

온톨로지와 개체관계 모델의 상호운용성에 대한 연구

이 동 훈* · 김 남 규** · 정 인 환***

An Investigation on the Interoperability between Ontology and the Entity-Relationship Model

Donghoon Lee* · Namgyu Kim** · Inhwan Jung***

Abstract

In developing information systems, conceptual modeling is among the most fundamental means. The importance attributed to conceptual modeling has not only given rise to a lot of modeling methods, but also to the “yet another modeling approach (YAMA)” syndrome and the “not another modeling approach (NAMA)” hysteria. Criticism of conceptual modeling methods usually targets their lacking of theoretical foundations. In response to such criticism, various approaches towards theoretical foundations of conceptual modeling have been proposed so far. One of the recent responses to the quest for theoretical foundations of conceptual modeling is the reference to the philosophical ontology. The currently most prominent of diverse approaches towards ontological foundations of conceptual modeling appears to be the Bunge-Wand-Weber (BWW) ontology. Recent approaches attempt to regard BWW ontology as another conceptual data model as well as a criterion for evaluating various conceptual models. However, unfortunately, relatively few researches have been made on interoperability between the Entity-Relationship (ER) model, which is the most dominant conceptual data model, and ontology based model. In this paper, we investigate the interoperability between ontology and the ER model. In detail we (i) reclassify components of ER model with respect to ontology concepts, (ii) identify some components that cannot be directly represented in ontology notation, and (iii) present alternative representations to the components to acquire ontologically clear ER diagrams. Additionally, we (iv) present a set of mapping rules for converting the ontologically clear ER diagram into the corresponding ontology. In a case study, we show the process of converting an ER diagram for a concise Project Management System (PMS) into the ontologically clear ER diagram and the corresponding ontology. We also describe an experiment that we undertook to test whether users understand the Ontologically-Clear ER diagram better.

Keywords : BWW Framework, Conceptual Modeling, Data Modeling, Entity-Relationship Modeling, Ontology

논문접수일 : 2011년 11월 09일 논문게재확정일 : 2011년 12월 06일

※ 본 연구는 한성대학교 교내연구비 지원과제임.

* 국민대학교 비즈니스IT전문대학원 석사과정, e-mail : ldhspace@yahoo.co.kr

** 교신저자, 국민대학교 경영정보학부 조교수, e-mail : ngkim@kookmin.ac.kr

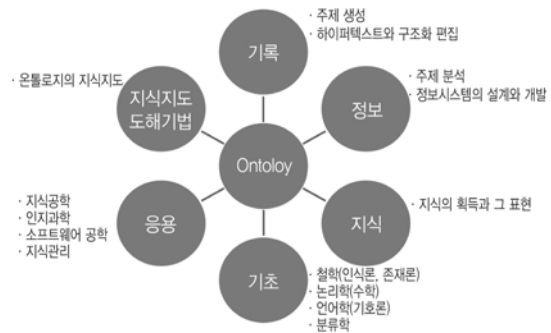
*** 한성대학교 컴퓨터공학과 부교수, e-mail : ihjung@hansung.ac.kr

1. 서론

정보시스템 개발에 있어서 개념 모델의 중요성이 강조되면서 다양한 개념 모델링 방법론들이 고안되어 왔으며, 계속해서 더 나은 모델을 고안하기 위한 연구들이 수행되고 있다. 개념 모델링에 대한 지나친 관심은 수많은 개념 모델을 양산하는 결과를 야기하였으며, 심지어 “계속해서 새로운 방법론을 찾는 신드롬”(YAMA : Yet Another Modeling Approach)[Oei et al., 1982]과 “더 이상 새로운 방법론은 없다는 히스테리”(NAMA : Not Another Modeling Approach)[Siau, 2003]를 일으키기도 하였다. 개념 모델링 분야가 직면한 이러한 어려움은, 개념 모델이 충분한 이론적 토대 위에서 생성되지 않았다는 비판과 무관하지 않다. 즉 실세계의 현상을 왜 특정 방식으로 표현해야 하는지에 대한 이론적 근거를 찾기가 어렵다는 것이다. 예를 들면 개념 데이터 설계에 가장 널리 사용되는 개체관계(ER : Entity-Relationship) 모델의 경우 동일한 실세계에 대해서도 설계자의 경험이나 직관에 따라 개체, 관계, 속성이 상이하게 나타날 수 있는데[Wand et al., 1995], 이러한 현상 역시 ER 모델의 구성 요소에 대한 충분한 이론적 토대를 찾기가 어렵기 때문이라는 것이다. 이러한 한계를 극복하기 위해 개념 모델의 이론적 토대를 마련하기 위한 많은 연구가 수행되고 있으며, 그 일환으로 개념 모델의 이론적 토대로서의 철학적 온톨로지(Ontology)에 대한 관심이 증대되고 있다.

온톨로지는 크게 의미의 기본 단위인 개념과 개념들 사이의 관계들로 표현되며, Gruber[1993]가 정의한 “어떤 관심 분야를 개념화하기 위해 명시적으로 정형화한 명세서”라는 개념으로 주로 사용된다. 철학에서 시작된 온톨로지는 그 배경인 역사학, 논리학, 수학, 언어학, 기호론 등에 의하여 보다 구체적으로 확립되고, 분류학이라

는 기본 원리로 집대성되었다. 이러한 온톨로지는 최근에는 정보과학, 인지과학, 지식공학, 기록정보학 등의 영역에서 연구되고 있으며, 시맨틱 웹[윤홍원 등, 2005; Berners-Lee et al., 2001], 자연어 처리, 지식관리 시스템, 정보검색 및 추출 등의 다양한 분야에 응용되고 있다(<그림 1> 참조).



<그림 1> 온톨로지의 주요 영역

정보시스템 모델링 분야에 온톨로지 개념을 도입한 가장 괄목할만한 시도는 Wand and Weber [1993]에서 찾을 수 있다. Wand and Weber[1993]는 정보시스템 모델링의 이론적 배경을 철학에서 찾고자 시도하였으며, 철학자 Mario Bunge가 정리한 온톨로지[Bunge, 1977; Bunge, 1979]가 정보시스템을 모델링하기에 적합한 특성을 갖고 있는 것으로 판단하였다. 이들은 Bunge의 온톨로지서 정보시스템 모델링에 필요한 개념을 차용하여 BWW(Bunge-Wand-Weber) 온톨로지 프레임워크를 설계하였는데, 대부분의 정보시스템 관련 연구에서 사용하는 온톨로지가 이 BWW 프레임워크에 기반하고 있다. BWW 프레임워크에 따르면 이 세계는 고유한 프로퍼티(Property)를 갖고 있는 사물(Thing)들로 구성되며, 모든 사물은 하나 이상의 프로퍼티를 갖는다. BWW 프레임워크는 사물과 프로퍼티를 기본으로 하여 상태, 이벤트, 상호작용, 시스템의 개념을 명확히 규정함으로써, 온톨로지가 정보시스템 모델링에 적용되기 위한

<표 1> 개념 모델과 실세계의 구성 요소가 일대일로 대응되지 않는 경우

유형	설명	예(ER 모델 기준)
Construct Overload	개념 모델의 한 구성 요소가 실세계의 둘 이상의 구성 요소를 표현할 수 있는 경우	ER 모델의 개체가 실세계의 사물 또는 사건을 표현함
Construct Redundancy	실세계의 한 구성 요소가 개념 모델에서 둘 이상의 구성 요소로 표현 가능한 경우	실세계의 사건이 ER 모델에서 개체 또는 관계(Relationship)로 표현됨
Construct Excess	개념 모델의 구성 요소가 실세계에서는 존재하지 않는 개념을 표현하는 경우	ER 모델의 선택적 속성(Optional Attribute)
Construct Deficit	실세계의 특정 개념을 표현하기 위한 구성 요소가 개념 모델에 존재하지 않는 경우	실세계의 사물의 상태(State of a Thing)

철학적 토대를 마련하였다.

이러한 BWW 프레임워크는 정보시스템의 각 분야에서 사용되어 온 주요 개념 모델들을 평가하고 개선하기 위한 기준으로 사용되고 있다. 예를 들어 데이터베이스 분야의 경우에는 관계형 구조로의 변환이 용이하다는 큰 장점을 갖는 ER 모델 [정일주, 1999; 정일주, 2005; Chen, 1976; Jajodia and Ng, 1983]이 개념적 설계 단계에 가장 널리 사용되어 왔다. ER 모델은 표현력이 매우 풍부하다는 장점을 갖는 동시에, 동일한 현상이 설계자의 경험이나 직관에 따라 서로 다르게 표현될 수 있다는 한계[Campbell et al., 1985; Chen, 1976; Wand et al., 1995]도 갖는다. 이러한 모호성은 “데이터베이스 설계는 과학(Science)이 아닌 예술(Art)의 영역에 속한다”는 주장이 의미하듯 개념 모델링이 갖는 본질적인 한계로 인식되어 왔다. 하지만 개념 모델이 갖는 모호성의 원인을 보다 이론적 측면에서 규명하고자 하는 지속적인 시도가 있었으며, 그 일환으로 ER 모델이 갖는 한계의 원인을 BWW 프레임워크에 기반하여 설명하기 위한 다양한 연구[Burton-Jones and Weber, 1999; Gemino and Wand, 2005; Shanks et al., 2003; Shanks et al., 2008]가 수행된 바 있다.

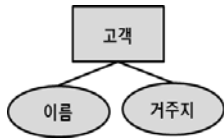
이들 연구에 따르면 사람이 개념 모델의 이해에 어려움을 겪는 현상은 개념 모델의 문법이나 표현이 존재론적 명확성(Ontological Clarity)을 갖지 못하는 경우에 발생하며, 존재론적 명확성의 결

여는 개념 모델의 각 구성 요소(Construct)와 실세계의 구성 요소 간에 일대일 대응이 이루어지지 않는 경우에 발생함을 알 수 있다. 예를 들면 ER 모델에서의 개체(Entity)는 실세계의 사물을 표현하는 데 사용될 뿐 아니라 사건(Event)을 표현하는 데에도 사용되기 때문에 존재론적 명확성이 떨어지게 된다는 주장이다. 개념 모델과 실세계의 구성 요소가 일대일로 대응되지 않는 경우는 Burton-Jones and Weber[1999]와 Weber [2003]에서 체계적으로 제시되었으며, <표 1>은 그 내용을 요약하여 보여준다.

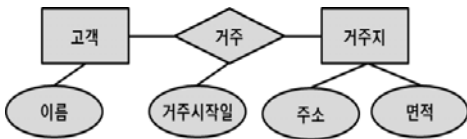
<표 1>에서 언급한 ER 모델과 실세계의 구성 요소가 일대일로 대응되지 않는 경우를 다음의 <그림 2>에 제시하였다. <그림 2>는 개념 중복(Construct Redundancy)을 보여주고 있으며, 설명에 불필요한 세부 요소(대응 수, 세부 속성, Key 등)은 모두 생략되어 있다. 우선 실세계의 “거주지”라는 개념이 <그림 2>(a)에서는 속성으로, <그림 2>(b)에서는 개체로 표현되는 개념 중복이 발생한다. 또한 실세계의 “거주”라는 개념이 <그림 2>(b)에서는 관계로, <그림 2>(c)에서는 개체로 표현되어 있다.

하지만 <그림 2>(a)와 <그림 2>(b) 간에 나타나는 개념 중복과 <그림 2>(b)와 <그림 2>(c) 간에 나타나는 개념 중복은 본질적으로 차이가 있다. 그 차이는 <그림 2>(a)와 <그림 2>(b)의 경우는 관심영역(Domain of Interest)이 서로 다

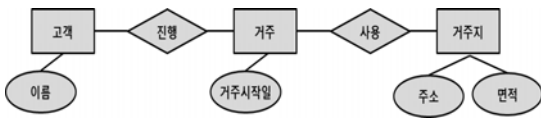
르지만 <그림 2>(b)와 <그림 2>(c)는 관심영역이 서로 동일하다는 것이다. 즉 <그림 2>(a)에서 “거주지”는 고객의 거주지 주소만을 의미하는 단순 속성이었지만, 고객의 거주 자체에 대한 관심이 발생함에 따라 “거주지”에 대한 추가 정보 관리의 필요성이 발생하고, 이에 따라 “거주지”는 <그림 2>(b)에서와 같이 개체로 격상된 것이다. 반면 <그림 2>(b)와 <그림 2>(c)는 관심영역에는 차이가 없으나, 설계자의 판단에 따라 서로 동일한 현상을 상이하게 표현하고 있다.



(a) “거주지”를 속성으로 표현



(b) 관심영역의 확장으로 인한 개체화



(c) 속성을 갖는 관계의 다른 표현
<그림 2> 관심영역의 확장과 개념 중복

본 연구에서는 <그림 2>(b)와 <그림 2>(c) 간의 개념 중복을 불필요한 개념 중복으로 규정하는 반면 <그림 2>(a)와 <그림 2>(b)간의 개념 중복은 불가피한 개념 중복으로 인정한다. 즉 본 연구에서는 개념 중복 자체를 ER 모델이 갖는 취약 요소로 파악하기 보다는, 개념 중복이 발생했을 때 어떠한 방식으로 표현하는 것이 온톨로지 관점에서 더욱 명확한지를 살펴보고자 한다. 특정 표현에 대한 온톨로지 관점의 명확성은 크게 다음의 두 가지에 의해 평가될 수 있다. 우선 모델링 결과를 이에 대응하는 온톨로지로 변환하는 작업이 쉽게 이루어져야 한다. 추후 본문에서 자세히 다루겠지만, <그림 2>(a)와 <그림 2>(c)는 이에 대응하는 온톨로지로의 변환이 쉽게 이루어지며 그 결과가 각각 <그림 3>(a)와 <그림 3>(b)에 나타나있다. 반면 <그림 2>(b)는 온톨로지로의 직접 변환이 어렵기 때문에, 온톨로지 관점에서 명확하지 않은 표현이라고 할 수 있다. 온톨로지 관점의 명확성을 평가하는 또 다른 기준은 사용자의 이해도를 평가하는 것이다. 이는 온톨로지 관점에서 명확한 표현일수록 사용자가 개념 모델을 이해하기 쉽다는 기존의 연구 결과[Burton-Jones and Weber, 1999; Gemino and Wand, 2005; Shanks et al., 2003; Shanks et al., 2008]에 근거한 것이다.



(a) <그림 2>(a)에 대한 온톨로지 표현



(b) <그림 2>(c)에 대한 온톨로지 표현

<그림 3> <그림 2>의 ER 모델에 대응되는 온톨로지 표현

본 연구에서는 위에서 언급한 두 가지 관점, 즉 온톨로지로의 변환이 쉽고 사용자가 쉽게 이해할 수 있어야 한다는 관점에서 ER 모델의 명확성을 높이기 위한 설계 방안을 제시한다. 구체적으로는 (i) ER 모델의 구성 요소들을 온톨로지 관점에서 재분류하고, (ii) 구성 요소 중 온톨로지 직접 변환이 불가능한 요소들을 식별하고자 한다. 또한 (iii) 이렇게 식별된 요소들에 대한 대체 표현법을 제안함으로써 온톨로지 쉽게 변환 가능한 ER Diagram(ERD)를 도출하고, 마지막으로 이렇게 도출된 ERD를 이에 대응되는 온톨로지 변환하기 위한 규칙을 제시하고자 한다. 본 연구에서는 실험을 통해 간단한 프로젝트 관리 시스템(PMS)에 대한 ERD를 온톨로지 관점에서 명확한 ERD로 변환하고, 이에 대응되는 온톨로지를 실제로 구축하는 과정을 보인다. 또한 일반 ERD와 온톨로지 관점에서 명확한 ERD 중 어떤 것에 대한 이해도가 더 높은지 살펴보기 위해, ER 모델에 대한 기본 지식을 갖고 있는 60명을 대상으로 실험을 수행하고 그 결과를 분석하고자 한다.

본 논문의 이후 구성은 다음과 같다. 다음 절인 제 2절에서는 ER 모델, BWW 프레임워크, 그리고 ER 모델과 온톨로지를 함께 다룬 기존의 연구들을 간단히 요약한다. 제 3절에서는 온톨로지 관점에서 명확한 ERD 설계 방안을 제시하고, 이러한 ERD를 온톨로지 직접 변환할 수 있는 변환 규칙을 제안한다. 기존 ERD와 제안하는 방법론에 의해 변환된 ERD에 대한 이해도 평가 결과는 제 4절에서 소개되며, 본 연구의 기여 및 한계는 마지막 절인 제 5절에서 요약한다.

2. 관련연구

2.1 ER 모델에 대한 연구

ER 모델은 Chen[1976]에 의해 제안되었으며, 풍부한 표현력을 가질 뿐 아니라 관계형 모델로의

변환이 매우 용이하다는 특성[Jajodia and Ng, 1983]으로 인해 현재 개념 데이터 모델링에 가장 널리 사용되고 있다. ER 모델은 대부분의 개념 모델이 Binary 관계만을 허용하는 것과 달리 N-ary 관계를 허용함으로써 매우 높은 표현력을 갖는다. 하지만 역설적으로 이로 인해 대부분의 설계자들이 설계 과정에서 개체의 식별에는 큰 어려움을 겪지 않지만 관계의 설정에 많은 어려움을 겪는 현상[Burton-Jones and Weber, 1999]도 발생하게 되었다. 또한 설계자에 따라 동일한 현상을 서로 다르게 표현할 수 있다는 표현의 유연성은, 반대로 사용자들이 ERD를 통해 주어진 도메인과 업무를 이해하는데 어려움을 줄 수 있다는 약점[Campbell et al., 1985; Chen, 1976; Wand et al., 1995]으로도 작용한다.

이러한 ER 모델의 한계들을 극복하기 위해 다양한 연구들이 수행되었으며, 이들 중 많은 연구는 체계적인 설계 방법론을 제안하거나 모델링 Tool을 고안하는 것을 목표로 수행되었다. Antony and Batra[2002]는 초심자의 개념적 모델링을 위한 도구로 CODASYS를 제안하였으며, 데이터베이스 설계의 주된 문제점으로 정규화되지 않은 테이블이 도출되는 것과 정규화는 되었지만 실세계와 다른 모델이 도출되는 현상을 지적하였다. 이 연구는 이러한 문제들을 중복된 관계의 제거를 통해 대부분 해결할 수 있음을 보였으며, ER 모델을 사용하여 4차 정규형을 직접 도출해 낼 수 있음을 보였다. Storey and Goldstein[1988]는 Prolog을 이용한 CASE Tool을 제안하였다. 제안된 시스템은 우선 사용자에게 개체로 무엇을 사용할 것인지 질문을 하고, 추후 사용자가 입력한 서술형 문장으로부터 관계를 추출하는 방식으로 설계를 지원한다. 또한 자주 사용되는 개체와 이 개체의 속성을 지식베이스에 저장하여 덤으로써 반복되는 입력 부담을 덜어줄 수 있도록 설계되었다. 하지만 이 시스템은 본질적으로 사용자가 개체와 관계를

정확히 입력하지 않으면 원하는 결과를 얻기 어렵다는 한계를 갖는다.

Tool을 고안하여 설계를 지원하기 위한 연구와 별도로, 개체와 속성을 구분하기 위한 많은 연구가 수행되었다. Kop and Mayr[1998]는 요구분석 단계와 개념적 설계 단계의 사이에 개념적 예비설계(Conceptual Predesign) 단계를 추가하였다. 즉 개념적 설계 단계를 분석 단계(Analysis Phase)와 변환 단계(Transform Phase)로 세분화 하였으며, 전 단계에서는 개체와 속성을 Things라는 타입으로 묶은 뒤, 이들의 개체/속성 구분을 이후 단계에서 수행하였다. 개체와 속성을 구분하기 위한 보다 체계적인 방법론은 Teorey et al.[1986]에서 찾을 수 있다. 이 연구에서는 특정 개념은 다음의 세 가지 요건 중 한 가지 이상의 요건을 만족시키는 경우에만 개체로 표현되어야 함을 주장하였다. 우선 식별자(Identifier) 이외의 추가 속성(Descriptor)을 갖고 있는 개념은 개체로 표현되어야 하며, 다중 속성(Multivalued Attribute)도 개체로 표현되어야 한다. 또한 일대다 관계가 속성을 포함하고 있는 경우도 개체로 표현되어야 함을 주장하였다.

2.2 BWW 프레임워크에 관한 연구

Wand and Weber[1993]는 Bunge의 온톨로지[Bunge, 1977; Bunge, 1979]에서 정보시스템 모델링에 필요한 개념을 차용하여 BWW(Bunge-Wand-Weber) 온톨로지 프레임워크를 설계하였으며, 대부분의 정보시스템 관련 연구에서 사용하는 온톨로지가 이 BWW 프레임워크에 기반하

고 있다. Bunge 온톨로지에 의하면 이 세계는 고유한 프로퍼티를 갖고 있는 사물들로 구성되며, 모든 사물은 하나 이상의 프로퍼티를 갖는다. 둘 이상의 사물의 조합을 통해 합성물(Composite Thing)을 형성할 수 있으며, 이 경우 합성물은 반드시 창발적 속성(Emergent Property)를 가져야 한다. 프로퍼티는 프로퍼티를 가질 수 없으며, 프로퍼티의 종류에는 하나의 사물에 연관되는 본질적(Intrinsic) 프로퍼티와 두 개의 사물에 연관되는 상호(Mutual) 프로퍼티가 있다. Bunge 온톨로지에서는 프로퍼티와 애트리뷰트(Attribute)를 구분한다. 프로퍼티는 사람이 인식할 수도 있고 인식하지 못할 수도 있는 속성인 반면, 애트리뷰트는 사람이 인식하여 사물에 부여한 속성을 일컫는다. 즉 모든 프로퍼티는 애트리뷰트로 표현 가능하지만, 반드시 모두 표현될 필요는 없다. 또한 애트리뷰트가 반드시 프로퍼티만을 표현해야 하는 것도 아니다. BWW 프레임워크는 이 외에도 상태(State), 이벤트(Event), 상호작용(Interaction), 그리고 시스템(System)에 대한 정의도 제시하였으며, 그 내용은 <표 2>에 요약되어 있다.

2.3 온톨로지와 ER 모델의 대응에 대한 연구

Weber[2003]는 온톨로지의 관점에서 다양한 개념 모델의 장단점을 평가하였다. 개념 모델이 명확하게 이해되기 위해서는 개념 모델의 구성 요소들이 실세계의 개념들과 일대일로 대응되어야 하며, 일대일 대응이 이루어지지 않는 경우를 Construct Overload, Construct Redundancy, Construct Excess, 그리고 Construct Deficit의 네 가

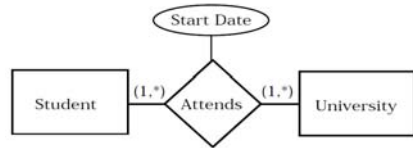
<표 2> BWW 프레임워크에서 상태, 이벤트, 상호작용, 시스템의 정의

용어	정의
상태	주어진 시점에 모든 속성이 갖는 값의 집합
이벤트	상태의 변화를 의미하며, <초기상태, 최종상태, 변환>의 Triple 형태로 표현됨
상호작용	X와 Y중 적어도 어느 하나가 상대방에게 영향을 미치는 경우, X와 Y간에는 상호작용이 존재한다고 함
시스템	상상적 상호작용 하는 사물들로 구성된 체계

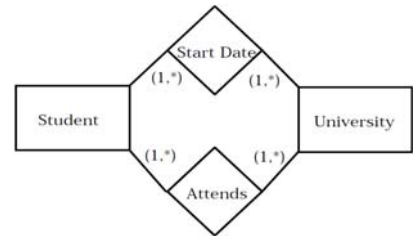
지로 제시하였다. 예를 들면 특정 개념을 개체로 표현할지 속성으로 표현할지를 결정해야 하는 어려움은, ER 모델의 개체와 속성이 실세계의 사물과 개념 중복(Construct Redundancy)을 일으키기 때문이라는 것이다.

한편 Burton-Jones and Weber[1999]는 온톨로지와 일대일 대응이 이루어지는 ERD를 Ontologically-Sound ERD라고 명명하고, 이러한 ERD를 도출하기 위한 방안을 제시하였다. 특히 이 연구는 온톨로지에서의 프로퍼티는 프로퍼티를 가질 수 없다는 점을 강조하며, ERD에서 속성을 갖는 관계를 사용하지 말 것을 주장했다. 속성을 갖는 관계를 사용하지 말아야 한다는 주장은 Wand et al.[1999]에서도 제기되었다. 이 연구에서는 이 주장과 함께 ER 모델의 선택적 속성도 사용하지 말 것을 주장했는데, 이는 실세계에서는 선택적 속성에 대응되는 개념이 존재하지 않기 때문이다. 이 연구는 <그림 4>(a)는 <그림 4>(b)로, <그림 5>(a)는 <그림 5>(b)로 변경하여 사용할 것을 주장한다.

하지만 <그림 4>와 <그림 5>에서 나타난 바와 같이, 온톨로지와 일대일로 대응되는 ERD가 오히려 해석할 뿐 아니라 해석이 어려운 측면이 있다. 따라서 Shanks et al.[2003]은 “Entity only” ERD, “Practice” ERD, 그리고 “Ontologically-Sound” ERD를 동시에 제시하여, 세 가지 유형의 ERD의 장단점을 비교하였다. 즉 “Ontologically-Sound” ERD는 기존 ERD에 비해 이해도는 높지만 현업에서 사용하는 ERD와 너무 차이가 크기 때문에 실용성이 없다는 것이다. 따라서 사용자의 이해를 받드시 필요로 하는 요구사항 분석 단계에서는 “Ontologically-Sound” ER 모델을 사용하고, 그 이후의 논리적 설계 단계부터는 “Practice” ER 모델을 사용할 것을 권장하였다. 하지만 이들 연구에서는 ER 모델의 각 구성 요소를 온톨로지로 변환하기 위한 구체적인 규칙은 다루지 않았다.



(a) 관계가 속성을 갖는 경우

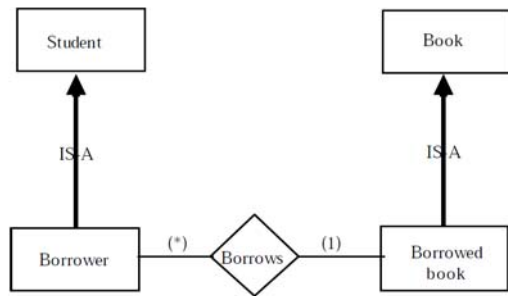


(b) 속성을 갖는 관계를 금지하는 경우

<그림 4> 속성을 가진 관계의 표현[Wand et al., 1999]



(a) 선택적 관계를 허용하는 경우



(b) 선택적 관계를 일반화로 변형하여 표현하는 경우

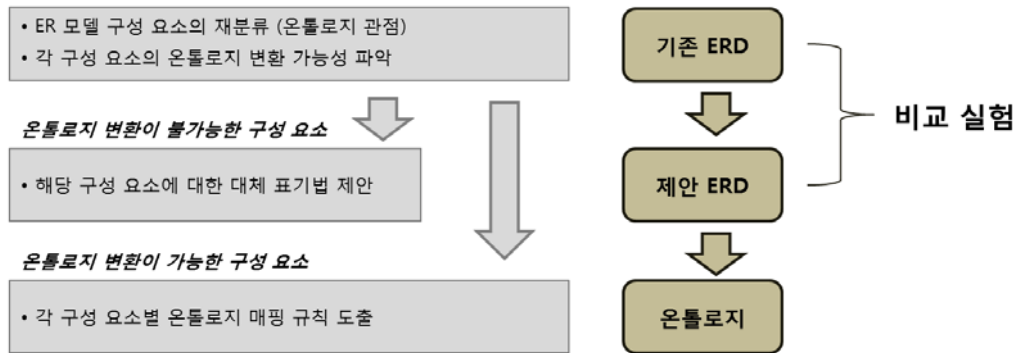
<그림 5> 선택적 관계의 표현[Wand et al., 1999]

3. 온톨로지와 상호운용 가능한 ER 모델 도출 방안

3.1 연구 범위 및 절차

본 절에서는 온톨로지와 상호운용 가능한 ER 모델 도출 방안을 제안하고, 제안된 방안에 따라 도출된 ERD를 온톨로지 변환하기 위한 매핑 규칙을 제안한다. 본 연구의 범위 및 절차는 <그림 6>에 간략히 소개되어 있다.

〈그림 6〉 연구 범위 및 절차



우선 ER 모델의 구성 요소를 전통적 관점이 아닌 온톨로지 관점에서 재분류하고, 이렇게 재분류된 각 구성 요소가 온톨로지에서 어떤 구성 요소로 대응되는지 살펴본다. 직접 대응이 어려운 구성 요소의 경우 동일한 현상을 표현할 수 있는 대체 표현법으로 수정하고, 이렇게 수정된 ERD(OC ERD : Ontologically-Clear ERD)를 온톨로지로 직접 변환할 수 있는 변환 규칙을 제안한다. OC ERD는 온톨로지의 구성 요소와 일대일로 대응된다는 점에서 기존 ERD에 비해 명확성이 높을 것으로 기대되지만, 실제로 사용자 입장에서 어떤 ERD의 이해도가 높은지를 확인하기 위해서는 별도의 실험이 필요하다. 본 연구에서는 이를 위해 기존 ERD와 OC ERD 간의 이해도 평가 실험을 수행하였으며, 그 결과는 다음 절인 제 4절에서 소개한다.

3.2 온톨로지 관점에서 명확한 ERD(OC ERD : Ontologically-Clear) 작성 규칙

본 부절에서는 온톨로지와 상호운용 가능한 ERD(OC ERD)를 작성하기 위한 규칙을 제안한다. 이를 위해 우선 기존의 연구를 통해 알려진 ER 구성 요소와 온톨로지 간의 변환 규칙을 살펴보기로 한다. <표 3>은 Wand et al.[1999]에서 정리된 변환 규칙의 일부를 발췌한 것으로, ER 모델의 주요 구성 요소와 온톨로지 구성 요소간의 대응 관계를 보여주고 있다.

<표 3>에서 개체, 속성, 관계, 그리고 일반화 등 ER 모델의 주요 구성 요소는 세부 표현 상황에 따라 서로 상이한 온톨로지 구성 요소에 대응됨을 알 수 있다. 예를 들어 동일한 이진 관계(Binary Relationship)라 하더라도, 속성을 갖지

〈표 3〉 ER의 주요 구성 요소와 온톨로지 구성 요소 간의 대응 관계[Wand et al., 1999]

ER Constructs		Ontology Constructs
Entity	Entity	Thing
Attribute	Attribute of an Entity	Intrinsic Property
	Attribute of a Relationship	No direct representation
Relationship	Binary Relationship without Attribute	Mutual Property
	Binary Relationship with Attribute	No direct representation
	N-ary Relationship without Attribute	No direct representation
	N-ary Relationship with Attribute	No direct representation
Generalization	Generalization	SSubclass of

않으면 상호 프로퍼티에 대응되고 속성을 갖는 경우엔 곧바로 대응되는 온톨로지 표현이 없음을 알 수 있다. 이처럼 동일한 구성 요소라도 온톨로지에서 서로 다른 요소로 표현되는 이유는, 기존의 전통적인 ER 구성 요소의 분류 기준이 온톨로지 관점에서는 적절하지 않기 때문이다. 따라서 본 연구에서는 이러한 기존의 분류를 재구성하고자 하며, 이를 위해 ER 모델의 각 구성 요소를 수학의 집합 관점에서 설명한 Batini et al.[1992]의 연구 성과를 사용하고자 한다.

Batini et al.[1992]는 ER 모델의 각 구성 요소를 집단화 (Aggregation) 관점에서 소개하였으며, 이에 따르면 관계는 개체 간 또는 개체와 속성간의 집단화로 정의된다. 즉 개체 A와 개체 B 간에 관계 R이 존재한다면, R은 A와 B의 집단화를 표현하는 개념으로 해석된다. 만약 관계 R이 자체 속성으로 속성 C를 갖는다면, 관계 R은 개체 A와 B, 그리고 속성 C를 집단화하는 개념을 나타낸다. 기존의 분류 방식에서 관계는 해당 관계에 연결된 속성의 유무에 관계없이 참여 개체의 개수가 몇 개인지에 따라서 차수가 결정된다. 하지만 온톨로지 관점에서는 관계에 연결된 구성 요소가 개체인지 속성인지는 크게 상관이 없다. 오히려 해당 관계가 몇 개의 구성 요소에 대한 집단화를 나타내는지가 더욱 중요하게 고려되어야 한다. 이는 ER 모델과 온톨로지와의 대응이 직접적으로 이루어지기 어려운 이유가 온톨

로지는 기본적으로 이진 Binary 모델에 기반하고 있는 반면 ER 모델은 N-ary 관계를 허용하기 때문이라는 사실을 반영한 것이다. 즉 속성을 갖는 이진 관계나 속성이 없는 삼진 관계(Ternary Relationship)는 모두 세 개의 구성 요소간의 집단화라는 점에서 서로 동일하게 취급되어야 한다는 것이다.

이러한 원리에 입각하여 ER의 구성 요소를 재분류한 결과가 <표 4>에 제시되어 있다. <표 4>에서 온톨로지 구성 요소는 W3C에서 제공하는 OWL(Ontology Web Language[Bechhofer et al., 2004; McGuinness and Harmelen, 2004; Patel-Schneider et al., 2004; Smith et al., 2004])를 기준으로 도출된 것이다.

<표 4>에서 음영으로 표시된 구성 요소, 즉 관계에 속한 속성과 N-ary 집단화를 제외한 모든 구성 요소는 온톨로지의 한 가지 구성 요소로 대응되기 때문에 Construct Overload나 Construct Redundancy를 발생시키지 않음을 알 수 있다. 한편 관계에 속한 속성과 N-ary 집단화 표현을 포함한 ERD의 경우는 온톨로지로 직접 변환이 불가능하기 때문에, 제안하는 방법론에서는 이러한 두 가지 상황이 발생하는 경우 다른 구성 요소로 대체하여 표현할 것을 규정한다.

우선 관계에 속한 속성이 온톨로지로 대응되지 않는 이유는 온톨로지에서 프로퍼티는 프로퍼티를 가질 수 없다는 특성에 기인한다. 따라서

<표 4> 재분류된 ER의 주요 구성 요소와 온톨로지 구성 요소 간의 대응 관계

ER Constructs		Ontology Constructs
Entity	Entity	Class
Attribute	Attribute of an Entity	Data Property
	Attribute of a Relationship	No direct representation
Binary Aggregation	Binary Relationship w/o Attribute	Object Property
N-ary Aggregation (N > 2)	Binary Relationship with Attribute	No direct representation
	N-ary Relationship w/o Attribute	
	N-ary Relationship with Attribute	
Generalization	Generalization	SSubClassOf

ER 모델에서 속성이 관계에 속하는 경우 이를 금지하고 대신 동일한 의미의 다른 표현으로 대체하고자 한다. 관계 중 가장 빈번하게 출현하는 1:N 이진 관계가 속성을 갖는 경우에 대한 예가 <그림 7>에 나타나있다. <그림 7>(a)는 “배정일”을 관계의 속성으로 나타낸 그림이며, 전술한 바와 같이 이러한 표현은 온톨로지로 직접 대응이 불가능하다. 한편 <그림 7>(b)에서는 “배정일”이 “직원”의 속성으로 표현되었다. 이 경우 물론 <그림 7>(b)는 관계형 테이블로 변환 시 <그림 7>(a)와 동일하게 나타나지만, ER 모델에서 “배정일” 속성을 발생시키는 주체가 명확하게 표현되지 않는다는 단점이 있다. 즉 “배정일”은 “직원”의 고유 속성이라기보다는, “직원”이 “부서”에 “소속”될 때 발생하는 창발적 속성이므로 <그림 7>(b)는 바람직하지 않다는 것이다. 따라서 제안하는 방법론에서는 속성을 갖는 1:N 이진 관계는 개체로 격상시켜 <그림 7>(c)과 같이 표현할 것을 규정한다. 동일한 이유로 1:1 이진 관계, M:N 이진 관계, 그리고 모든 N-ary 관계가 속성을 갖는 경우도 개체로 격상시킨다 (ERD 수정 규칙 1).

(ERD 수정 규칙 1) : 속성을 갖는 모든 관계는 개체로 격상시킨다.

<표 4>에 제시된 N-ary 집단화는 속성을 갖는 이진 관계, 속성이 없는 N-ary 관계, 그리고 속성을 갖는 N-ary 관계로 구분된다. 하지만 ERD 수정 규칙 1에 의해 수정된 ERD에서는 속성을 갖는 관계는 더 이상 존재하지 않는다. 따라서 N-ary 집단화 중 추가 수정이 필요한 경우는 속성이 없는 N-ary 관계 하나뿐이다. <그림 8>은 특정 “프로젝트”를 위해 특정 “공급업체”에 특정 “부품”을 주문하는 경우를 나타낸다. 전통적인 ER 모델에서는 이러한 경우 <그림



(a) 1:N 이진 관계가 속성을 갖는 경우

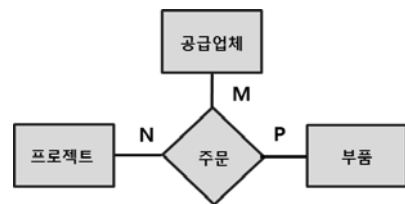


(b) 관계의 속성을 참여 개체의 속성으로 이동

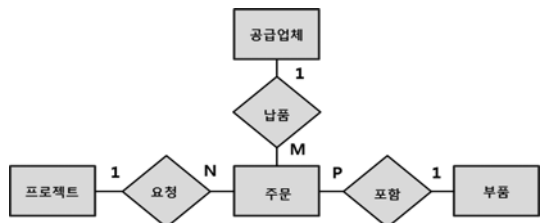


(c) 속성을 갖는 관계를 개체로 격상

<그림 7> 속성을 갖는 1:N 이진 관계의 대체 표현



(a) “주문”을 삼진 관계로 표현



(b) 관계 “주문”을 개체로 격상

<그림 8> N-ary 관계의 대체 표현

8>(a)와 <그림 8>(b)의 두 가지 설계 대안이 존재한다. 하지만 <그림 8>(a)의 경우는 <표 4>에 제시된 바와 같이 온톨로지로 직접 대응이 불가능하므로 온톨로지 관점에서 명확성이 떨어진다고 할 수 있다. 따라서 제안하는 방법론에서는

N-ary 관계의 경우 해당 관계를 개체로 격상시켜서 <그림 8>(b)와 같이 표현할 것을 규정한다 (ERD 수정 규칙 2).

(ERD 수정 규칙 2) : N-ary 관계 (N > 2)의 경우 속성을 갖는지 여부에 관계없이 개체로 격상시킨다.

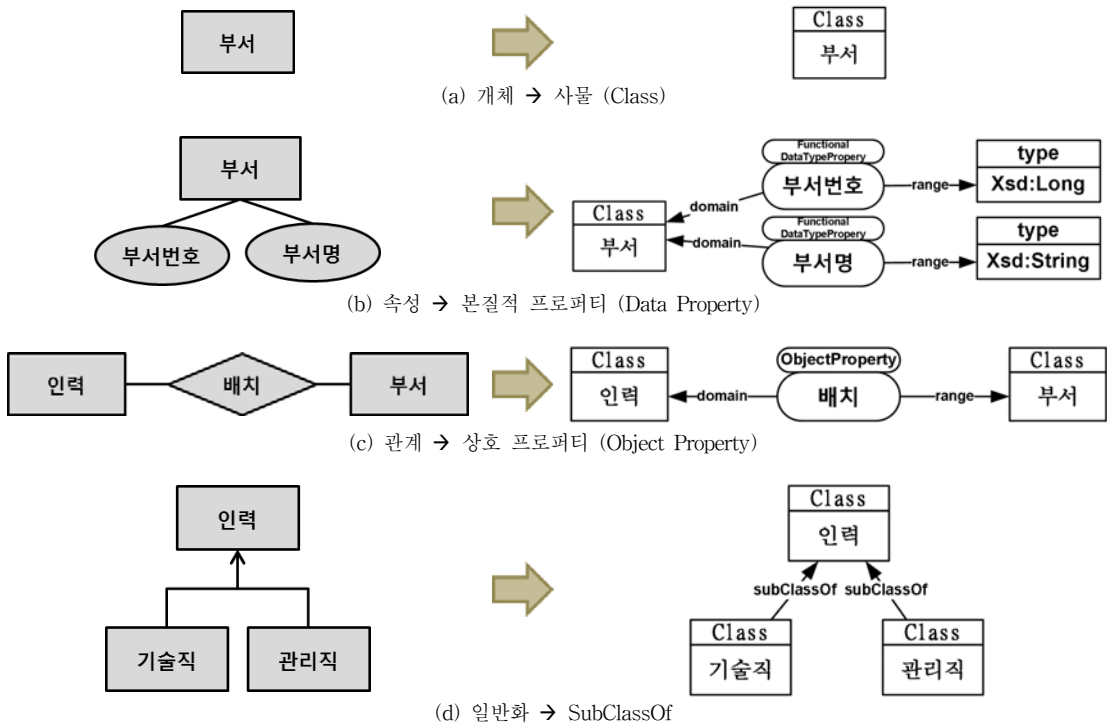
이상의 ERD 수정 규칙 1과 규칙 2에 의해 도출된 OC ERD는 수정 전의 ERD의 정보를 손실 없이 표현할 수 있으며, 간단한 변환 규칙에 따라 온톨로지기로 변환 가능하다. 온톨로지로의 변환 규칙은 다음 부절에서 설명한다.

3.3. OC ERD의 온톨로지 변환

본 부절에서는 OC ERD를 이에 대응되는 온

톨로지기로 변환하기 위한 규칙을 도출한다. 본 연구에서는 <표 4>에 제시된 ER 모델의 핵심 구성 요소(개체, 관계, 속성, 일반화)에 대한 온톨로지 변환 과정을 살펴본다(<그림 9> 참조). <그림 9>에서 Class, Data Property, Object Property, 그리고 SubClassOf의 표현은 W3C의 권고안[Bechhofer et al., 2004; McGuinness and Harmelen, 2004; Patel-Schneider et al., 2004; Smith et al., 2004]에 따른 것이다.

<그림 9>(a)에서 개체 “부서”는 온톨로지의 Class “부서”로 변환되며, <그림 9>(b)에서 개체 “부서”의 속성 “부서번호”와 “부서명”은 각각 Class “부서”를 정의역으로 갖고 자료형으로 정수형과 문자형을 갖는 Data Property로 변환된다(온톨로지 변환 규칙 1, 규칙 2). <그림 9>(c)에서 개체 “부서”와 “인력”간의 관계 “배치”는 Class “부서”와 “인력”을 각각 정의역과 치역으로 갖는 Object



<그림 9> ER 모델과 온톨로지간 구성 요소 매핑

Property로 변환된다(온톨로지 변환 규칙 3). 마지막으로 <그림 9>(d)에서 일반화는 온톨로지의 SubClassOf로 변환된다(온톨로지 변환 규칙 4). <그림 9>에서 표현된 온톨로지의 OWL Code는 <표 5>에 정리되어 있다. 이상의 온톨로지 변환 규칙들을 정리하면 다음과 같다.

(온톨로지 변환 규칙 1) : 개체는 Class로 변환한다.

(온톨로지 변환 규칙 2) : 개체의 속성은 해당 개체에 대응되는 Class를 정의역으로 갖는 Data Property로 변환한다.

(온톨로지 변환 규칙 3) : 관계는 참여 개체에 대응되는 Class들을 각각 정의역과 치역으로 갖는 Object Property로 변환한다.

(온톨로지 변환 규칙 4) : 일반화는 Sub-ClassOf로 변환한다.

<그림 9>(c)는 관계를 이에 대응되는 Object Property로 변환하는 경우를 보여준다. 이 예에는 대응수가 명시되어 있지 않지만, 실제로 대응수는 ER 모델에서 매우 중요하게 다루어지는 특성 중 하나이기 때문에 대응수에 따른 변환 규칙이 세부적으로 제시되어야 한다. 이를 설명하기 위한 예가 <그림 10>에 제시되어 있다. <그림 10>은 이진 관계가 가질 수 있는 N:1, 1:N, 1:1, M:N의 대응수를 보이고 있다. <그림 10>(a)는 하나의 “인력”이 하나의 “부서”에만 대응되는 상황을 나타낸다. 이에 대응되는 온톨로지가 “인력”을 정의역으로 갖고 “부서”를 치역으로 갖는 경우, “인력” → “부

<표 5> <그림 9>의 온톨로지 표현에 대한 OWL Code

온톨로지 구성 요소	OWL Code
(a) Class “부서”	<owl : Class rdf : about = “#부서”/>
(b) Data Property “부서명”, “부서번호”	<owl : Class rdf : about = “#부서”/> <owl : DatatypeProperty rdf : about = “#부서명”> <rdf : type rdf : resource = “&owl:FunctionalProperty”/> <rdfs : domain rdf : resource = “#부서”/> <rdfs : range rdf : resource = “&xsd:string”/> </owl : DatatypeProperty> <owl : DatatypeProperty rdf : about = “#부서번호”> <rdf : type rdf : resource = “&owl:FunctionalProperty”/> <rdfs : domain rdf : resource = “#부서”/> <rdfs : range rdf : resource = “&xsd:long”/> </owl : DatatypeProperty>
(c) Object Property “배치”	<owl : Class rdf : about = “#인력”/> <owl : Class rdf : about = “#부서”/> <owl : ObjectProperty rdf : about = “#배치”> <rdfs : domain rdf : resource = “#인력”/> <rdfs : range rdf : resource = “#부서”/> </owl : ObjectProperty>
(d) Generalization “인력”	<owl : Class rdf : about = “#인력”/> <owl : Class rdf : about = “#기술직”> <rdfs : subClassOf rdf : resource = “#인력”/> </owl : Class> <owl : Class rdf : about = “#관리직”> <rdfs : subClassOf rdf : resource = “#인력”/> </owl : Class>

서”의 방향으로 함수 관계가 성립하게 된다. 따라서 “배치”에 해당되는 Object Property는 Functional 제약 조건을 갖게 된다(온톨로지 변환 규칙 5.1). <그림 10>(b)는 이와 반대의 경우를 나타낸다. 즉 “발주”에 대응되는 Object Property는 “고객사”를 정의역으로, “프로젝트”를 치역으로 갖는다. 이 예에서는 치역인 “프로젝트”로부터 정의역인 “고객사”의 방향으로 함수 관계가 성립하게 되므로 “고객사” → “프로젝트” 방향의 “발주”는 Inverse Functional 제약 조건을 갖게 된다(온톨로지 변환 규칙 5.2). 한편 <그림 10>(c)의 경우는 어떤 방향으로도 함수 관계가 성립하지 않으므로 별도의 제약이 존재하지 않는다(온톨로지 변환 규칙 5.3). 마지막으로 <그림 10>(d)의 경우는 정의역과 치역 간에 양 방향의 함수 관계가 존재하므로 “부서장” → “부서”의 Object Property는 Functional과 Inverse Functional의 제약을 동시에 갖게 된다(온톨로지 변환 규칙 5.4). <그림 10>에서 표현된 온톨로지의 실제 OWL Code는 <표 6>에 정리되어 있다.

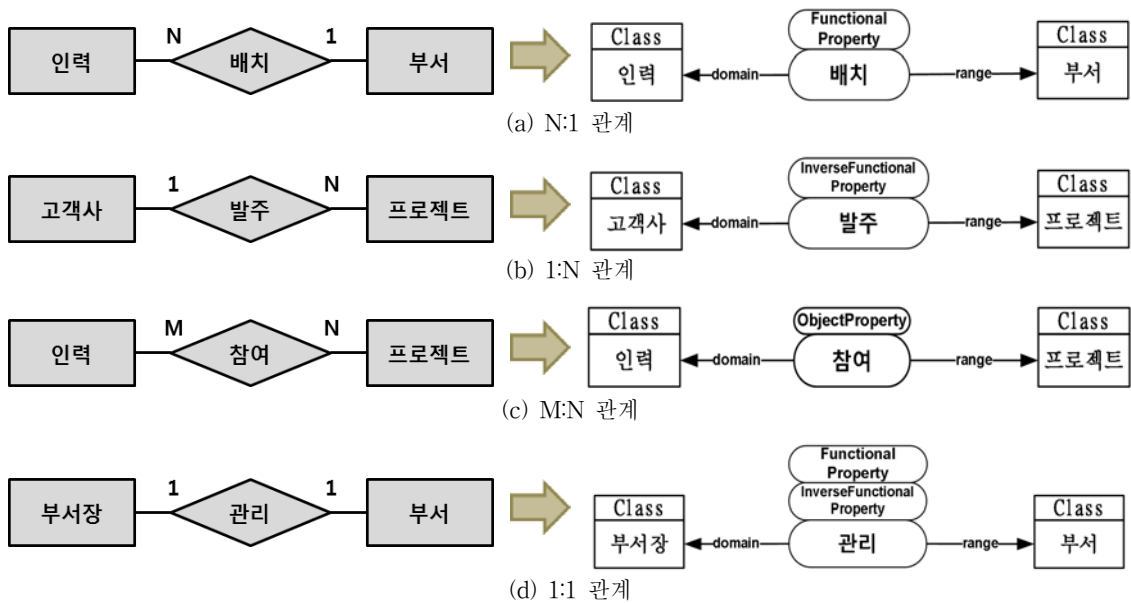
(온톨로지 변환 규칙 5) : 이진 관계의 대응수에 따라 다음의 온톨로지 변환 규칙 5.1~5.4가 선택적으로 적용된다.

(온톨로지 변환 규칙 5.1) : 이진 관계의 대응수가 $N : 1$ 인 경우, 이에 대응되는 Object Property는 Functional Property의 제약을 갖는다.

(온톨로지 변환 규칙 5.2) : 이진 관계의 대응수가 $1 : N$ 인 경우, 이에 대응되는 Object Property는 Inverse Functional Property의 제약을 갖는다.

(온톨로지 변환 규칙 5.3) : 이진 관계의 대응수가 $M : N$ 인 경우, 이에 대응되는 Object Property는 Functional Property 또는 Inverse Functional Property의 제약을 갖지 않는다.

(온톨로지 변환 규칙 5.4) : 이진 관계의 대응수가 $1 : 1$ 인 경우, 이에 대응되는 Object Property는 Functional Property와 Inverse Functional Property의 제약을 동시에 갖는다.



<그림 10> 대응수에 따른 제약조건 설정

<표 6> <그림 10>의 온톨로지 표현에 대한 OWL Code

온톨로지 구성 요소	OWL Code
(a) N : 1 관계	<pre> <owl : Class rdf : about = "#인력"/> <owl : Class rdf : about = "#부서"/> <owl : ObjectProperty rdf : about = "#배치"> <rdf : type rdf : resource = "&owl; FunctionalProperty"/> <rdfs : domain rdf : resource = "#인력"/> <rdfs : range rdf : resource = "#부서"/> </owl : ObjectProperty> </pre>
(b) 1 : N 관계	<pre> <owl : Class rdf : about = "#고객사"/> <owl : Class rdf : about = "#프로젝트"/> <owl : ObjectProperty rdf : about = "#발주"> <rdf : type rdf : resource = "&owl;InverseFunctionalProperty"/> <rdfs : domain rdf : resource = "#고객사"/> <rdfs : range rdf : resource = "#프로젝트"/> </owl : ObjectProperty> </pre>
(c) M : N 관계	<pre> <owl : Class rdf : about = "#인력"/> <owl : Class rdf : about = "#프로젝트"/> <owl : ObjectProperty rdf : about = "#참여"> <rdfs : domain rdf : resource = "#인력"/> <rdfs : range rdf : resource = "#프로젝트"/> </owl : ObjectProperty> </pre>
(d) 1 : 1 관계	<pre> <<owl : Class rdf : about = "#부서장"/> <owl : Class rdf : about = "#부서"/> <owl : ObjectProperty rdf : about = "#관리"> <rdf : type rdf : resource = "&owl; FunctionalProperty"/> <rdf : type rdf : resource = "&owl;InverseFunctionalProperty"/> <rdfs : domain rdf : resource = "#부서장"/> <rdfs : range rdf : resource = "#부서"/> </owl : ObjectProperty> </pre>

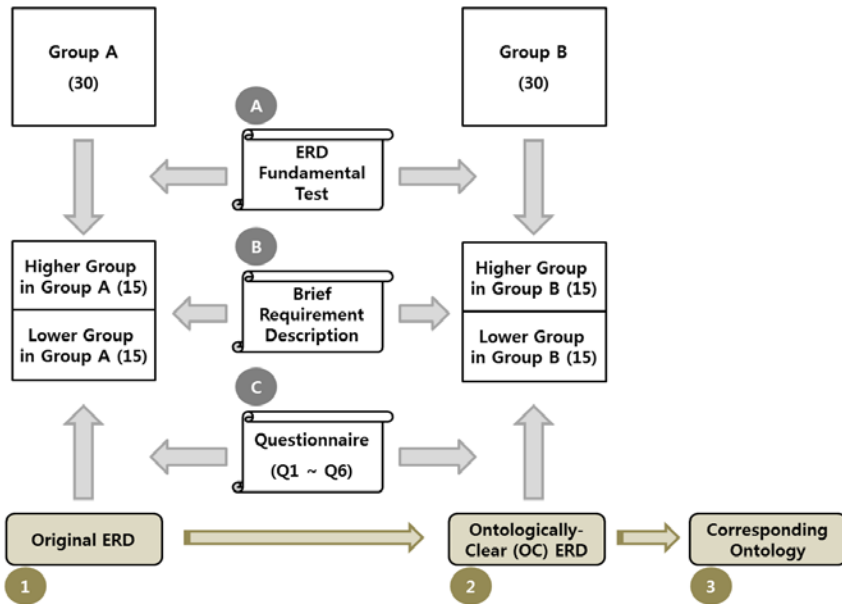
앞에서 살펴본 ERD 수정 규칙 1, 2와 온톨로지 변환 규칙 1~5에 의해 모든 ERD를 이에 대응되는 온톨로지로 변환 가능함을 살펴보았다. 다음 절인 제 4절에서는 주어진 ERD를 제안하는 ERD 수정 규칙에 따라 수정하고 이를 온톨로지 변환 규칙에 따라 온톨로지로 변환한 실제 예를 보이고, 두 가지 버전의 ERD에 대한 이해도 비교 실험을 통해 온톨로지와 상호운용 가능한 ER 모델의 효용을 살펴보고자 한다.

4. 실험

4.1 실험 절차 및 사전 실험

본 절에서는 일반적인 ERD와 OC ERD와의 이

해도 비교를 위한 실험을 수행하고 그 결과를 분석한다. 본 절에서 수행하고자 하는 실험의 전체 개요는 <그림 11>과 같다. 전체 실험은 Original ERD, Ontologically-Clear(OC) ERD, 그리고 이에 대응되는 온톨로지를 준비하기 위한 프로세스 (1)~프로세스 (3)의 프로세스와, 두 가지 버전의 ERD의 이해도를 비교하기 위한 (A)~(C)의 프로세스로 구성되어 있다. 우선 프로세스 (1)은 일반적 ER 모델 설계 방법에 따라 ERD를 도출하는 과정을 나타낸다. 본 실험에서는 이 과정을 위한 요구명세서로 한국데이터베이스진흥원 외 여러 기관이 2010년 공동으로 주최한 제7회 DA 설계공모대전에서 제시되었던 프로젝트관리시스템 (PMS) 설계 문제를 단순화하여 사용하였다. 다



〈그림 11〉 실험 전체 개요

음으로 프로세스 (2)는 도출된 ERD를 수정하여 Ontologically-Clear ERD를 획득하는 과정으로, 본 연구에서 제안된 ERD 수정 규칙 1과 규칙 2가 이 단계에 적용된다. 또한 프로세스 (3)은 Ontologically-Clear ERD로부터 온톨로지를 구축하는 단계로, 본 연구에서 제안된 온톨로지 변환 규칙 1~5가 이 단계에 적용된다. 프로세스 (1)~프로세스 (3)의 결과물이 각각 <그림 12> ~ <그림 14>에 나타나있다.

두 가지 버전의 ERD 이해도 비교 실험은 데이터베이스 과목을 수강하고 있는 학생들을 대상으로 정규 시험의 일부로 수행하였다. 일부 문항은 ER 모델에 대한 응답자의 사전 지식을 평가하기 위한 문항으로 구성되었으며, 일부 문항은 주어진 ERD에 대한 이해도를 평가하기 위한 문항으로 구성되었다. 본 실험은 데이터베이스에 대한 개념을 어느 정도 이해하고 있지만 전문가의 수준은 아닌 응답자를 대상으로 하였다. 점에서 본 연구의 취지에 부합되는 실험이라고 판단된다. 또한 단순 설문조사가 아니라 정규 시험

의 일부로 수행되었다는 점에서, 응답자들이 비교적 성실하게 응답에 임했을 것으로 기대된다.

우선 프로세스 (A)는 두 가지 목적으로 수행되었다. 첫 번째 목적은 실험 대상인 두 그룹 간에 ER 모델에 대한 사전 지식의 차이가 존재하는지 여부를 살펴보는 것이다. 이 사전 실험은 ER 모델의 기본 문법 및 개념에 대한 문항 평가를 통해 수행되었으며, 사전 실험 결과 Group A와 Group B의 ER 모델에 대한 사전 지식의 정도는 다르다고 할 수 없는 것으로 나타났다(<표 7> 참조). 또한 프로세스 (A)를 통해 ER 모델에 대한 사전 지식수준에 따라 각 그룹 내에서 상위 50%와 하위 50%의 수강생을 구분하였다. 이는 이후 실험에서 ER 모델의 사전 지식수준에 따라 두 가지 버전의 ERD에 대한 이해도에 차이가 발생하는지를 살펴보기 위함이다. 상위 50%에 해당되는 학생들에 대한 각 그룹 간의 수준 차이, 그리고 하위 50%에 해당되는 학생들에 대한 각 그룹 간의 수준 차이를 평가하기 위한 사전 실험도 수행되었으며, 그 결과도 역시 <표 7>에

〈표 7〉 ER 모델에 대한 두 그룹 응답자의 사전 지식 동질성 평가

통계량	전체수강생		상위 50%		하위 50%	
	Group A	Group B	Group A	Group B	Group A	Group B
평균	73.5	75.8	94.7	87.3	52.33	64.33
관측수	30	30	15	15	15	15
t-value	-0.357		2.039		-1.332	
p-value(양측)	0.723		0.052		0.196	

**p < 0.05, *p < 0.01.

함께 요약되어 있다. 이상의 모든 사전 실험을 통해 Group A와 Group B 간의 ER 모델에 대한 사전 지식수준은 통계적으로 유의한 차이가 없음을 알 수 있었다(유의수준 : 0.05).

다음으로 프로세스(B)는 응답자들로 하여금 설계 대상 도메인에 대해 간략히 소개하는 과정으로, 프로젝트 관리 시스템의 대상 범위에 대한 간략한 기술서를 제공하였다. 자세한 내용은 이후 제공되는 ERD를 통해서만 파악할 수 있도록 하기 위해, 기술서에서는 매우 간단한 내용만을 다루도록 하였다.

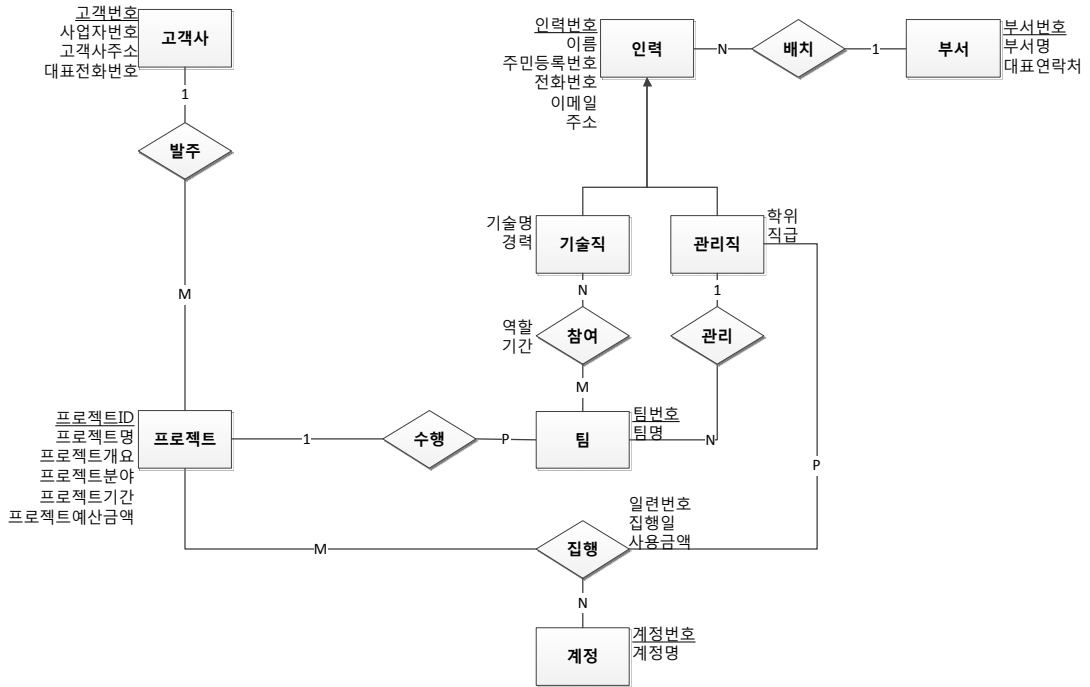
다음으로 프로세스(C)는 본 실험의 가장 핵심이 되는 부분이다. 모든 응답자들에게 ERD를 제공한 뒤, 해당 ERD에 대해 주어진 질문에 대한 답의 정확성을 평가함으로써 ERD에 대한 이해도를 측정하는 실험이다. 질문 문항은 총 6가지로 구성되었으며, 응답자 60명에게는 모두 동일한 문제지가 주어진다. 문항은 크게 관계의 속성 표현에 관한 문제(ERD 변환 규칙 1에 해당)와

N-ary 관계의 표현에 관한 문제(ERD 변환 규칙 2에 해당), 그리고 일반화에 관한 문제로 구분되며 각 문제들은 세부적으로 2개의 문제로 구성된다. 일반화 문제는 두 가지 버전의 ERD에서 동일하게 표현되므로, 실험이 정상적으로 수행되었는지를 살펴보기 위한 지표 문항으로 사용된다. 각 문항에 대해 응답자는(Yes, No, Uncertain) 중 하나로 답하는 형태로 실험에 참여하게 되며, 실험에 사용된 문항은 <표 8>에 나타나있다.

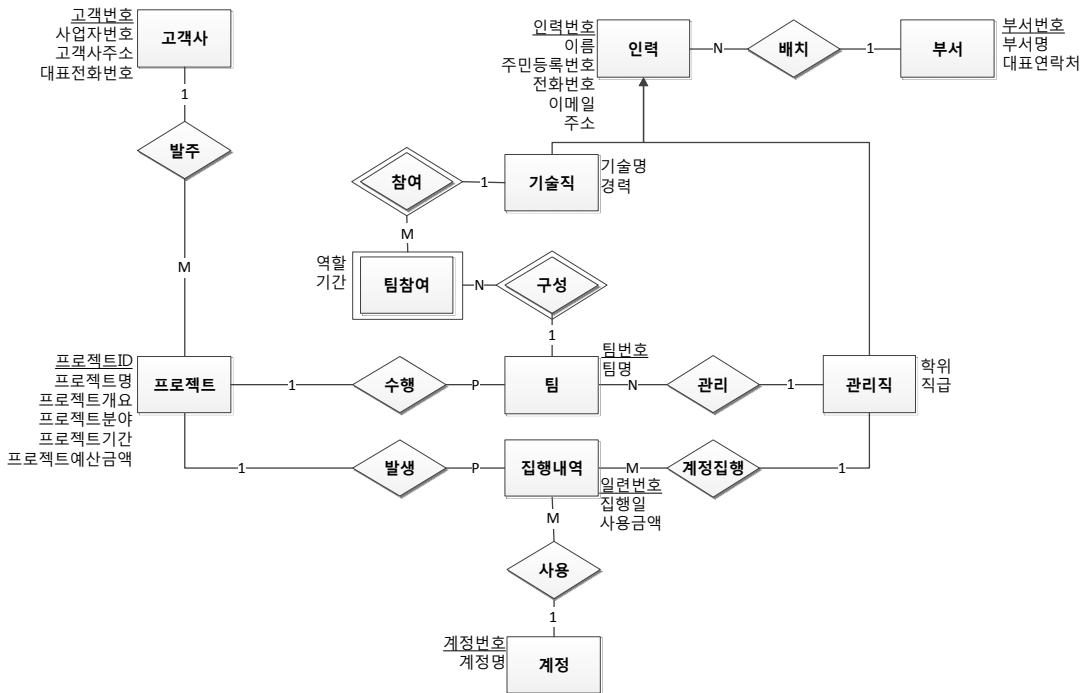
<표 8>의 문항은 모든 실험 참여자에게 동일하게 주어지지만, 각 실험 참여자는 소속 그룹에 따라 서로 다른 ERD를 받게 된다. 즉 Group A의 경우 일반적인 ER 모델링 규칙에 의해 도출된 Original ERD(<그림 12> 참조)를 받게 되며, Group B의 경우 본 논문에서 제안하는 방법론에 따라 수정된 OC ERD(<그림 13> 참조)를 받게 된다. <그림 14>는 OC ERD에 대응되는 온톨로지이며, Visio-OWL 표기법[Flynn, 2004]을 사용하여 작성되었다.

〈표 8〉 Original ERD와 OC ERD의 이해도 비교를 위한 문항

범주	문항번호	문항
관계의 속성	Q1	한 명의 기술직 인력이 어떤 팀에 참여했을 때, 그 팀에서의 역할은 단 하나로 정해진다.
	Q2	하나의 프로젝트를 두 개의 팀이 공동으로 수행하고 있다. 이 때 어떤 기술직 인력은 두 개의 팀에 동시에 소속되어 서로 다른 역할로 동일한 프로젝트에 참여할 수 있다.
일반화	Q3	한 명의 기술직 인력이 둘 이상의 부서에 배치될 수 있다.
	Q4	모든 인력은 자신이 투입된 프로젝트에 대해서 계정을 집행할 수 있다.
N-ary 관계	Q5	한 건의 계정집행내역이 있을 때, 이 건은 둘 이상의 프로젝트에 의해서 발생되었을 수 있다.
	Q6	한 건의 계정집행내역이 어느 팀에서 사용되었는지 조회한 결과 두 개의 팀이 조회되었다. 이 두 팀의 관리자는 동일인물이다.



<그림 12> Original ERD(Group A에게 주어짐)

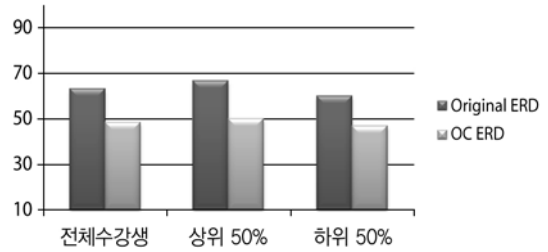


<그림 13> Ontologically-Clear ERD(Group B에게 주어짐)

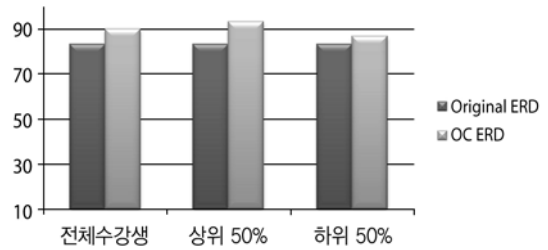
4.2 결과 해석

본 연구를 위해 수행한 실험 중 프로세스(A)의 사전실험 결과는 앞의 <표 7>에서 이미 설명하였다. 본 부절에서는 프로세스 (C)에 해당되는 실험, 즉 Original ERD와 OC ERD 간의 이해도 비교 실험에 대한 결과를 분석하고자 한다. 우선 세 가지 평가 항목에 대한 각 그룹의 평균 득점을 단순 비교한 결과가 <그림 15> ~ <그림 17>에 제시되어 있다. 그림에서 가로 축은 응답자의 구분, 세로 축은 해당 그룹의 점수 평균을 나타낸다. 또한 Original ERD는 Group A에 주어진 일반적인 ERD를, OC ERD는 Group B에 주어진 Ontologically-Clear ERD를 나타낸다.

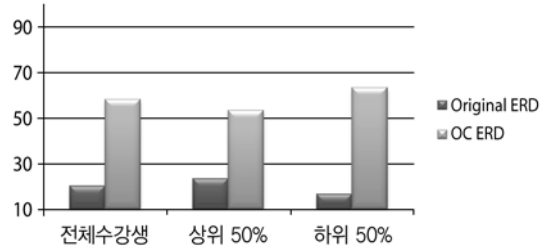
<그림 15> ~ <그림 17>에서 파악된 바는 다음과 같다. 우선 관계의 속성 표현에 대한 실험의 경우 Original ERD가 OC ERD에 비해 오히려 이해도가 높은 것으로 나타났다. 한편 일반화의 경우 OC ERD의 이해도가 근소하게 높은 것으로 나타났으나, 그 차이는 미미한 것으로 나타났다. 마지막으로 N-ary 관계 표현의 경우 OC ERD의 이해도가 Original ERD에 비해 매우 높은 것으로 나타났다. 이상의 차이에 대한 통계적 유의성 분석 결과가 <표 9>에 요약되어 있다.



<그림 15> 관계의 속성 표현에 대한 두 그룹의 이해도 비교



<그림 16> 일반화 표현에 대한 두 그룹의 이해도 비교



<그림 17> N-ary 관계 표현에 대한 두 그룹의 이해도 비교

<표 9> 두 그룹 간 이해도 차이의 통계적 유의성 분석 결과

		관계의 속성	일반화	N-ary 관계	
전체 수강생	평균	Group A	63.3	83.3	20
		Group B	48.3	90	58.3
	t-value		2.298	-0.941	-5.139
	p-value(단측)		0.013**	0.175	0.000002***
상위 50%	평균	Group A	66.7	83.3	23.3
		Group B	50	93.3	53.3
	t-value		1.581	-1.090	-3.367
	p-value(단측)		0.063*	0.144	0.002***
하위 50%	평균	Group A	60	83.3	16.7
		Group B	46.7	86.7	63.3
	t-value		1.673	-0.302	-3.862
	p-value(단측)		0.053*	0.383	0.0003***

* p < 0.1, ** p < 0.05, *** p < 0.01.

위의 분석 결과를 요약하면 다음과 같다. 우선 일반화에 대한 문항의 경우 전체 수강생, 상위 50%, 하위 50% 모든 경우에 대해 Group A와 Group B 간의 유의한 평균 차이는 나타나지 않았다(전체 수강생 : $p = 0.175$, 상위 50% : $p = 0.144$, 하위 50% : $p = 0.383$). 이는 일반화 부분은 두 버전의 ERD에서 동일하게 표현된 점을 감안하면 당연한 결과라고 할 수 있다. 한편 관계의 속성에 대한 문항의 경우 Original ERD가 OC ERD에 비해 오히려 이해도가 높은 것으로 나타났다. 이러한 현상은 전체 수강생($p = 0.013$), 상위 50%($p = 0.063$), 하위 50%($p = 0.053$)에서 모두 유의하게 나타났다(유의수준 : 0.1). 이러한 현상의 원인은 속성을 갖는 이진 관계를 모든 응답자들이 충분히 접해 본 경험이 있다는 측면에서 찾을 수 있다. 즉 관계가 속성을 갖는 표현이 온톨로지 관점에서는 명확성이 떨어진다고 할 수 있지만, ER 모델에 익숙한 많은 사용자들은 당초 예상과 달리 이러한 표현에 대해 큰 어려움을 느끼지 않았던 것으로 해석할 수 있다.

반면 N-ary 관계 표현에 대한 문항의 경우 OC ERD의 이해도가 Original ERD에 비해 높은 것으로 나타났으며, 그 차이도 매우 크게 나타났다(전체 수강생 : $p = 0.000002$, 상위 50% : $p = 0.002$, 하위 50% : $p = 0.0003$). 이러한 결과는 ER 모델링에서 N-ary 관계는 해석이 어려움으로 가급적 사용하지 않는 것이 좋다는 일부의 주장과 같은 맥락에서 해석될 수 있다. 즉 N-ary 관계는 온톨로지 관점에서 명확성이 떨어질 뿐 아니라 이해하기도 어려우므로, 모든 N-ary 관계

를 개체로 격상시켜 표현하는 것이 바람직하다는 결론을 얻을 수 있다. 이상의 실험을 통해 얻은 결론이 <표 10>에 요약되어 있다.

5. 결 론

개념 모델링은 정보시스템 개발의 가장 중요한 요소 중 하나로 인식되어 왔으며, 개념 모델링의 이론적 토대를 찾기 위한 연구가 지속적으로 수행되어 왔다. 특히 최근에는 개념 모델링의 이론적 근거를 온톨로지에서 찾으려는 움직임이 활발하게 이루어지고 있으며, 그 가운데에서도 Bunge-Wand-Weber(BWW)의 프레임워크를 기반으로 ER 모델 등 기존의 개념 모델을 평가하고 보완하기 위한 연구가 꾸준히 수행되고 있다. 본 연구에서는 개념 데이터 모델링에 가장 널리 사용되는 ER 모델과 온톨로지 간의 상호운용성을 살펴보았다. 즉 ER 모델에서 일부 표현이 온톨로지 관점에서 명확하지 않기 때문이 이해도를 감소시킬 수 지적하고, 이러한 표현을 다른 표현으로 대체하여 온톨로지 관점에서 명확한 ERD(OC ERD : Ontologically-Clear ERD)를 사용할 것을 제안하였다. 또한 OC ERD를 이에 대응되는 온톨로지로 직접 변환하기 위한 변환 규칙을 제안하고 간단한 프로젝트 관리 시스템(PMS)에 대해 ERD를 온톨로지로 변환하는 과정을 보였다.

OC ERD와 기존의 ERD의 이해도에 대한 평가를 위해 데이터베이스 과목을 수강하는 학생들을 대상으로 실험을 수행하였다. 이 실험은 수강생들을 대상으로 실시된 정규 시험의 일부로

<표 10> 실험 결과 요약(이해도 높음 > 이해도 낮음)

	관계의 속성	일반화	N-ary 관계
전체 수강생	Original ERD > OC ERD	유의한 차이 없음	Original ERD << OC ERD
상위 50%	Original ERD > OC ERD	유의한 차이 없음	Original ERD << OC ERD
하위 50%	Original ERD > OC ERD	유의한 차이 없음	Original ERD << OC ERD

수행되었으며, ER 모델에 대한 응답자의 사전 지식에 따라 상위 그룹과 하위 그룹을 구분하여 실험 결과를 분석하였다. 실험에서는 관계의 속성 그리고 N-ary 관계를 나타내는 경우 일반 ERD와 OC ERD 중 어떤 표현을 응답자들이 더욱 정확하게 이해하는지 평가하였다. 실험 결과 이진 관계가 속성을 갖는 경우는 이러한 표현이 온톨로지 관점에서 명확하지 않음에도 불구하고 응답자들이 비교적 쉽게 이해함을 알 수 있었다. 반면 N-ary 관계의 경우 이 관계를 개체로 격상시켜서 온톨로지 관점에서 명확하게 변경하여 주었을 때 표현의 이해도가 매우 향상됨을 알 수 있었다.

본 연구에서는 ER 모델의 구성 요소 중 개체, 속성, 관계, 그리고 일반화 등의 주요 요소에 대한 온톨로지 변환만 다루었다. 추후 연구에서는 키 속성, 강성 개체/약성 개체, 선택적 관계/강제적 관계 등 ER 모델의 세부 요소들에 대한 변환 규칙이 반드시 제시되어야 할 것이다. 또한 본 실험에는 한 그룹 당 30명씩 총 60명이 참여하였다. 또한 각 그룹은 ER 모델에 대한 기본 지식에 따라 다시 상위 50%와 하위 50%로 나뉘어져서 각 세부 그룹에는 15명의 응답자만 포함되어 있다. 이는 결과의 통계적 유의성을 분석하기에 충분한 숫자가 아니므로, 추후 보다 많은 응답자를 대상으로 한 추가 실험을 통해 본 연구 결과의 신뢰성을 향상시킬 수 있을 것으로 기대한다.

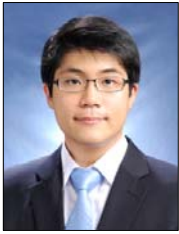
참 고 문 헌

- [1] 윤홍원, 이중화, 김정원, “변경 집합을 이용한 온톨로지 버전 관리 기법”, *Journal of Information Technology Applications and Management*, Vol. 12, No. 3, 2005, pp. 27-39.
- [2] 정일주, “개체관계 모형의 설계 특성과 성과에 관한 실험적 연구”, *Journal of Information Technology Applications and Management*, Vol. 6, No. 2, 1999, pp. 45-69.
- [3] 정일주, “E-R 모델과 자동생성기를 이용한 응용시스템의 구축 과정에 관한 연구”, *Journal of Information Technology Applications and Management*, Vol. 12, No. 4, 2005, pp. 133-155.
- [4] Antony, S. R. and Batra, D., “CODASYS : A Consulting Tool for Novice Database Designers”, *ACM SIGMIS Database*, Vol. 33, No. 3, 2002, pp. 54-68.
- [5] Batini, C., Ceri, S., and Navathe, S. B., “Conceptual Database Design : An Entity-Relationship Approach”, Benjamin-Cummings Publ. Co., Inc., Redwood City, 1992.
- [6] Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A., “OWL Web Ontology Language Reference”, W3C Recommendation, 2004(available at : <http://www.w3.org/TR/2004/REC-owl-ref-20040210>).
- [7] Berners-Lee, T., Hendler, J., and Lassila, O., “The Semantic Web”, *Scientific Am*, 2001, pp. 34-43.
- [8] Bunge, M., “Treatise on Basic Philosophy, Ontology I : The Furniture of the World”, Boston, MA : Reidel, Vol. 3, 1977.
- [9] Bunge, M., “Treatise on Basic Philosophy, Ontology II : A World of Systems”, Boston, MA, Reidel, Vol. 4, 1979.
- [10] Burton-Jones, A. and Weber, R., “Understanding Relationships with Attributes in Entity-Relationship Diagrams”, in *Proceedings of the 20th international conference on Information Systems*, 1999, pp. 214-228.
- [11] Campbell, D. M., Embley, D. W., and Czejdo,

- B. D., "A Relationally Complete Query Language for an Entity-Relationship Model", in *Proceedings of the 4th International Conference on Entity-Relationship Approach*, 1985, pp. 90-97.
- [12] Chen, P. P. S., "The Entity-Relationship Model-Toward a Unified View of Data", *ACM Transactions on Database Systems (TODS)*, Vol. 1, No. 1, 1976, pp. 9-36.
- [13] Flynn, J., "MS Visio VisioOWL Stencil", SemWebCentral, 2004(available at : <http://projects.semwebcentral.org/projects/visioowl>).
- [14] Gemino, A. and Wand, Y., "Complexity and Clarity in Conceptual Modeling : Comparison of Mandatory and Optional Properties", *Data and Knowledge Engineering*, Vol. 55, No. 3, 2005, pp. 301-326.
- [15] Gruber, T. R., "A Translational Approach to Portable Ontology Specifications", *Knowledge Acquisition*, Vol. 5, No. 2, 1993, pp. 199-220.
- [16] Jajodia, S. and Ng, P. A., "On the Representation of Relational Structures by Entity-Relationship Diagrams", in *Proceedings of ER*, 1983, pp. 249-263.
- [17] Kop, C. and Mayr, H. C., "Conceptual Pre-design-Bridging the Gap between Requirements and Conceptual Design", in *Proceedings of the 3rd International Conference on Requirements Engineering : Putting Requirements Engineering to Practice(ICRE '98)*, 1998, pp. 90-98.
- [18] McGuinness, D. and Harmelen, F., "OWL Web Ontology Language Overview", W3C Recommendation, 2004(available at : <http://www.w3.org/TR/owl-features/>).
- [19] Oei, J. L. H., Van Hemmen, L. J. G. T., Falkenbert, E., and Brinkkemper, S., "The Meta Model Hierarchy : A Framework for Information Systems Concepts and Techniques", *Technical Report, Department of Information Systems*, University of Nijmegen, 1982, pp. 92-17.
- [20] Patel-Schneider, P. F., Hayes, P., and Horrocks, I., "OWL Web Ontology Language Semantics and Abstract Syntax", W3C Recommendation, 2004(available at : <http://www.w3.org/TR/2004/REC-owl-semantics-20040210>).
- [21] Shanks, G., Nuredini, J., Tobin, D., Moody, D. L., and Weber, R., "Representing Things and Properties in Conceptual Modelling : an Empirical Evaluation", in *Proceedings of the 11th European Conference on Information Systems(ECIS 2003)*, 2003, pp. 1-17.
- [22] Shanks, G., Tansley, E., Nuredini, J., Tobin, D., and Weber, R., "Representing Part-Whole Relations in Conceptual Modeling : An Empirical Evaluation", *MIS Quarterly*, Vol. 32, No. 3, 2008, pp. 553-573.
- [23] Siau, K., "The Psychology of Information Modeling", *Advanced Topics in Database Research*, Vol. 1, Idea Group Publishing, 2003, pp. 106-118.
- [24] Smith, M. K., Welty, C., and McGuinness, D. L., "OWL Web Ontology Language Guide", W3C Recommendation, 2004(available at : <http://www.w3.org/TR/2004/REC-owl-guide-20040210>).
- [25] Storey, V. C. and Goldstein, R. C., "A Methodology for Creating User Views in Database Design", *ACM Transactions on Database Systems(TODS)*, Vol. 13, No. 3, 1988,

- pp. 305-338.
- [26] Teorey, T. J., Yang, D., and Fry, J. P., "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model", *ACM Computing Surveys (CSUR)*, Vol. 18, No. 2, 1986, pp. 197-222.
- [27] Wand, Y., Monarchi, D., Parsons, J., and Woo, C., "Theoretical Foundations for Conceptual Modelling in Information Systems Development", *Decision Support Systems*, Vol. 15, No. 4, 1995, pp. 285-304.
- [28] Wand, Y., Storey, V. C., and Weber, R., "An Ontological Analysis of the Relationship Construct in Conceptual Modeling", *ACM Transactions on Database Systems*, Vol. 24, No. 4, 1999, pp. 494-528.
- [29] Weber, R., "Conceptual Modelling and Ontology : Possibilities and Pitfalls", *Journal of Database Management*, Vol. 14, No. 3, 2003, pp. 1-20.

■ 저자소개



이 동 훈

한국방송통신대학교 컴퓨터과학과에서 학사 학위를 취득하였으며, 현재 국민대학교 비즈니스 IT전문대학원에서 비즈니스IT 전공 석사 과정에 재학 중이다.

주요 관심분야는 데이터 모델링, 데이터 베이스, 빅 데이터, 데이터 마이닝 및 온톨로지 등이다.



김 남 규

현재 국민대학교 경영정보학부에서 조교수로 재직 중이다. 서울대학교 컴퓨터공학과에서 학사 학위를 취득하고, KAIST 테크노경영대학원에서 Database

와 MIS를 전공하여 경영공학 석사 및 박사학위를 취득하였다. 한국지능정보시스템학회 이사, JITAM 편집위원 및 한국정보시스템학회, 한국경영정보학회 종신회원으로 활동 중이다. 주요 관심분야는 시멘틱 데이터 관리, 데이터베이스 설계 및 데이터 마이닝 등이다.



정 인 환

현재 한성대학교 컴퓨터공학과에서 부교수로 재직 중이다. 한양대학교 원자력공학과에서 학사 학위를 취득하고, KAIST 정보 및 통신공학과에서 분산데이

터베이스를 전공하여 정보 및 통신공학 석사 및 박사학위를 취득하였다. 한국정보과학회, 한국정보처리학회 및 한국멀티미디어학회 종신회원으로 활동 중이다. 주요 관심분야는 분산데이터베이스에서의 트랜잭션 관리, 분산시스템최적화 및 멀티미디어 통신 등이다.