

유비쿼터스 환경의 상황인식 프레임워크를 위한 온톨로지 모델링

The Ontology Modeling for Situation-Awareness Framework in Ubiquitous Environment

이기철¹, 이지형²

성균관대학교 컴퓨터공학과

E-mail: ¹ lee77@skku.edu, ² jhlee@ece.skku.ac.kr

요 약

온톨로지는 뛰어난 확장성과 다양한 표현력 등 많은 장점을 갖고 있기 때문에 이를 이용한 유비쿼터스 환경 구축이 최근 제기되고 있고 이러한 유비쿼터스 환경에서의 다양한 정보(컨텍스트)들을 수집하고 분석하기 위해서 상황인식 시스템의 필요성이 제기되고 있다. 이러한 이유로 몇몇 프로젝트에서 온톨로지를 이용하여 상황인식 미들웨어를 제작하였지만 이러한 기존 미들웨어의 온톨로지 모델은 추론과 학습 서비스에 대한 고려가 부족하였고 또한 W3C에서 최근 제안한 SWRL(Semantic Web Rule Language)[1]이 고려되지 않았다.

그러므로 본 논문에서는 유비쿼터스 환경에서의 상황인식 프레임워크의 학습 및 추론서비스 향상을 위해 컨텍스트의 타입을 정의하였고 또한 SWRL을 이용하여 규칙을 표현할 수 있도록 온톨로지를 모델링 하였다.

Key Words : 유비쿼터스, Situation-Awareness, Ontology, SWRL

1. 서 론

현재 W3C(World Wide Web Consortium)에서는 OWL과 Rule-ML을 이용하여 SWRL이라는 언어를 제안하였다. 이는 기존의 OWL만으로는 표현할 수 없었던 규칙을 포함 하여 표현할 수 있도록 확장한 것이다 [1].

또한 유비쿼터스 환경은 사용자 중심의 환경이나 주변 상황으로부터 획득한 컨텍스트를 수집하고 분석하여 사용자의 요구에 맞추어 지능적인 서비스를 할 수 있도록 하기 위한 상황인식 시스템이 필요하다.

이러한 상황인식 시스템을 구축하기 위한 다양한 방법론이 제기 되었지만 최근에는 온톨로지 기반의 상황인식 시스템이 제기되고 있다.

다음 2장에서는 온톨로지에서의 사용하는 언어와 이의 활용한 추론 그리고 이러한 온톨로지를 활용한 상황인식 프레임워크에 대해 살펴본 후, 3장에서 본 논문에서 제안하는 컨텍스트 타입의 정의 그리고 상황인식 프레임워크를 위한 온톨로지의 모델과 SWRL의 정의에 대해 논하고 마지막 결론 및 향후 연구로써 활용방안에 대해 논하도록 하겠다.

온톨로지는 현재 W3C를 중심으로 세계 각국의 기관 및 기업에서 활발히 연구 중인 분야이다. 이는 수많은 웹상의 정보들 간의 관계를 이용하여 웹에 의미를 부여하고 이를 이용하고자 하는 시멘틱 웹을 목표로 하여 발전해 가고 있다. 이러한 연구는 XML(eXtensible Markup Language)기반인 RDF(Resource Description Framework), OWL, Rule-ML, SWRL와 같은 여러 가지 언어를 만들어 내었고 이러한 언어를 각종 규칙기반 추론엔진(Rulebase Inference Engine), 로직(Loginc)언어 등을 이용하여 추론 서비스를 제공하기 위한 여러 가지 연구가 진행되고 있다.

또한 유비쿼터스 컴퓨팅 환경에서 사용자 중심적 서비스를 제공하기 위해서는 사용자의 현재 상황과 주변 상황정보를 이용해서 사용자가 원하는 서비스가 무엇인지 파악하는 역할을 담당하는 상황인식 어플리케이션이 필요하다.

이러한 상황인식 어플리케이션은 다양한 정보를 수집하는 센서와 각종 어플리케이션과 사용자간의 중간 역할을 담당하게 된다. 또한 각종센서와 어플리케이션으로부터의 정보를 기반으로 학습하고 추론하고 가공하여 사용자가 필요로 하는 서비스를 제공해 줄 수 있도록 한다.

2. 관련 연구

2.1 온톨로지 언어

온톨로지를 개발하기 위한 언어로 W3C에서 제안

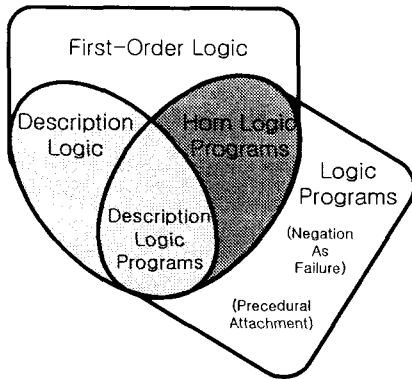
하는 여러 가지 언어가 있다. 그중 SWRL은 OWL과 Rule-ML을 확장한 언어이며 OWL에 규칙에 대한 표현을 할 수 있도록 확장한 언어이다. 예를 들면, [그림1]과 같은 형태의 규칙을 SWRL언어로 표현할 수 있다. OWL과 SWRL형태로 표현한 온톨로지와 추론엔진을 이용하여 새로운 정보를 추론하는데 활용할 수 있다.[1]

Rule-3 :
hasParent(?x1,?x2) ∧ hasBrother(?x2,?x3) → hasUncle(?x1,?x3)

[그림 1] SWRL Rule

2.2 온톨로지에서의 SWRL추론

온톨로지의 추론을 위한 로직언어의 표현 범위는 [그림2]와 같이 크게 LP(Logic Programs)와 FOL(First Order Logic)로 나눌 수 있다. 두 가지 언어 모두 Description Logic Program에 대한 표현은 가능하지만, First-Order Logic과 Logic Program은 각각의 표현의 범위를 갖고 있다. 이러한 이유로 로직 언어를 온톨로지 상에 적용할 수 있도록 온톨로지언어를 확장하는 연구가 활발히 진행 중이며 그 연구 결과의 하나로 SWRL과 같은 언어를 들 수 있을 것이다.[2]



[그림 2] 논리 지식의 표현범위[2]

SWRL을 통하여 추론 기능이 확장 된 온톨로지의 예를 들면, 다음 [그림 3]와 같은 Family 온톨로지를 예로 들 수 있다. 이 예는 OWL 온톨로지를 이용하여 DL-Rule을 정의하고 SWRL의 Rule을 정의한 것이다. 이렇게 구성한 DL과 Rule을 이용해서 온톨로지 상의 개체(Individual)에 대한 추론이 가능하다. 여기서는 DL추론을 위하여 RACER를 이용하였고 SWRL의 추론을 위하여 Jess를 이용하였다. 이를 이용한 추론 결과는 [표 1]과 같다. 이 온톨로지는 Man과 Woman클래스에 남자 10명(M1~M10)과 여자 10명(F1~F10)의 개체를 초기화 하였다. 표의 왼쪽 부분은 RACER를 이용한 추론 결과이다. 또한 Jess를 이용하여 SWRL Rule을 추론한 결과가 오른쪽에 표시되어 있다. 이와 같이 DL언어를 이용한 추론과 이에 SWRL 규칙을 포함하여 그 규칙에 대한 추론결과를 확인 할 수 있다[3].

이러한 온톨로지 언어의 확장으로써 좀 더 다양한 정보의 온톨로지를 추론하여 사용자에게 제공할 수 있게 되었다.

OWL ontology	
-1- Person	:= Man ∪ Woman
-2- Parent	:= Person ∩ hasChild ≥ 1
Child	:= Person ∩ hasParent ≥ 1
(hasChild) ⁻¹	= hasParent
-3- Father	:= Parent ∩ Man
Mother	:= Parent ∩ Woman
-4- Son	:= Child ∩ Man
Daughter	:= Child ∩ Woman
-5- Brother	:= Sibling ∩ Man
Sister	:= Sibling ∩ Woman
-6- Nephew	:= Man ∩ (hasUncle ≥ 1 ∪ hasAunt ≥ 1)
-7- Relative	:= Child ∪ Parent ∪ Aunt ∪ Nephew ∪ Niece ∪ Uncle ∪ Sibling
SWRL rules	
Initial rule base	
-8-	hasParent(?x1,?x2) ∧ hasConsort(?x2,?x3) ⇒ hasParent(?x1,?x3)
-9-	hasParent(?x1,?x2) ∧ hasSister(?x2,?x3) ⇒ hasAunt(?x1,?x3)
-10-	hasParent(?x1,?x2) ∧ hasParent(?x3,?x2) ∧ differentFrom(?x1,?x3) ⇒ hasSibling(?x1,?x3)
-11-	hasSibling(?x1,?x2) ∧ hasDaughter(?x2,?x3) ⇒ hasNiece(?x1,?x3)
Rules mirroring the ontology knowledge	
MR1-	hasSibling(?x1,?x2) ∧ Man(?x2) ⇒ hasBrother(?x1,?x2)
MR2-	hasSibling(?x1,?x2) ∧ Woman(?x2) ⇒ hasSister(?x1,?x2)
MR3-	hasParent(?x1,?x2) ∧ Man(?x2) ⇒ hasFater(?x1,?x2)

[그림 3] Family OWL 온톨로지와 SWRL Rule

Class	Facts after RACER	After RACER and Jess
		Initial SWRL rule base
1. Person	M1->M10, F1->F10 [20]	M1->M10, F1->F10 [20]
2. Man	M1->M10 [10]	M1->M10 [10]
3. Woman	F1->F10 [10]	F1->F10 [10]
4. Parent	M1, M2, M3, M4, M6, M7, M8 [7]	M1, M2, M3, M4, M6, M7, M8, F1, F3, F4, F6, F7, F8, F10 [14]
5. Father	M1, M2, M3, M4, M6, M7, M8 [7]	M1, M2, M3, M4, M5, M6, M7, M8 [7]
6. Mother		F1, F3, F4, F6, F7, F8, F10 [7]
7. Child	M2, M3, M4, M5, M6, M9, M10, F2, F3, F5, F6, F9 [12]	M2, M3, M4, M5, M6, M9, M10, F2, F3, F5, F6, F9 [12]
8. Son	M2, M3, M4, M5, M6, M9, M10 [7]	M2, M3, M4, M5, M6, M9, M10 [7]
9. Daughter	F2, F3, F5, F6, F9 [5]	F2, F3, F5, F6, F9 [5]
10. Sibling		M2, M3, M5, M6, F2, F3, F5, F6 [8]
11. Brother		M2, M3, M5, M6 [4]
12. Sister		F2, F3, F5, F6 [4]
13. Relative	M1->M10, F2, F3, F5, F6, F9 [15]	M1->M10, F1->F10 [20]
		Additional mapping rules
14. Uncle		M2, M5, M6 [3]
15. Aunt		F2, F3, F5, F6 [4]
16. Nephew		M3, M5, M6, M9, M10 [5]
17. Niece		F5, F6, F9 [3]

[표 1] RACER and Jess Inferences

2.3 상황인식 프레임워크에서의 온톨로지

상황인식 프레임워크에서 온톨로지를 활용함으로써 얻을 수 있는 장점은 다음과 같다.

- 컨텍스트 정보의 공유
- 컨텍스트 정보의 재사용성
- 컨텍스트 정보의 추론
- 확장성과 상호운용성

이러한 온톨로지의 장점을 이용한 온톨로지 기반의 상황인식 아키텍처로는 CoBra4[4](Context Broker Architecture)와 SOCAM[5](Service-Oriented Context-Aware Middleware) 등이 있다.

이 두 가지 상황인식 프레임워크 모두 온톨로지를 사용하여 온톨로지의 장점인 정보공유, 재사용성, 추론, 상호운용성 등을 갖는다. 하지만 최근 제안한 SWRL이 포함되어있지 않아 상황인식의 컨텍스트 정보에 대한 추론 기능에 제한이 있고 학습과 추론을 고려하지 않은 형태의 온톨로지 모델을 갖고 있기 때문에 지능형 서비스에 대한 제한이 있다고 할 수 있다.

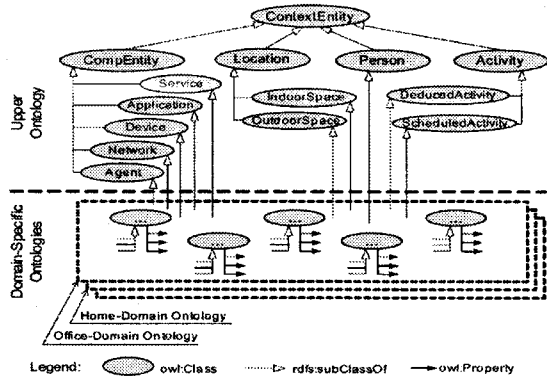
이와 같이 온톨로지를 상황인식 시스템에 사용하기 위해서는 상황인식 시스템에 적당한 온톨로지의 모델링이 필요하다. 이러한 상황인식 컨텍스트의 모델링을 위한 연구는 크게 세 가지로 나눌 수 있다.

첫 번째로, Application-Oriented Approach 이다. 이는 가장 원시적인 형태의 상황인식 컨텍스트 모

델링의 방법으로써 지식의 표현력, 정형성 (Formality)이 부족하다. 이러한 형태의 상황인식 미들웨어로는 상황인식 미들웨어의 출발점인 Context-Toolkit[6]이 있다.

두 번째는, Model-Oriented Approach이다. 이는 개념적인 모델링 기법을 이용하여 모델링 하는 방법으로, ER Model 또는 UML Diagram을 이용하는 방법이다. 이는 일반적인 어플리케이션의 모델링 기법으로써는 적당하지만 지능적인 상황인식 어플리케이션에는 정보의 표현력이 부족하다.

마지막으로 Ontology-Oriented Approach는 온톨로지의 장점인 확장성, 정형화된 모델, 다양한 지식의 표현, 정보의 추론 등을 모두 갖게 되어 지능형 상황인식 어플리케이션에 가장 적당하다고 볼 수 있다. 예를 들면, [그림 6]과 같은 Upper-Domain 모델의 형태로 상위온톨로지를 정의하고 이를 하위도메인 레벨의 온톨로지에 활용하는 방법이다. 이는 상위 온톨로지를 통하여 일반적인 상황인식 정보를 표현하고, 하위 도메인온톨로지에서는 세부적인 상황인식 정보를 표현하게 된다. 또한 이러한 Upper-Domain 온톨로지 모델은 도메인 온톨로지를 쉽게 구성할 수 있으며, 상황인식 어플리케이션 제작에 시간을 절약할 수 있는 장점이 있다.[5]



[그림 4] Upper-Domain Ontology Model

3. 상황인식 온톨로지

앞에서 논의한 상황인식 프레임워크와 온톨로지의 내용을 종합하여보면 현재까지의 상황인식 프레임워크를 위한 온톨로지는 OWL과 RDF를 이용하여 온톨로지를 표현하였고, 이를 이용하여 온톨로지를 모델링하였다. 하지만 최근, 이러한 온톨로지 언어에 대한 표현의 한계로 인하여 SWRL언어가 제안되었고 이를 활용하기 위한 연구들이 진행되고 있다. 그러므로 본 논문에서는 상황인식 프레임워크를 위한 온톨로지를 SWRL을 이용하고 위에 정의한 컨텍스트 타입을 통한 온톨로지를 모델링하여 이를 상황인식 시스템에 이용하기 위한 방법을 제안하고자 한다.

3.1 상황인식 프레임워크에서의 컨텍스트 정의

상황인식 어플리케이션을 구축하기 위해서 컨텍스트를 정의할 필요가 있다. 본 논문에서는 컨텍스트의 타입을 다음과 같이 크게 네 가지 형태로 정의하였다.

- Sensed : 센서를 통해 얻는 정보
- Combined : 센서된 컨텍스트의 계산에 의한 정보

- Inferred : 추론엔진을 통한 추론된 컨텍스트
- Learned : 사용자의 성향 및 선호도 등을 고려하기 위한 학습된 컨텍스트

위와 같이 정의한 컨텍스트는 온톨로지 모델중 Context클래스의 하위 클래스로 분류되어 각각의 특징에 따른 처리방식을 정할 수 있도록 하였다. 예를 들어 [그림 5]에서 보듯이 Sensed클래스의 경우 PersonID, Name, Weight, Height등을 하위 클래스로 정의하여 이러한 컨텍스트의 정보들을 센서로부터 입력받는 경우 일관된 처리를 할 수 있으며, Combined의 경우 StandWeight클래스를 하위로 정의하여 SWRL규칙에 의해 계산된 정보가 새로운 컨텍스트가 될 수 있도록 하였다.

이러한 컨텍스트 타입의 정의는 사용자에게 지능적인 서비스를 제공하는 상황인식 어플리케이션을 좀 더 쉽고 일관된 형태의 표준을 이용하여 제작할 수 있도록 하여 온톨로지의 특징인 확장성과 재사용성을 최대한 이용할 수 있도록 한다.

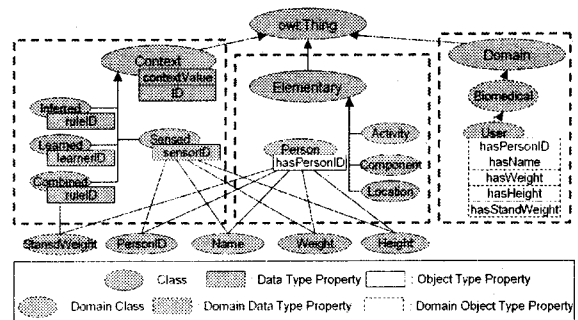
3.2 UED(Upper-Elementary-Domain) Model

SOCAM에서 정의한 온톨로지의 모델인 Upper-Domain온톨로지를 확장하여 UED Model 형태의 온톨로지를 구성하였다. [그림 7]에서 보듯이 상위(Upper-Level) 온톨로지와 하위(Domain-Level)온톨로지로 나뉘게 되며 상위는 실선으로, 하위온톨로지는 점선으로 나타내었다.

상위 온톨로지의 Context 클래스는 contextValue와 ID라는 속성을 갖게되며 ID는 Object타입의 속성으로써 모든 컨텍스트의 정보가 유일성을 갖을 수 있도록 정의하였다. 또한 Context클래스를 Sensed, Combined, Inferred, Learned의 네 가지 클래스로 나누어 컨텍스트의 정보를 분류하였다. 이는 이전 장에서 분류한 컨텍스트의 정보를 표현함으로써 각 컨텍스트의 분류형태에 따라 특징에 맞는 처리 과정을 수행할 수 있도록 분류하였고 각각의 속성을 정의하였다. StandWeight, PersonID, Name, Weight, Height와 같은 기본적인 컨텍스트 정보를 컨텍스트의 성질에 따라 각각 하위클래스로 구성하였다.

다음으로 Elementary 클래스의 하위 클래스로 Person, Activity, Component, Location으로 분류하여 SOCAM에서 정의한 Upper-Domain온톨로지 모델링 형태를 취함으로써 컨텍스트의 정보를 다양하게 분류할 수 있도록 하였다.

Domain클래스는 [그림 8]에서와 같이 하위도메인 온톨로지를 쉽게 구성할 수 있고, 어플리케이션 제작에 시간을 절약할 수 있도록 하였다.



[그림 5] UED Ontology Model

UED Model의 가장 큰 특징은 User클래스를 도메인 온톨로지에 해당하는 클래스의 예로써 상위 온톨로지 및 도메인 온톨로지에서의 정의한 각각의 컨텍스트 클래스(StandWeight, PersonID, Name, Weight, Height)의 Object-Type의 속성을 갖게 되며 이러한 속성은 컨텍스트 클래스의 인스턴스를 값으로 갖게 된다. 이러한 확장 방식을 통하여 도메인 온톨로지에서의 컨텍스트 정보를 확장해 나갈 수 있다. 또한 StandWeight클래스의 정의와 같이 다음 장에서 이야기할 SWRL규칙을 이용하여 Combined컨텍스트와 Inferred컨텍스트의 정보를 지능적으로 처리할 수 있다는 장점이 있다.

3.3 SWRL규칙 정의

위와 같은 형태의 온톨로지를 구성한 후 StandWeight클래스에 대한 SWRL규칙의 정의가 필요하다 즉, StandWeight클래스의 인스턴스는 SWRL규칙에 의해서 생성되는 것이다.

이러한 SWRL의 정의는 StandWeight뿐만 아니라 어떠한 도메인의 상황인식 온톨로지를 작성하여 Combined또는 Inferred컨텍스트로 분류하여 정의함으로써 규칙에 대한 정보의 표현이 가능하며 이와 같이 정의한 SWRL규칙에 의해서 상황인식 프레임워크의 향상된 지능적인 서비스가 가능하다고 할 수 있다.

작성한 규칙은 다음과 같다.

```
Height(?a) ^ contextValue(?a,?y) ^
PersonID(?x) ^ hasPersonID(?a,?x) ^
StandWeight(?d) ^ hasPersonID(?d,?x) ^
swrlb:subtract(?b,?y,100.0) ^ swrlb:multiply(?c,?b,0.8)
→ contextValue(?d, ?c)
```

이는 다음과 같은 표준폼무게(StandWeight)의 값을 구하기 위한 규칙이다.

표준폼무게 = (키 - 100) * 0.8

3.4 규칙기반 추론엔진

우선, 이러한 시스템을 구축하기 위해서 규칙기반 추론엔진(Rule-Based Inference Engine)이 필요하다. 본 논문에서 작성한 온톨로지 모델의 특징 중 하나는 OWL기반의 온톨로지를 구성하고 SWRL를 이용하여 규칙을 표현하였다는 것이다. 이와 같은 OWL기반의 SWRL온톨로지를 추론하기 위한 추론엔진으로는 Bossam[7]과 Jess[8]를 이용하는 방법이 있다.

Bossam은 한국전자통신연구소(ETRI)에서 제작한 규칙기반 추론엔진으로써 RDF, OWL, SWRL의 온톨로지를 추론할 수 있으며 셸명령(Shell-Command)의 추론과 Java Embedded.프로그래밍이 가능하다.

Jess또한 규칙기반 추론엔진으로써 OWL과 SWRL형태의 온톨로지를 추론하기 위해서는 Protege의 플러그인중 하나인 OWL-Plugin의 SWRLTab을 이용하는 방법이다. SWRLTab은 SWEL Editor, SWRL Factory, SWRL Bridge, SWRL Built-in Bridge를 통하여 Jess추론엔진과 연동하여 추론을 수행할 수 있으며 이 또한 Java API를 제공하여 Java Embedded 프로그래밍이 가능하다.[9]

4. 결론 및 향후연구

두 추론엔진을 이용하여 본 논문에서 정의한 UED온톨로지 모델을 Biomedical 도메인을 통하여 테스트 하였고 User클래스의 인스턴스를 생성하여 위 두 엔진 중 Bossam을 이용하여 수행한 결과 StandWeight의 contextValue로 계산식에 의한 결과 값이 생성되는 것을 확인할 수 있었다. 하지만 SWRLTab의 경우는 현재 SWRL에서 사용하고 있는 계산식(swrl:subtract, swrl:multiply)을 통한 추론이 불가능한 상태이다.

본 논문에서는 상황인식 온톨로지의 모델을 정의하였고 이를 이용하여 Biomedical도메인의 사용자를 정의하고 SWRL을 정의하여 이를 이용한 추론을 수행하였다. 이를 이용하여 향후, UED모델 온톨로지과 SWRL규칙을 이용한 지능형 상황인식 미들웨어를 구현할 예정이다.

참고 문헌

- [1] SWRL: A Semantic Web Rule Language Combining OWL and SWRL (<http://www.w3.org/Submission/SWRL/>)
- [2] B.Groszof and M.Dean, "Semantic Web Rules with Ontologies, and their E-Business Applications," ISWC2004-Tutorial (http://sweetrules.projects.semwebcentral.org/iswc_tutorial.pdf)
- [3] C.Golbreich, "Combining Rule and Ontology Reasoners for the Semantic Web," Rules and RuleMarkup Languages for the Semantic Web (RuleML2004), p.6~22, Nov. 2004.
- [4] Harry Chen, Tim Finin, and Anupam Joshi, "An Ontology for Context-Aware Pervasive Computing Environments," Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, Vol.18, pp.197-207, September 2003
- [5] T.Gu et al, "An Ontology Based Context Model in Intelligent Environments," Proc. of Communication Networks and Distributed Systems Modeling and Simulation Conference, 2004
- [6] A.K.Dey, G.D.Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Application," Human Factors in Computing Systems, pp.434-411, 1999
- [7] Bossam Rule/OWL Reasoner (<http://mknows.etri.re.kr/bossam/>)
- [8] The Rule Engine for the Java Platform (<http://herzberg.ca.sandia.gov/jess/>)
- [9] M.J.O'Connor, H.Knublauch, S.W.Tu, B.Groszof, M.Dean, W.E.Grosso, M.A.Musen, "Supporting Rule System Interoperability on the Semantic Web with SWRL," Fourth International Semantic Web Conference(ISWC2005), Galway, Ireland, 2005.