

온톨로지 기반의 용어 정의 비교 및 유사도를 고려한 의미 매핑

정원철*, 이재현**, 서효원***

Semantic Mapping of Terms Based on Their Ontological Definitions and Similarities

Jung, W.C.*, Lee, J.H.** and Suh, H.W.***

ABSTRACT

In collaborative environment, it is necessary that the participants in collaboration should share the same understanding about the semantics of terms. For example, they should know that 'COMPONENT' and 'ITEM' are different word-expressions for the same meaning. In order to handle such problems in information sharing, an information system needs to automatically recognize that the terms have the same semantics. So we develop an algorithm mapping two terms based on their ontological definitions and their similarities. The proposed algorithm consists of four steps: the character matching, the inferencing, the definition comparing and the similarity checking. In the similarity checking step, we consider relation similarity and hierarchical similarity. The algorithm is very primitive, but it shows the possibility of semi-automatic mapping using ontology. In addition, we design a mapping procedure for a mapping system, called SOM (semantic ontology mapper).

Key words : Ontology, Ontology Mapping, Semantic Mapping

1. 서 론

협업 환경은 제품 설계, 엔지니어링, 제조, 구매, 판매, 마케팅 그리고 서비스와 같은 제품의 수명 주기와 관련된 모든 문제들을 통합한 개념이다. 협업 환경에서는 방대한 양의 정보 교환이 발생한다^[1]. 효과적인 협업을 위해서는, 무엇보다도 제품의 수명 주기와 관련된 모든 참여자들 사이에 이러한 정보를 원활히 교환하는 것이 필요하다. 그러나 제품의 수명 주기와 관련된 참여자들은 보통 서로 다른 용어를 사용하기 때문에 제품의 수명주기와 관련된 정보시스템을 통합하는데 어려움을 갖게 된다.

모든 도메인에서 사용되는 용어들이 온톨로지적 정의를 갖는다는 가정 하에서, 본 연구는 의미 공유 문제에 대한 해결책으로 온톨로지를 기반으로 한 의미

매핑 알고리즘을 제안한다. 본 알고리즘은 문자열 비교(Character Matching), 추론(Inferencing), 정의 비교(Definition Comparing)와 유사도 측정(Similarity Checking)의 4단계로 구성된다. 정의 비교 단계에서는 전체 매핑 알고리즘이 계층적으로 사용된다. 정의 비교 단계에서 사용자의 판단이 필요한 경우, 용어의 유사도 측정을 통해 사용자의 매핑 참여를 지원한다. 또한, 본 논문에서 제안된 의미 매핑 알고리즘을 활용하여 설계한 Semantic Ontology-based Mapper(SOM) 프로토타입 시스템을 소개한다.

일반적으로 서로 다른 온톨로지 간의 매핑 접근방법은 전문가가 직접 수행하거나 반자동화를 통해 수행할 수 있다. 우리는 온톨로지 매핑에 관한 기존 연구들을 인스턴스 수준, 데이터 모델 수준, 온톨로지 수준의 매핑 접근방법으로 분류하였다. 각 매핑 수준에 대한 기존 연구들은 2장에서 상세히 설명한다. 본 연구는 온톨로지 수준의 매핑을 반자동화 하는 접근방법을 제안한다. 본 연구는 두 도메인의 용어들에 대한 정의와 공리들이 논리적으로 정의되었을 때 이를 바탕으로 추론을 통하여 매핑하는 접근방법을 제안한

*LG생산기술연구원 시스템 기술그룹

**학생회원, 한국과학기술원 산업공학과 박사과정

***교신저자, 종신회원, 한국과학기술원 산업공학과 교수
- 논문투고일: 2004. 10. 26
- 심사완료일: 2005. 12. 29

다. 또한, 추론 결과 사용자의 관여가 필요할 때 두 용어의 온톨로지적 구조의 유사도를 측정하고 두 용어가 매핑될 가능성을 평가하여 사용자의 매핑 판단을 지원한다.

용어의 정의와 공리를 바탕으로 두 온톨로지를 매핑하는 기존 연구들은 전문가가 직접 매핑하는 접근 방법을 취하고 있다. 반자동화 접근방법을 취하는 연구로는 본 연구의 선행 연구¹⁾가 있다. 김경영, 서효원²⁾의 매핑 프로세스는 정의를 이용하여 매핑하는 부분적 과정을 보인 반면에, 본 연구는 위 매핑 프로세스에 온톨로지적 구조의 유사도를 측정하여 사용자의 판단을 지원하는 반자동화 매핑 단계를 확장 정의하고, 이를 바탕으로 SOM 프로토 시스템을 설계하였다. 본 연구는 다양한 용어 설명(공리 및 정의)을 바탕으로 매핑 관계를 추론한다. 하지만, 공리를 포함한 많은 논리를 활용하여 매핑 관계를 추론하기 때문에 추론 과정에 한계점들이 있다. 본 연구의 한계점에 대해서는 6장 결론에서 상세히 논의한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고 3장에서는 온톨로지의 구조 및 표현언어에 대해서 설명한다. 4장에서는 의미 매핑 알고리즘을 제안하며, 5장에서는 제안된 매핑 알고리즘을 바탕으로 설계된 SOM 프로토 시스템을 소개한다.

2. 관련 연구

자동 정보 교환은 상이한 데이터 모델 사이의 매핑을 필요로 한다. 기존 연구들은 검토한 결과 이러한 매핑은 인스턴스, 데이터 모델 또는 온톨로지를 바탕으로 구현되었다. 하지만, 인스턴스 또는 데이터 모델을 바탕으로 구현한 매핑 방법은 두 도메인의 데이터 모델이 유사하게 설계되었다는 가정이 필요하거나, 사용자가 직접 매핑을 수행해야 하는 단점이 있다. 예를 들어, Lukasz Kurgan은 두 개의 다른 모델 간의 XML 인스턴스에 기반을 둔 자동 매핑 방법을 제안했다³⁾. 그는 태그의 이름, 값, 위치, 그 외 수치의 유사성을 바탕으로 귀납적 기계 학습을 이용한 매핑방법을 제안했다. 그러나 이러한 접근은 인스턴스들이 유사한 태그와 구조로 표현되어 있다는 것을 가정한다. R. Vdovjak와 G. J. Houben⁴⁾은 XML을 다른 XML로 변형시키는 언어인 XSLT를 이용하는 데이터 모델 매핑 아키텍처를 제안했다. 또한 B. Omelayenko과 D. Fensel⁵⁾도 이와 유사한 접근법을 제안했다. 그러나 그들의 제안은 사용자들로 하여금 데이터 모델들 사이의 매핑 규칙을 디자인하도록 요구한다는 한계가

있다.

이러한 한계를 극복하기 위하여 많은 연구들이 온톨로지 매핑을 통한 데이터 모델 매핑방법을 제안하였다. 그 이유는 온톨로지가 데이터 모델의 의미를 논리적으로 명세하기 때문에 모델 간의 매핑 시 의미 수준에서 매핑 관계의 옳고 그름을 판단할 수 있는 근거가 되기 때문이다. 일반적으로 온톨로지 기반 정보 교환시, 두 종류의 매핑을 고려해야 한다. 하나는 특정 도메인의 데이터 모델과 동일한 도메인의 온톨로지 사이 매핑이고, 다른 하나는 다른 두 도메인의 온톨로지 사이의 매핑이다⁶⁾. 두 가지 매핑 종류 중 본 연구는 서로 다른 두 도메인의 온톨로지 사이의 매핑을 하기 위한 방법을 제안하고자 한다.

서로 다른 두 온톨로지 간의 매핑에 대한 접근법으로는 'defined mapping', 'lexical relations', 'top-level grounding', 'semantic correspondence' 등이 있다⁶⁾. 'defined mapping' 접근법은 KRAFT⁷⁾에서 사용되었다. KRAFT에서는, 전문가에 의해서 정의된 매핑 규칙을 포함하는 특화된 중재자가 서로 다른 온톨로지 간의 매핑을 지시한다. 'lexical relations' 접근법은 매핑 관계를 설명하기 위해서 매핑 온톨로지를 제공한다. OBSERVER에서는 서로 다른 온톨로지의 서로 다른 용어들 사이의 관계를 포착하기 위해서 관계 기술어 - 일종의 매핑 온톨로지 - 를 사용한다⁸⁾. 'top-level grounding' 접근법은 데이터 모델의 온톨로지가 공통의 최상위 레벨 온톨로지로부터 개념(Concept)을 상속받게 하는 것이다. 본 접근법은 공통의 상위 클래스에 근거한 서로 다른 온톨로지 개념간의 구조적 관계를 갖도록 하며, 결국 단일화된 온톨로지를 구축할 수 있다. 'semantic correspondence' 접근법은 개념의 의미에 근거하여 직접적인 매핑을 가

Table 1. SOM과 기존 연구와의 비교

시스템	비교요건	온톨로지 매핑 접근법*
SIMS [10]		None
TSIMMIS [11]		M.M.
OBSERVER [8]		L.R.
KRAFT [7]		D.M.
DWQ [12]		T.L.G.
PICSEL [13]		S.C.
BUSTER[9]		S.C.
SOM		S.C. and Similarity

*M.M= Manual Mapping, L.R.=Lexical Relations, D.M.=Defined Mapping, T.L.G.=Top-level Grounding, S.C.=Semantic Correspondence.

능케 한다. 예를 들면, BUSTER는 본 접근법을 용어 매핑에 적용하였다^[4]. Table 1은 본 접근법인 SOM 접근법과 다른 온톨로지 매핑 시스템을 비교한 결과를 보여주고 있다.

위 네 가지 접근 방법 중 ‘defined mapping’과 ‘lexical relations’ 접근방법은 사용자가 직접하는 매핑방법이고, ‘top-level grounding’과 ‘semantic correspondence’ 접근방법이 (반)자동화 매핑 접근방법이다.

본 연구는 ‘semantic correspondence’ 접근방법을 바탕으로 용어의 정의와 공리 및 관계의 유사성을 바탕으로 매핑의 반자동화를 추구한다.

3. 온톨로지 구조 및 표현언어

임의의 도메인을 온톨로지로 표현하기 위해서는 온톨로지의 구조 그리고 표현 언어의 결정이 선행되어야 한다. 온톨로지 구조는 지식 엔지니어마다 조금씩 다른 형태를 활용한다. 본 연구에서는 최근에 활용되는 KAON^[4] 온톨로지 구조를 활용하였다. 이 구조의 기본 요소는 다음과 같다.

$$O := \{C, R, H^c, rel, A^O\}$$

C: Concept, *R*: Relation, *H^c*: Concept Hierarchy

rel: Functions of relations, *A^O*: Axiom

온톨로지 구조에서 개념(Concept)과 관계(Relation)는 원시(Primitive) 개념/관계와 정의된(Defined) 개념/관계로 구분할 수 있다. 원시 개념/관계는 자연적으로 존재하는 것으로 그 존재성을 증명하기 어려운 것이고, 정의된 개념/관계는 원시 개념/관계들을 활용하여 논리적으로 정의할 수 있는 개념/관계를 의미한다. 따라서, 원시 개념/관계는 존재함으로써 발생하는 필요조건만을 가질 수 있지만, 정의된 개념/관계는 존재를 위한 필요충분조건(즉, 개념/관계의 정의)을 가질 수 있다. 예를 들어, ‘도끼’라는 원시 개념은 ‘털이 있고, 귀가 길다’라는 필요조건을 갖을 수 있지만, ‘도끼’를 식별할 수 있는 필요충분조건은 없다. 반면에, ‘산도끼’라는 정의된 개념은 ‘도끼’라는 원시 개념을 이용하여 정의를 제공할 수 있다.

본 연구에서는 온톨로지 표현 언어로서 지식 표현 능력이 우수한 First Order Logic(FOL)을 선택하였다. Table 2는 FOL로 표현된 온톨로지의 예이다.

이러한 온톨로지 구조와 표현언어를 바탕으로 본 논문에서는 온톨로지 간 매핑 알고리즘을 제안한다.

Table 2. FOL로 온톨로지를 표현한 예

요소	예제
Concept	$Part(x), endProduct(x)$
Relation	$isSubPartOf(x, y)$
Concept Hierarchy	$\forall x endProduct(x) \rightarrow part(x)$
Relation Function	$\forall x, y isSubPartOf(x, y) \rightarrow part(x) \wedge part(y)$
Axiom	$\forall x, y isSubPartOf(x, y) \leftrightarrow hasSubPart(y, x)$

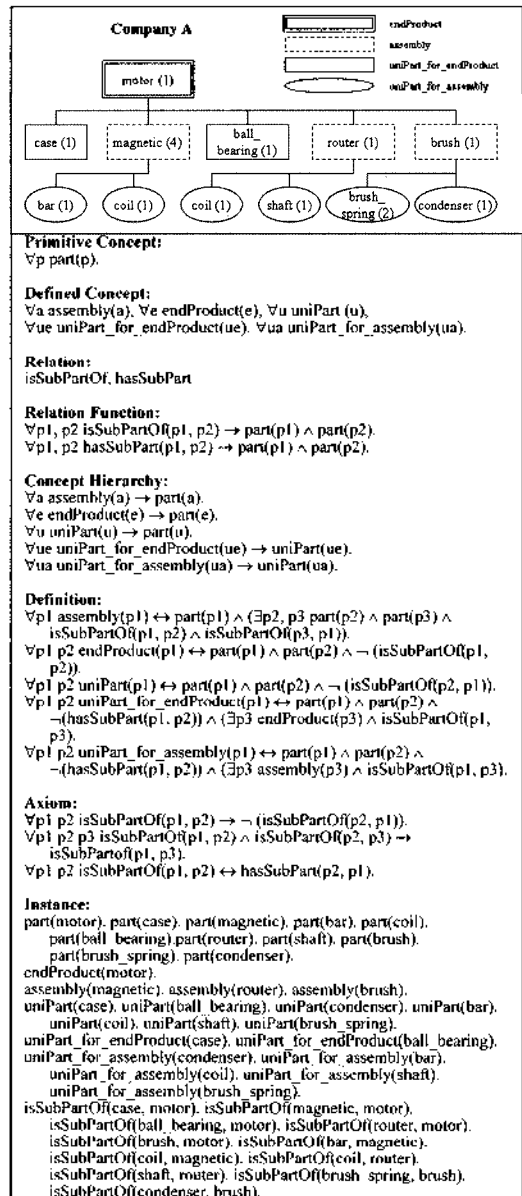


Fig. 1. 'A' 회사 용어로 모터 BOM을 표현한 예제.

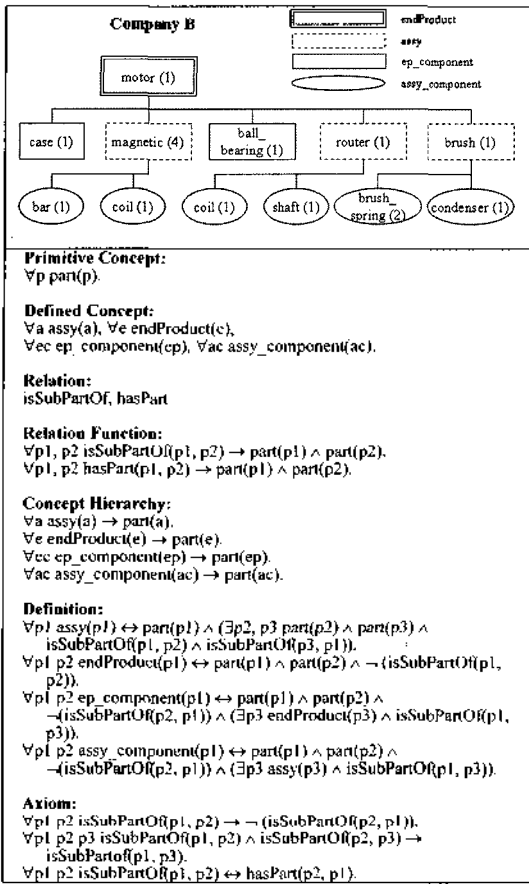


Fig. 2. 'B' 회사 용어로 모터 BOM을 표현한 예제.

4장에서는 알고리즘을 논리적으로 설명하면서, 모터 BOM(Bills Of Materials) 예제를 활용하여 알고리즘에 대한 설명을 돕는다. Fig. 1과 Fig. 2는 서로 다른 회사에서 작성된 모터의 BOM에 대해서 온톨로지와 인스턴스를 표현한 예이다.

논리적으로 Fig. 1에서 $\text{isSubPartOf}(p1, p2)$ 관계를 갖지 않은 $\text{part}(p)$ 들은 모두 $\neg(\text{isSubPartOf}(p1, p2))$ 관계를 갖는다고 가정한다. 본 연구의 목적은 온톨로지 기반 매핑 결과를 얻는 것이다. 이 매핑 결과는 궁극적으로 'A' 회사의 인스턴스를 'B' 회사 용어로 표현된 인스턴스로 변환하는데 활용된다.

4. 의미 매핑 알고리즘

본 연구에서는 1) 문자열 비교, 2) 추론, 3) 정의 비교, 4) 유사도 측정의 4단계를 순차적으로 수행하는 의미 매핑 알고리즘을 제안하였으며 상세 절차는 Fig. 3과 같다.

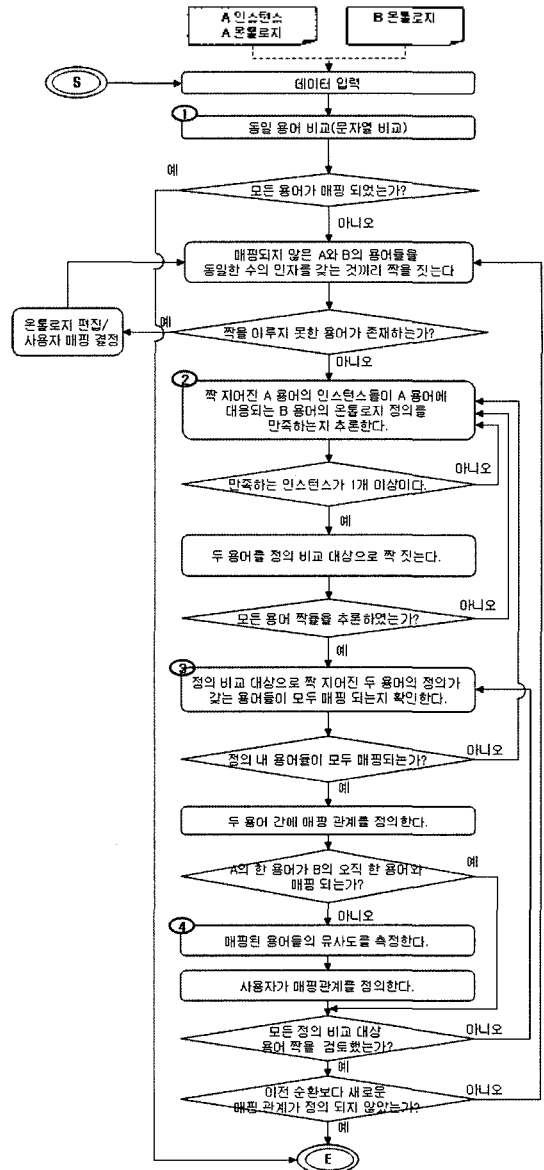


Fig. 3. 의미 매핑 알고리즘.

4.1 문자열 비교(Character Matching)

먼저, 임의의 두 도메인에서 사용하는 용어에 대해서 다음과 같이 정의한다.

A 도메인 용어: $T_A = \{a_1, \dots, a_n\}$.

B 도메인 용어: $T_B = \{b_1, \dots, b_m\}$.

또한, A도메인의 용어 중에서 개념에 해당하는 것들의 집합을 C_A 로 관계에 해당하는 것들의 집합을 R_A 로 정의한다. A회사와 B회사에서 사용하는 용어들이 Fig. 1과 Fig. 2와 같다면, 다음과 같은 결과를 얻는다.

$$T_A = \{part, endProduct, assembly, uniPart\ for_endProduct, uniPart, uniPart_for_assembly, isSubPartOf, hasSubPart\}.$$

$$C_A = \{part, endProduct, assembly, uniPart, uniPart_for_endProduct, uniPart_for_assembly\}.$$

$$R_A = \{isSubPartOf, hasSubPart\}.$$

$$T_B = \{part, endProduct, assy, ep_component, assy_component, isSubPartOf, hasPart\}.$$

$$C_B = \{part, endProduct, assy, ep_component, assy_component\}.$$

$$R_B = \{isSubPartOf, hasPart\}.$$

문자열 비교에서는 두 용어의 문자열이 같으면 동일한 의미라는 가정을 한다. 이 가정 하에서 다음과 같은 매핑된 용어의 집합, M_c^A 와 M_c^B 를 얻을 수 있다.

$$M_c^A = \{a_1, \dots, a_n\}, 1 \leq k \leq \min(n, m).$$

$$M_c^B = \{b_1, \dots, b_m\}, 1 \leq k \leq \min(n, m).$$

앞의 예에서, 매핑된 용어의 집합은 아래와 같다.

$$M_c^A = \{part, endProduct, isSubPartOf\}.$$

$$M_c^B = \{part, endProduct, isSubPartOf\}.$$

또한 문자열 비교를 통해서 매핑이 되지 않은 용어들의 집합 NM_c^A 와 NM_c^B 를 정의할 수 있다.

$$NM_c^A = \{a_{k+1}, \dots, a_n\}.$$

$$NM_c^B = \{b_{k+1}, \dots, b_m\}.$$

앞의 예에서, NM_c^A 와 NM_c^B 는 다음과 같이 표현할 수 있다.

$$NM_c^A = \{assembly, uniPart, uniPart_for_endProduct, uniPart_for_assembly, hasSubPart\}.$$

$$NM_c^B = \{assy, ep_component, assy_component, hasPart\}.$$

4.2 추론(Inferencing)

4.2.1 용어의 인자 개수에 따른 용어쌍 생성

용어의 인자 개수를 반환하는 함수 $G(a_p)$ 를 정의한다. 예를 들어, $part(x)$ 와 $isSubPartOf(x, y)$ 에 대하여 $G(part) = 1$ 이고, $G(isSubPartOf) = 2$ 이다. NM_c^A , NM_c^B 의 용어 중 인자 개수가 l ($l = G(a_p)$)인 용어들끼리 짝을 지은 집합 P_l 을 다음과 같이 정의한다.

$$P_l = \{(a_p, b_q) | a_p \in NM_c^A, b_q \in NM_c^B, G(a_p) = l, G(b_q) = l\}.$$

앞의 예로부터 인자 개수가 1 또는 2인 경우의 P_l 을 다음과 같이 나타낼 수 있다.

$$P_1 = \{(assembly, assy), (assembly, assy_component), (assembly, ep_component), (uniPart, assy), (uniPart, assy_component), (uniPart, ep_component), \dots\}.$$

$$P_2 = \{(hasSubPart, hasPart)\}.$$

만약 NM_c^A , NM_c^B 의 용어 중 P_l 에 포함되지 않는 용어들이 있다면, 이 용어들은 주어진 온톨로지를 바탕으로는 자동 매핑이 안되는 것이기 때문에, 지식 엔지니어가 온톨로지를 수정하거나 용어 매핑 결정을 해야한다.

4.2.2 용어쌍 별 추론 적용

P_l 에 속하는 용어쌍 중에서 A회사 용어가 ' a_p '인 용어쌍들의 집합을 $S_{a_p}^l$ 로 정의한다.

$$S_{a_p}^l = \{(a_p, b_q) | b_q \in NM_c^B, a_p \in NM_c^A\}.$$

A회사 용어 ' $uniPart_for_assembly$ '에 대해서 다음과 같은 결과를 얻는다.

$$S_{uniPart_for_assembly}^{l=1} = \{(uniPart_for_assembly, assy), (uniPart_for_assembly, ep_component), (uniPart_for_assembly, assy_component)\}.$$

A회사 용어 ' a_p '의 인스턴스 ($I_{A(a_p)}$)를 반환하는 함수 $I_A(a_p)$ 를 다음과 같이 정의한다.

$$I_A(a_p) = \{I_{a_1}^s, I_{a_2}^s, \dots, I_{a_n}^s\}, s \geq 1.$$

A회사 용어의 인스턴스가 Fig. 1과 같을 때, ' $uniPart_for_assembly$ ' 용어에 대해서 다음과 같은 결과를 얻는다.

$$I_A(uniPart_for_assembly) = \{bar, \dots, condenser\}.$$

$S_{a_p}^l$ 의 각 원소들은 A회사 용어인 a_p 와 B회사 용어인 b_q 의 짝으로 이루어져 있다. 추론과정에서는 각 용어쌍에 대해서 a_p 의 인스턴스 중에서 b_q 의 정의를 만족하는 것이 적어도 한 개 이상 있는지를 질의하고, 질의에 역방향 추론(Backward Chaining)^[13]을 적용하여 True/False를 반환한다. a_p 가 ' $uniPart_for_assembly$ '이고 b_q 가 ' $assy_component$ '인 경우에 아래와 같은 질의를 하게 된다.

$$FOL-BC-ASK(KB, assy_component(bar), \{\})$$

...

FOL-BC-ASK(KB, *assy_component(condenser)*, {})

$H(I_A(a_p), b_q)$ 는 적어도 한 개 이상의 인스턴스 ' $I_A(a_p)$ '가 ' b_q '의 정의를 만족하면 *True*를 반환하고, 전부 만족하지 않을 경우 *False*를 반환하는 함수로서 다음과 같이 정의할 수 있다.

$$\begin{aligned} H(I_A(a_p), b_q) &= \text{True, if } \exists I_{A^k} \text{ s.t. } b_q(I_{A^k}) = \text{True} \\ &\text{for } I_{A^k} \in I_A(a_p), 1 \leq k \leq s. \\ H(I_A(a_p), b_q) &= \text{False, otherwise.} \end{aligned}$$

각각의 용어쌍들을 $H(I_A(a_p), b_q)$ 에 적용하였을 때, *True*를 갖는 용어쌍들의 집합을 TSa_p' 로 정의한다. TSa_p' 는 최대 $|S_a_p|$ 개의 원소를 가질 수 있으며 정의는 다음과 같다. ($|S|$ = 집합 S 의 원소 개수)

$$TSa_p' = \{(a_p, b_q) \mid H(I_A(a_p), b_q) = \text{True}\}.$$

a_p '가 '*uniPart_for_assembly*'인 경우, $TS_{uniPart_for_assembly}'$ 는 다음과 같다.

$$\begin{aligned} TS_{uniPart_for_assembly}' \\ = \{(uniPart_for_assembly, assy_component)\}. \end{aligned}$$

4.3 정의 비교

정의 비교 단계에서는 추론 단계의 결과를 바탕으로 용어쌍의 정의를 비교한다. 두 용어의 정의 비교를 위해서 각 정의에서 사용되는 용어들이 모두 매핑이 되어야 한다. 이를 위해 두 용어의 정의에 활용된 용어들에 대해서 문자열 비교 및 추론 매핑 단계를 재귀적으로 적용할 필요가 있다. 예를 들어, Fig. 1과 Fig. 2의 '*uniPart_for_assembly*'와 '*assy_component*'의 정의는 다음과 같다.

$$\begin{aligned} \forall p1, p2 \text{ uniPart_for_assembly}(p1) &\leftrightarrow \text{part}(p1) \wedge \\ &\text{part}(p2) \wedge \neg(\text{hasSubPart}(p1, p2)) \wedge (\exists p3 \\ &\text{assembly}(p3) \wedge \text{isSubPartOf}(p1, p3)). \\ \forall p1, p2 \text{ assy_component}(p1) &\leftrightarrow \text{part}(p1) \wedge \text{part}(p2) \\ &\wedge \neg(\text{isSubPartOf}(p2, p1)) \wedge (\exists p3 \text{ assy}(p3) \wedge \\ &\text{isSubPartOf}(p1, p3)). \end{aligned}$$

그러면, 우리는 4.1 문자열 비교 단계의 T_A, T_B 와 같은 T_A', T_B' 를 두 용어의 정의로부터 얻을 수 있다.

$$\begin{aligned} T_A' &= \{\text{part, assembly, isSubPartOf, hasSubPart}\}. \\ T_B' &= \{\text{part, assy, isSubPartOf}\}. \end{aligned}$$

TSa_p' 의 모든 용어쌍들에 대해서 정의 비교는 다음의 3단계로 수행된다.

- 단계 1: T_A' 와 T_B' 의 용어들 간에 문자열 비교를 수행한다.

- 단계 2: 문자열 비교를 통해서 매핑이 이루어지지 않는 두 용어들이 이미 매핑이 이루어졌는지를 확인한다. 만약, 두 용어가 이미 매핑이 되어 있다면 단계 3으로 이동한다. 그러나 매핑이 되어 있지 않다면, 두 용어에 대해서 추론 이전의 알고리즘을 재귀적으로 적용하여 같은 의미를 갖는지를 평가하여야 한다. 본 연구에서는 알고리즘의 안정성을 위해 다음 두 가지를 가정한다. 1) 알고리즘 상의 루프가 없다. 2) 용어의 정의에 있는 용어들의 표현 순서대로 알고리즘을 수행한다.

위의 예에서, '*assembly*'와 '*assy*'가 매핑이 되어 있다면 그 결과를 '*uniPart_for_assembly*'와 '*assy_component*'를 매핑하는데 활용할 수 있다. 만약 사전 매핑 결과에 '*assembly*'와 '*assy*'가 매핑이 안되어 있다면, 이들에 대해서 추론 및 정의 비교를 재귀적으로 적용한다. 재귀적으로 호출되는 정의 비교는, 사용자가 반복(iteration)할 횟수를 정하여, 정의 내 활용된 용어의 또 다른 정의를 비교하는 복잡도를 조절할 수 있다.

- 단계 3: 만약, 앞 단계의 두 용어가 같은 의미를 갖는다는 결론을 얻게 된다면, 문자열 비교를 통해서 매핑이 이루어지지 않은 다른 용어에 대해서 앞의 과정을 반복적으로 수행하게 된다. 두 용어의 정의내 모든 용어들이 매핑이 되고, 공리(Axiom)을 활용하여 두 용어의 정의가 동일하다고 판단되면, 두 용어는 서로 같은 의미를 갖는다고 볼 수 있다.

'*uniPart_for_assembly*'와 '*assy_component*' 예에서, Fig. 1에서 보는 바와 같이 '*assembly*'와 '*assy*'의 정의가 동일하기 때문에 같은 의미를 갖는다는 결론을 얻을 수 있다. 그리고, '*uniPart_for_assembly*'의 정의에서 '*hasSubPart*'가 '*isSubPartOf*'와 역관계라는 공리를 통해, 우리는 '*uniPart_for_assembly*'와 '*assy_component*'가 같은 의미라는 결론을 이끌어낼 수 있다.

TSa_p' 의 모든 용어 쌍들에 대해서 앞의 단계 1, 2, 3을 적용하면, 같은 의미를 갖는 것으로 판명된 용어 쌍들의 개수를 확인 할 수 있다. 'A'의 한 용어 a_p 에 대하여 같은 의미를 갖는다고 판명되는 'B'의 용어 개수를 반환하는 함수 $f(TSa_p')$ 를 정의하고, $f(TSa_p')$ 의 값 n 에 따라 다음과 같이 용어 매핑을 수행한다.

- $n = 0$: 유사도 측정 후 사용자가 용어를 선택한다.
- $n \geq 1$: 두 용어를 자동으로 매핑 시킨다.

'A'의 한 용어 a_p 에 매핑된 용어들로 구성된 용어쌍의 집합을 MTa_p 라 정의하면, $n \geq 1$ 인 경우 매핑된 용어쌍들인 MTa_p 는 매핑 테이블에 저장된다. 그러나 $n = 0$ 인 경우, 지식 엔지니어가 직접적으로 매핑 결정을 해야 한다. 엔지니어의 매핑 결정 시 수치화한 두 용어의 유사도를 제공함으로써 엔지니어의 의사결정을 지원할 수 있다. 용어의 유사도 측정 방법은 4.4절에서 설명한다.

위의 예에서, $f(TS_{uniPart_for_assembly})$ 의 값은 1이다. 따라서 다음과 같은 결과를 얻는다.

$$MTa_{uniPart_for_assembly} = \{(uniPart_for_assembly, assy_component)\}.$$

반면에, a_p 가 'uniPart'인 경우, $TS_{uniPart}$ '와 $f(TS_{uniPart})$ 는 다음과 같다.

$$TS_{uniPart} = \{(uniPart, assy_component), (uniPart, ep_component)\}.$$

$$f(TS_{uniPart}) = 0.$$

이와 같은 경우 유사도 측정을 하여 사용자가 직접 매핑을 하거나 온톨로지를 수정하도록 지원한다.

4.4 유사도 측정

매핑 가능성이 있는 용어들 간의 유사도는 온톨로지 구조의 함수관계와 구조관계를 이용한다. 함수 관계의 변역(domain)과 치역(range)을 이용하여 관계 유사도 (Relation Similarity)를 구하고, 개념의 구조관계를 이용하여 유사도 (Hierarchical Similarity)를 구한다. 본 연구는 TSa_p '에 해당하는 모든 용어쌍들을 대상으로 두 종류의 유사도를 구하고, 이들의 가중치 합을 최종 유사도로 사용한다.

4.4.1 관계성 유사도

관계성 유사도는 두 개념이 갖는 관계가 유사하다면 서로 매핑될 가능성이 높다는 가정을 활용한다. 관계성 유사도는 'A'의 용어가 갖을 수 있는 관계들 중 'B' 용어가 갖을 수 있는 관계와 매핑된 관계들의 비율로 수치화 한다. 관계 ' r_j '에 대하여 $dom(r_j)$ 는 관계 ' r_j '의 변역을 반환하는 함수로, $range(r_j)$ 는 관계의 치역을 반환하는 함수로 정의한다. 또한, 개념 c 에 대하여 개념 c 자체와 c 의 상위 개념들로 구성된 집합을 반

환하는 함수 $Sup(c)$ 를 정의한다. '≡'는 두 용어가 의미적으로 동일하다는 것을 나타내는 부호로 사용한다. 용어 a_p 를 변역 혹은 치역으로 갖는 관계들의 집합 Ra_p 를 다음과 같이 정의하고, 같은 방법으로 Rb_q 를 정의할 수 있다.

$$Ra_p = \{r_j | dom(r_j) \subseteq Sup(a_p) \text{ or } range(r_j) \subseteq Sup(a_p), r_j \in R_A, 1 \leq j \leq n\}.$$

$$Rb_q = \{r_h | dom(r_h) \subseteq Sup(b_q) \text{ or } range(r_h) \subseteq Sup(b_q), r_h \in R_B, 1 \leq h \leq m\}.$$

만약 a_p 가 'uniPart'라면,

$$TS_{uniPart} = \{(uniPart, ep_component), (uniPart, assy_component)\},$$

이것, 이 때 b_q 를 'ep_component'라 가정하면,

$$Sup(uniPart) = \{part, uniPart\}$$

$$dom(isSubPartOf) = \{part\} \subseteq Sup(uniPart).$$

$$range(isSubPartOf) = \{part\} \subseteq Sup(uniPart).$$

$$\therefore R_{uniPart} = \{isSubPartOf\}.$$

$$Sup(ep_component) = \{part\}$$

$$dom(isSubPartOf) = \{part\} \subseteq Sup(ep_component).$$

$$range(isSubPartOf) = \{part\} \subseteq Sup(ep_component).$$

$$\therefore R_{ep_component} = \{isSubPartOf\}.$$

a_p 와 b_q 가 갖는 관계들 중 매핑이 된 관계에 대하여 변역에 해당하는 용어 혹은 치역에 해당하는 용어가 같은 관계들의 짝을 원소로 갖는 집합 SRa_p, b_q '를 다음과 같이 정의한다.

$$SRa_p, b_q = \{(r_j, r_h) | dom(r_j) \equiv dom(r_h) \text{ for } range(r_j) = a_p \text{ and } range(r_h) = b_q \text{ or } range(r_j) \equiv range(r_h) \text{ for } domain(r_j) = a_p \text{ and } domain(r_h) = b_q, r_j \in Ra_p, r_h \in Rb_q, (a_p, b_q) \in TSA_p\}.$$

만약, $|Ra_p| = x$ 이고 $|SRa_p, b_q| = y$ 이면, 관계성 유사도, $Sim_1(a_p, b_q)$ 는 다음과 같이 구해진다.

$$Sim_1(a_p, b_q) = 0, \text{ if } x = 0 \text{ and } y \neq 0.$$

$$Sim_1(a_p, b_q) = \alpha, \text{ if } x = 0 \text{ and } y = 0. \text{ 사용자의 판단에 따라 } \alpha \text{에 값을 부여함.}$$

$$Sim_1(a_p, b_q) = y/x, \text{ 그 외의 경우.}$$

앞 예제에서, $|R_{uniPart}| = 1, |SR_{uniPart, ep_component}| = 1$ 이므로 $Sim_1(uniPart, ep_component) = 1$ 이 된다.

4.4.2 계층적 유사도

계층적 유사도는 두 개념의 부모 개념 혹은 자식 개념 간에 서로 매핑이 되어 있다면, 두 개념 간에도 매핑이 될 가능성이 높다는 가정을 활용한다. 계층적 유사도는 'A'의 용어가 갖는 부모/자식 개념들 중에서 'B'의 용어가 갖는 부모/자식 개념과 매핑된 개념의 비율로 수치화한다. 우선, a_p 의 직접적인 상위 개념에 해당하는 용어들의 집합 Ca_p^{super} 와 a_p 의 직접적인 하위 개념에 해당하는 용어들의 집합 Ca_p^{sub} 를 다음과 같이 정의할 수 있다. 마찬가지로 b_q 의 직접적인 상위 개념에 해당하는 용어들의 집합 Cb_q^{super} 와 b_q 의 직접적인 하위 개념에 해당하는 용어들의 집합 Cb_q^{sub} 를 다음과 같이 정의할 수 있다.

$$Ca_p^{super} = \{ca_p^{w1} \mid HF(ca_p^{w1}, a_p), 0 \leq w1 \leq n\}.$$

$$Ca_p^{sub} = \{ca_p^{w2} \mid HF(a_p, ca_p^{w2}), 0 \leq w2 \leq n\}.$$

$$Cb_q^{super} = \{cb_q^{w3} \mid HF(cb_q^{w3}, b_q), 0 \leq w3 \leq m\}.$$

$$Cb_q^{sub} = \{cb_q^{w4} \mid HF(b_q, cb_q^{w4}), 0 \leq w4 \leq m\}.$$

앞의 예에서, a_p 가 'uniPart'인 경우의 $TS_{uniPart}^{l=1} = \{(uniPart, ep_component), (uniPart, assy_component)\}$ 이므로, 다음과 같은 결과를 얻을 수 있다.

$$C_{uniPart}^{super} = \{part\}.$$

$$C_{uniPart}^{sub} = \{uniPart_for_endProduct, uniPart_for_assembly\}.$$

$$C_{ep_component}^{super} = \{part\}.$$

$$C_{ep_component}^{sub} = \emptyset.$$

$$C_{assy_component}^{super} = \{part\}.$$

$$C_{assy_component}^{sub} = \emptyset.$$

집합 $SHRa_p, b_q^{super}$ 는 Ca_p^{super} 에 해당하는 용어 중에서 Cb_q^{super} 에도 동시에 속하는 용어, 즉, 서로 매핑이 된 용어쌍을 갖는 집합이며 다음과 같이 정의한다.

$$SHRa_p, b_q^{super} = \{(ca_p^{w1}, cb_q^{w3}) \mid ca_p^{w1} \equiv cb_q^{w3} \text{ for } cb_q^{w3} \in Cb_q^{super}\}, ca_p^{w1} \in Ca_p^{super}.$$

만약, $|Ca_p^{super}| = x$, $|Cb_q^{super}| = y$ 이고 $|SHRa_p, b_q^{super}| = z$ 이면, 'Sim₂(a_p, b_q)'로 표현되는 상위 개념과 관련된 유사도는 다음과 같이 얻어진다.

$$Sim_2(a_p, b_q) = 0, \text{ if } x = 0 \text{ and } y \neq 0.$$

$$Sim_2(a_p, b_q) = \beta, \text{ if } x = 0 \text{ and } y = 0. \text{ 사용자의 판단에 따라 } \beta \text{에 값을 부여함.}$$

$$Sim_2(a_p, b_q) = z/x, \text{ 그 외의 경우.}$$

유사한 방법으로, 하위 개념과 관련된 유사도

$SHRa_p, b_q^{sub}$ 를 다음과 같이 정의할 수 있다.

$$SHRa_p, b_q^{sub} = \{(ca_p^{w2}, cb_q^{w4}) \mid ca_p^{w2} \equiv cb_q^{w4} \text{ for } cb_q^{w4} \in Cb_q^{sub}\}, ca_p^{w2} \in Ca_p^{sub}.$$

하위 개념과 관련된 유사도 'Sim₃(a_p, b_q)'도 Sim₂(a_p, b_q)와 같은 방법으로 얻어진다. 만약, $|Ca_p^{sub}| = x$, $|Cb_q^{sub}| = y$ 이고 $|SHRa_p, b_q^{sub}| = z$ 이면,

$$Sim_3(a_p, b_q) = 0, \text{ if } x = 0 \text{ and } y \neq 0.$$

$$Sim_3(a_p, b_q) = \gamma, \text{ if } x = 0 \text{ and } y = 0. \text{ 사용자의 판단에 따라 } \gamma \text{에 값을 부여함.}$$

$$Sim_3(a_p, b_q) = z/x, \text{ 그 외의 경우.}$$

앞의 예에서, 다음과 같은 결과를 얻을 수 있다.

$$SHR_{uniPart, ep_component}^{super} = \{(part, part)\},$$

$$SHR_{uniPart, ep_component}^{sub} = \emptyset,$$

$$SHR_{uniPart, assy_component}^{super} = \{(part, part)\},$$

$$SHR_{uniPart, assy_component}^{sub} = \emptyset,$$

$$Sim_2(uniPart, ep_component) = 1 / 1 = 1,$$

$$Sim_3(uniPart, ep_component) = 0 / 2 = 0,$$

$$Sim_2(uniPart, assy_component) = 1 / 1 = 1,$$

$$Sim_3(uniPart, assy_component) = 0 / 2 = 0.$$

최종적인 두 용어 간의 유사도는 'Sim(a_p, b_q)'로 표현하며 아래와 같은 방법으로 얻어진다. 각 유사도에 대한 가중치는 도메인 전문가의 판단에 따라서 값을 부여하게 된다.

$$Sim(a_p, b_q) = \mu_1 \times Sim_1(a_p, b_q) + \mu_2 \times Sim_2(a_p, b_q) + \mu_3 \times Sim_3(a_p, b_q).$$

$\mu_1 = 0.4$, $\mu_2 = 0.3$, $\mu_3 = 0.3$ 와 같이 가중치를 부여하였다면, 최종 유사도는 다음과 같다.

$$Sim(uniPart, ep_component) = 0.4 \times 1 + 0.3 \times 1 + 0.3 \times 0 = 0.7$$

$$Sim(uniPart, assy_component) = 0.4 \times 1 + 0.3 \times 1 + 0.3 \times 0 = 0.7$$

최종적으로, $TS_{a_p}^i$ 에 속하는 모든 b_q 에 대한 최종 유사도가 구해지면, 유사도와 함께 b_q 에 해당하는 용어들이 사용자에게 제시된다. 사용자는 이들 중에서 매핑될 용어를 선택하거나, 적절한 매핑 대안이 없다면 은 톨로지를 적절히 수정해야 한다. 앞 예제의 경우, 'uniPart'에 대해서 'ep_component', 'assy_component'가 유사도와 함께 사용자에게 제시된다. 두 용어 모두 동일한 유사도를 갖기 때문에 사용자는 'uniPart' 용

어가 두 용어와 유사한 관계가 있음을 알 수 있고, 이 중에 매핑될 용어를 선택할 수 있다. 온톨로지 수정을 통해 'uniPart'와 매핑이 되는 'B' 회사의 용어를 추가하거나 'uniPart' 용어를 매핑 대상에서 제외 시킬 수 있다.

5. SOM 프로토 시스템 설계

5.1 SOM 프로토 시스템 아키텍처

SOM 프로토 시스템 아키텍처는 크게 네 부분으로 구성 된다. 첫 번째 Core Mapper는 본 연구에서 제안된 매핑 알고리즘이 구현되는 부분이다. 두 번째 온톨로지 번역기는 XML을 Prolog로 혹은 그 반대로 변환 시키고 온톨로지 베이스에 저장하는 기능을 수행하는 부분이다. 다음으로, 온톨로지 베이스는 온톨로지 구성요소들(개념, 관계, 개념 구조관계, 함수 관계, 공리)과 용어에 대한 정의를 갖고 있으며, 매핑 결과가 저장되는 저장소를 의미한다. 마지막으로 추론모듈은 매핑 알고리즘 수행 중 역방향 추론을 수행하는 부분이다. Core Mapper 부분과 온톨로지 번역기 부분은 C#과 같은 프로그램 언어로 구현이 가능하며, 온톨로지 베이스는 SQL Server와 같은 데이터베이스로 구현이 가능하다. 추론모듈은 Prolog와 같은 역방향 추론엔진을 활용하여 구현이 가능하다. Fig. 4는 SOM 프로토 시스템의 아키텍처를 그림으로 표현하고 있다.

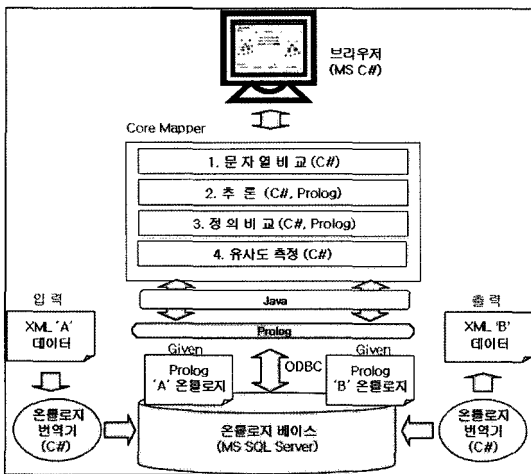


Fig. 4. SOM 프로토 시스템 아키텍처.

5.2 SOM 프로토 시스템 프로시저

SOM 프로토 시스템은 A회사의 용어로 표기된

XML 형식의 문서를 읽어 들여서, 매핑된 B 회사의 용어로 표기된 XML 형식의 문서를 작성한다. SOM 프로시저를 간단히 설명하면 다음과 같은 하위 과정들로 나눌 수 있다.

선행 조건: 입/출력 정보로 사용되는 XML문서의 정합성을 검증할 수 있도록 XML 스키마가 주어져야 한다. 또한 두 회사의 용어의 온톨로지가 미리 온톨로지 베이스에 정의되어 있어야 한다.

입력 문서 처리: SOM 프로토 시스템은 인스턴스 정보를 갖는 XML 형식의 입력 문서를 XML 스키마를 참조하여 검증한다. 그리고 SOM 프로토 시스템은 입력 문서의 정보를 미리 정의된 온톨로지를 이용하여 검증을 하며, 이 과정을 마치면 Prolog 형식의 파일이 작성된다. Prolog 프로그램에서 추론을 수행하기 위해서는 Prolog 코드의 작성 순서가 매우 중요하다. Prolog 코드는 다음과 같은 순차로 작성된다.

- 1) Prolog 리스트를 다루는 일반 함수 코드
- 2) A 회사 개념의 인스턴스 리스트
A 회사 관계의 인스턴스 리스트
- 3) B 회사의 개념 리스트
B 회사의 관계 리스트
B 회사의 개념 구조 리스트
- 4) B 회사의 개념과 관계에 대한 공리

이렇게 작성된 Prolog 파일은 사용자에 의해서 일부수정 후 Prolog 프로그램에서 컴파일 되고 저장된다.

매핑 수행: Prolog 파일에 있는 두 회사의 용어에 대한 온톨로지 정의를 이용하여, 매핑 알고리즘은 용어 매핑을 수행한다. 두 회사 용어간의 매핑 정보는 온톨로지 베이스의 매핑 테이블에 저장된다.

출력 문서 작성: 매핑 정보를 이용하여, A 사의 인스턴스 정보를 B회사의 용어를 사용하여 표현할 수 있다. 우선, 온톨로지 정합성 검증을 거쳐서 Prolog 형식의 출력 문서를 작성하게 된다. 이 문서는 B회사의 XML 스키마를 참조하여 정합성을 검증한다. 검증이 끝나면 B회사의 용어로 표현된 XML형식의 문서가 최종적으로 작성된다.

Fig. 5는 SOM 시스템에서 사용하는 입력문서의 예이고, Fig. 6은 Fig. 5의 입력문서를 SOM 프로시저에 따라 수행한 출력문서이다. 매핑 결과에 따라 A

회사 용어의 인스턴스들이 B 회사용어로 표현되었음을 알 수 있다. Fig. 7은 SOM 프로토 시스템의 상세한 전체 프로시저를 보여준다.

6. 결 론

본 연구에서는 서로 다른 용어를 매핑시키는 의미 매핑 알고리즘을 제안하였으며, SOM(Semantic Ontology Mapper) 프로토 시스템의 프로시저를 개발하였다. 매핑 알고리즘은 문자열 비교, 추론, 정의 비교, 유사도 측정의 4단계로 이루어진다. SOM 프로토 시스템은 용어 매핑 결과를 바탕으로 입력 문서가 갖고 있는 A 회사의 정보를 B사의 용어로 변환시켜준다. SOM 프로토 시스템은 특별한 표준에 따르지 않고도, 기업간 정보 공유를 가능하게 한다.

```

</endProduct><motor</endProduct>
<uniPart_for_wildProduct>case</uniPart_for_endProduct>
<uniPart_for_endProduct>ball_bearing</uniPart_for_endProduct>
<uniPart_for_assembly>bar</uniPart_for_assembly>
<uniPart_for_assembly>coil</uniPart_for_assembly>
<uniPart_for_assembly>shaft</uniPart_for_assembly>
<uniPart_for_assembly>brush_spring</uniPart_for_assembly>
<uniPart_for_assembly>condenser</uniPart_for_assembly>
<assembly>magnet</assembly>
<assembly>router</assembly>
<assembly>brush</assembly>
</Concept>
-Relation>
  
```

Fig. 5. SOM 프로토 시스템 입력문서.

```

</endProduct><motor</endProduct>
<assy>magnet</assy>
<assy>router</assy>
<assy>brush</assy>
<ep_component>case</ep_component>
<ep_component>ball_bearing</ep_component>
<assy_component>bar</assy_component>
<assy_component>coil</assy_component>
<assy_component>shaft</assy_component>
<assy_component>brush_spring</assy_component>
<assy_component>condenser</assy_component>
</Concept>
-Relation>
  
```

Fig. 6. SOM 프로토 시스템 출력문서.

본 알고리즘은 반자동화 매핑방법으로 사람의 의사 결정을 필요로 한다. 재귀적으로 알고리즘을 활용하는 정의 비교 단계에서도 일정 횟수 이상 반복을 하지 않도록 하였다. 따라서, 본 알고리즘은 일정 시간 이내에 종료될 수 있다. 또한 용어의 논리적 정의에 따라 자동 매핑이 이루어지고, 자동 매핑이 되지 않는 용어들은 사용자의 판단에 따라 매핑이 되므로 이상적으로는 잘못된 매핑 결과가 도출되지 않는다.

하지만 제안된 알고리즘은 다음과 같은 한계점을

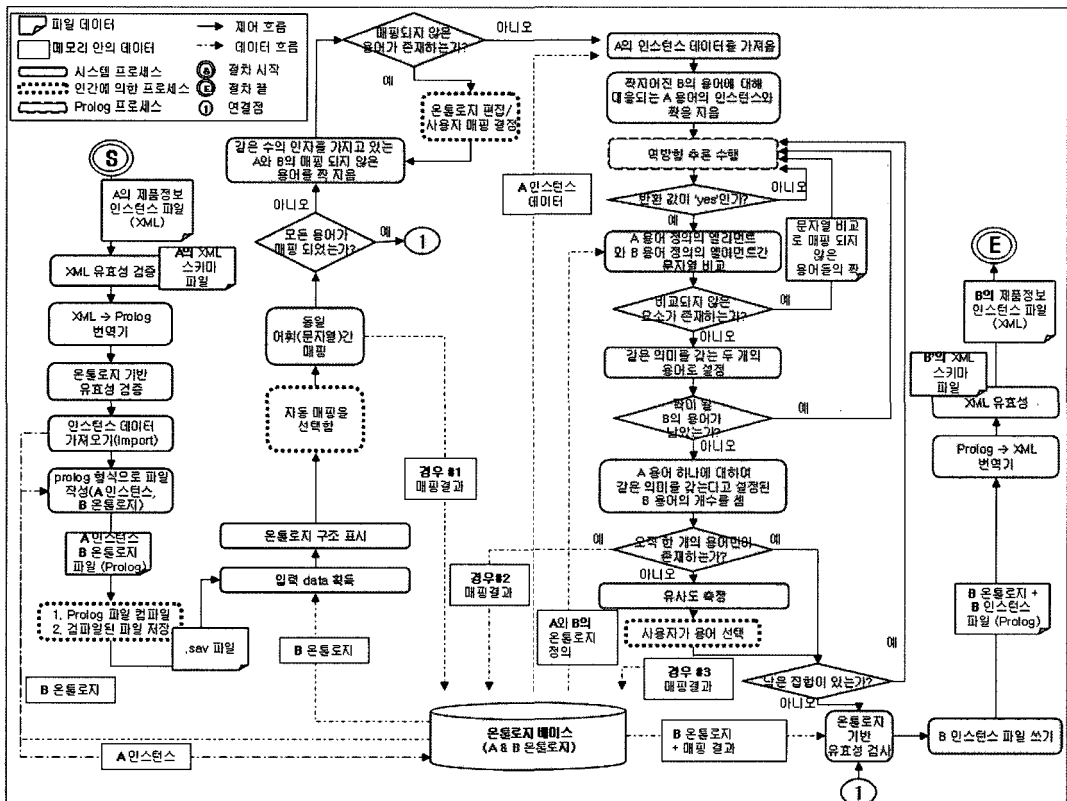


Fig. 7. SOM 프로토 시스템 프로시저.

갖고 있다. 첫 번째로 용어의 정의 및 공리가 포함하는 논리 요소들을 비교하여 재귀적으로 매핑관계를 추론하기 때문에 논리 요소들이 증가할수록 정의를 비교하는 프로세스의 복잡도가 상당히 커지게 된다. 두 번째로 두 도메인간 동일한 용어는 동일한 의미를 갖는다는 가정을 두었기 때문에 현실에서 두 도메인 간에 동일한 용어를 서로 다른 의미로 사용할 경우 매핑 프로세스가 정상적으로 수행될 수 없다는 한계가 있다. 또한, 사용자의 매핑 의사결정이 올바르게 없을 경우 이후 자동 매핑 과정에서 오류가 발생할 수 있다. 마지막으로 본 알고리즘이 두 도메인의 공리와 정의에 전적으로 의존하기 때문에 두 도메인의 온톨로지가 상세하고 일관된 형태로 정리되지 않으면 매핑 프로세스가 정상적으로 수행될 수 없다는 한계가 있다.

따라서, CPC 환경에서의 자유로운 정보공유를 위해서는 매핑 알고리즘에 대한 추가적인 연구가 필요하다. 현재의 알고리즘이 현장에 활용되기 위해서는 휴리스틱한 매핑 방법과 결합하여 정의를 비교하는 프로세스를 단순화 시키는 접근방법의 개발이 필요하다. 또한, 현재의 알고리즘은 매핑의 많은 부분을 사용자의 판단에 맡기고 있다. 따라서 자동화된 정보 공유를 위해서는 본 연구의 문제점들을 해결하면서 일반적인 경우에 적용할 수 있는 알고리즘의 개발이 필요하다. 또한, 매핑 기술의 적용을 위해서는 각 기업이 온톨로지를 구축해야 한다. 그러나 온톨로지 디자인은 방대한 양의 지식공학 작업, 많은 시간과 노력을 필요로 한다. 따라서 이에 대한 간편하고 효율적인 접근법이 개발되어야 할 것이다.

참고문헌

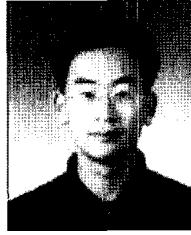
1. Aberdeen Group, "Collaborative Product Commerce: Delivering Product Innovations at Internet Speed", *Market View Point*, Vol. 12, No. 9, 1999.
2. Kim, K. Y. and Suh, H. W., "An Approach to Semantic Mapping using Product Ontology for CPC Environment", 한국 CAD/CAM 학회집, Vol. 9, No. 3, pp. 192-202, 2004.
3. Kurgan, L., Swiercz, W. and Cios, K. J., "Semantic Mapping of XML Tags using Inductive Machine Learning", Proceedings of the 2002 International Conference on Machine Learning and Applications (ICMLA'02), Las Vegas, Nevada, pp. 99-109, 2002.
4. Vdovjak, R. and Houben, G. J., "RDF Based Archi-

5. tecture for Semantic Integration of Heterogeneous Information Sources", *Workshop on Information Integration On the Web*, 2001.
5. Omelayenko, B. and Fensel, D., "Scalable Document Integration for B2B Electronic Commerce", *Special Issue of Electronic Commerce Research Journal on B2B Research*, 2001.
6. Wache, H., Vögele, T., Vissler, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hübner, S., "Ontology-Based Integration Information: A Survey of Existing Approaches", *IJCAI Workshop on Ontologies and Information Sharing*, 2001.
7. Preece, A., Hui, K., Gray, A. and Martí, P., "The KRAFI Architecture for Knowledge Fusion and Transformation", *Proceedings of the 19th SGES International Conference on Knowledge-Based Systems and Applied Artificial Intelligence(ES'99)*, 1999.
8. Mcna, E., Kashyap, V., Sheth, A. and Illaramendi, A., "OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies", *Proceedings 1st IFCIS International Conference on Cooperative Information Systems*, 1996.
9. Stuckenschmidt, H., "Using OIL for Intelligent Information Integration", *Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods at ECAI*, 2000.
10. Arens, Y., Hsu, C. N. and Knoblock, C. A., "Query Processing in the SIMS Information Mediator", *Advanced Planning Technolog*, 1996.
11. Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J. and Widom, J., "The TSIMMIS Project: Integration of Heterogeneous Information Sources", *Conference of the Information Processing Society Japan*, 1994.
12. Calvanese, D., Giacomo, G. D., Lenzerini, M., Nardi, D. and Rosati, R., "Description Logic Framework for Information Integration", *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference*, 1998.
13. Goasdoue, F., Lattes, V. and Rousset, M. C., "The Use of Carin Language and Algorithms for Information Integration: The PICSEL System", *International Journal of Cooperative Information Systems (IJCIS)*, 2000.
14. Bozak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S. and Stojanovic, L., "KAON - Towards a Large Scale Semantic Web", *Lecture Note Computer Science*, No. 2455, pp. 304-313, 2002.
15. Stuart Russell and Peter Norvig, "Artificial Intelligence", Prentice Hall, pp. 185-216, 1995.



정 원 철

2002년 한국과학기술원 산업공학과 학사
2004년 한국과학기술원 산업공학과 석사
2004년~현재 LG 전자 생산기술 연구원
시스템 기술 그룹
관심분야: 온톨로지 기반 정보 교환, 온
톨로지 기반 지식관리시스템



이 재 현

1999년 한국과학기술원 산업공학과 학사
2000년 한국과학기술원 산업공학과 석사
2001년~현재 한국과학기술원 산업공학과
박사과정 재학
관심분야: 제품 정보 관리, 온톨로지기반
지식관리시스템



서 효 원

1981년 연세대학교 기계공학과 학사
1983년 한국과학기술원 기계공학과 석사
1991년 West Virginia University 산업
공학과 박사
1983년~1987년 대우중공업(주) 중앙연구
소 주임연구원
1992년~1995년 생산기술연구원 생산시
스템센터 수석연구원

1996년~현재 한국과학기술원 산업공학과 교수
관심분야: CE/PDM/CPC/PLM, Worklow Management/BPM,
Ontology/Knowledge Based System