.2 규칙의 표현

우리는 기본적으로 규칙의 표현을 위해서 2.2절에서 논의된 OLP(Ordinary Logic Program)[20,9] 버전의 RuleML 버전 0.8을 선택한다. 그러나 OLP 버전의 RuleML은 2.2절에서 언급되고 있듯이 절차적인 행위를 위한 프리미티브들의 부재와 규칙 충돌문제 해결의 부재와 같은 제한점들을 가진다. 따라서 우리는 이러한 제한점을 극복하고 보다 표현력 있는 규칙들을 제공하기 위하여 SCLP 버전의 RuleML을 이용하여 규칙들을 표현하였다.

3.1.3 DAML-S의 내부 또는 외부에 규칙 임베딩하기

표현된 규칙을 어디에 위치시킬 것인가가 우리의 연구에서 결정할 중요한 문제였다. DAML-S 온톨로지 내부에 규칙을 포함시킬 것인지 아니면 DAML-S 외부에 즉, 별도의 지식 베이스를 구성하여 규칙을 저장할 것인지에 대해 결정해야만 했다. 최종적으로 우리는 DAML-S 내부에 규칙을 표현하기로 결정하였다. 이와 같은 이유는 SCLP 버전의 RuleML이 DAML+OIL의 클래스와 속성들로 표현이 가능하기 때문이다. SCLP 버전의 RuleML을 위한 DAML+OIL을 DamiRuleML이라 한다. 우리는 DamiRuleML을 이용하여 SCLP 버전의 RuleML을

DAML-S 온톨로지에 임베딩 시킨다. 이와 같은 접근법은 DAML-S가 DAML+OIL 언어와 행위 묘사를 위한 부가적인 구조물들로 구성되었다는 측면에서 가장 자연스러운 접근법이다. 뿐만 아니라 DAML-S와 규칙표현과의 호환성문제도 자연스럽게 해결될 수 있는 장점을 가진다.

3.1.4 규칙의 저장과 질의

다음으로 우리가 직면한 문제점은 DamlRuleML로 보완된 DAML-S 온톨로지 정보에서 규칙들을 저장하고 저장된 규칙들을 활용하기 위한 질의에 관한 것이었다. 우리는 3.2절에서 언급되고 있는 시나리오를 생성하고 구현함에 있어 저장과 질의를 위하여 규칙추론 엔진인 Jess(Java Expert System Shell)[22]를 이용하기로 결정하였다. Jess는 공개된 규칙 시스템으로서 여타의 다른 연구들에서도 많이 사용되고 언급된 효과적인 추론 엔진이다.

3.1.5 규칙과 DAML-S 스펙과의 호환성

이제 우리는 DamlRuleML로 표현된 SCLP 버전의 RuleML을 DAML-S 온톨로지에 포함시킬 때 호환성 문제를 더 이상 고려할 필요가 없다. 왜냐하면 우리가 사용할 SCLP RuleML이 DAML+OIL의 문법을 따르기 때문이다.

3.1.6 규칙 충돌

마지막으로 우리가 DAML-S를 규칙기반 프레임워크로 확장시킬 때 당면했던 문제점은 표현된 규칙들이 단조성(monotonic)을 가진다면 어떻게 처리할 것인가에 대한 것이었다. 그러나 이미 규칙충돌에 대한 해결책은 SCLP 버전의 RuleML에서 명시적으로 제공되고 있다.

3.2 시나리오

우리는 이번 섹션에서 [23]에서 소개된 시나리오 중 자동적인 웹 서비스 구성을 위한 예를 보인다. 그 이유는 구성을 위한 예가 발견과 실행의 경우를 모두 포함하고 있기 때문이다. 다음은 자동적인 웹 서비스 구성을 위한 예와 관련된 가정들이다.

○ 시나리오: 어떤 사람 Bob은 샌프란시스코에서 몬테리로 DARPA가 후원하는 연구 프로젝트와 관련하여 여행을 하길 원한다. Bob에게는 2개의 제약사항들이 있다. 하나는 개인적인 것이고 하나는 프로젝트와 관련된 것이다. 즉 개인적인 제약사항으로는 비행시간이 3시간이 넘지 않을 경우에는 자가운전을 선호한다는 것이고, 업무상 제약사항은 미국 비행기를 이용해야만 한다는 것이다.

우리는 더 나아가 상기의 복합적인 서비스 구성을 위한 최종 목적이 다음의 각 단계들을 순서대로 행함으로써 수행된다고 가정한다.

- 비행기 또는 자동차 예약
- 숙소 예약

FindAirlineTicketingService는 항공사 이름, 날짜와 시간 및 출발지와 목적지를 입력으로 받아 가격과 탑승 시간을 반환하는 서비스이다(지면관계상 코드는 생략한다). 이와 유사하게 다른 서비스들의 프로세스 온톨로지도 작성할 수 있을 것이다. 이와 같은 프로세스와 프로세스 흐름을 가지는 서비스를 발견 및 구성하는 것이 우리의 최종 목적이다.

현재의 DAML-S에서 Bob이 이와 같은 서비스를 자동으로 찾기 위해서는 서비스 프로파일 온톨로지 또는 필요에 따라 프로세스 온톨로지를 읽고 처리하여 찾고자 하는 서비스를 찾게 된다. 이와 같은 서비스 매치메이킹 절차는 본 논문 범위 밖의 문제로서 우리는 가장 간단하지만 강력한 다음의 일반적인 방법을 사용한다.

DAML-S 온톨로지 정보가 IOPE에 따라 SVO와 같은 트리플(triple)형태로서 Jess와 같은 규칙추론 엔진에 저장되어 있다고 가정한다. Bob이 찾고자 하는 서비스를 SVO의 형태로 Jess 추론엔진에 입력하게 되면 추론 엔진은 IOPE 타입정보에 합당한 서비스들의 목록을 결과로 반환하여 준다. Bob은 결과의 서비스들 중 자기의 기호에 맞는 서비스를 선택하게 된다.

상기에서 살펴보듯이 Bob의 기호에 대한 정보를 DAML-S 온톨로지가 포함하고 있지 않기 때문에 Bob은 자기에게 꼭 맞는 서비스를 발견하기 위해서 부가적인 서비스 필터링 작업을 거치고 난 후에야 원하는 서비스를 찾을 수 있게 되는 불편한 점을 가진다. 그러나 우리가 제안한 규칙정보를 포함한 DAML-S 온톨로지 정보를 이용하여 보면 상기와 같은 문제점들이 다음과 같이 해결될 수 있다.

먼저 상기의 'FindAirlineTicketingService' 서비스의 모델 온톨로지 정보 이외에 다음과 같은 규칙정보를 서비스 광고 시에 부가적으로 명시한다.

Bob은 UAL 항공편을 업무상 이용해야 한다는 제약조건은 다음과 같이 DamlRuleML로 표현된다.

```
<damlRuleML:fact>
  <damlRuleML:_riab>Bobsconstraint</damlRuleML:_riab>
  <damlRuleML:_head>
    <damlRuleML:_atom>
      <damlRuleML:_opr>
        <damlRuleML:rel>UAL<damlRuleML:rel>
      </damlRuleML:_opr>
      <damlRuleML:ind>Bob</damlRuleML:ind>
    </damlRuleML:_atom>
  </damlRuleML:_head>
</damlRuleML:fact>
```

그리고, 비행거리가 3시간 미만일 경우에는 자가운전을 한다와 같은 개인적인 선호에 대한 규칙은 지면상 생략한다.

Bob이 이와 같은 제약사항 또는 개인적인 선호사항들을 서비스 발견을 수행할 때 온톨로지 정보에 포함시킬 경우에는 서비스 매칭의 결과들에서 Bob의 추가적인 선별작업이 요구되지 않을 것이다. 또한 규칙이 포함되기 이전의 DAML-S 온톨로지 정보만을 가지고 서비스 구성을 수행하는 함에 있어 만약 비행기 예약이 취소되었을 경우 복합 서비스의 실행은 중단될 수밖에 없을 것이다. 그러나 규칙정보로 보완된 DAML-S 온톨로지의 경우에 있어서는 SCLP RuleML 버전의 구조물 중 'sensor'와 'effect' 문장을 통하여 서비스 광고들이 존재하는 지식 베이스를 질의할 뿐만 아니라 규정된 액션을 취할 수 있는 규칙을 표현할 수 있다. 우리는 다음과 같은 'sensor' 및 'effect' 문장을 DamlRuleML로 표현하고 이를 DAML-S 온톨로지 정보에 부가적으로 첨가할 경우 Bob의 비행 예약 서비스가 중단되었을 경우 자동적으로 자동차 예약 서비스를 호출하게 할 수 있다.

```
<damlRuleML:effe>
  <damlRuleML:_opr>
    <damlRuleML:rel>callAutomobileReservation</damlRuleML:rel>
  </damlRuleML:_opr>
  <damlRuleML:_aproce>
    <damlRuleML:_jproc>
      <damlRuleML:clas>AutomobileReservation</damlRuleML:clas>
      <damlRuleML:meth>RentalCar</damlRuleML:meth>
    </damlRuleML:_jproc>
    <damlRuleML:path>iis.korea.ac.kr.AutomobileReservation</damlRuleML:path>
  </damlRuleML:_aproce>
</damlRuleML:effe>
```

즉, 상기 DamlRuleML 코드에서 비행시간이 3시간 미만일 경우에는 'callAutomobileReservation' 자바 메소드가 호출되게 되어 서비스 구성이 멈추지 않고 계속해서 작동하게 된다.

우리는 이상의 시나리오와 보기에서 DAML-S의 온톨로지 정보로서는 불가능했던 서비스 발견, 실행 및 구성의 자동화가 규칙정보로 보완된 온톨로지 정보를 이용하여 해결되는 과정을 살펴보았다.

4. 결론 및 향후과제

우리는 본 논문에서 시맨틱 웹을 근간으로 하면서 웹

서비스의 의미적인 표현을 위한 마크업 언어인 DAML-S를 규칙기반 프레임워크로의 확장 및 통합에 대한 연구결과를 제시하였다. 제안된 프레임워크는 SCLP 버전의 RuleML을 자연스럽게 DAML-S에 포함시킨다. 뿐만 아니라 확장 이전의 DAML-S 스펙으로서는 불가능했던 한계점들이 3장의 구체적인 시나리오들을 통해서 살펴봤던 것처럼 확장된 규칙기반의 프레임워크에서는 가능하게 되었다. 구체적으로 언급하면 2.2절에서 언급된 17개의 한계점들 중 10개의 한계점들이 아래에서 언급되듯이 해결 또는 개선될 수 있음을 알 수 있었다.

- L1: DAML-S 서비스 온톨로지들 사이의 불분명한 연계가 규칙을 통해서 해결될 수 있었다. 즉, 규칙들을 통해서 보완된 온톨로지들은 원래 버전의 DAML-S만을 가지고 표현되었을 경우보다 훨씬 더 명확한 연계를 가진다.
 - L3: DAML-S를 WSDL로 매핑할 때 발생하였던 표현력의 제한이 규칙들을 부가함으로써 해서 보완되었다.
 - L5: IOPE 정보뿐만 아니라 제약조건들 및 개인번호 사항들까지 규칙들로 보충된다.
 - L6: 부가적인 논리가 바로 규칙이다.
 - L9: 우리의 시나리오 예제들에서 살펴봤던 것처럼 서비스의 발견과 구성을 위해서는 IOPE 정보 이외에 규칙과 같은 부가적인 정보가 요구된다.
 - L10: DAML-S의 IOPE 정보를 보다 정확하게 RuleML로서 표현할 수 있다.
 - L11: 로직 프로그램에서는 변수를 취할 수 있다.
 - L12: QoS(Quality of Service)와 관련된 프리미티브를 제공하지는 않지만 RuleML에서는 이러한 역할을 할 수 있는 메커니즘을 제공한다.
 - L13: 조기 종료와 같은 조건들은 규칙으로 표현하기에 매우 알맞다.
 - L14: SCLP 버전의 RuleML이므로 반사성을 SLP를 이용하여 해결할 수 있다.
- 이와 같은 우리의 접근법은 현재 시맨틱 웹에서 추진하고 있는 온톨로지와 규칙의 레이어를 고려해왔을 때 매우 바람직하고 의미 있다. 뿐만 아니라 이와 같은 접근법은 규칙들을 사용함에 있어 기존 온톨로지들에서 정의된 용어들과 속성들을 활용할 수 있는 장점과 더불어 규칙들을 활용하여 온톨로지 자체를 보완할 수 있다는 장점을 가진다.

향후과제로서 우리는 DL(Description Logic)기반의 온톨로지와 FOL(First Order Logic)에 기반을 둔 RuleML의 표현력을 현재 OLP 버전의 RuleML과 SCLP 버전의 RuleML이상으로 이끌어 낼 수 있는 가능성에 대하여 연구할 것이다. 또 다른 향후과제로서는 DAML-S 서비스 온톨로지에 SCLP 버전의 RuleML을 DamlRuleML로 인코딩함에 있어 발견된 사항과 관련된 것이다. 즉 마크업 할 대상의 정보가 온톨로지 정보로 표현될 것인지 아니면 규칙 정보로 표현될 것인지를 명확하게 구분할 수 있는가에 대한 연구이다. 실제로 우리의 경험에서 우리는 몇몇 규칙들이 상대적으로 긴 표현을 가지더라도 DAML-S에서 제공되는 선택스를 이용하여 표현될 수 있음을 발견하였다. 마지막으로 우리가 고려하고 있는 향후과제는 현재의 바텀 업(bottom up) 방식으로 시맨틱 웹과 규칙에 대한 통합을 언급하고 있는 연구방향과는 다른 방법으로 접근하는 것이다. 즉 소프트웨어 공학적인 접근방법에서 탑 다운으로 DAML-S 서비스 온톨로지와 규칙들을 모델링하고 이에 대응되는 프로그램들을 구현하는 접근법이다. 이와 같은 접근법에서 개발된 서비스는 보다 일반적인 템플릿 형태의 모델을 가지게 될 것이고, 일반적인 템플릿 형태로 서비스가 모델링 되고 구현된다면 온톨로지 및 규칙정보를 이용하여 커스터마이징이 가능하게 될 뿐만 아니라 재사용 또한 가능하게 되는 잠재적인 특징들을 가진다고 판단된다.

참고문헌

- [1] SOAP: <http://www.w3.org/TR/SOAP/>
- [2] WSDL: <http://www.w3.org/TR/wsdl>
- [3] UDDI: <http://www.uddi.org/>

- [4] WSCI: <http://www.w3.org/TR/wsci/>
- [5] BPEL4WS: <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [6] DAML-S: <http://www.daml.org/services/daml-s/0.9/>
- [7] 양진혁, 공유근, 김지영, 정인정, “지능형 웹 서비스를 위한 플랫폼에 관한 연구: 시맨틱 웹 기반의 웹 서비스 마크업 언어 설계 전략들”, 2003 가을 학술발표 논문집, pp. 118-120, 한국정보과학회, 2003. 10
- [8] <http://www.ruleml.org/indtd0.8.html>
- [9] Benjamin N. Grosf, “Representing E-Commerce Rules Via Situated Courteous Logic Programs in RuleML”, Electronic Commerce Research and Applications (ECRA) journal, 2003
- [10] <http://userpages.umbc.edu/~mgandh1/2002/06/DamlRuleML/>
- [11] Marta Sabou, Debbie Richards, and Sander van Splunter, An experience report using DAML-S, In Proceedings of the Twelfth International World Wide Web Conference Workshop on E-Services and the Semantic Web, 2003.
- [12] Mithun Sheshagiri, Marie desJardins, and Tim Finin, A Planner for Composing Services Described in DAML-S, In Proceedings of the AAMAS Workshop on Web Services and Agent-based Engineering, 2003.
- [13] Debbie Richards, Sander van Splunter, Frances M.T. Brazier and Marta Sabou, Composing Web Services using an Agent Factory, In Proceedings of the AAMAS Workshop on Web Services and Agent-based Engineering, 2003.
- [14] Simon Thompson and Ion Constantinescu, The Agentcities DAML-S style guide, Technical report, Agentcities, 2003.
- [15] Joanna J. Bryson, David Martin, Sheila I. McIlraith, and Lynn Andrea Stein, Agent-Based Composite Services in DAML-S: The Behavior-Oriented Design of an Intelligent Semantic Web, Web Intelligence, pages 37-58. Springer, 2003.
- [16] Srinin Narayanan and Sheila A. McIlraith, Semantic web services: Simulation, verification and automated composition of web services, In Proceedings of the Eleventh International World Wide Web Conference, pages 77-88, ACM Press, 2003.
- [17] Anupriya Ankolekar, Frank Huch, and Katia Sycara, Concurrent Execution Semantics for DAML-S with Subtypes, In The First International Semantic Web Conference, 2002.
- [18] <http://www.daml.org/services>
- [19] Benjamin Grosf, Mahesh D. Gandhe, and Timothy W. Finin, SweetJess: Translating DamlRuleML to Jess, Proc. International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, Sardinia (Italy), June 2002.
- [20] Benjamin N. Grosf, Ian Horrocks, Raphael Volz, and Stefan Decker, Description Logic Programs: Combining Logic Programs with Description Logic, In Proc. of 12th Intl. Conf. on the World Wide Web (WWW-2003), Budapest, Hungary, May 20-23, 2003.
- [21] The Semantic Web, Tim Berners-Lee Academic discussion, Japan Prize 2002
- [22] <http://herzberg.ca.sandia.gov/jess/>
- [23] T.C. Son S. McIlraith and H. Zeng, Semantic Web Services. IEEE Intelligent Systems, Special Issue on the Semantic Web, 16(2):46-53, 2001.