

# 트리 유사도: 상호운용성 평가도구

정성훈<sup>○</sup>, 배재학\*  
울산대학교 IT융합학부

gomgom9kr@outlook.kr, jhjbae@ulsan.ac.kr

## Tree Similarity: Interoperability Evaluation Tool

Seonghoon Jeong<sup>○</sup>, Jae-Hak J. Bae\*  
School of IT Convergence, University of Ulsan

### 요 약

현대사회에 존재하는 다양한 시스템들이 병합될 때는 병합을 위해서 여러 가지 방법을 사용해 볼 수 있다. 이때 시스템의 성격에 따라 더 적절한 병합 방법론이 존재할 수 있지만, 어떤 방법이 해당 시스템을 통합하는데 더 적절한지를 판단하기는 쉽지 않다. 본 논문에서는 서로 다른 시스템을 통합할 때, 그 상호운용성을 평가하기 위한 수단으로 트리의 유사도를 측정하는 방안을 제시한다. 이렇게 측정된 유사도는 0 이상 1이하의 값을 가지며, 정확한 수치로 제시되기 때문에 서로 다른 통합 방법론을 평가하기 위한 계량적 근거로 사용될 수 있다. 다만 트리 구조로 나타낼 수 없는 일부 시스템들에 대해서는 적용할 수 없는 한계를 가진다.

**주제어:** 상호운용성, 트리, 유사도, 온톨로지

## 1. 서론

### 1.1. 연구배경

현대사회에서는 다양한 조직이 존재하고, 그 조직들은 고유의 시스템을 가지고 있다. 이러한 조직들은 필요에 따라서 조직들간의 병합이나 분리과정을 거치게 된다.

조직들이 병합할 때에는 그 조직들이 가지고 있는 고유한 시스템들 또한 같은 과정을 거치는데, 이러한 통합을 위한 방법론은 여러 가지가 존재한다. 온톨로지 분야를 예로들면 언어적인 방법(Lexical), 구조적인 방법(Structural), 인스턴스에 의한 방법(Instance-Based), 간접적으로 정렬하는 방법(Mediated), 의미론적 유사성에 의존하는 방법(Semantic Similarity)등의 방법이 존재하며[1], 이를 위한 소프트웨어도 다수 존재한다[2]. 흔히 사용되는 관계형 데이터베이스 또한 기본적인 join 연산을 통한 병합 외에도 객체지향적인 개념의 통합 설계방법론이 존재한다[3].

이렇듯 통합방법론이 한가지로 수렴하지 않고 여러 가지가 존재하는 것은, 각 방법론마다 고유의 장점과 약점이 존재하기 때문이다. 하지만 실제로 개별 시스템에 어떤 방법이 더 적합한지를 판단하는 것은 매우 어려운 일이다.

### 1.2. 연구의 필요성

개별 시스템에 더 적절한 방법론을 판단하는 것은 어려운 일이지만, 또한 매우 중요한 일이기 때문에 정부와 기업들 모두 해당 주제를 긴 시간동안 활발히 연구해왔다[4]. LISI, i-Score등의 평가도구들은 모두 이러한 노

력의 결과로 고안된 것 들이다[4].

하지만 이러한 도구들은 고도의 수학적 지식이 필요할 뿐 아니라 복잡한 연산과정이 필요하기 때문에 비교적 많은 비용을 요구한다.

### 1.3. 연구목표

본 연구에서는 시스템 사이의 통합의 정도를 평가하는 방법으로 트리 유사도를 소개한다. 트리로 표현된 통합 시스템과 기존의 시스템간의 유사도를 절대적인 수치로 표현할 수 있다면, 이는 상호운용성을 평가하는 계량적인 근거로 사용될 수 있을 것이다. 또한 트리의 순회와 정수 연산과 같은 비교적 단순한 연산기법을 사용하여 비교적 적은 자원으로 평가에 필요한 연산을 가능하게 한다.

## 2. 관련 연구

### 2.1. 유물 분류 시스템 통합

이 논문은 서로 다른 박물관에서 사용하는 온톨로지로 표현된 유물 분류체계를 통합하는 방법[5]에 대해서 설명한다. 해당 논문에서는 확실히 동일한 것으로 간주할 수 있는 실제 유물에 대한 분류계(Classification)를 채널(Channel)로 두고 기존의 시스템들을 각 레벨별로 서로에 대한 제약식을 구하는 과정을 반복하는 것으로 두 박물관 분류체계간의 상호운용성을 확보하는 과정을 거친다.

\* 교신저자

2.2. 온톨로지 통합 방법에 관한 연구

온톨로지 정렬(Alignment)[2]에는 언어적인 방법, 구조적인 방법, 인스턴스에 의한 방법, 간접적으로 정렬하는 방법, 의미론적 유사성에 의존하는 방법 등 복수의 온톨로지를 정렬하는 여러 방법론이 개발되어 있다. 각 방법은 고유의 장단점을 가지고 있다.

2.3. 상호운용성 평가에 관한 메타분석

상호운용성(Interoperability)은 사용되는 맥락에 따라서 여러 가지로 정의할 수 있다[4]. 이를 바탕으로 상호운용성의 종류와 평가를 위한 여러 접근법을 고찰해 볼 수 있다. 본 연구에서는 상호운용성을 “둘 이상의 시스템이 이질적인 네트워크에서 정보를 교환하고 사용하는 능력” [4][6]으로 정의한다.

3. 제약식과 온톨로지구조 표현

시스템을 병합할 때는 Information flow 이론[7]에 기초를 두면 유리한데, 이는 간단한 수학적 구조로 나타낼 수 있을 뿐 아니라 구조가 유연해서 큰 수정 없이 여러 경우에 일반적으로 적용할 수 있기 때문이다. Information flow 이론에서는 직접적으로 트리를 나타내기 보다는 제약식(Constraint)의 형식을 사용하여 논리적 구조를 표현한다. 표1에서는 제약식의 몇 가지 예시를 보여준다.

표 1 : 제약식의 특별한 경우 [7]

제약식	의미
$a \vdash \beta$	a가 논리적으로 $\beta$ 를 수반한다.
$\vdash a$	a는 항상 참이다.
$a \vdash$	어떠한 토큰(token)도 a에 속하지 않는다.
$\vdash a, \beta$	모든 토큰이 a또는 $\beta$ 에 속한다.
$a, \beta \vdash$	a와 $\beta$ 는 상호배타적이다.

비록 이러한 제약식은 원래 논리관계를 설명하기 위한 도구이지만, 트리로 표현된 시스템 또한 묘사할 수 있다. 예를 들어 “ $\vdash a$ ”는 트리에서 a가 최상위에 존재하는 노드라는 것을 나타내는데 사용될 수 있고, “ $a \vdash \beta$ ”는 a가  $\beta$ 의 자식노드를 나타내는 것으로 이해할 수 있다. 또한 “ $a, \beta \vdash$ ”는 a와  $\beta$ 가 서로 다른 부모 노드에 포함되었다고 이해할 수 있을 것이다.

온톨로지 구조 또한 이와 유사하게 트리 구조로 표현이 가능하다. a의 입장에서 “ $a \vdash \beta$ ”를 “subclass of  $\beta$ ”로, “ $a, \beta \vdash$ ”를 “disjoint with  $\beta$ ”로 표현한 것으로 이해할 수 있다.

표 2는 2.1.논문에서 제시된 통합된 시스템을 나타내는 제약식들이고, 그림 1은 이를 바탕으로 구성한 트리의 모습이다.

표 2 : 통합된 박물관 시스템에서의 제약식 [5]

	$ADV \vdash DV$
	$ACV \vdash DV$
$FCV \vdash BGV$	
$AV \vdash BGV$	
$MPV \vdash BGV$	$ADV \vdash BGV$
$BGV, WV \vdash$	$ACV \vdash BGV$
$CV = FCV$	$J \vdash AV$
$MPV = MV$	$Z \vdash AV$
$AV = DV$	$J = ADV$
	$Z = ACV$

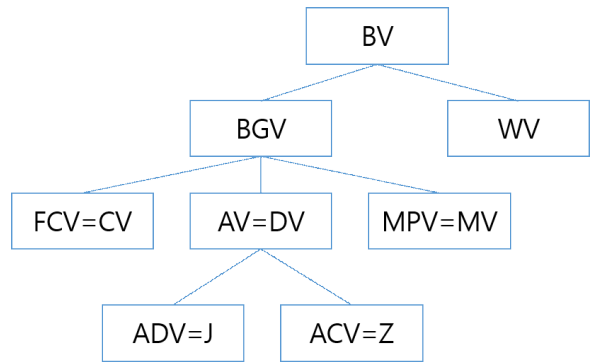


그림 1 : 통합된 청동기 유물 분류계

4. 구조적 유사도 평가

본 장에서는 2.1.논문에서의 트리를 사용하여 실제로 트리의 유사도를 구하고 상호운용성을 평가해본다. 아래의 그림 2와 그림 3은 통합되기 전의 트리를 나타낸다.

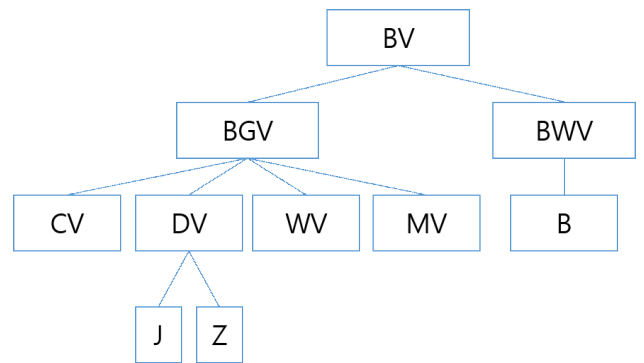


그림 2 : 청동기에 대한 유물 분류계 A [5]

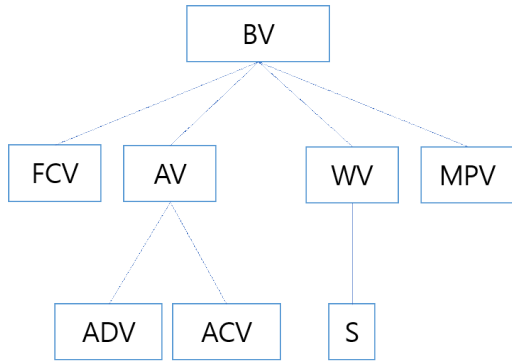


그림 3 : 청동기에 대한 유물 분류계 B [5]

이하의 과정에서 위 분류계 A와 통합 분류계, 또 분류계 B와 통합 분류계 간의 유사도를 구할 것이다. 계산과정은 최종적으로 유사도가 0이상 1이하의 값을 가지도록 고안되었으며, 두 트리가 조금도 유사하지 않을 경우 0, 완전히 동일한 경우 1의 값을 가지게 된다.

실제 계산은 영향을 주는 요인에 대해서 점수를 할당하고 얻은 수치의 합을 구하고, 획득할 수 있는 최대 점수를 나누는 것으로 이루어진다.

4.1. 동일한 노드의 존재유무 확인

트리의 유사도를 계산할 때 가장 먼저 고려해야 할 것은 비교하는 트리 사이에 같은 노드가 얼마나 존재하는가 하는 점이다.

이때 점수는 각 노드에 대해서, 비교하는 트리에 동일한 노드가 존재한다면 1점, 존재하지 않는다면 0점을 부여하고, 획득할 수 있는 최대점수는 비교하는 두 트리의 노드의 개수의 합이 될 것이다.

4.2. 동일한 부모 노드 보유 여부

비교대상 트리에 동일한 노드가 존재하는 것을 확인했을 때 양쪽 노드의 부모 노드를 확인하여 부모가 동일한 노드인지 체크하여 점수를 부여할 수 있다. 이는 개별 노드의 입장에서 보면 단순히 부모를 체크하는 작업이지만, 전체적으로 보면 트리 내에서 노드의 상대위치를 파악하는 효과를 얻을 수 있다.

상대위치를 파악하는 것은 자식을 비교하는 것으로도 가능하지만, 일반적으로 부모 노드는 최상위 노드를 제외하고 전부 보유하고 있으며, 특별한 경우가 아니면 자식노드는 다수가 존재할 수 있으나, 부모 노드는 오직 하나만 존재하기 때문에 계산하기에 용의하다.

점수는 비교트리에 동일한 노드가 존재하고 그 노드의 부모 노드까지 동일하면 1점, 노드가 존재하지 않거나 존재하지만 부모 노드가 다르다면 0점을 부여하고, 획득할 수 있는 최대점수는 위의 과정과 마찬가지로 비교하는 두 트리의 노드의 개수의 합이 될 것이다.

4.3. 유사도 계산

편의상 위 두 과정을 별개의 과정으로 서술하였지만, 구현의 난이도나 효율성 측면에서 보면 두 과정을 동시에 두고 보는 것이 더 합리적이다. 따라서 트리를 탐색하면서 두 점수를 동시에 파악하고, (두 트리의 노드의 개수 합 \* 2)를 나누는 식으로 계산을 한다.

이 때 4.1.과 4.2.에서 모두 동일한 노드의 존재유무를 체크하기 때문에 해당항목이 과대평가 되는 것으로 보일 수 있다. 그러나 동일한 노드가 없다면 총 2점 중 한쪽 트리에서만 0점을 받게 되고, 동일한 노드가 존재하지만 부모만 다르다면 양쪽 트리에서 동일한 노드에 대해 각각 1점씩만 부여하게 된다. 전체적으로 보면 두 경우 모두 2점을 감점하는 것이 되고, 따라서 둘에 같은 가중치를 주는 것으로 볼 수 있다.

트리를 탐색하는 방법은 중복 없이 모든 노드를 탐색하는 알고리즘이라면 무엇이든 상관없다. 그림4에서는 BFS(너비 우선 탐색)[8]알고리즘을 사용하는 계산과정을 개략적으로 보여준다.

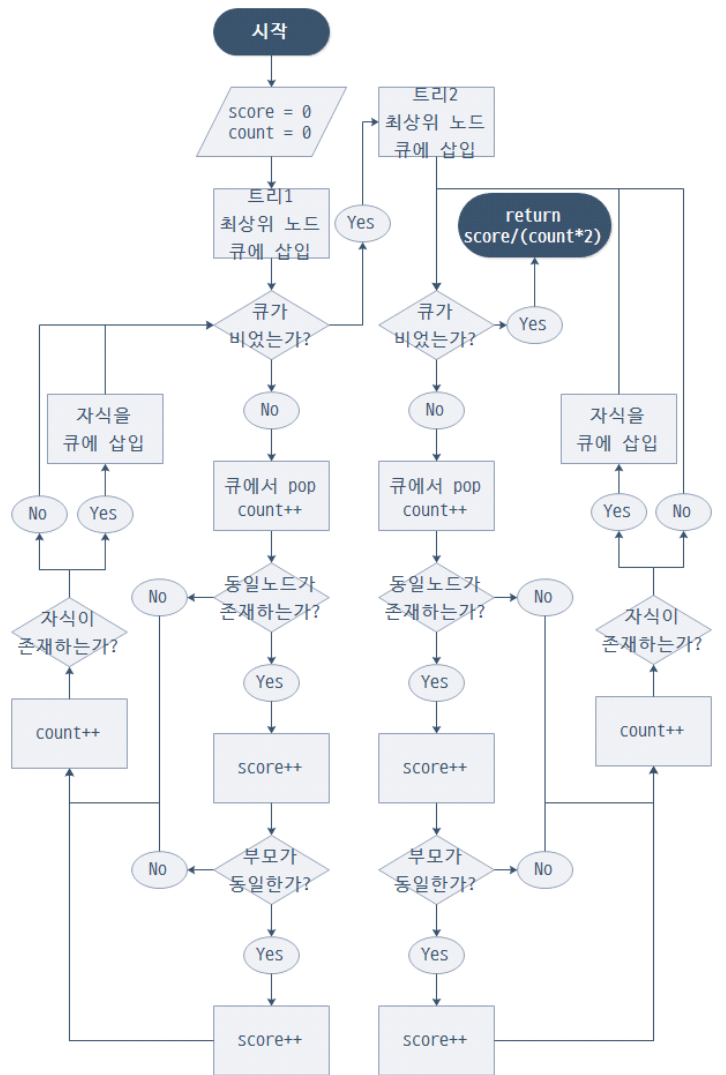


그림 4 : 유사도 계산과정

4.4. 유사도 평가

지금까지 보인 계산과정을 토대로, 본문에서 언급된 통합된 박물관 트리와 기존의 청동기 유물 분류 트리의 유사도를 비교해보면 표3 과 같다.

표 3 : 통합 트리와 기존 트리의 유사도 비교 결과

트리 A / 통합 트리	트리 B / 통합 트리
0.8889	0.6875

비교 결과 트리 A는 약 89%, 트리 B는 약 68%정도의 유사도를 가지므로 위 예시에서 병합된 트리는 기존 트리들의 상호운용성을 비교적 잘 반영하고 있다고 할 수 있다. 만약 2.1. 논문에서 제안한 것과 다른 병합 방법을 사용하고자 한다면, 위의 결과보다 더 높은 유사도를 보이는 방법을 찾아야 할 것이다.

5. 의미론적 유사도 평가

앞서 살펴본 방법은 구조상의 유사도를 평가할 수 있지만, 만약 트리 구조 자체와 상관없는 기능적인 역할이 시스템에 존재했을 경우 오직 구조만을 평가하고 의미론적인 유사도는 평가에 반영하기 힘들게 된다. 이러한 경우의 예시로는 온톨로지의 속성(Property), 객체지향 프로그램에서의 함수(Method)등을 들 수 있다. 이러한 것들은 트리의 구조와는 별개로 평가해줄 필요가 있다. 이러한 기능들은 기본적으로 함수의 꼴을 취할 수 있기 때문에 동일한 입력(Parameter)에 대해서 유사한 출력(Return)을 가지는 정도를 측정하는 것으로 의미론적인 유사도의 부재를 보완할 수 있다.

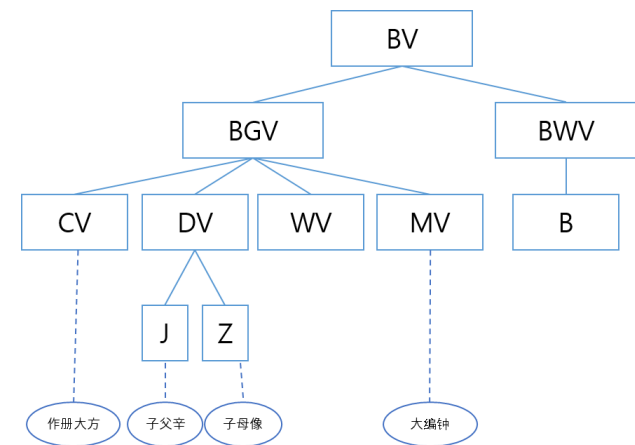


그림 5 : 인스턴스를 포함한 유물 분류계 A [5]

위에서 예로 들었던 유물 분류 시스템에서 이러한 예를 찾는다면 유물의 검색기능을 예로 들 수 있다. 이 기능을 함수 꼴로 생각한다면, 입력은 실재 존재하는 유물 인스턴스(Instance)가 되고, 출력은 그 유물이 속해있는 클래스(Class)가 될 것이다.

표 4 분류계A에서의 getClassification()

호출 형태	결과 값
getClassification(作册大方)	CV
getClassification(子父辛)	J
getClassification(子母像)	Z
getClassification(大编钟)	MV

표 5 통합된 분류계에서의 getClassification()

호출 형태	결과 값
getClassification(作册大方)	FCV = CV
getClassification(子父辛)	ADV = J
getClassification(子母像)	ACV = Z
getClassification(大编钟)	MPV = MV

표4와 표5의 결과를 비교하면 위의 예시는 의미론적으로 유의미한 차이가 없는 것을 볼 수 있다. 이 예시에서는 분류계A 에 대해서만 비교를 해봤지만 분류계B에 대해서도 동일한 결과가 나오는데, 이는 예시가 의미론적 유사성에 기반을 두고 병합을 했기 때문이다.

만약 함수의 실행결과에 차이가 있는 경우라면, 구조적 유사도를 구할 때와 유사한 방법으로 그 차이를 전체 정의역과 치역의 가짓수로 나누어서 0에서 1사이의 값으로 나타낼 수 있을 것이다.

6. 결론

본 연구에서는 통합된 시스템과 기존의 시스템들의 유사도를 측정하는 것으로 두 트리간의 상호운용성을 수치적으로 평가하는 방법을 제시한다.

본문에서는 동일 노드의 존재유무와 트리 내에서의 상대적 위치(동일한 부모 노드)만을 고려하였고, 이 두 요소가 같은 중요도를 가진다고 가정했지만, 실제로 어떤 요소에 더 가중치를 뒀어야 할지, 또한 어떤 변수를 고려해야 할지 여부는 시스템의 성격에 따라 달라질 수 있다. 예를 들어, 군대조직과 같이 실제 수행하는 업무의 내용보다 계급의 높낮이가 더 중요한 조직을 병합할 때는 부모 노드의 비교를 통해 상대위치를 이용하기 보다는 차수(Degree)를 비교해서 절대위치를 비교하는 것이 더 합리적일 것이다.

본 연구에서는 트리의 유사도를 평가도구로 사용하기 때문에 일부 트리 구조로 나타낼 수 없는 시스템을 평가할 때는 적용할 수 없으며, 결국 평가를 위한 개별 항목의 가중치는 일정부분 직관에 의존할 필요가 있다는 한계를 가진다. 이러한 한계들은 후속연구에서 보충되어야 할 것이다.

참고문헌

- [1] Dargie, Walteneagus, ed. "Context-aware computing and self-managing systems, CRC Press", 2009.
- [2] Natalya F. Noy, Mark A. Musen, "The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping," International Journal of Human-Computer Studies, Vol. 59, No. 6, pp. 983-1024, December 2003.
- [3] 주경수, 조도형, "관계형 데이터베이스 응용시스템을 위한 통합 설계방법론 개발" 한국컴퓨터정보학회 논문지 , 16(11), 25-34, 2011.
- [4] Ford, Thomas C., et al. "Survey on Interoperability Measurement" , AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH, 2007.
- [5] Hongzhe Liu, Hong Bao and Junkang Feng, "IF Based Semantic Interoperability for Distributed Digital Museums" , COMPUTING AND INFORMATION SYSTEMS, 10, 1, 2006.
- [6] Morris, E., et al., "System of Systems Interoperability (SoSI): Final Report," Carnegie-Mellon University-Software Engineering Institute, Pittsburgh, PA, Tech. Rep. CMU/SEI-2004-TR-004, Apr. 2004.
- [7] Barwise J and Seligman J, Information Flow: the Logic of Distributed Systems, Cambridge University Press, Vol. 44, 1997.
- [8] E. F. Moore, "Shortest path through a maze" in Annals of the Computation Laboratory of Harvard University, Mass., Cambridge:Harvard University Press, vol. 30, pp. 285-292, 1959.