

시나리오 테스트를 통한 온톨로지 기반 지식 평가 기법¹⁾이건수⁰ 김민구아주대학교 정보통신전문대학원
{lks7256⁰, minkoo}@ajou.ac.kr

Ontology-based Knowledge Evaluation Method using Scenario Testing

Keonsoo Lee⁰, Minkoo Kim

Graduate School of Information and Communication, Ajou University

요 약

오늘날 컴퓨팅 기능의 급격한 발달은, 시스템의 효율뿐만 아니라 지능화에 대한 요구를 증대시키고 있다. 지능화된 서비스를 성공적으로 제공하기 위해서는, 제공하려는 도메인에 대한 지식 처리 기능이 필수적이다. 시스템의 지능적인 행동은 그 시스템이 보유하고 있는 지식의 질(質)에 영향을 받기 때문에, 사용되는 지식의 질적 우수성을 보장하려는 시도가 이루어지고 있지만, 문제는 지식의 질이 높아질수록, 지식 관리의 비용이 증가한다는 점이다. 따라서 품질과 비용 사이의 균형점을 맞추기 위한 노력이 필요하다. 이에 본 논문에서는 사용하려는 지식의 품질과 시스템에서 요구하는 지식의 품질을 비교함으로써, 이 균형점을 맞추는 방법을 제안한다. 이를 위해, 시스템이 제공하려는 서비스 시나리오를 대상 지식으로 표현 및 처리한다. 시스템 관리자는 이 결과를 바탕으로 채택하려는 지식과 자신의 시스템 사이의 괴리를 손쉽게 판단할 수 있고, 그 차이를 없애기 위해 지식의 확장 및 축소의 부분을 결정하고, 수정할 수 있다.

1. Introduction

오늘날 컴퓨팅 기능의 급격한 발달은 유비쿼터스 컴퓨팅으로 대표되는 새로운 컴퓨팅 환경을 탄생시켰다. 소위 5ANY로 정의되는 유비쿼터스 컴퓨팅 환경은 시/공간에 대한 제약 없이 사용자가 자유롭게 네트워크에 접근하여 원하는 서비스를 제공받을 수 있는 환경을 의미한다 [1]. 이러한 유비쿼터스 컴퓨팅 환경의 특성 중 하나는, 사용자로부터 명시적인 요청뿐만 아니라, 동적으로 변화하는 서비스 수행 환경을 인지하여, 서비스가 수행되는 시점의 상황을 반영하여 서비스의 동작 과정을 적절히 조절하는 것이다. 소위 상황 인지(Context Aware)라고 표현되는 이러한 특성은 시스템의 지능화를 부추기는 주요 원인 중에 하나로 꼽을 수 있다 [2]. 가령, 방안의 사용자가 추위를 느껴 온도를 높여달라는 서비스 요청을 했다고 가정해 보자. 이 경우, 사용자가 추위를 느끼게 되는 원인을 없애주면 해결되는 문제로 볼 수 있다. 그러나 실제 환경에서 서비스 수행의 목표가 되는 원인의 종류의 다양함은 기존의 정적 컴퓨팅의 한계를 보여준다. 가령 사용자가 감기에 걸렸을 수도 있고, 환기를 위해 열어놓은 창문이 그 원인일 수도 있다. 혹은 단순히 날이 추워졌기 때문일 수도 있다. 이처럼 사용자의 요청을 수행하려는 그 순간의 상황이 서비스 수행의 중요한 고려사항이 된다면, 또 그 상황이 동적으로 변화 가능하다면, 정적 조건 이상의 도메인 모델이 필요하고,

이러한 도메인 모델 관리는 도메인 정보의 지식화 과정을 통해 그 효율을 높일 수 있다.

이처럼, 지식화를 통한 동적 환경에서의 시스템의 지능적인 행동을 지원하려고 할 때, 지식화의 수준을 고려해야 한다. 이는 지식화의 수준과 그 지식의 관리 비용 사이의 비례관계가 존재하기 때문이다. 가령, Description Logic을 사용한 지식 관리의 경우, SHIQ의 표현 범위를 갖는 경우, ALC로 표현된 지식의 관리보다 기하급수의 복잡도를 갖는다 [3]. 따라서, 시스템의 지능적 행동에 필요한 지식의 수준과 그 지식을 관리하기 위한 비용 사이의 균형점(Trade-off)을 찾기 위한 방법이 필요하다. 그 시스템에서 필요로 하는 지식의 수준을 찾는 것은 곧 시스템의 운용 효율을 결정짓는 조건이 되기 때문이다. 이에 본 연구에서는 지식화를 위한 후보 모델들 중에서 각 시스템이 필요로 하는 지식수준을 만족시키는 가를 평가하고, 부족한 지식을 확장하거나 초과되는 지식을 간결화 시키기 위해 지식의 확장성을 검증하는 방법을 제안한다.

2. Knowledge Engineering

지식 공학은 지식 기반 시스템(Knowledge-based System)을 개발, 유지 및 관리하기 위한 방법론으로 통상 다음의 7단계의 개발 흐름을 따른다 [4]. 이러한 지식 공학 방법론은 기존의 소프트웨어 공학 기법과 그 대상만을 달리할 뿐 유사한 흐름을 따르고 있다. 다만, 지식이라는 설계 대상의 차이를 가진다. 지식은 “체계화되고 개념화된 정보”로 정의되며, 이러한 정보를 표현, 관리하기 위한 다양한 기법들이 존재한다 [5]. 이때, 해결하려는 문제 도메인에 따라 지식의 형태는 달라지고, 또

1) This research is supported by the ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

지식의 형태에 따른 표현 방식이 달라지고, 표현 방식의 차이에 따라 그 지식을 활용 및 추론 하는 기법들이 달라진다. 즉 지식 공학의 목표는 문제를 해결하기 위해, 어떤 지식을 어떻게 표현하고, 그 표현된 지식을 어떻게 활용할 것인가를 결정하는 것이라고 볼 수 있다.

<표 1> 지식 공학의 개발 단계

개발 단계	개발 내용
문제 분석	시스템이 해결할 문제를 정의 한다
지식 기반 시스템 설계	시스템의 디자인 및 구현
지식 베이스 설계	시스템에서 사용할 지식의 형태를 결정 한다
관련 세부 정보 습득	지식 베이스에 저장하기 위한 관련 정보를 수집하고 이를 토대로 지식화 한다
지식 검증	생성된 지식의 유효성을 검증한다
시스템 운용	검증된 지식을 바탕으로 실제 시스템을 사용한다
시스템 개정	새로운 지식으로 확장시킨다

3. Ontology as Knowledge Representation

시스템의 지능적 행동을 보장하기 위해 어떤 지식을 사용할 것인지를 결정하는 것은 도메인 전문가의 몫이다. 지식의 대상이 결정되면, 이 지식의 표현 방법을 선택하게 된다. 지식 표현 시 고려해야 하는 것은 표현의 범위와 표현 기법이다. 일반적으로 표현 범위가 상세화 될수록, 지식의 활용도는 높아지지만, 관리/유지비용은 증가하게 된다. 또한 지식 활용의 대상에 의해 표현 기법이 정해지고, 이렇게 정해진 지식 표현의 방법에 따라 추론의 기법이 결정된다.

지식의 표현 방법은 크게 FOPC(First Order Predicate Calculus)기반의 논리를 사용하는 것과 Frame 기반의 객체지향 기법을 사용하는 것으로 구분할 수 있는데, 전자의 경우, 명제논리(Propositional Logic), 술어논리(Predicate Logic), Production System 등이 대표적인 예로 표현이 쉽고, 추론하는 방식이 명확하며 다양한 규칙들을 만들 수 있다는 장점이 있지만, 표현 범위의 자유도가 높기 때문에 기존 지식의 재활용이 어렵다. 반면 후자의 경우, 객체들과 객체들 사이의 관계를 정의하는 것으로 지식을 표현한다. Frame, 의미망(Semantic Network), 온톨로지(Ontology) 등이 대표적인 예다. 이 방법은 지식 자체를 표현하는 ABox(Assertional Box)와 지식을 표현하기 위한 틀을 표현하는 TBox(Terminological Box)로 구성된다. 명시적인 틀이 정해져 있기 때문에, 지식 재활용 및 지식 공유에 유용하지만, 틀을 벗어나는 지식이나 추론이 불가능하다는 단점이 있다. 가령, “이 혼을 하면 부부가 아니다” 라는 지식은 Frame 기반 지식 표현 기법에서는 표현할 수 없다. 본 연구에서는 이처럼 다양한 지식 표현의 방법 중, 온톨로지로 표현된 지식을 대상으로 평가 방법을 제안한다.

온톨로지는 개념에 대한 명시적인 상세 명세서로 정의된다 [6]. 원래 철학에서 형이상학을 지칭하는 의미로 쓰

였지만, 오늘날 인식의 틀로서 개념과 개념들 사이의 관계를 설명하기 위한 표현 방법으로 사용되고 있다. 온톨로지는 주체(Subject), 서술어(Predicate), 그리고 대상(Object)의 삼자 관계 구조 (Triple Structure)로 표현되는데, 주체와 대상을 객체(Concept)이라고 하고, 서술어를 관계(Role)이라고 지칭한다. 객체와 관계의 표현 범위는 Description Logic에 따라 그 표현 범위가 달라지는데, 일반적으로 ALC에서 SHIQ의 범위 안에서 주로 사용된다. <표 2>는 Description Logic의 표현 범위에 따른 단계를 보여준다.

<표 2> Description Logic Family

DL Family	expressive power	semantics
S	AL	atomic concept universal concept bottom concept atomic negation intersection value restriction limited existential quantification
	C	negation of arbitrary concepts
	R+	transitive roles
U	$C \sqcup D$	union
E		full existential quantification
N	$\geq nR, \leq nR$	number restriction
H		role hierarchy
I	R^{-}	inverse role
Q	$\geq nRC, \leq nRC$	qualified number restriction
O	$ = 1$	nominal

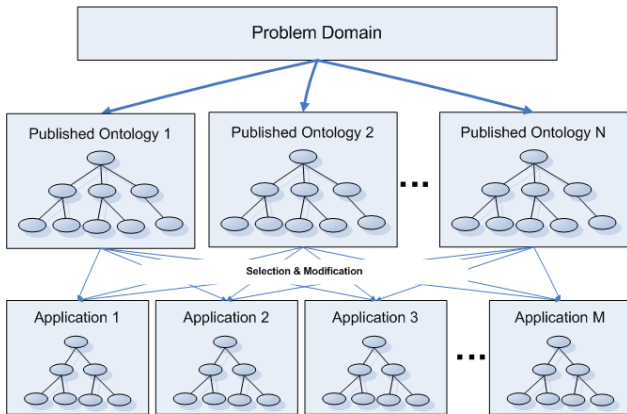
이러한 온톨로지는 시멘틱 웹의 등장과 함께 지식 공학의 방안으로 각광 받고 있다. 이는 온톨로지의 TBox를 공유하는 것만으로 다른 시스템에서 사용하는 지식의 의미를 손쉽게 채용할 수 있기 때문이다.

4. Scenario-Testing for Knowledge Evaluation

4.1 Ontology Evaluation

온톨로지를 평가함에 있어 크게 2가지 측면에서 바라볼 수 있는데, 그 첫 번째 관점은 온톨로지 자체의 가치를 평가하는 것이다. 두 번째 측면은 온톨로지를 사용함에 있어, 적용 도메인에서의 온톨로지 활용 측면에서의 가치를 평가하는 것이다. 첫 번째 관점에서는 온톨로지가 얼마나 잘 만들어졌는가를 판단한다 [7, 8]. 이때 사용되는 판단 기준으로는 1. 어떤 객체들이 정의되었는지, 2. isA 관계가 적합한지, 3. hasA 관계가 적합한지, 4. 문법에 맞는 정의가 이루어졌는지, 5. 온톨로지의 구조가 올바른지 등을 평가한다. 평가 방법으로는 소위 Golden Ontology라고 불리는 정답 온톨로지가 있다고 가정하고 이 정답에 얼마나 유사한지를 판단하거나, 관련 온톨로지들을 수집하여 얼마나 평균에 가까운지를 그 기준으로 삼는다. 그러나, 지식 공학의 측면에서 특정 도메인에서

의 Golden Ontology가 존재한다고 가정하기 힘들고, 특정 도메인에서 발생하는 모든 문제에 적합한 일반 온톨로지 (General Ontology)의 효용성에도 문제가 존재한다 [9]. 실제 동일 도메인에서도 다양한 정보 제공자들은 자신의 온톨로지를 공개하고, 각각의 어플리케이션들은 그 온톨로지를 참조하여 타인의 지식을 공유 및 활용한다. <그림 1>은 특정 도메인에서 공개된 다양한 온톨로지들을 참조하여 자신의 어플리케이션에 맞도록 수정/보완 과정을 거쳐 사용하는 과정에 대한 개념도이다.



<그림 1> 온톨로지를 통한 지식 공유

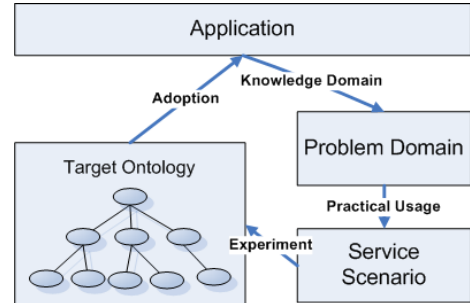
4.2 Scenario Testing for Domain Relevance

이처럼, 온톨로지 자체가 올바르게 혹은 정확하게 만들어졌는지 평가하는 것보다, 공개된 온톨로지를 참조하여 자신의 용도에 맞게 재활용한다는 관점에서 기존의 다양한 온톨로지중에서 가장 적합한 온톨로지를 찾아내는 방법으로 시나리오 테스트 기법을 제안한다. 시나리오 테스트 기법은 소프트웨어 검증과 유사한 흐름을 갖는다. 소프트웨어 검증은 프로그램의 버그를 찾아내기 위한 기법으로, 프로그램 명세서의 기능을 올바르게 수행하는가를 확인하는 과정이다 [10]. 이 과정에서 버그로 인지되는 동작은 다음의 네 가지 경우로 구분된다. 첫째, 명세서에 기술되었지만, 수행되지 않는 경우. 둘째, 명세서에 기술되지 않은 동작을 수행하는 경우. 셋째, 명세서에 수행하지 않는다고 기술된 동작을 수행하는 경우. 마지막으로 명세서에 기술되어 있지는 않지만, 당연히 수행되어야 하는 동작을 수행하지 않는 경우. 이처럼, 검증의 기준은 명세서가 기준이 된다.

지식의 경우라면, 명세서는 지식을 이용해 해결하려는 문제, 즉 시스템의 수행 시나리오가 되고, 명세서 확인 과정은 그 시스템이 계획된 서비스를 수행하는 과정에서 그 지식이 해결하려는 문제를 정확히 표현할 수 있는가를 확인하는 과정이 된다. <그림 2>는 어플리케이션이 자신이 필요로 하는 온톨로지를 확인하는 과정을 보여준다.

어플리케이션은 자신의 지식 도메인에서 해결하려고 하는 문제를 분석하여 정리한다. 정리된 문제를 해결하기 위한 시나리오를 만들고, 그 시나리오를 기존의 배포된 온톨로지들을 적용하여 표현한다. 적용 결과를 통해

기존 온톨로지를 사용할 때, 부족한 지식이나, 불필요한 지식을 찾아내게 되고, 이를 통해 참조 온톨로지를 어플리케이션에 맞게 수정하고, 수정된 온톨로지를 채택하여 지식을 활용한다.



<그림 2> 시스템 관점에서의 온톨로지의 적합성 확인 과정

시나리오 작성 시 사용되는 어휘는 검사하려는 온톨로지가 가지고 있는 객체를 이용해야 함을 가정하지만, 온톨로지간 어휘 맵핑 방법을 적용하여 시나리오 명세 과정에 있어 표현 자유도를 높일 수 있다.

5. Simulation

제안된 평가 기법을 사용하여, 개별 어플리케이션에서 지식으로 활용하려는 온톨로지의 적합성을 확인하기 위해 다음의 실험 환경을 가정하였다. 우선 대상 온톨로지로는 W3C에서 제공하는 포도주 온톨로지를 선택하였다 [11]. 이 온톨로지를 사용하려는 가상 어플리케이션은 포도주 추천 시스템으로 음식점에서 고객의 식사 주문시 적절한 와인을 추천해 주는 서비스를 제공한다고 가정하였다. 가상 어플리케이션의 서비스 시나리오는 <표 3>과 같다.

<표 3> 가상 서비스 시나리오

1. 고객이 식사를 주문하면, 그에 어울리는 와인을 추천해준다
2. 고객의 행사에 어울리는 와인을 추천해준다
3. 새로운 와인이 들어온 경우, 그 와인 취향에 맞는 고객에게 시음을 추천해준다

이상의 가상 서비스를 수행하려고 할 때, 포도주 온톨로지로서 표현된 지식이 적합한지를 판단한다. 이를 위해 제안된 기법에 따라 서비스 명세의 어휘를 온톨로지의 객체와 관계로 변환한다. 변환된 결과는 다음과 같다.

<표 4>의 변환된 시나리오에서 “NULL”은 대상 온톨로지서 표현되어 있지 않음을 의미하고, 그 밖의 변환 대상의 표현은 EOL 문법에 따라 표현하였다 [12]. 변환 결과에 의하면, 가상 어플리케이션에서 서비스를 수행하기 위해서는 첫 번째 시나리오의 경우, 고객과 주문에 대한 개념 정의가 필요함을 확인할 수 있고, 두 번째 시나리오의 경우, 고객과 행사에 대한 개념 정의. 그리고 마지막 시나리오의 경우 고객과 취향에 대한 개념 정의가 부족함을 확인할 수 있다. 어플리케이션 관리자는 이

온톨로지를 사용하려고 한다면, 부족한 개념에 대한 추가 온톨로지를 확장 시키든지, 아니면 다른 온톨로지를 사용할 것인지를 선택할 수 있다. 또한 와인을 표현하기 위해 중압감, 색, 맛, 당도, 제조사, 제조 위치, 생산 과일로 정의하는 것이 자신의 시나리오에 맞는지를 확인할 수 있다.

<표 4> 제안 기법을 적용한 시나리오 변환

1. 고객(NULL)이 식사((at_least 1 food:course food:MealCourse ((at_least 1 food:hasDrink) and (at_least 1 food:hasFood)))를 주문(NULL)하면, 그에 어울리는 와인(and ((hasBody) and ((hasColor) and ((hasFlavor) and ((hasMaker) and ((hasSugar) and ((locatedin) and (madeFromGrape)))))))을 추천해준다
2. 고객(NULL)의 행사(NULL)에 어울리는 와인(and ((hasBody) and ((hasColor) and ((hasFlavor) and ((hasMaker) and ((hasSugar) and ((locatedin) and (madeFromGrape)))))))을 추천해준다
3. 새로운 와인(and ((hasBody) and ((hasColor) and ((hasFlavor) and ((hasMaker) and ((hasSugar) and ((locatedin) and (madeFromGrape)))))))이 들어온 경우, 그 와인 취향(NULL)에 맞는 고객(NULL)에게 추천해준다

6. Conclusion

지능형 시스템을 구현함에 있어 지식화는 그 시스템의 지능 정도를 결정하는 주요 요소가 된다. 그렇지만, 방대한 도메인 정보들 속에서 문제 해결에 필요한 지식을 추출해내기란 쉬운 일이 아니다. 더욱이, 불필요한 지식이나, 적절하지 못한 표현 방식은 시스템의 지식 처리 비용을 증가시킴으로써, 지식 활용의 이점을 무위로 돌릴 위험성이 존재한다. 이에 본 연구에서는 기존 지식의 재 활용 및 타 시스템과의 지식 공유에 탁월한 성능을 발휘하는 온톨로지를 사용하여 개별 어플리케이션에 적합한 온톨로지 지식을 판단하기 위한 시나리오 테스트 기법을 제안하였다. 어플리케이션은 지능적 서비스 수행을 위한 시나리오를 작성하고, 그 시나리오를 대상 온톨로지를 사용해 변경시킨다. 변경된 결과는 그 어플리케이션이 필요로 하는 지식의 범주와 사용하려는 온톨로지의 차이를 명확히 보여줄 수 있기 때문에, 어플리케이션 관리자로 하여금, 불필요한 온톨로지의 간결화, 조악한 부분의 세밀화 부분에 대한 선택을 보조해 줌으로써, 필요로 하는 지식을 명확히 정의하도록 도와주게 된다.

Reference

[1] M. Weiser, "The computer for the 21st century", Scientific American vol.265 no.3 pp94-104, 1991
 [2] K. Henricksen, J. Indulska, and A. Rakotonoraony "Modeling Context information in Pervasive Computing Systems" In Proc of Pervasive 2002 LNCS 2414, pp.167-180, 2002
 [3] U. Hustadt, B. Motik, and U. Sattler. "Data complexity of reasoning in very expressive description logics" In Proc. of IJCAI-05, 2005.

[4] en.wikipedia.org/wiki/knowledge_engineering
 [5] J. Giarratano, and G. Riley "Expert System: Principles and Programming" PWS Publishing Company, 1998
 [6] T. R. Gruber. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993
 [7] J. Brank, M. Grobelnik, and D. Mladenic "A Survey of Ontology Evaluation Technoques" IST World PASCAL Project 2005
 [8] N.F. Noy, D. McGunness "Ontology Development 101:A Guide to Creating Your First Ontology" Stanford Knowledge System Laboratory Technical Report KSL-01-05 2001
 [9] K. Lee, W. Kim, M. Kim: Effective Information Sharing Using Concept Mapping. KES (3) 2005: 177-183
 [10] R. Patton "Software Testing" Sams Publishing ISBN: 0-672-32798-8, 2005
 [11] http://www.w3.org/TR/owl-guide/wine.rdf
 [12] K. Lee, I. Hong, and M. Kim "EOL:Epistemological Ontology Language with SUNHI Expression Power for Ubiquitous Computing Environment" 추계정보과학회 2006