

온톨로지 기반 지식추론 기법

이정원⁰, 박세형^{**}, 이언경^{**}, 방건동^{**}, 백두권^{*}

고려대학교 소프트웨어 시스템 연구실*, 한국과학기술연구원(KIST)**
(jungwonlee, sehyung)@kist.re.kr, tomayto@syslab.korea.ac.kr, pahng@kist.re.kr, baik@swsys2.korea.ac.kr

Ontology-based Method for automatic Knowledge Reasoning

Jung-Won lee^o, Sehyung Park^{**}, Eon-Kyung Lee^{**}, Gun-Dong Francisco Pahng^{**}, Doo-Kwon Baik^{*},
Software System Lab. Dept. of Computer Science & Engineering Korea University*
&
Korea Institute of Science and Technology(KIST)**

요 약

제품 개발을 담당하는 부서에서, 다양한 이유로 핵심인력이 빠져나간다. 이때마다 제품개발 현장에서는 심각한 지식 누수현상이 나타나게 된다. 따라서 지식 누수 현상을 방지하기 위한 방법은 기업 내부에 존재하는 핵심인력의 노하우를 형식지로 저장해 관리하는 것이다.

제품을 개발을 담당하는 부서에는 수많은 문서들이 존재한다. 특히 품질 관리 문서는 제품개발과 관련된 핵심인력의 노하우가 농축되어있는 지식이다. 그래서 많은 기업에서는 기업내부에서 발생하는 그 지식을 관리하고 재활용하고자, 문서관리를 위한 시스템을 도입; 사용하고 있다. 그러나 설계 지식의 공유를 지원하는 시스템을 갖추었어도, 단순히 설계지식을 저장해놓은 경우가 많다. 개발자는 필요한 자료를 다시 선정해야 하는 문제가 발생한다. 이는 개발자에게 있어서 부담이 되며, 풍부한 지식활용률을 떨어뜨리게 만드는 한 요인이 된다.

본 논문에서는 이 문제를 해결하기 위해, 온톨로지를 기반으로 문서를 분류하고, 이 온톨로지에 정의된 키워드를 바탕으로 새로운 지식을 자동적으로 추론하여, 제품에 대한 기술적 지식을 가지고 검색하게 함으로서 필요 없는 검색 결과를 최소화 하고, 설계자의 지식 활용률을 높이고자 하였다.

1. 서론

오늘날 제조업 분야에서의 다품종 소량생산은 제품 개발 사이클을 단축시켰다. 그리하여 생산기술의 축적에 의존한 제품경쟁력유지는 한계에 다다랐다. 결국 경쟁력 있는 제품 생산을 위해서는 제품개발기간을 단축해야 한다. 한편 제품 개발 시 발생하는 설계 오류는 제품 개발기간 및 비용에 많은 영향을 준다. 그런데 제품 개발 시에 발생하는 설계 오류의 50%가 반복적으로 발생한다. 이는 제품 개발 노하우가 체계적 관리되지 않고, 숙련된 설계자가 보유한 설계 지식의 공유가 이루어지지 않았기 때문이다. 비록 설계 지식의 공유를 지원하는 시스템을 갖추었어도, 단순히 설계지식을 저장해놓은 경우가 많아, 설계자가 키워드를 통하여 검색을 하게 되면, 원하지 않는 결과까지 제공되는 문제점이 있다.

설계지식의 분류는 다양하지만 크게 background 와 foreground 지식 또는 formal 과 informal 지식으로 나누어 질 수 있다 [1-4]. 의미상의 차이가 다소 있기는 하지만, foreground 와 formal 지식은 설계업무 상에 일부로 되어 있는 각종 도면, 설계문서 등을 의미한다. 이에 반해, background 와 informal 지식은 문서상으로 표현되기 어려운 개인의 메모나 설계일지 또는 회의 등에서 도출되는 설계의 노하우를 의미한다.

한편 최근에 개발되고 있는 KMS(Knowledge

Management System)는 자산화 하고자 하는 지식의 대부분이 비정형적인 자료를 대상으로 한다. 이 비정형적인 지식은 서술적인 정보형태가 많을 뿐만 아니라, 다양한 Case 가 발생한다. 그러나 본 논문에서 다루고자 하는 설계 지식은 한정된 도메인이고, 로직이 내포된 정보 형태이며, 비교적 반복적인 Case 가 많다. 따라서 설계 지식의 효과적인 관리를 위하여 차별화된 지식관리 방법이 필요하다.

본 논문에서는 이 문제를 해결하기 위해, 온톨로지를 기반으로 문서를 분류하고, 이 온톨로지에 정의된 키워드를 바탕으로 새로운 지식을 추론하여, 제품에 대한 기술적 지식을 가지고 검색하게 함으로서 필요 없는 검색 결과를 최소화 하고, 설계자의 지식 활용률을 높이고자 하였다.

2. 관련연구

2.1 Ontology

온톨로지는 특정 분야의 지식을 표현하기 위한 기본 지식 체계를 제공한다. 온톨로지는 문서가 내포하고 있는 정보를 동일용어로 표현한다는 측면에서 “키워드”와 유사하고, 특정분야의 지식을 표현하는 체계를 제공한다는 측면에서는 “Taxonomy”와 유사하다. 그러나, 온톨로지의 가장 큰 특징은 온톨로지로 정의된 용어간의 상호연관관계를 체계적으로 정의한다는 것이다.

온톨로지는 'Concept' 과 'Relation', 두 가지로 구성 된다. 'Concept' 은 온톨로지 내에 정의된 키워드를 말하며, 'Relation' 은 온톨로지 내에 정의된 개념간의 관계를 정의한 것을 말하고, 이를 이원(二元)관계(binary relation)라 한다[5,6].

예를 들면 그림 1은 개념간의 계층구조를 나타내고 있다. 그림에서 문제점 및 부품파손, 기능불량, 외관 불량 등은 개념에 해당하고, 그들 간의 상호연관성은 축적된 설계문서의 분석을 통하여, 온톨로지를 구성하는 개념간 관계를 정의할 수 있다. 가령 설계문서의 분석을 통해 문제점 온톨로지서 진동은 설계개선 온톨로지서 설계변경의 치수변경을 통하여 개선할 수 있다는 관계가 얻어질 수 있으며, 이러한 관계는 효과적인 지식관리 및 새로운 지식의 유추에 활용될 수 있다.

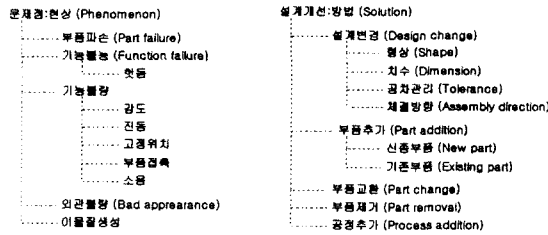


그림 1 온톨로지 예

2.2 온톨로지의 Concept 들의 관계정보를 바탕으로 한 자동추론 알고리즘

온톨로지의 Concept 들의 관계정보를 바탕으로 새로운 지식을 자동생성하기 위한 알고리즘은 이미 정의된 온톨로지 키워드를 바탕으로 해야 한다는 제약이 따른다.

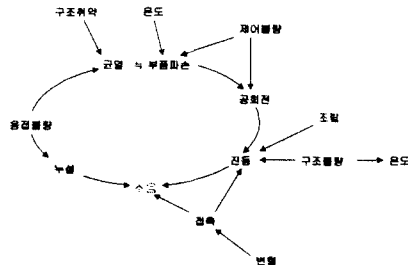


그림 2 온톨로지서 정의된 concept 간 인과 관계도

우선 그림 2는 프로젝트에 사용했던 품질관리 문서 중 24 개의 문서를 분석하여, 그 문서에 사용된 온톨로지 Concept 간의 인과관계를 직접 수작업으로 분석하여 도표로 나타낸 것이다.

설계자가 위 정보의 일부를 품질문서를 입력하면서, 제한한다고 가정하자. 예를 들어 해당 입력항목에 문제점, 원인 순으로 입력한다고 할 때, 제품의 특정부위에서 진동이 발생하여 알아본 결과, 공회전에 의해서 나타난 현상이라고 하자. 그럼 설계자는 입력시, 문제점 항목엔 '진동' 이라는 Concept 을, 원인 항목엔 '공회전' 이라는 Concept 을 입력할 것이다. 이때 DB에는 이들 Concept 정보가 저장되게 된다.

추론을 하기 위한 일단계는 우선 아래처럼 Concept

을 바탕으로 매트릭스를 만들어야 한다. 그리고 위 예처럼 사용자가 입력한 정보(예 "공회전은 진동을 유발한다") 정보를 바탕으로 왼쪽 항목을 원인으로 하여 해당 교차점에 1을 입력한다.(그림 3 참조)

	구조해석	부품파손(공회전)	기능불량	진동	소음	공회전	변형	조립불량	구조불량	공회전	외관불량
구조해석											
부품파손(공회전)		1									1
기능불량			1								
진동				1							
소음					1						
공회전						1					
변형							1				
조립불량								1			
구조불량									1		
공회전										1	
외관불량											1

그림 3 온톨로지의 Concept 간의 관계 테이블 다음으로 위의 매트릭스에서 행과 열 방향으로 모든 값을 합산 한다. 행의 합산 값은 "from score" 항목에, 열의 합산값은 "to score" 항목에 각각 기입한다.(그림 4 참조)

	구조해석	부품파손(공회전)	기능불량	진동	소음	공회전	변형	조립불량	구조불량	공회전	외관불량	from score
구조해석												0
부품파손(공회전)		1										1
기능불량			2									2
진동				1								1
소음					3							3
공회전						1						1
변형							1					1
조립불량								1				1
구조불량									1			1
공회전										1		1
외관불량											1	1
to score	0	4	0	1	3	1	0	1	0	2	0	2

그림 3 행과 열의 합산결과

그 후 그림 4 처럼 행의 합산 결과를 "from" 항목에 열의 합산결과를 "to" 항목으로 입력하고, "form" 과 "to" 항목의 합산결과를 sum 항목에 입력한 표를 만든다.

	from	to	sum
구조해석	0	0	0
부품파손(공회전)	1	4	5
기능불량	2	0	2
진동	1	1	2
소음	0	3	3
공회전	1	1	2
변형	1	0	1
조립불량	1	3	4
구조불량	1	0	1
공회전	2	2	3
외관불량	2	0	2

그림 4 Concept 간 릴레이션 값 합산테이블

이어서 그림 4 에서 sum 을 중심으로 내림차순 정리를 한다.

	from	to	sum
부품파손(공회전)	1	4	5
진동	1	3	4
소음	0	3	3
공회전	1	2	3
기능불량	2	0	2
진동	1	1	2
공회전	1	1	2
구조불량	2	0	2
외관불량	2	0	2
구조해석	1	0	1
변형	1	0	1
조립불량	1	0	1

그림 5 Sum 값을 바탕으로한 내림차순 정렬

그림 5 에서 sum 의 수는 지식 추론시 사용할 수 있는 Concept 의 총수이다. 예를 들어 부품파손 Concept 은 지식추론을 위해 5 번을 사용할 수 있다는 것이다. 이제 새로운 지식을 추론해 보면 다음과 같다. 우

선 그림 5 에서 내림차순 정렬로 정렬된 Concept 순서로 원인이 되어, 그림 2 를 이용 룰을 찾아낸다. 예를 들어 부품파손(균열)은 그림 2 에서 보듯이 2 행에 존재한다. 2 행을 따라가면서, '1'이 있는 항목을 찾는다. 여기서의 공회전이 이에 해당한다. 그림 '부품파손(균열)은 공회전을 일으킨다' 라는 규칙을 하나 찾아낼 수 있다. 이어서 "공회전" 항목을 찾아 마찬가지로 찾아보면, "공회전" 항목은 "진동"과 교차하는 지점에 1 이 존재한다. 따라서 "부품파손은 공회전을 일으키고, 공회전은 진동을 발생시킨다" 라는 규칙을 얻을 수 있다. 다시 "진동" 항목은 "소음" 항목에 '1' 이 존재하며, 우리는 "부품파손(균열)->공회전->진동->소음" 이라는 인과 관계를 얻을 수 있다. 이어서 소음을 보면 '1' 이 존재하지 않으므로, 여기서 루프를 끝내게 되고, 위 4 개의 concept 을 하나의 룰 그룹으로 묶는다. 다시 그림 5 에서 다음 항목인 진동부터 다시 시작한다. 그러나 이때부터는 약간의 조정이 필요하다. 우선 진동 행을 따라가면 소음과 연결되어있다. 이때 '진동' 이라는 Concept 은 행의 합의 값 만큼 룰형성을 위해 사용할 수 있다. 예를 들어 현재 진동은 행의 합산결과가 '1' 이므로(그림 5 참조) 2 번째 루프에서는 소음을 다시 사용할 수가 없다. 따라서 진동->소음이라는 규칙을 생성하지 않는다. 즉 이렇게 함으로써 앞의 예에서 생성한 "진동은 소음을 유발한다" 라는 규칙이 다시 생성되는 것을 막는다. 이와는 다르게 그림 3 에서 "구조불량" 항목을 "from" 항목의 값이 2 이므로 룰을 형성하기 위해 2 번에 걸쳐서 사용할 수 있다. 다시 말해 구조불량이 원인이 되어서, 2 개의 룰을 연속적으로 생성한다. 즉 "구조불량->부품파손(균열)", 과 "구조불량->소음" 의 2 가지 규칙을 생성하게 된다. 이렇게 반복하면 아래와 같은 서로 다른 색으로 표시된 다양한 규칙 그룹을 만들 수 있다.

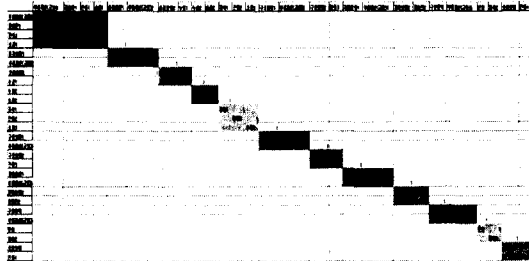


그림 6 다양한 규칙 그룹

그림 6 은 위에서부터 아래방향으로, 순차적으로 진행하여 그룹화 한 것이다. 그림 6 에서 찾아진 다양한 규칙들은 관련 있는 것끼리 묶으면 새로운 지식이 형성된다.

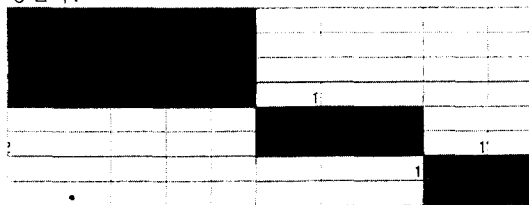


그림 7 다양한 규칙 그룹 발체

예를 들어 그림 7 의 규칙들을 연결해보자, 3 개중 중간의 규칙 그룹에서 "용접불량" 이 "부품파손"

원인이 되고, 첫번째 규칙에서 부품파손은 공회전에 앞선다.이렇듯 체인을 형성하여 "용접불량->부품파손->공회전->진동->소음" 이라는 새로운 지식이 형성되며, 그림 1 에서 직접 수작업으로 확인한 결과와 동일하다는 것을 알 수 있다.

3. 결론

제품 개발 시 발생하는 설계 오류는 제품 개발기간 및 비용에 많은 영향을 준다. 그런데 제품 개발 시에 발생하는 설계 오류의 50%가 반복적으로 발생한다. 이는 제품 개발 노하우가 체계적 관리되지 않고, 숙련된 설계자가 보유한 설계 지식의 공유가 이루어지지 않았기 때문이다

한편 제품을 개발을 담당하는 부서에는 수많은 문서들이 존재한다. 특히 품질 관리 문서는 제품개발과 관련된 핵심인력의 노하우가 농축되어있는 지식이다. 그래서 많은 기업에서는 기업내부에서 발생하는 그 지식을 관리하고 재활용하고자, 문서관리를 위한 시스템을 도입, 사용하고 있다. 그러나 설계 지식의 공유를 지원하는 시스템을 갖추어도, 단순히 설계지식을 저장해놓은 경우가 많아, 개발자는 필요한 자료를 다시 선정해야 하는 문제가 발생한다. 이는 개발자에게 있어서 부담이 되며, 풍부한 지식활용률을 떨어뜨리게 만드는 한 요인이 된다.

본 논문에서는 위와 같은 해결하기 위해, 온톨로지를 기반으로 문서를 분류하고, 이 온톨로지에 정의된 키워드를 바탕으로 새로운 지식을 추론하여, 제품에 대한 기술적 지식을 가지고 검색하게 함으로써 필요한 검색 결과를 최소화 하고, 설계자의 지식 활용률을 높이고자 하였다.

4. 참고 문헌

[1]. F. Kimura and P. Suzuki, "Representing Background Information for Product Description to Support Product Development Process," Annals of the CIRP, 44(1)113-116, 1995
 [2]. H. Suzuki, F. Kimura, B. Moser, and T. Yamaha, "Modeling Information in Design Background for Product Development Support," Annals of the CIRP, 45(1)141-144, 1996.
 [3]. G. Toye, M. Cutkosky, J. Tenenbaum, J. Glicksman, "SHARE: A Methodology and Environment for Collaborative Product Development," Proceedings of Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise, 33-47. 1993
 [4]. E. Conklin, "Designing Organizational Memory: Preserving Intellectual Assets in a Knowledge Economy," White paper, Group Decision Support Systems, 1996, <http://www.gdss.com/DOM.htm>.
 [5]. M. Uschold and M. Gruninger, "Ontologies: principles, methods and applications," The Knowledge Engineering Review, 11(2)93-136, 1996.
 [6]. T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," Int. J. Humna-Computer Studies, 43:907-928, 1995.